

信息中心

V7.1.3.5 本信息处于开发状态，可以随时进行更改；将在 *ibm.com* 上提供已发布版本。

注

在使用本资料及其支持的产品之前，请阅读第 517 页的『声明』中的信息。

目录

第 1 章 关于 IBM Rational Build Forge	1
Build Forge 产品修订版	1
过程自动化	1
Build Forge 的概念和对象	3
第 2 章 开始使用 Build Forge	7
访问和使用控制台	7
对列表进行过滤和排序	10
创建 hello world 项目	10
设置服务器	11
定义项目	13
运行作业	13
查看作业结果	14
项目示例	14
第 3 章 IBM Rational Build Forge 文 档声明	17
第 4 章 需求	21
硬件、软件和数据库需求	21
数据库需求	21
Installation Manager 需求	21
产品版本的许可需求	22
在安装期间指定许可证文件	22
为 Build Forge 配置 Rational 许可证服务器	22
获取许可证密钥并设置 Rational 许可证服务器	24
实现 IPv6 支持的联网需求	25
本地语言支持需求	26
管理控制台和代理程序的语言设置	26
数据库主机的国际数据支持	27
更改管理控制台中的用户语言首选项	27
确定 UNIX/Linux 主机的语言/字符集	28
确定 Windows 主机的语言代码页	28
第 5 章 规划安装	29
组件	29
部署类型	30
安装方法	30
配置选项	30
与其他产品集成	31
第 6 章 安装前设置	33
国际数据设置	33
数据库备份 (必需)	34
数据库设置	34
DB2 Express 安装	35
DB2 设置	36
Microsoft SQL Server 设置	39
MySQL 设置	43
Oracle 设置	46

安全性设置	53
第 7 章 安装管理控制台	55
使用启动板启动 Installation Manager	55
从产品 DVD 中启动启动板	56
从已下载的软件包打开启动板	56
Installation Manager 中的安装步骤	57
安装后核对表	60
增加 Linux 文件句柄的数量	61
第 8 章 备选安装方法	63
使用您自己的组件安装	63
先决条件	63
国际数据支持	63
数据库安装和配置	64
Apache HTTP Server 安装和配置	65
PHP 安装和配置	67
Apache Tomcat 安装和配置	70
手动安装 Installation Manager	72
启动 Installation Manager	72
指定存储库 URL	72
执行产品组件的静默安装	73
使用 Installation Manager 创建响应文件	73
以静默方式安装和运行 Installation Manager	74
执行产品组件的静默升级	75
使用 Installation Manager 创建更新响应文件	75
以静默方式运行更新安装	76
在 VMware 上安装 Build Forge 系统	76
在 Linux on System z 上安装管理控制台	77
第 9 章 在管理控制台中配置其他功能	83
Build Forge 配置文件 (buildforge.conf)	83
更新 buildforge.conf 文件	83
Buildforge.conf 引用	84
将管理控制台配置为使用备用端口	85
配置冗余	86
关于冗余	86
安装冗余系统	86
使用冗余	87
启用 IPv6 网络支持	87
修改 httpd.conf	87
设置 FLEXlm 客户机	88
安全性功能	88
实施单点登录	89
启用 SSL 和 HTTPS	101
启用密码加密	120
第 10 章 安装代理程序	131
在 Windows 平台上安装代理程序	131
用户方式代理程序	131
在 Windows 操作系统上执行静默代理程序安装	132

在 UNIX 和 Linux 系统上安装代理程序	133
在 System i 平台上安装代理程序	134
在 System z 平台上安装并运行代理程序	135
对 z/OS 上的代理程序安装进行故障诊断	137
代理程序更新和部署实用程序	138
受支持的操作系统	138
安装	139
运行 UNIX 或 Linux 的目标计算机的连接设置	139
运行 Windows 的目标计算机的连接设置	139
用法概述	142
在 XML 文件中指定目标系统	142
运行 agentupdate 实用程序	146
agentupdate 实用程序引用	147
XML 文件参数、命令行选项和环境变量的优先顺序	148
通过日志文件查看实用程序操作	149
故障诊断	149
运行代理程序	154
在 Windows 上运行代理程序	154
在 UNIX、Linux 和 MacOS 上运行代理程序	154
在 System i 上运行代理程序	154
bfagent 引用	156
配置代理程序	156
查找代理程序配置文件	156
更改代理程序端口	157
配置不同的 shell	157
在网络共享文件系统 (Windows) 上运行代理程序	157
命令	157
配置代理程序之间的直接文件传输	158
触发器变量和代理程序性能	158
bfagent.conf 引用	159
对代理程序进行故障诊断	167
测试主机名解析	167
测试连接	167
对 Windows 上的代理程序进行故障诊断	169
对 UNIX、Linux 或 MacOS 上的代理程序进行故障诊断	169

第 11 章 安装后任务 171

启动和停止引擎	171
在 Windows 上启动和停止引擎	171
在 UNIX 或 Linux 系统上启动和停止引擎	171
设置用户	172
Root 用户	172
创建和编辑用户	172
验证安装	174
服务器认证	174
配置服务器	174
创建测试项目	175
运行项目	175
日志的位置和设置	176
对常见问题进行故障诊断	177

第 12 章 从先前版本升级 179

升级的先决条件	179
升级 V7.1 控制台	179

执行更新安装	179
在升级到 V7.1.3.1 后复制适配器	180
升级 V7.0.2.x 控制台	181
升级代理程序	181

第 13 章 卸载产品组件 183

使用 Installation Manager 卸载产品	183
如果 Installation Manager 安装失败, 请手动卸载	184
该产品	184
卸载后清除 Build Forge 文件	184
使用 Installation Manager 重新安装产品并使用现有的 DB2 Express	184
卸载 Windows Build Forge 代理程序	185
卸载 UNIX 或 Linux Build Forge 代理程序	185

第 14 章 发行说明 189

发行说明 - IBM Rational Build Forge V7.1.3.5	189
--	-----

第 15 章 “仪表板”选项卡 191

第 16 章 管理 193

关于管理	193
访问组	193
访问概述	195
访问示例: 对组授予运行作业的能力	196
团队和项目安全性规划	196
管理访问属性	196
用户	197
用户属性	197
管理用户	198
使用用户类型控制访问	199
帐户锁定	199
Root 用户	200
API 用户	201
许可权	201
许可权练习	201
LDAP 集成	202
关于 LDAP 集成	202
LDAP 域属性	203
任务	206
系统配置设置	208
其他系统设置	217
审计日志记录	217
消息	221
已翻译的消息	222
预订 RSS 数据订阅源以获取作业状态	222
过滤 RSS 数据订阅源以获取系统消息	223
“安全”面板	223
密钥库面板	225
SSL 面板	226
SSO 面板	227
管理许可证	227
输入新的许可证密钥	228
令牌许可证日志	228
管理引擎	229
暂停引擎	229

启动和停止引擎	229
管理数据库	230
删除数据库日志文件	230
错误消息	230
没有活动步骤	230
许可证密钥无效或 Build Forge 许可证密钥已损坏或丢失	230
需要数据库许可证	230

第 17 章 服务器 231

关于服务器资源	231
关于“服务器”面板	231
创建服务器	232
测试服务器	234
更改服务器	234
启用和禁用服务器	234
限制服务器上的并发作业	234
复位作业计数	235
将服务器上的作业计数复位为 0	235
将所有服务器上的作业计数复位为 0	235
服务器状态的 RSS 数据订阅源	236
服务器认证	236
关于服务器认证	236
创建服务器认证	237
覆盖服务器认证	238
允许使用受限的服务器认证	238
选择器	239
关于选择器	239
选择器设置实践	240
选择器变量类型	240
选择器变量比较规则	241
针对符合条件的服务器资源的选择器评估	241
收集器	242
关于收集器	242
关于收集器面板	242
创建收集器	242
清单和动态服务器选择	243
查看清单	245
手动刷新清单	245
设置服务器清单的更新频率	245
静态和动态服务器选择的示例设置	246
属性引用	246
使用快照新建选择器实例	251
选择器快照概述	251
选择器快照计划	252
从现有选择器或选择器快照创建选择器快照	253
更改缺省选择器快照	253
更改选择器快照的快照名称	254
访问和查看选择器快照集中的快照	254
删除选择器快照	255

第 18 章 使用环境 257

关于环境	257
环境继承	257
继承的特例	258
启动作业时更改项目变量	258

关于变量	259
步骤中的变量解释	259
未声明变量的解析	260
关于“环境”面板	260
“详细信息”选项卡	261
“快照”选项卡	263
创建环境	263
使用变量	263
为变量创建下拉列表	264
包括其他环境	265
在步骤执行期间更改变量值	265
映射 Windows 驱动器	265
在变量中使用点命令	267
系统变量引用	267
触发器变量引用	270
环境快照	274
环境快照概述	275
环境快照计划	276
创建环境快照	276
更改缺省环境快照	277
更改环境快照名称	277
访问和查看快照	278
删除环境快照	278

第 19 章 使用项目 279

关于项目	279
关于“项目”面板	279
更改项目属性	282
复制项目	283
删除项目	283
使步骤固定在服务器中	284
链: 其他项目或库的条件执行	284
链式项目中的环境变量继承	285
启用等待时取消链式项目	285
定义标记	286
编辑项目的标记格式	286
同步标记	287
标记的系统定义变量	288
创建或编辑标记变量	288
库	289
关于库	289
复制库	290
日志过滤器	290
关于日志过滤器	290
创建日志过滤器	291
向步骤指定日志过滤器	292
过滤器模式	292
过滤器操作	293
过滤器通知	294
错误阈值	294
错误和警告计数	295
类	295
关于类	295
设置通知	297
关于通知模板	298
配置 SMTP 服务器	300

设置项目和步骤的通知属性	300
通知练习	300
定制通知模板	301
内联项目的通知	304
使用快照新建项目实例	304
项目快照概述	304
项目快照计划	306
项目快照选项	306
验证和编辑访问组的快照许可权	307
从现有项目或项目快照创建项目快照	307
更改缺省项目快照	308
更改项目快照的快照名称	309
访问和查看项目快照集中的快照	309
启动缺省项目快照的作业	310
启动非缺省项目快照的作业	310
删除项目快照	311

第 20 章 使用步骤 313

关于步骤	313
关于步骤面板	313
添加步骤	316
编辑步骤	317
禁用步骤	317
其他步骤操作	317
控制执行流程	318
步骤如何运行	318
内联: 包含项目或库的步骤	319
步骤的通过链和失败链	320
线程化: 并行运行步骤	320
将步骤广播到多个服务器	321
条件步骤执行	322
While 循环执行	322
条件函数	323
从步骤启动项目	325
定制日志输出	325
标记步骤的日志输出	325
将步骤输出突出显示为彩色或活动链路	326
使用作业数据	327
在项目文件中嵌入构建号	327
作业期间更改构建标记	327
作业期间更改环境变量的值	328
使用寄存器	329
项目寄存器	330
向/从步骤的服务器资源复制文件	331
在服务器资源上启用文件复制	331
从服务器获取文件	331
将文件置于服务器	331
配置直接文件传输	332
对步骤处理进行故障诊断	332
执行 ANT 构建命令后作业不会处理任何步骤命令	332
执行 Windows 批处理命令后步骤中的命令不运行	332
点命令参考	333
点命令语法	333
.bom	333
.bomexport	334

.break	335
.bset	335
.buildstatus	336
.date	336
.defect	338
.drill	339
.edit	340
.email	340
.export	341
.get	341
.load	341
.lock	344
.mkdir	344
.monitor	344
.pack	344
.pop	345
.poptag	346
.purge	346
.push	346
.put	346
.rem	347
.report	347
.retag	347
.retry	347
.rget	348
.rmdir	348
.rput	348
.run 和 .runwait	349
.scan	350
.semget	351
.semput	351
.set	351
.sleep	352
.snapshot	352
.source	353
.stop	354
.strsub	354
.test	355
.tset	355
.unlock	355

第 21 章 使用作业 357

关于作业	357
关于主面板	357
关于“作业”面板	357
运行作业和查看结果	359
启动作业	359
查看作业结果	361
作业状态的 RSS 数据订阅源	361
重新启动失败的作业	362
使用材料清单	363
调度作业	365
调度参数	367
调度作业类清除	369
管理作业	369
锁定作业	369

删除作业	370
作业的工作目录	371
信号量	373
第 22 章 使用报告	377
关于报告	377
在报告输出中显示数据的先决条件	377
显示报告中数据的访问许可权	378
将内置报告结果导出到 CSV 文件	378
“性能”的标准报告	378
查看项目的作业性能统计信息	378
查看项目的作业持续时间	379
查看项目的步骤和服务器性能	379
“查询”的预定义报告	380
查看选择器使用情况历史记录	380
查看按服务器列出的当前服务器清单	381
查看作业通过/失败/警告结果	381
查看某个日期范围内作业的服务器使用情况	382
使用作业文件的 MD5 值搜索作业文件	382
使用快速报告创建报告	383
快速报告的报告组许可权	383
快速报告的报告类型引用	384
快速报告的报告格式和显示引用	389
样本报告引用	393
使用提供的报告类型创建报告	394
将报告输出添加到作业 BOM	395
在快速报告中修改和管理报告	395
第 23 章 使用实用程序	399
访问和运行命令行实用程序	399
导出项目	399
bfexport 参考	399
使用 .export	401
导入项目	402
使用“导入”实用程序导入项目和其他对象	402
bfimport 参考	403
访问组如何指定给导入的对象	406
访问组指定和安全性	406
导入时重命名和替换对象	406
第 24 章 在“UI 配置”选项卡中链接到 Web 资源	409
第 25 章 Build Catalyst	411
受支持的操作系统	411
安装概述	411
在 Linux 和 Solaris 操作系统上安装 Build Catalyst	412
在 Windows 上安装 Build Catalyst	413
Build Catalyst 示例	413
rafmake 实用程序引用	416
Build Catalyst 环境变量	418
第 26 章 Rational Build Forge 集成方法	421
使用命令行界面集成	421

使用适配器进行集成	422
适配器需求	422
适配器	422
适配器任务概述	430
核心适配器任务	433
更新 ClearQuest 构建记录	438
高级适配器任务	439
适配器参考	444
适配器 XML 引用	446
第 27 章 Rational Build Forge 和 Rational ClearCase	455
第 28 章 Rational Build Forge 和 Rational ClearQuest	461
第 29 章 Rational Build Forge 和 CVS	467
第 30 章 IDE 集成	469
关于 IDE 集成	469
测试项目的特殊变量	469
Eclipse 和 Rational Application Developer 的插件	470
安装 Eclipse 或 Rational Application Developer 插件	471
SSL 启用时的备选安装	472
使用 Eclipse 和 Rational Application Developer 的插件	473
Rational Team Concert 插件	473
使用 Rational Team Concert 插件	474
对 Rational Team Concert 插件进行故障诊断	475
第 31 章 Rational Build Forge 和 Performce	477
第 32 章 Rational Build Forge 和 Rational Quality Manager	479
第 33 章 Rational Build Forge 和 Rational Team Concert	481
Rational Team Concert V3.x.	481
Rational Team Concert V1.x 和 V2.x.	481
安装 Rational Team Concert 服务器扩展	482
为 Rational Team Concert 安装客户机插件	483
配置 Rational Team Concert 适配器	485
第 34 章 Rational Build Forge 和 StarTeam	487
第 35 章 Rational Build Forge 和 Subversion	489
第 36 章 Rational Build Forge 和 WebSphere 产品	491

使用 WebSphere Application Server 而非 Apache Tomcat	491	步骤日志 (step log)	511
配置 Websphere Application Server 以运行 Build Forge 应用程序	491	插件 (plug-in)	511
在 Websphere Application Server 中为 Build Forge 配置 Java 2 安全性	494	动态 (dynamic)	511
在 Websphere Application Server 中为 Build Forge 配置 SSL 和密码加密支持	494	代理程序 (agent)	511
使用 Websphere Application Server 运行 Build Forge 应用程序的启动顺序	495	服务器 (server)	511
使用 IBM HTTP Server 而不是 Apache HTTP Server	495	服务 (services)	512
设置预编译的 PHP 文件	496	访问组 (access group)	512
编辑 IBM HTTP Server 配置文件	497	管理控制台 (Management Console)	512
识别 PHP 中的代理服务器	498	归档 (archive)	512
为 IHS 配置 SSL	498	环境 (environments)	512
Windows: 替换 DLL	500	接口 (interface)	512
AIX 系统: 额外设置	500	静态 (static)	512
第 37 章 使用 API	501	快照 (snapshot)	512
为 API 程序创建 Build Forge 用户	501	库 (library)	512
Java 客户机 API	501	类 (class)	513
获取 Java 客户机 API 软件包	502	拦截器 (interceptor)	513
安装 Java 客户机 API	502	破坏 (clobber)	513
Perl 客户机 API	502	清单 (manifest)	513
获取 Perl 客户机 API 软件包	503	轻量级目录访问协议 (Lightweight Directory Access Protocol)	513
安装 Perl 客户机 API	503	数据库 (database)	513
第 38 章 确定管理控制台版本号	505	收集器 (collector)	513
第 39 章 性能和可伸缩性	507	适配器 (adaptor)	513
第 40 章 随产品安装的可执行命令	509	通知模板 (notification templates)	514
第 41 章 词汇表	511	握手 (handshake)	514
步骤 (step)	511	信号量 (semaphore)	514
		选择器 (selector)	514
		项目 (project)	514
		线程技术 (threading)	514
		引擎 (engine)	514
		用户 (user)	515
		作业 (job)	515
		BOM	515
		附录. 声明	517
		商标	518

第 1 章 关于 IBM Rational Build Forge

本部分描述了产品修订版和基本概念。

Build Forge 产品修订版

提供了产品的以下修订版:

- IBM Rational Build Forge Standard Edition
- IBM Rational Build Forge Enterprise Edition
- IBM Rational Build Forge Enterprise Plus Edition

下表列出了这些版本的不同组件或功能。

组件或功能	Standard Edition	Enterprise Edition	Enterprise Plus Edition
管理控制台 (Management Console)	Microsoft Windows、UNIX 或 Linux	Windows、UNIX 或 Linux	Windows、UNIX 或 Linux
数据库	所有受支持的数据库	所有受支持的数据库	所有受支持的数据库
代理程序	所有受支持的操作系统	所有受支持的操作系统	所有受支持的操作系统
许可证服务器	需要 (25 个并发用户)	需要 (150 个并发用户)	不需要 (250 个并发登录)
适配器工具包	受支持	受支持	受支持
快速报告	受支持	受支持	受支持
API (Perl 和 Java)	受支持	受支持	受支持
动态服务器管理	不受支持	受支持	受支持

过程自动化

过程自动化使企业能够编排涉及不同应用程序、人员和系统的过程，以消除低效情况、优化成本、确保合规性，以及提高生产力。

Build Forge 可以对产品开发生产线中的过程进行自动化、编排、管理和跟踪。它常用于使软件构建和软件包过程实现自动化。

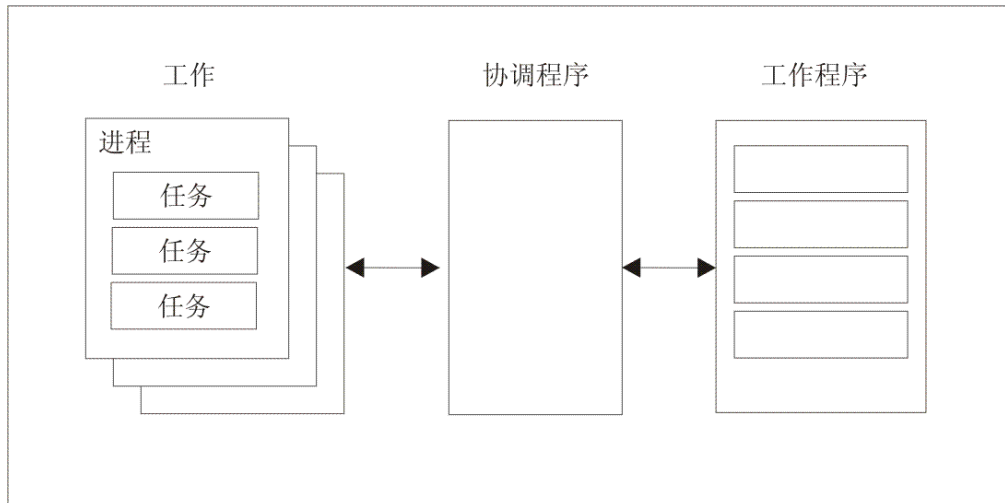
本部分描述针对过程的人员模型，并将该模型映射到 Build Forge 功能。

人员模型

以下角色构成了基于人员的过程自动化模型。

- **工作:** 要完成的工作是一组已记录的过程。每个过程由若干步骤组成。

- **协调程序：**协调程序负责“运行”工作。协调程序选择要运行的过程，然后按顺序检查任务。对于每个任务，过程会定义应当执行该任务的工作程序。协调程序将任务移交给工作程序并等待结果。
- **工作程序：**工作程序执行过程中定义的任务。



Build Forge 模型

基于人员的模型映射到 Build Forge 模型，如下所示：

- **工作：**工作过程在 Build Forge 项目对象中定义。每个项目包含一个或多个步骤对象。项目中的步骤对应于人员模型的过程中的任务。
- **协调程序：**协调程序是正在运行的 Build Forge 软件。它包含称为过程引擎的组件。在 Build Forge 中启动项目时，过程引擎会将该项目作为作业对象运行。作业只是正在运行的项目。
- **工作程序：**工作程序是主计算机。在 Build Forge 中，它们由服务器对象表示。主机必须安装代理程序。

这是一个非常简化的模型。

- Build Forge 包含许多其他对象类型来支持过程自动化。支持服务器和项目的用户、权限和其他对象都存储在数据库中。Build Forge 的某些部件也存储在数据库中，例如 UI 窗口小部件。因此，数据库必须正在运行，Build Forge 才能运行。
- 项目和步骤可配置为以多种不同的方式运行，以支持具有复杂依赖性和工作流程的复杂过程。项目可以运行其他项目，步骤自身可以运行项目。
- 步骤和项目可以根据服务器定义中包含的条件数据，动态地选择服务器。例如，步骤可以根据是运行 Windows 还是 UNIX/Linux，选择在哪个服务器上运行。服务器也可以配置为组成池的资源，以便步骤可以根据服务器的当前可用性 or 负载情况进行选择。

要安装的 Build Forge 组件

要使用 Build Forge，必须安装以下组件：

- **数据库：**Build Forge 使用数据库来存储它所使用的全部对象。

- **Build Forge:** 一组技术，用于监管对数据库的访问、运行 UI 代码和执行正在运行的作业的工作。
- **代理程序:** 您需要在每台将执行工作的主机上安装代理程序软件。
- **Web 浏览器:** 用户通常通过 Web 浏览器中提供的控制台来访问 Build Forge。

Build Forge 的概念和对象

Build Forge 使用一组简单的对象来构造过程自动化的描述。

本部分概述了关键概念和对象，以及它们如何关联。为方便起见，还包含了针对更详细说明的链接。

用户和角色

要访问 Build Forge，用户必须为自己设置一个用户对象。在生产系统中，这通过将 Build Forge 配置为访问身份管理系统 (LDAP) 来实现。

Build Forge 使用授权系统来控制用户对所有对象的访问。访问组对象包含许可权的列表。用户属于一个或多个访问组。

您可以创建自己的访问组，也可以修改所提供的访问组。系统中定义的访问组包括：

- **构建工程师** - 定义过程（创建项目和步骤）
- **系统管理员** - 管理服务器和其他系统范围功能
- **安全性** - 管理用户、访问权和安全性
- **开发者** - 帮助开发过程、运行作业和查看结果
- **操作员** - 复制项目、运行作业和查看结果
- **访客** - 复制项目、运行作业和查看结果

服务器

服务器对象定义一个可用于运行项目和步骤的场所。项目和步骤可以使用相同的服务器，也可以独立地选择服务器。

服务器对象表示执行工作的主机。主机必须运行 Build Forge 代理程序，以从系统接收工作并返回结果。

其他对象与服务器相关，需要在定义项目和步骤前进行设置：

- **服务器认证:** 服务器认证存储登录信息，以供服务器用于访问由服务器指定的主机。必须先创建服务器认证，然后才能创建使用该服务器认证的服务器。在服务器定义中，您可以从系统中定义的所有服务器认证的列表中选择服务器认证。
- **收集器:** 收集器对象负责收集指定的服务器属性。数据存储在清单中。服务器指定了一组缺省的属性。这些内置的属性包括有关主机体系结构、网络连接和资源（CPU、内存和负载）的信息。您可以通过定义收集器对象来添加其他属性。必须先创建收集器，然后才能将其添加到服务器。在服务器定义中，您可以从系统中定义的所有收集器的列表中选择收集器。
- **选择器:** 选择器对象定义如何选择供项目或步骤使用的服务器。必须先创建选择器，然后才能将其添加到项目中，以供该项目使用。在项目定义中，您可以从系统中定义的所有选择器的列表中选择选择器。

- 环境：环境对象是一组可供步骤使用的变量。在作业执行期间，会在步骤运行前在服务器主机上设置变量。环境可与服务器对象、项目对象和步骤对象关联。如果在不同环境中，将同一变量设置为不同的值，那么继承方案会确定要使用的值。必须先创建环境，然后才能将其添加到服务器、项目或步骤。在那些对象定义中，您可以从系统中定义的所有环境的列表中选择环境。

环境

环境是一组变量。可以针对服务器、项目和步骤对象指定环境。步骤运行时，在所有这些对象中设置的环境会组合起来提供变量，以供该步骤使用。请参阅第 257 页的『关于环境』。

可以在步骤运行时更改变量。（请参阅第 265 页的『在步骤执行期间更改变量值』。）变更作用域可以是步骤本地、项目本地或永久（变量在存储环境中更改）。

可以使用预定义的系统变量以及自己定义的变量。

项目

项目定义了要在过程中完成的工作。项目启动时，会作为一项作业运行。

要完成的工作包含在步骤列表中。

其他对象与项目相关：

- 选择器：选择器确定在何处启动项目。选择器称为库，如果未对项目指定选择器，那么选择器不会独立运行。必须先定义选择器，才能将其分配给作业。
- 环境：环境对象是一组可供步骤使用的变量。在作业执行期间，会在步骤运行前在服务器主机上设置变量。环境可与服务器对象、项目对象和步骤对象关联。如果同一变量设置为不同的值，那么继承方案会确定要使用的值。必须先创建环境，然后才能创建使用该环境的服务器、项目或步骤。在那些对象定义中，您可以从系统中定义的所有环境的列表中选择环境。
- 通知模板：通知模板定义如何发出有关作业活动（作业启动、作业通过、作业失败，等等）的通知。通知对象定义通过访问组通知的人员。您可以在系统设置中指定 SMTP 服务器的位置。
- 类：类对象用于对项目进行分组，以供维护。通常，类用于定期清除或归档已完成的作业。所有使用此项目运行的作业都会受到该类的影响。
- 适配器：适配器定义与外部系统（通常是源代码管理系统）的集成。我们提供了几个样本模板作为起点。您需要进一步配置这些模板，以便管理连接并执行系统中指定的操作。
- 适配器链接：适配器链接定义适配器和使用该适配器的项目间的关系。

步骤

步骤定义要完成的工作的最小单元。（请参阅第 313 页的『关于步骤』。）它的关键组件是其“命令”属性，其中包含要在所选服务器上运行的命令。

“命令”属性还可用于运行点命令。点命令是在过程引擎上运行并提供额外功能的命令。

其他对象与项目相关：

- 选择器：步骤可以具有自己的选择器。如果未指定，步骤会使用其项目的选择器。
- 环境：步骤可以具有自己的环境。由服务器、项目和步骤提供的环境会组合在一起。缺省情况下，这些环境按上述顺序应用，这样由步骤的环境定义的任何变量优先于同一变量的定义。优先顺序可以在系统设置中控制。
- 日志过滤器：在步骤的“结果”属性中，可以将其分配给一个日志过滤器。日志过滤器对象用于指定一些条件，以表明步骤是通过还是失败。可以使用 Perl 表达式来扫描日志以查找某个特定模式。通常，会使用命令的退出状态，但日志过滤器提供了一种备选方法。必须先创建日志过滤器，才能在步骤中指定该过滤器。在步骤的“结果”属性中，您可以从系统中定义的所有日志过滤器的列表中选择日志过滤器。
- 通知模板：通知模板定义如何发出有关步骤活动（步骤启动、步骤通过、步骤失败，等等）的通知。通知对象定义通过访问组通知的人员。您可以在系统设置中指定 SMTP 服务器的位置。
- Build Catalyst：步骤可运行 `rafmake` - Build Catalyst 中的一个关键实用程序。Build Catalyst 有助于加速基于 `make` 的构建。除 Build Forge 代理程序外，Build Catalyst 也必须安装在运行 `make` 构建的同一主机上。

作业

作业指正在运行的项目。作业启动后，过程引擎会将该作业放入队列然后将其运行。您可以在“作业”面板中检查其状态。作业完成后，会提供以下信息：

- 结果：您可以通过打开作业来检查所有步骤的结果。您还可以打开正在运行的作业以监控其进度。
- 步骤日志：步骤日志记录有关步骤运行方式的详尽信息，包括有关清单、环境设置以及执行结果等的信息。单击“结果”页面中的步骤结果链接可查看日志。
- 材料清单：材料清单 (BOM) 包含有关作业步骤和步骤清单的信息。可以使用 `.bom` 点命令对 BOM 的其他信息进行格式化，并将数据写入其中。可以使用 `.scan` 点命令向 BOM 添加基线和检查点。

您可以取消和重新启动作业。您可以向计划安排添加项目，以使这些项目按计划的时间运行。

第 2 章 开始使用 Build Forge®

这些主题提供对系统的简短介绍。

安装

开始之前，如果您尚未安装系统，请先进行安装。您可以在单一主机上安装该系统（管理控制台和代理程序）以简化入门过程。

1. 检查需求
 - 第 21 页的『数据库需求』
2. 请完成以下步骤:
 - a. 第 29 页的第 5 章，『规划安装』（独立、正常生产、分布式等）
 - b. 第 33 页的第 6 章，『安装前设置』（国际数据、数据库和安全性）
 - c. 第 55 页的第 7 章，『安装管理控制台』
 - d. 第 131 页的第 10 章，『安装代理程序』

访问和使用控制台

本主题提供有关使用系统菜单和选项卡的基本信息。

访问控制台

要访问控制台，请执行以下操作：

1. 打开浏览器窗口。
2. 输入控制台的 URL。

`http://host:port/`

Host 是运行控制台的主机的标准网络名或 IP 地址。

仅当未使用 HTTP 缺省端口时，*Port* 才是必需的。除非启用了 SSL，否则缺省端口为 80。缺省 HTTPS 端口为 443。仅当控制台配置为使用 SSL 时，才使用 HTTPS。

示例：

- 如果您正在安装了控制台的主机上运行浏览器，那么可使用 `http://localhost/` 或 `http://127.0.0.1/`。
 - `http://my.company.com/`
 - `http://my.company.com:81/`：由于控制台安装为使用端口 81，因此必须指定该端口号。
3. 登录。提供以下信息，然后单击**登录**：
 - 用户名
 - 密码
 - 域

缺省情况下，会将您重定向至另一个 URL 以输入您的凭证，然后在您成功登录后反向重定向。该 URL 适用于服务组件。它提供了用于认证的安全 servlet。它的用法是可配置的，如果将其关闭或将 LDAP 用于用户认证时将不会使用它。有关更多信息，请参阅第 102 页的『关于缺省登录安全性』。

如果作为 root 用户或仅在 Build Forge 中定义的其他用户登录，那么不会提示您输入域。最佳做法是将 Build Forge 配置为使用 LDAP 来认证用户。如此设置时，将显示“域”字段。它引用用于认证用户的 LDAP 域。

另请参阅第 202 页的『LDAP 集成』。

用户会话

用户会话由最初在 cookie 中生成的会话标识进行跟踪。生成后，会话标识将存储在数据库中。每个用户交互都对该会话标识进行检查。

一般情况下，可以使用同一登录打开多个浏览器窗口，但这些浏览器窗口必须位于同一主机上且使用同一类型的浏览器（FireFox 和 Internet Explorer）。

在以下任一条件下，会话将关闭：

- 同一用户帐户用于从另一台主机登录。
- 同一用户帐户用于从同一台主机登录但使用不同类型的浏览器。
- 用户关闭包含“管理控制台”会话的浏览器窗口或选项卡。
- 达到会话时间限制。时间限制在“自动注销”系统设置中设置。缺省情况下，其设置为 0，表示没有时间限制。

约束和技巧

请注意以下约束和使用技巧：

- 将显示器设置为不低于 1024 x 768 分辨率。要获得最佳效果，请使用 1280 x 1024 或更高分辨率。
- 请勿将浏览器窗口缩小到 1024 x 768 以下。
- 使用浏览器设置控制字体大小、颜色和其他辅助功能选项的功能。若要正确显示新浏览器设置，可能需要刷新页面。
- 如果字体过大或者您缩小浏览器，那么在 Microsoft Internet Explorer 中的作业 > 启动页面上将不会显示管理控制台 (MC)。

使用控制台窗口




单击右上部分的选项卡以查看应用程序。您可以单击仪表板（缺省值）、**UI 配置**、**控制台**或**报告**。系统会记住您注销时所在的选项卡，并在您下次登录时显示。

如下在**控制台**应用程序中进行导航：

- 左侧显示了一个菜单，右侧显示了主要视图面板。
- 单击菜单项可看到面板或打开面板子菜单。
- 对于比查看区域还要长的面板，请使用右上部分中的翻页控件：



- 查看列表时请使用“过滤器”字段。输入字符串并单击**过滤器**时，将更新该列表。它仅显示名称中包含该字符串的项。
- 拖动菜单的右侧边缘可调整其大小。
- 您可用来创建或编辑数据的面板通常包含以下控件：
 - 面板的上半部分可用来查看和选择项。单击一个项名称可查看其内容。单击**编辑**图标以编辑项目。
 - 面板的下半部分显示选定项的内容。
- 在某些情况下，当您选择列表中的某项时，在左侧的菜单下将显示其他信息。示例：
 1. 打开**管理 > 用户**
 2. 单击某个用户。

在左侧的菜单下将显示该用户的许可权。

对列表进行过滤和排序

您可以对列表进行快速过滤和排序。

要对列表进行过滤，请执行以下操作：

1. 在“过滤器”文本框中输入文本。该条目区分大小写。
2. 单击**过滤**。
3. 查看结果。您可以选择查看所有结果，也可以对这些结果进行页码标记。

技巧：

- 过滤器对这些条目链接到的所有列（蓝色的和带下划线的）都发挥作用。不能对其值以普通黑色文本显示的列进行过滤。
- 要只对一列进行过滤，请在过滤字符串前输入列的名称。例如，如果有一个“选择器”列，那么可以输入 **Selector: Select All**。
- 系统将保留您输入的过滤字符串。单击**过滤器**框旁边的箭头可显示您或其他人输入的过滤器列表。要删除过滤字符串，请突出显示该字符串并单击其右侧的**废纸箱**图标。
- **全部显示**选项始终显示在过滤字符串的列表中。
- 如果您输入过滤文本而没有与过滤条件相匹配的条目，那么将不显示任何条目。

要对列表进行排序，请执行以下操作：

1. 单击列标签旁边的双箭头。
2. 查看结果。您可以选择查看所有结果，也可以对这些结果进行页码标记。

技巧：

- 再次单击双箭头可按反向顺序排序。

创建 hello world 项目

本主题描述如何创建并运行简单项目来验证是否正确设置了构建系统。

以下几项提供了创建 Hello World 项目所需任务的概述。

1. 设置服务器：
 - a. 创建服务器认证（登录名和密码），以便 Rational Build Forge 能够在服务器上访问和运行命令
 - b. 创建服务器定义，以表明服务器安装了代理程序
 - c. 为 HelloWorld 项目创建要使用的选择器，以确定运行时所使用的服务器
 - d. 测试服务器连接，以确保服务器上的主控制台和代理程序能够互相通信
2. 定义项目：
 - a. 将项目命名为 HelloWorld，使用选择器指定运行该项目的服务器
 - b. 创建步骤，以运行显示 Hello World 消息的命令
3. 运行作业：
 - a. 启动项目的实例，也称为作业
 - b. 单击左侧面板中的**作业**项，以检查作业状态
4. 查看作业结果：

- a. 在已完成作业的列表中，单击作业标记以显示该作业步骤的列表（或者，在 HelloWorld 示例中，只有一个步骤）
- b. 单击该作业的步骤以显示步骤日志
- c. 在步骤日志中查找命令的 Hello World 输出

设置服务器

本主题描述如何设置“Hello World”练习中要使用的服务器以验证是否正确设置了构建系统。

创建服务器认证

Rational Build Forge 使用服务器认证（登录名和密码）以连接到安装了代理程序的服务器。

关于此任务

在将该服务器定义为要使用的 Rational Build Forge 资源之前，必须针对服务器创建服务器认证。与登录名和密码相关联的特权确定 Rational Build Forge 项目在服务器上运行时使用的特权。

过程

1. 在 Rational Build Forge 的左侧面板中，单击**服务器 > 服务器认证**。

“管理控制台”在底部显示一个空的“服务器认证详细信息”面板。

2. 单击**添加服务器认证**。
3. 在**名称**中，输入认证名称。使用登录名作为认证名称。
4. 在**登录**中，指定服务器上某个帐户的登录名。

注：如果登录名来自域用户，那么必须在该字段中包含域。例如，输入：
MYDOMAIN/joeuser。

5. 在**密码**字段中，输入密码。
6. 在**验证**字段中，重新输入密码。
7. 单击**保存服务器认证**。

结果

新的服务器认证将在顶部面板中显示，并可用于定义服务器。

创建服务器

“管理控制台”中的服务器代表您可运行项目或步骤的主机。

过程

1. 在 Rational Build Forge 的左侧面板中，单击**服务器**。
2. 单击**添加服务器**。
3. 提供服务器详细信息。

- a. **名称:** 指定服务器名称为 **HelloWorldServer**。稍后在定义选择器时，您将使用此相同名称。
- b. **路径:** 指定当服务器创建项目和作业目录时使用的目录，例如，**C:\buildforgeprojects**。系统创建构建目录时，使用此路径值作为起始点。

提示: 系统不会创建服务器路径。在构建尝试访问服务器之前，该路径必须存在。如果该路径不存在，构建将失败。

- c. **主机:** 提供运行代理程序的物理计算机的主机名。如果要将“管理控制台”计算机定义为服务器，那么使用值 **localhost**。（代理程序还必须安装在 **Management Console** 上。）

注: 请勿在主机名前包含协议。例如，请不要使用 **http://**。

- d. **认证:** 选择要用于此服务器的服务器认证。
- e. 保留其他字段的缺省值。

4. 单击**保存**。您的新服务器将在内容面板顶部的服务器列表中显示。
5. 要验证是否已正确配置了服务器，请在列表中选择服务器，然后单击**测试连接**。

若无法与服务器通信，系统将报错。如果收到错误，请确保代理程序正在服务器上运行。

创建选择器

选择器确定运行项目或步骤的服务器。

关于此任务

创建选择器，以使 **HelloWorld** 项目可以确定完成其步骤的位置。

过程

1. 在 **Rational Build Forge** 的左侧面板中，单击**服务器 > 选择器**。
2. 单击**添加选择器**。
3. 在**名称**中，输入 **HelloWorldSelector**。
4. 单击**保存**。面板底部部分进行了更改，使您能够定义确定要选择的服务器的选择器属性。
5. 使用“第 11 页的『创建服务器』”中指定的服务器名称来设置要在哪台服务器上运行。在将**名称**设置为 **BF_NAME** 并将运算符设置为 **EQ**（缺省值）后，在**值**中输入 **HelloWorldServer**。
6. 单击**保存**。

定义项目

本主题描述如何在“Hello World”练习中创建一个简单的项目，以验证是否正确设置了构建系统。

创建项目

开始之前

必须在 HelloWorld 项目定义中设置要使用的服务器和选择器。

过程

1. 选择项目。这样会在主内容面板底部显示项目详细信息面板。
2. 在名称字段中，输入 HelloWorld。
3. 将选择器设置为 HelloWorldSelector。
4. 单击保存。系统为项目显示空步骤列表和空的步骤详细信息页。定义在『创建步骤』中说明的步骤。

创建步骤

开始之前

要在 HelloWorld 项目中定义第一个步骤，必须已开始定义项目。

过程

1. 在步骤详细信息页面的名称字段中，输入 EchoHelloWorld。
2. 在命令字段中，输入将 Hello World 写到所选服务器标准输出的命令。例如，此命令可以在 Windows®、Solaris、Linux®、UNIX® 和 Apple Macintosh OS X 操作系统上运行：

```
echo Hello World
```
3. 单击保存步骤。这样该步骤会显示在步骤列表中。

运行作业


本主题描述如何在“Hello World”练习中启动作业（正在运行的项目的实例），并检查其状态，以验证是否正确设置了构建系统。

启动作业

开始之前

必须定义 HelloWorld 项目。

过程

1. 选择项目重新显示项目列表。
2. 单击 HelloWorld 项目旁边的快速启动图标 。

检查作业状态

过程

1. 单击作业，然后单击正在运行选项卡，以查看列为“正在运行”的 HelloWorld 作业。作业完成后，会转至已完成选项卡页面。

注：如果未列出 HelloWorld 作业，请继续下一步。

2. 单击**已完成**选项卡。 HelloWorld 作业会在已完成作业的列表中显示。

查看作业结果

本主题描述如何在“Hello World”练习中查看作业结果，以验证是否正确设置了构建系统。

开始之前

您必须定义并运行 HelloWorld 项目。

查看已完成的作业列表

过程

1. 单击**作业**。
2. 单击**已完成**选项卡。 在 HelloWorld 作业成功完成后，已完成的作业列表将显示该作业。

查看步骤日志

过程

1. 单击**作业**。
2. 单击**已完成**选项卡。
3. 单击 HelloWorld 作业的作业标记。 初始作业的缺省作业标记是 BUILD_1。 系统显示作业步骤的名称。在此示例中，作业仅有一个步骤 EchoHelloWorld。
4. 单击 EchoHelloWorld 步骤以检查其日志。 在大多数 Hello World 示例中，您将会在控制台窗口或弹出窗口中看到“Hello World”文本。“管理控制台”通过向目标服务器上的代理程序进程发送命令来发挥作用；然后，代理程序将这些命令的输出发回“管理控制台”，后者将输出存储在步骤日志中。 日志具有许多部分，与之相关的是最后的 EXEC 部分。您可以通过清除所有复选框，然后选中 EXEC 复选框并单击“刷新”链接，以便只显示 EXEC 部分。下面显示了命令结果。

```
80 04/19/10 11:06AM EXEC Locale set to 'English_United States.1252'
253 04/19/10 11:06AM EXEC Locale set to 'English_United States.1252'
354 04/19/10 11:06AM EXEC Performing variable expansion on command line
356 04/19/10 11:06AM EXEC start [C:\buildforgeprojects\HelloWorld\BUILD_1@mcsystem]
357 04/19/10 11:06AM EXEC Hello World
358 04/19/10 11:06AM EXEC end [C:\buildforgeprojects\HelloWorld\BUILD_1@mcsystem]
```

结果

该项目说明您已经正确配置了系统，即项目可以成功地访问服务器，在服务器上运行和生成输出。您可以将 echo 命令替换为可在目标服务器上运行的任何命令。

项目示例

特提供项目示例，以帮助您熟悉系统。

项目示例包含在一下目录中：

`<bfinstall>/samples/projects/`

缺省情况下，Build Forge 安装目录或 `<bfinstall>` 为 C:\Program Files\IBM\Build Forge（在 Windows 上）和 /opt/buildforge（在 UNIX 和 Linux 上）。

要使用项目示例，请完成以下操作：

- 使用 `bfimport` 命令将样本项目导入管理控制台。
- 运行项目。

第 3 章 IBM® Rational® Build Forge 文档声明

© Copyright IBM Corporation 2003, 2013.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，将由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节 (DBCS) 信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：International Business Machines Corporation“按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证，因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

这些信息中包含日常商业运营中使用的样本数据和报告。为了尽可能完整地说明这些内容，样本中包含个人、公司、品牌和产品的名称。这些名称全部都是虚拟的，如果与实际的企业使用的名称和地址类似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。样本程序“按现状”提供，不附有任何种类的保证。IBM 对于因使用样本程序所造成的任何损害赔偿均不负责。

凡这些样本程序的每份拷贝或其任何部分或任何演绎作品，都必须包括如下版权声明：

©（贵公司的名称）（年）。此部分代码是根据 IBM 公司的样本程序衍生出来的。© Copyright IBM Corp. 2003, 2013.

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

商标声明

请访问 www.ibm.com/legal/copytrade.shtml。

第 4 章 需求

Rational® Build Forge® 产品组件具有硬件和软件需求。

需求显示在以下区域中:

- 硬件、软件和数据库
- 安装管理器
- Licensing
- 实现 IPv6 支持的联网需求
- 集成
- 本地语言支持

硬件、软件和数据库需求

特定产品的详细系统需求站点的查询报告中提供了硬件、软件和数据库需求。您提供所使用的产品、产品修订版、版本和操作系统。所述需求针对硬件、操作系统、数据库、代理主机和 Web 客户机。

数据库需求

您必须已安装数据库才能使用 Rational Build Forge。请检查受支持数据库的 Clearing House 中的硬件和软件需求。

随产品提供的数据库权利

DB2 Workgroup V9.7 数据库可从 Passport Advantage 单独下载。您的产品许可证包括本产品的许可证。如果打算在本产品中使用 DB2 Workgroup 9.7，必须从 Passport Advantage 下载，然后在安装控制台前安装该版本。使用 Rational Build Forge 时没有平台或主机位置限制。

注: DB2 9.1.1 原先随本产品一起打包。它仍受到支持，但是不再与本产品捆绑在一起。

Installation Manager 需求

如果安装了 Installation Manager，那么它必须满足安装 Build Forge 的版本需求。

安装程序尝试查找已安装的 Installation Manager。它必须满足版本需求。

如果未找到 Installation Manager，安装程序将进行安装。

- 支持的最低版本为 1.3.0。
- 如果您要使用 Solaris 11 进行控制台安装，请使用 Installation Manager 1.4.4。

产品版本的许可需求

您使用的许可机制取决于您拥有的产品版本。请参阅第 1 页的『Build Forge 产品修订版』。

版本	许可证机制
Rational Build Forge Standard Edition	许可证服务器
Rational Build Forge Enterprise Edition	许可证服务器
Rational Build Forge Enterprise Plus Edition	许可证文件

此主题包含有关为 Build Forge 设置 Rational 许可证服务器的详细信息。可以通过 Build Forge UI 指定新的许可证服务器。不支持通过 IBM Installation Manager 管理许可证。

在安装期间指定许可证文件

本主题仅适用于使用 Build Forge Enterprise Plus Edition 的客户，这需要基于文件的许可。

要在安装期间指定许可证文件，请完成以下步骤：

1. 从 Passport Advantage 获取许可证密钥文件 `irbf_license.properties`。然后将其存放在根安装目录中。

Windows	C:\Program Files\IBM\Build Forge
UNIX/Linux	/opt/buildforge

2. 在 Installation Manager 安装程序中的“许可证服务器配置”页面上，选择**许可证文件**。
3. 单击**浏览**以在根安装目录中找到许可证密钥文件。
4. 双击 **irbf_license.properties** 文件以选择它并单击**下一步**以继续安装。

为 Build Forge 配置 Rational 许可证服务器

许可证管理员设置 Rational 许可证服务器，并为您提供您在安装期间指定的许可证服务器主机名。

本部分仅适用于需要基于服务器的许可的产品版本。

- Build Forge Standard Edition
- Build Forge Enterprise Edition

Build Forge 对许可证服务器的配置需求

为 Build Forge 配置许可证服务器之前，请复审以下需求。

- 环境中的所有管理控制台主计算机都必须能够连接到 Rational 许可证服务器主计算机。
- UNIX/Linux 或 Windows Rational 许可证服务器必须设置为自动启动并作为服务运行。
- 要安装和配置产品以使用许可证服务器，许可证管理员必须向您提供 Rational 许可证服务器的主机名和端口号。
- 与许可证服务器通信所需的 FLEXlm 客户机软件在安装管理控制台时进行安装和配置。

注：在 Build Forge V7.1.3.4 中，受支持的 Rational License Key Server 版本从 7.x 更改为 8.1.2。如果您要从低于 7.1.3.4 的版本升级，那么还必须升级许可证服务器。请查看详细的系统需求报告，以获取更多信息。

在安装期间指定许可证服务器

在安装期间，提供 Rational 许可证服务器的主机名和 TCP/IP 端口。从许可证管理员处获取此信息。

要配置基于服务器的许可证，请执行以下操作：

1. 在 Installation Manager 安装程序中，在“许可证服务器配置”页面上，选择**许可证服务器**。
2. （UNIX 和 Linux）在 **Build Forge 应作为哪个用户身份运行？**处，接受缺省用户（root 用户）或指定其他用户。如果指定了其他用户，那么仍必须以 root 用户身份启动 Build Forge，但在此安装步骤中，Build Forge 会以指定用户身份启动引擎和 Apache Tomcat。用户必须具备在 Installation Manager 的“数据库配置”页面中指定的数据库和 JDBC jar 文件的读和执行许可权。
3. 在**许可证服务器**上，为 Rational 许可证服务器提供有效主机名。

如果提供了主机名，此信息将自动添加到 Build Forge 系统设置中。

如果不知道主机名，那么在安装完成之后，在此字段中输入任何字符或值并在 UI 中更新 Build Forge 系统设置。

要点：不要将此字段保留为空。将此字段保留为空白可能会导致安装不完整且产品不可用。

4. 在**端口**中，提供许可证服务器的 TCP/IP 端口。缺省端口为 27000。
5. 单击**下一步**以继续安装。

更改管理控制台的许可证服务器

要更改管理控制台的 Rational 许可证服务器，请对产品许可证服务器配置执行以下修改。如果出现以下情况，必须执行此过程：

- 在安装期间输入了不正确的主机名或其他值。
 - 在安装期间许可证服务器主机名变灰，这表示 FLEXlm 客户机已注册了该主机的许可证服务器。
1. 该步骤仅适用于 Build Forge V7.1 至 7.1.3.3。对于更高版本，请继续步骤 2。针对您的操作系统，更改 RATIONAL_LICENSE_FILE 变量的值。

Windows	RATIONAL_LICENSE_FILE 在注册表中的 HKEY_LOCAL_MACHINE\SOFTWARE\FLEXlm License Manager 处
UNIX/Linux	RATIONAL_LICENSE_FILE 在 .flexlmrc 文件中设置，该文件位于正在运行 Build Forge 的用户的主目录中

将此变量设置为许可证服务器的正确主机名：

port@hostname | @hostname

如果许可证服务器端口为 27000（缺省值），那么不需要端口。

2. 在管理控制台 UI 中，选择**管理** → **系统**。

3. 查找许可证服务器设置，并将其值设置为新 Rational 许可证服务器的主机名。

使用以下某种格式。

`<host_name>:<port> | <host_name> | <port>:<host_name>`

如果许可证服务器端口为 27000（缺省值），那么不需要端口。

4. 单击 Web 浏览器上的刷新以验证管理控制台是否能够连接至新的许可证服务器。

获取许可证密钥并设置 Rational 许可证服务器

如果现有许可证服务器不可用，下表指示了许可证管理员为获取许可证密钥和设置 Rational 许可证服务器而要完成的一般任务。

要安装和配置 Rational 许可证服务器，请查看您的 Rational 许可证服务器软件版本的相应文档。转至 <http://www-306.ibm.com/>，选择支持与下载，然后搜索 Rational License Management Guide。

许可证管理员任务	资源
从 Rational 许可证密钥中心获取许可证密钥。	Rational License Management Guide 位于以下地址的 Rational 许可证密钥中心的入门指南： http://www-306.ibm.com/software/rational/support/licensing
验证管理控制台主机和 Rational 许可证服务器的网络连接。	Rational License Management Guide
通过以下任一方式获取所需 Rational 许可证服务器软件版本： <ul style="list-style-type: none">访问产品分发随附的软件。从以下地址的 IBM Rational Download and Licensing Center 下载软件： https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=rational	Rational License Management Guide
安装 Rational 许可证服务器。	Rational License Management Guide
按如下所述安装许可证密钥或将其导入 Rational 许可证服务器中： <ul style="list-style-type: none">在 Windows 上，使用随许多 IBM Rational 产品和 Rational License Server 软件一起安装的 IBM Rational License Key Administrator (LKAD)。在 UNIX/Linux 上，使用以下位置的 IBM Rational Download and Licensing Center 的 license_setup 脚本和许可可执行文件： https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=rational	Rational License Management Guide

许可证管理员任务	资源
<p>按如下所述启动 Rational 许可证服务器并将 Rational 许可证服务器设置为自动启动并作为服务运行:</p> <ul style="list-style-type: none">在 Windows 上, Rational License Server 软件自动设置为在计算机启动时作为服务启动。如果它不自动启动, 请参阅有关在 Windows 上自动启动许可证服务器的指示信息。在 UNIX/Linux 上, 使用提供的模板启动脚本创建启动脚本, 并针对您的安装对其进行修改。从以下位置的 IBM Rational Download and Licensing Center 获取模板: <p>https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=rational</p>	<p>Rational License Management Guide</p>

实现 IPv6 支持的联网需求

Build Forge 系统可以在使用因特网协议 V6 (IPv6) 和混合使用 IPv6-IPv4 的网络中运行, 但有某些限制。

需要正确配置计算机和网络才能支持 IPv6。网络配置问题会使在 Build Forge 系统中指定的主机名和地址无法正确得以解析。

必须手动配置 Build Forge 以支持 IPv6。要执行此操作, 请参阅第 87 页的『修改 httpd.conf』。

在 Build Forge 中输入 IP 地址时, 请使用正确的地址格式。在 Build Forge 中, 管理员或用户仅在几个位置输入主机名或 IP 地址:

- 在安装期间, 管理员指定管理控制台用于与许可证服务器 (标准版和企业版) 和数据库通信的主机名或 IP 地址。
- 在代理程序配置 (bfagent.conf 文件) 中, 可选设置限制为一个或一组与代理程序连接的特定地址。
- 用户在浏览器中输入 URL 以查看管理控制台用户界面。该 URL 包含正在运行管理控制台的服务器的主机名或 IP 地址。例如, 要访问安装在名为 BFmachine 且配置了 IPv4 和 IPv6 地址的服务器上的管理控制台, 用户可以在 Web 浏览器中输入以下任何地址:
 - `http://BFmachine/`
 - `http://localhost/`
 - `http://127.0.0.1/` (IPv4 回送地址)
 - `http://::127.0.0.1/`、`http://0:0:0:0:0:0:127.0.0.1/` (IPv4 回送地址的 IPv6 缩写), 或者只是 `http://::1/` (回送地址的 IPv6 压缩表示法)

IPv4 和 IPv6 地址在格式以及长度上都不同。

- IPv4 格式:** 长度为 32 位。地址指定为四个以小数点分隔的十进制值, 例如, 255.255.255.255

- **IPv6 格式：**长度为 128 位。地址指定为八个以冒号分隔的十六进制值，例如 FE80:0000:0000:0000:0202:B3FF:FE1E:8329。关于高阶区间的使用，存在一系列约定。还有一些缩写规则。Build Forge 不执行对 IP 地址的任何解析，这些地址将直接传递至网络。因此合法且有效的缩写应该起作用。请参阅其他参考资料，以获取有关 IPv6 地址约定的更多信息。

/etc/hosts 中避免 localhost 别名

如果您使用 IPv4 和 IPv6，那么在 UNIX 和 Linux 系统上请勿使用 localhost 作为 /etc/hosts 中的别名。使用服务器的标准域名或 IP 地址。

如果必须使用 localhost 作为 /etc/hosts 中的主机名，那么应只定义 IPv4 地址 127.0.0.1。

不支持 IPv6 的组件

不支持 IPv6 的组件必须安装在使用 IPv4 地址的主计算机上。在具有 IPv4 地址的计算机上安装以下组件：

- Rational License Server 7.0，它是包含在产品分发中的 Build Forge Standard Edition 和 Enterprise Edition 的许可证服务器。

要支持 IPv6 地址，必须安装 Rational License Server 7.1。从 IBM Rational Support 或 IBM Rational Download and Licensing Center 获取该软件。还必须配置支持 IPv6 的 FLEXlm 客户机软件版本。请参阅第 88 页的『设置 FLEXlm 客户机』。

- DB2 数据库：PHP 数据库驱动程序尚不支持 IPv6。
- MySQL 数据库：PHP 数据库驱动程序尚不支持 IPv6。
- Oracle 数据库（11g 除外）：PHP 数据库驱动程序尚不支持 IPv6。

本地语言支持需求

Build Forge 产品提供了对法语、德语、意大利语、巴西葡萄牙语、西班牙语、日语、韩语、简体中文和繁体中文的本地化支持。

本部分提供有关 Build Forge 中语言支持的信息。

管理控制台和代理程序的语言设置

本主题描述了 Build Forge 如何为管理控制台、代理程序和引擎组件配置语言设置。

管理控制台语言设置

当前用户的语言设置确定了用于在管理控制台中显示界面控件的语言。

如下为用户帐户设置管理控制台语言：

Root 用户

安装程序创建管理员用于首次登录至控制台的 root 用户帐户。

在初次登录时，root 用户的语言设置基于 Build Forge 引擎主机的操作系统语言。

其他用户的语言缺省值

对于管理员创建的用户帐户，缺省控制台语言最初是使用用户登录的 Web 浏览器的语言首选项设置的。（这是 Web 浏览器的配置设置，而不是 Build Forge 的。）

如果管理员未更改用户的语言首选项，那么将从 Web 浏览器中继承语言首选项。

如果管理员想更改用户的语言，请完成此过程。在 UI 中，单击**管理 > 用户 > 语言**，然后选择一种语言。

LDAP 创建的用户帐户始终使用为 Web 浏览器配置的语言首选项。

更改缺省用户语言

登录到控制台后，管理员可以为各个 Build Forge 用户选择不同的语言设置，方法是：从 Build Forge 窗口的左侧导航面板中选择**管理 > 用户 > 语言**。

如果用户的语言首选项是以这种方式配置的，那么管理控制台将使用为该用户选定的语言显示界面控件，而不考虑为 Web 浏览器配置的语言。

代理程序和 Build Forge 引擎语言设置

Build Forge 引擎主机的操作系统语言指定：

- Build Forge 引擎使用的语言
- Build Forge 代理程序的缺省语言

代理程序语言设置控制系统消息和作业输出的语言。

无论管理控制台语言通过 Web 浏览器设置的还是设置为 Build Forge 首选项，代理程序都会使用 Build Forge 引擎主机的操作系统语言来记录系统消息和作业输出的数据。

要避免在管理控制台界面中混合使用语言，请确保为管理控制台选择的语言与 Build Forge 引擎主机使用的语言相匹配。

语言设置

要确保显示在管理控制台中的语言与代理程序用于记录系统消息和作业输出的数据的语言相匹配，请对 Web 浏览器、管理控制台主机上的操作系统和 Build Forge 用户使用同一语言。

数据库主机的国际数据支持

要显示和处理国际数据，请针对管理控制台数据库对主计算机进行如下配置：

- 使用 Unicode UTF-8 字符集。
- 安装您打算用于显示数据的字体。

更改管理控制台中的用户语言首选项

所有管理控制台用户的缺省语言最初设置为 Web 浏览器的语言。

要更改 Build Forge 用户的语言设置，请选择**管理 > 用户 > 语言**。

管理控制台使用为用户选择的语言显示用户界面控件，但是代理程序继续使用 Build Forge 引擎主机的操作系统语言来记录系统消息和作业输出数据。

确定 UNIX/Linux 主机的语言/字符集

如果管理控制台或代理程序安装在 UNIX/Linux 主机上，请如下使用 `locale` 命令。

- 要确定操作系统当前使用的语言/字符集:

```
$ locale
```

- 要确定可用于操作系统的语言/字符集组合:

```
$ locale -a
```

- 要设置登录时的语言环境，请使用 `rc` 或概要文件脚本。

确定 Windows 主机的语言代码页

如果管理控制台或代理程序安装在 Windows 主机上，请如下使用 `chcp` 命令:

- 要确定活动代码页编号，请输入:

```
> chcp
```

- 要设置代码页，请输入对应于该语言的编号:

```
> chcp code_page
```

下表列出了 Build Forge 支持的 NLV1 语言的 Windows 字符编码:

语言	代码页
英语	1252
法语	
西班牙语	
意大利语	
德语	
葡萄牙语	
日语 Shift-JIS	932
韩国语	949
简体中文 GBK	936
繁体中文 Big5	950

第 5 章 规划安装

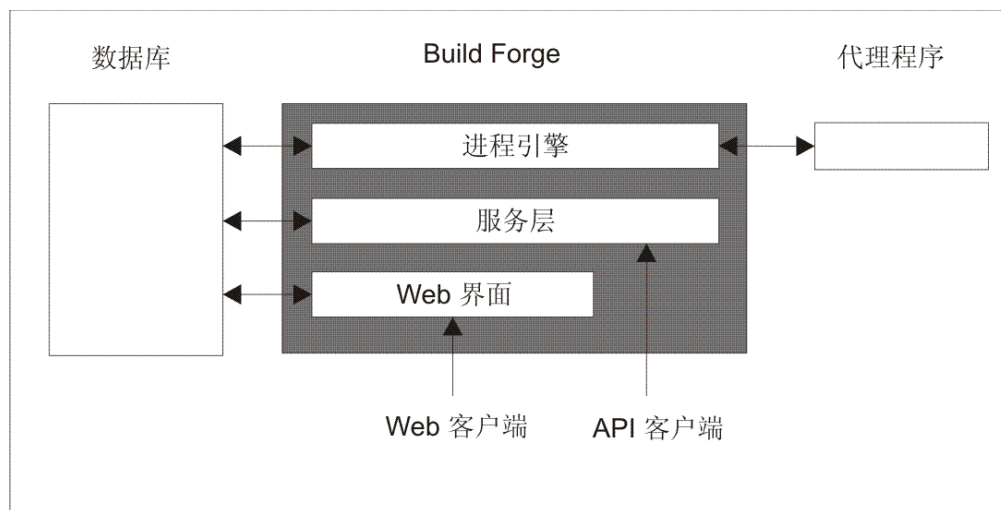
安装 Rational® Build Forge® 需要您考虑几个产品组件。

该主题描述了安装 Rational® Build Forge® 产品组件所需的规划。您可以将其用作路线图，以选择要安装和配置的对象以及安装和配置方法。

另请参阅第 507 页的第 39 章，『性能和可伸缩性』中描述的计划资源。

组件

Build Forge® 系统具有以下组件：



- **Web 客户机：**供使用 Web 浏览器访问系统的用户和管理员使用。浏览器客户机访问 **Web 接口** 组件。
- **API 客户机：**使用 Java API 或 Perl API 来访问 Build Forge 的任何程序。API 客户机直接访问 **服务层**。
- **Build Forge：**系统的集合术语。安装期间，系统显示为由 **核心产品功能** 组成：
 - **Web 接口：**也称为管理控制台或控制台。此组件由一组 PHP 模块组成。
 - **进程引擎：**也称为引擎。引擎管理作业调度和执行。
 - **服务层：**数据库抽象层，API 客户机、Web 接口和进程引擎通过它提出请求。

在简单安装中，所有三个组件都安装在相同主机上。它们可以安装在单个主机上。在这种情况下，必须修改 Build Forge 配置文件 `buildforge.conf` 组件才能通信。

- **数据库：**系统的信息存储器。数据库中存储项目定义、系统配置和用户配置。
- **代理程序：**安装在主机上的程序。代理程序必须安装在您希望管理控制台作为服务器资源使用的每个主机上。代理程序接收到执行工作（步骤）的请求并在安装该代理程序的主机上运行这些工作（步骤）。

这些组件可以通过各种方法部署，其部署范围可以是：将所有组件部署在单个主机上，到将所有组件部署在使用集群控制台和大量分布式服务器资源的系统上。

部署类型

本部分描述以下部署类型：

- 独立
- 常规生产安装
- 分层安装

独立

独立部署会将数据库、管理控制台和一个代理程序部署在单个主机上。该部署通常用于评估或开发目的。

常规

常规安装将数据库、管理控制台和代理程序置于不同的主机上。该部署通常多用于生产环境。

分层安装

在分层部署中，通常有三个常规部署：

- 开发（也称为从头开始或沙箱）：供开发人员创建新项目。也用作升级的初始区域。
- 测试：供质量保证和开发人员测试系统。
- 生产：供所有人员在日常业务中运行作业。

Build Forge 数据从一层导出，并导入至下一层。

要点：这三个部署应该是对称的。对于每个部署，使用相同的操作系统、数据库和组件。

安装方法

本部分描述适合您使用的安装方法。

- Installation Manager（交互）：在安装时提供输入
- Installation Manager（静默）：将输入保存到文件，然后在安装期间使用该文件提供输入
- 安装以使用您自己的组件：继续使用 Build Forge 需要的现有组件
- 虚拟映像上的安装：阅读有关在 VMware 上安装的指南
- IBM System z 上的安装：了解在 SUSE Linux on System z 上安装控制台所需的工具和信息

配置选项

本主题描述需要您更改产品配置的功能。

- 端口指定：了解如何使用除缺省端口 80 外的其他端口

- 第 102 页的『关于缺省登录安全性』：在安装期间提供密钥库密码。安全登录机制（缺省情况下提供）和 SSL（需要其他设置）均使用密钥库。如果需要，可以禁用安全登录。
- 密码加密：对 Build Forge 引擎、代理程序、服务层和数据库使用的密码加密
- SSL 和 HTTPS：了解在整个 Build Forge 系统中使用 SSL 所需的步骤
- 单点登录：在维护安全性的同时简化登录
- 使用控制台冗余来支持更高吞吐量：设置多个控制台来使用一个数据库
- 使用 Build Catalyst 加速基于 make 的构建：设置单系统并行构建和多系统分布式构建

与其他产品集成

- WebSphere: 使用 WebSphere Application Server 而非 Apache Tomcat
- Rational Team Concert IDE: 将 Build Forge 设置为 RTC 构建服务器并执行其他任务
- Eclipse 和 Rational Application Developer IDE: 学习如何从 Eclipse 和 Rational Application Developer IDE 访问控制台
- 使用适配器来访问源：使用提供的适配器并设计您自己的适配器来与其他产品交互

第 6 章 安装前设置

本部分描述了运行安装程序之前需要执行的安装前设置。

注： 如果已经安装了 Build Forge V7.1 或更早版本，请参阅第 179 页的第 12 章，『从先前版本升级』。

- 安装角色（必需）。必须以管理级别用户的身份安装软件。
- 国际数据支持（必需）。安装 Build Forge 之前，必须将 Web 浏览器和数据库设置为支持 UTF-8 字符集。
- 设置数据库（必需）。设置数据库以用于 Build Forge。通常需要创建数据库对象（包括数据库用户和密码），在 Build Forge 运行的主机上安装数据库客户机，并收集在 Build Forge 安装期间出现的提示信息。
- 安全性（可选）。如果打算使用 SSL/HTTPS，那么需要提供证书，或计划让 Build Forge 安装自签名证书。

国际数据设置

必须设置 Build Forge 组件以支持国际数据。

过程

1. 配置 Web 浏览器。
 - a. 设置语言。
 - b. 请确保安装了正确的字体。
2. 将代理程序主机配置为使用 UTF-8 字符集。

在 Windows 上，使用 chcp 命令来检查代码页：

```
> chcp
```

在 UNIX 或 Linux 上，使用以下命令来检查语言环境和字符集：

```
locale
```

您应当看到指示语言和字符集的值。以下示例来自使用美国英语和 UTF-8 字符集的 Solaris 系统：

```
LANG=en_US.UTF-8  
LC_CTYPE="en_US.UTF-8"
```

3. 配置数据库以使用支持国际数据的 UTF-8 字符集和字体。
 - **DB2:**
 - a. 设置代码集和地域。示例：CREATE DATABASE USING CODESET UTF-8 TERRITORY US（或在控制中心中选择相应的代码集和地域）。
 - b. 在管理控制台计算机上将 DB2CODEPAGE 环境变量设为 1208。

在 Windows 上，请使用以下命令：

```
set DB2CODEPAGE=1208
```

在 UNIX 或 Linux 上, 请使用以下命令:

```
export DB2CODEPAE=1208
```

如果现有数据库中含有数据, 而您需要将它迁移到 UTF-8, 那么以下文档可能对您有所帮助: <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/t0024033.htm>

- **Microsoft SQL Server** (仅用于与 Windows 上的管理控制台一起使用): 不支持国际数据。Microsoft SQL Server 使用 UCS-2 以存储 Unicode 数据且不支持 UTF-8。
- **MySQL**: 设置服务器字符集和整理。如果您的 MySQL 安装当前不支持国际数据, 那么可以从代码源重新编译它, 并使用 `./configure --with-charset=utf8 --with-collation=utf8_bin`。如果未配置此支持, 那么 Build Forge 引擎不会启动。
- **Oracle**: 安装时, 请在实例上将字符集设置为 **UTF8 - Unicode 3.0**。在“数据库配置助手”中, 此设置是在“字符集”选项卡上的“初始化参数”步骤中进行的。如果使用命令行, 请将字符集设置为 **AL32UTF8**。

4. (可选) 配置 Build Catalyst 功能的消息。

在 Microsoft® Windows® 平台上, Build Catalyst 消息自动显示为正确的译文。

要在 UNIX 和 Linux 平台上查看这些消息, 请将 NLSPATH 环境变量设置为以下值:

```
/opt/rational/buildforge/buildcatalyst/lib/nls/%L/utf8/%N.cat:$NLSPATH
```

其中:

- %L 代替 LANG 环境变量的值
- %N 代替传递到 catopen(3C) 的名称参数的值

或者, 您可以显式地指定译文。以下值针对于 ja_JP 译文:

```
/opt/rational/buildforge/buildcatalyst/lib/nls/ja_JP/utf8/%N.cat:$NLSPATH
```

数据库备份 (必需)

继续全新安装、重新安装或升级前, **备份数据库**至关重要。

如果在安装期间出现问题, 数据库备份使您能够将数据库恢复至尝试安装前的最后状态。数据库备份是所提供回滚的唯一方式。

数据库设置

本部分包含每个受支持数据库的设置指示信息。

数据库可以与管理控制台驻留在同一台主机上, 也可以驻留在不同的主机上。

注: Rational Build Forge 7.1 发行版需要使用空数据库来安装。如果数据库不为空, 安装将失败。

根据您的数据库, 其设置需要您完成以下某些或所有任务:

- 创建数据库对象以供 Build Forge 使用 (数据库、数据库用户)。

- 安装必需的客户机软件以供 Build Forge 使用。
- 确定在安装期间需要的其他信息。通常该信息是 Apache Tomca 应用程序服务器将使用的 JDBC 驱动程序的位置。
- 配置 UTF-8 字符集支持和相应的整理。通常在创建数据库时指定国际数据的支持；不能在数据库创建之后再配置国际数据支持。必须安装打算用于显示数据的字体。**Build Forge 需要使用国际数据（UTF-8 字符集）。**

DB2 Express 安装

使用本部分来设置 DB2 Express 以用于 Build Forge。

DB2 Express 的数据库对象

运行安装程序之前，必须先定义以下对象：

- 用户：必须在 Windows 中的数据库主机上定义用户帐户。该帐户应该专门由 Build Forge 用来访问数据库。
- 数据库对象：在 DB2 Express 实例中创建以下对象。
 - 数据库：创建数据库并将其命名为 BUILD。
 - 模式：创建数据库模式，并将其命名为 BUILD。您创建的用户应该具有对该模式的权限。

在安装期间所需的 DB2 Express 信息

安装期间，对于数据库配置面板上的安装 **DB2 Express**，请选择“否”，因为正在使用现有的 DB2 Express 实例。将显示以下字段。

在数据库配置面板中需要您提供其他信息：

数据库配置

- 数据库类型：选择 **DB2**。
- 是否要在安装时填充此数据库？：如果希望 Build Forge 在安装时自动进行填充，请选择“是”，而选择“否”就要在安装后手动填充。
- 数据库主机：输入 DB2 Express 安装所在的主机名。
- 数据库名称：输入您创建以供 Build Forge 使用的数据库的名称。
- 数据库模式名称：输入您创建以供 Build Forge 使用的模式的名称。
- 数据库端口：缺省值为 50000。如果使用的端口号不同，请输入该端口号。
- 数据库用户名：供 Build Forge 在访问数据库时使用的用户名。输入您创建以供 Build Forge 使用的 Windows 帐户。
- 密码：该数据库用户名的密码。输入该用户帐户的密码。
- 确认密码：输入该用户帐户的密码。

测试数据库配置

- DB2 客户机库的路径：缺省情况下，这些库在您安装 DB2 Express 的驱动器上的 C:/Program Files/IBM/SQLLIB/java 中。Build Forge 需要使用以下文件：db2jcc.jar 和 db2jcc_license_cu.jar。
- JDBC 驱动程序位置：缺省情况下，这些库在您安装 DB2 Express 的驱动器上的 C:/Program Files/IBM/SQLLIB/java 中。JDBC 驱动程序由 Apache Tomcat 用来访问数据库。

注：对是否要在安装时填充此数据库？选择“否”时，测试仅检查 JDBC 驱动程序的位置是否正确。

DB2 设置

使用此过程可设置对 DB2 的支持。

针对 DB2 的 Red Hat 和 SuSe Linux 6 需求 开始之前

在安装 Rational Build Forge 之前，需要对 Red Hat Linux 6 上的 DB2 进行额外设置。

过程

- 安装兼容性共享库。必须安装以下软件包：
 - libgcc-4.4.4-13.i686
 - glibc.i686，包含 nss-softokn-freebl.i686。
 - gtk2.i686，包含许多其他文件：atk.i686、audit-libs.i686、avahi-libs.i686、cairo.i686、cups-libs.i686、dbus-libs.i686、expat.i686、fontconfig.i686、freetype.i686、gamin.i686、glib2.i686、gnutls.i686、和 zlib.i686
 - libXtst.i686
 - libuuid.i686

- 所有文件按字母顺序排列的列表如下所示。请注意，文件扩展名针对 Intel 平台。对于 System z 平台，请使用 .s390 扩展名。

```
atk-1.28.0-2.el6.i686
audit-libs-2.0.4-1.el6.i686
avahi-libs-0.6.25-8.el6.i686
cairo-1.8.8-3.1.el6.i686
cups-libs-1.4.2-35.el6_0.1.i686
dbus-libs-1.2.24-4.el6_0.i686
expat-2.0.1-9.1.el6.i686
fontconfig-2.8.0-3.el6.i686
freetype-2.3.11-6.el6_0.2.i686
gamin-0.1.10-9.el6.i686
glib2-2.22.5-5.el6.i686
glibc-2.12-1.7.el6_0.5.i686
gnutls-2.8.5-4.el6.i686
gtk2-2.18.9-4.el6.i686
jasper-libs-1.900.1-15.el6.i686
keyutils-libs-1.4-1.el6.i686
krb5-libs-1.8.2-3.el6_0.7.i686
libcom_err-1.41.12-3.el6.i686
**libgcc-4.4.4-13.el6.i686
libgcrypt-1.4.5-3.el6.i686
libgpg-error-1.7-3.el6.i686
libjpeg-6b-46.el6.i686
libpng-1.2.44-1.el6.i686
libselinux-2.0.94-2.el6.i686
libstdc++-4.4.4-13.el6.i686
libtasn1-2.3-3.el6.i686
libthai-0.1.12-3.el6.i686
libtiff-3.9.4-1.el6_0.3.i686
libuuid.i686
libX11-1.3-2.el6.i686
libXau-1.0.5-1.el6.i686
```

```
libxcb-1.5-1.el6.i686
libXcomposite-0.4.1-2.el6.i686
libXcursor-1.1.10-2.el6.i686
libXdamage-1.1.2-1.el6.i686
libXext-1.1-3.el6.i686
libXfixes-4.0.4-1.el6.i686
libXft-2.1.13-4.1.el6.i686
libXi-1.3-3.el6.i686
libXinerama-1.1-1.el6.i686
libXrandr-1.3.0-4.el6.i686
libXrender-0.9.5-1.el6.i686
libXtst-1.0.99.2-3.el6.i686
nss-softokn-freebl-3.12.8-1.el6_0.i686
pango-1.28.1-3.el6_0.5.i686
pixman-0.18.4-1.el6_0.1.i686
zlib-1.2.3-25.el6.i686
```

- 如果要在 SuSe Linux 64 位系统上运行，那么还必须安装 pam.i686 软件包。

DB2 的数据库对象

过程

1. 在操作系统中，创建一个用户。管理控制台将使用该用户的名称登录数据库。示例：用户名 **BUILD**，密码 **BUILD**。

在 DB2 中执行其余步骤。要点：您必须对数据库名称和模式名称全部使用大写。

2. 创建名为 **BUILD** 的数据库
3. 创建名为 **BUILD** 的模式。

结果

注：Build Forge 使用用户的模式访问数据库。

DB2 SQL 命令脚本样本

在 DB2 命令行处理器中使用以下命令来创建模式和数据库。

```
// 创建数据库和模式
db2 "CREATE DATABASE BUILD USING CODESET UTF-8
    TERRITORY US PAGESIZE 16 K AUTOCONFIGURE
    USING MEM_PERCENT 20 APPLY DB ONLY"
db2 "CONNECT TO BUILD"
db2 "CREATE SCHEMA BUILD"
db2 "CONNECT RESET"

// 授予用户许可权
db2 "CONNECT TO BUILD"
db2 "GRANT CREATEIN,DROPIN,ALTERIN ON SCHEMA BUILD
    TO USER BUILD WITH GRANT OPTION"
db2 "GRANT DBADM,CREATETAB,BINDADD,CONNECT,
    CREATE_NOT_FENCED_ROUTINE,IMPLICIT_SCHEMA, LOAD,
    CREATE_EXTERNAL_ROUTINE,QUIESCE_CONNECT
    ON DATABASE TO USER BUILD"
db2 "commit work"
db2 "CONNECT RESET"db2 "terminate"
```

调整针对 DB2 建议的参数

关于此任务

使用现有的 DB2 数据库设置 DB2 调整参数可以提高 Build Forge 系统的性能和可伸缩性。

注：如果在已安装和运行 Build Forge 后更改这些参数，请在更改之前停止 Build Forge。重新启动 DB2 后，请重新启动 Build Forge。

过程

1. 设置调整参数。运行以下 DB2 命令：

```
db2set DB2_EVALUNCOMMITTED=ON
db2set DB2_SKIPDELETED=ON
db2set DB2_SKIPINSERTED=ON
```

请参阅 DB2 文档了解关于这些设置的效果的更多信息。

2. 重新启动 DB2。必须执行此步骤才能使参数生效。请先确保数据库上没有正在运行的会话。

```
db2stop force
db2start
```

DB2 客户机驱动程序

在安装该产品前，必须先主机上安装 IBM DB2 数据库客户机驱动程序。在 UNIX 或 Linux 上，请使用 32 位驱动程序。

过程

1. 安装 DB2 Data Server Client 以提供驱动程序。
2. 安装 DB2 客户机后重新启动计算机。否则产品安装失败。

在安装期间所需的 DB2 信息

关于此任务

在安装期间，将要求您在数据库配置面板中提供以下信息：

数据库配置

- **数据库主机：**安装 DB2 的主机。
- **数据库端口：**对于 DB2，Build Forge 将在此字段中放入缺省端口 50000。如果使用其他端口，请准备输入该端口号。
- **数据库名称：**供 Build Forge 使用的数据库的名称。您在先前的设置步骤中创建了该数据库。
- **数据库模式名称：**供 Build Forge 使用的模式的名称。
- **数据库用户名：**供 Build Forge 在访问数据库时使用的用户名。您在先前的配置步骤中创建了该用户。
- **密码：**该数据库用户名的密码。

测试数据库配置

- **指向 DB2 客户机库的路径** - DB2 客户机库所在的目录。

要点：在 UNIX 或 Linux 上安装 Build Forge 时，该目录必须是包含 32 位客户机驱动程序库的目录。

- **JDBC 驱动程序位置** - JDBC 驱动程序所在的目录。该驱动程序由 Apache Tomcat 用来访问数据库。典型位置：
 - Windows: <db2install>/IBM/SQLLIB/java
 - UNIX 或 Linux: 请参阅对应于您的系统的文档。

将显示以下信息:

- 必需的驱动程序 **JAR** 文件 - 显示必需的驱动程序 JAR 文件。对于 DB2, 这些文件是 db2jcc.jar 和 db2jcc_license_cu.jar。
- 必需的 **JDBC** 驱动程序类 - 显示必需的 JDBC 驱动程序类。对于 DB2, 这是 com.ibm.db2.jcc.DB2Driver。

Microsoft SQL Server 设置

使用这些过程可安装并配置从基于 Windows® 的管理控制台对 Microsoft SQL server 数据库的访问权。

Microsoft SQL Server 的数据库对象

关于此任务

在这些步骤中, 您将创建作为数据库所有者的用户, 以及 Build Forge 要使用的数据库。如果需要, 您还可以创建模式以供 Build Forge 使用。

过程

1. 创建充当数据库所有者的用户。 Build Forge 使用该用户名登录至数据库。除非必须使用其他名称, 否则使用 **build**。该用户必须具有完整许可权。
 - a. 打开 SQL Server Management Studio。
 - b. 在“对象资源管理器”(左面板)中打开数据库服务器。
 - c. 右键单击**安全**文件夹并选择**新建 > 登录**。
 - d. 在“登录 - 新建”对话框中, 指定登录名并选择选项(如下所示)。重要信息: 取消选中“用户必须在下次登录时更改密码”。
 - 选择 **SQL Server 认证**并提供密码。
 - 取消选中**强制密码到期**
 - 取消选中**用户必须在下次登录时更改密码**
2. 创建数据库。 必须使用混合方式的认证。以下交互式示例和脚本示例假定您将使用 DBO 的缺省模式。
 - a. 打开 SQL Server Management Studio。
 - b. 在“对象资源管理器”(左面板)中打开数据库服务器。
 - c. 右键单击**数据库**文件夹, 并选择**新建数据库**。
 - d. 在“新建数据库”对话框中, 指定数据库的参数:
 - 指定数据库名称。除非必须使用其他名称, 否则使用 **build**。该名称区分大小写。数据和日志文件的名称将在“数据库文件”框中自动更新。
 - 指定数据库所有者。
 - 单击字段右边的 **[..]** 控件
 - 在“选择数据库所有者”对话框中, 单击**浏览**。
 - 选中所创建的用户名, 然后单击**确定**。
 - 在“选择数据库所有者”对话框中单击**确定**。
 - 指定“数据库文件”参数。在“数据库文件”表中, 执行以下操作:
 - 对于这两个文件: 将“初始大小”设置为 500 (以 MB 为单位)

- 对于这两个文件：请设置“自动增大”。在“自动增大”列中，单击 **[..]** 控件以打开对话框。选中**启用自动增大**框，将增大值设置为 500 MB，选择**不受限增大**，然后单击**确定**。

或者，也可以使用以下脚本来创建数据库。

```
CREATE DATABASE [build] ON PRIMARY
( NAME = N'build',
  FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL.2\MSSQL\DATA\build.mdf' ,
  SIZE = 2048KB , FILEGROWTH = 1024KB )
LOG ON
( NAME = N'build_log',
  FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL.2\MSSQL\DATA\build_log.ldf' ,
  SIZE = 1024KB , FILEGROWTH = 10%)
GO
EXEC dbo.sp_dbcmtlevel @dbname=N'build', @new_cmptlevel=90
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [build].[dbo].[sp_fulltext_database] @action = 'disable'
end
GO
ALTER DATABASE [build] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [build] SET ANSI_NULLS OFF
GO
ALTER DATABASE [build] SET ANSI_PADDING OFF
GO
ALTER DATABASE [build] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [build] SET ARITHABORT OFF
GO
ALTER DATABASE [build] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [build] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [build] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [build] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [build] SET CURSOR_CLOSE_ON_COMMIT ON
GO
ALTER DATABASE [build] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [build] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [build] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [build] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [build] SET READ_COMMITTED_SNAPSHOT ON
GO
ALTER DATABASE [build] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [build] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [build] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [build] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [build] SET READ_WRITE
GO
ALTER DATABASE [build] SET RECOVERY FULL
GO
ALTER DATABASE [build] SET MULTI_USER
GO
ALTER DATABASE [build] SET PAGE_VERIFY CHECKSUM
```

```
GO
USE [build]
GO
IF NOT EXISTS (SELECT name FROM sys.filegroups WHERE is_default=1 AND name = N'PRIMARY') \
    ALTER DATABASE [build] MODIFY FILEGROUP [PRIMARY] DEFAULT
GO
```

注: READ_COMMITTED_SNAPSHOT 属性必须设置为 ON。

要对 READ_COMMITTED_SNAPSHOT 进行测试, 请运行以下 SQL 语句。

```
SELECT is_read_committed_snapshot_on FROM sys.databases WHERE name='build'
```

如果未使用 build, 请替换您的数据库名称。此语句必须返回“1”。

如果语句返回“0”, 请运行以下 SQL 语句。

```
ALTER DATABASE [build] SET READ_COMMITTED_SNAPSHOT ON
```

3. 设置用户的缺省数据库。

- a. 打开 SQL Server Management Studio。
 - b. 在“对象资源管理器”(左面板)中打开数据库服务器。
 - c. 在“对象资源管理器”中, 打开安全性 > 登录。
 - d. 右键单击您创建的用户, 并选择属性。
 - e. 在“常规”页面中, 选择缺省数据库。选择所创建的数据库。
 - f. 单击确定。
4. 根据需要创建模式。 如果可以使用缺省模式 **DBO**, 那么无需另行创建模式。建议使用缺省模式。如果您需要使用其他模式, 请在 Build Forge 数据库中创建该模式, 并为您创建的数据库用户分配所有者。以下是创建模式 bf_schema 并设置 build 作为所有者的示例:

```
USE build;
GO
CREATE SCHEMA bf_schema AUTHORIZATION build;
GO
```

Microsoft SQL Server 上的 TCP/IP 设置

关于此任务

必须在 Microsoft SQL Server 上启用 TCP/IP, 才能将其用于管理控制台。

要在 MS SQL Server 上启用 TCP/IP, 请执行以下操作:

过程

1. 打开 MSSQLServer 配置管理器。
2. 在 **SQL Server 网络配置**下, 单击 **MSSQLSERVER** 的协议。
3. 右键单击 **TCP/IP**, 然后从菜单中选择启用。

Microsoft SQL Server 客户机和 JDBC 驱动程序安装

关于此任务

您需要在管理控制台主机上安装 Microsoft SQL Server 客户机和 JDBC 驱动程序。管理控制台使用它们来访问数据库。

过程

1. 安装 SQL Native Client。 V2005.90.4035.00 是必需的。它包含在 Microsoft SQL Server 的 service pack 3 中。
2. 安装 JDBC 驱动程序。

- 对于版本 2005: JDBC 库 V1.2 是必需的。从 Microsoft 的以下位置进行下载:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=C47053EB-3B64-4794-950D-81E1EC91C1BA&displaylang=en>

完成 Microsoft 安装过程后, JAR 的位置如下:

/sqljdbc_1.2/enu/sqljdbc.jar

- 对于版本 2008: JDBC 库 V3.0 是必需的:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=%20a737000d-68d0-4531-b65d-da0f2a735707&displaylang=en>

针对 Microsoft SQL Server 的 ODBC 数据源设置

关于此任务

通过以下步骤可创建供 ODBC 数据源, 该数据源供基于 Windows 的管理控制台用于访问您在 Microsoft SQL Server 中创建的数据库。无论您正在运行 32 位的 Windows 系统还是 64 位的 Windows 系统, 都必须创建 32 位的 ODBC 数据源。

过程

1. 启动数据源管理器。
 - 对于 32 位 Windows 系统: 从 Windows 开始菜单中选择 **设置 > 控制面板 > 管理工具 > 数据源**。
 - 对于 64 位 Windows 系统: 双击文件 C:\Windows\SysWOW64\odcbad32.exe。请注意, 这样会打开“数据源管理器”对话框, 以允许您在 64 位系统上创建所需的 32 位 ODBC 数据源。

将出现 **ODBC 数据源管理器** 对话框。

2. 在系统 **DSN** 选项卡上, 单击 **添加**。 将出现驱动程序列表。
3. 从驱动程序列表中选择 **MS SQL Server**, 然后单击 **完成**。 在出现的 **ODBC 设置** 对话框中, 输入以下信息:
 - **数据源名称:** 此数据源的名称, 必须与数据库名称相同, 不能和与该数据库关联的模式名称相同。
 - **描述:** 对此数据源的描述。
 - **服务器名称:** 安装 MS SQL Server 数据库的主机的主机名。
 - **数据库名称:** 以上创建的数据库名称。

单击 **确定** 关闭 **ODBC 设置**, 然后单击 **确定** 关闭 **ODBC 数据源管理器**。这就创建了数据源。

结果

请记录以下信息。安装管理控制台时, 安装程序将请求提供这些信息。

- **数据源名称:** 创建 ODBC 数据源时由您指定
- **数据源类型:** SQL Server
- **用户名:** 为数据库创建的用户名 (例如 **build**)
- **密码:** 用户名的密码 (例如 **build**)

要点: 请对“数据库名称”和“数据源名称”使用相同的值。由于 JDBC 驱动程序的局限，因此需要此约束。如果这两者不同，那么“快速报告”报告功能以及 Java 和 Perl 的服务层 API 将不起作用。

安装期间需要的 Microsoft SQL Server 信息

关于此任务

在安装期间，将要求您在**数据库配置**面板中提供以下信息：

数据库配置

- **数据库主机:** 安装 SQL Server 的主机。
- **数据库端口:** 对于 SQL Server, Build Forge 将缺省端口 1521 置于此字段中。如果使用其他端口，请准备输入该端口号。
- **数据库名称:** 供 Build Forge 使用的数据库的名称。您在先前的设置步骤中创建了该数据库。
- **数据库模式名称:** 供 Build Forge 使用的模式的名称，可以是缺省值 DBO（建议值），也可以是您在先前的设置步骤中创建的模式名称。
- **ODBC 数据源名称:** ODBC 数据源的名称。
- **数据库用户名:** 供 Build Forge 在访问数据库时使用的用户名。您在先前的配置步骤中创建了该用户。
- **密码:** 该数据库用户名的密码。

测试数据库配置

- **SQL Server 客户机库的路径** - SQL Server 客户机库所在的目录。

要点: Microsoft SQL Server 仅在 Windows 平台上受支持。

- **JDBC 驱动程序位置** - JDBC 驱动程序所在的目录。该驱动程序由 Apache Tomcat 用来访问数据库。
- 2005 版本: 使用 2005 版本随附的驱动程序。
- 2008 版本: 使用版本 3.0 驱动程序。

将显示以下信息：

- **必需的驱动程序 JAR 文件** - 显示必需的驱动程序 JAR 文件。对于 SQL Server, 它是 /sqljdbc_1.2/enu/sqljdbc.jar。
- **必需的 JDBC 驱动程序类** - 显示必需的 JDBC 驱动程序类。对于 SQL Server, 它是 com.mysql.jdbc.Driver。

MySQL 设置

使用此过程可安装并配置对 MySQL 的支持。

开始之前

安装并配置以下各项。请使用以下各节中的指示信息。

MySQL 的 Red Hat Linux 4 需求

开始之前

在配置 Rational Build Forge 之前，需要对 Red Hat Linux 4 上的 MySQL 进行其他设置。

过程

- 安装 ncurses-libs.i686。它必须存在，这样才能在下一步中安装文件。
- 安装兼容性共享库。需要安装程序包或 RPM（但不要两者均安装）。

```
MySQL-client-community-5.0.51a-0.rhel5.i386
MySQL-shared-compat-5.0.51a-0.rhel5.i386
```
- 重命名启动脚本。需要对启动脚本进行重命名，从而使它在启动过程中比缺省情况下更早的装入 MySQL。以下示例应该可将其足够提前。

```
mv /etc/rc3.d/S99mysql /etc/rc3.d/S50mysql
```

MySQL 的数据库对象

过程

1. 创建名为 **build** 的空数据库。
2. 创建与其相关联的用户（用户名为 **build**，密码为 **build**）。

结果

可以使用以下命令来创建数据库 **build** 并创建以密码 **build** 作为“识别条件”（identified by）的用户 **build@localhost**:

```
mysql -u root
mysql> create database build;
mysql> grant all on build.* to build@localhost
-> identified by "build";
```

MySQL 库和客户机驱动程序

开始之前

安装 Build Forge 之前，必须在 Build Forge 主机上安装 MySQL 本机客户机驱动程序。在 UNIX 或 Linux 上，请使用 32 位驱动程序。

如果您要使用 MySQL 5.1，那么必须使用 MySQL 5.0 客户机。

如果您要将 MySQL 安装到 Build Forge 主机上，请安装以下项:

```
MySQL database server
MySQL-client-community-5.0.51a-0.rhel5.i386 - client drivers
MySQL-shared-compat-5.0.51a-0.rhel5.i386 - required compatibility libraries
Connector/J JDBC driver
```

如果您要在不同的主机上安装 MySQL，请在 Build Forge 主机上安装以下项:

```
MySQL-client-community-5.0.51a-0.rhel5.i386 - client drivers
MySQL-shared-compat-5.0.51a-0.rhel5.i386 - required compatibility libraries
Connector/J JDBC driver
```

仅限于 **AIX** 系统:

您必须安装 Connector/J 驱动程序和 5.0-51a 客户机驱动程序。它们与 AIX V6.1 兼容。您可以在 MySQL web 站点的 Archives 区域找到它们。

安装程序尝试重新打包 MySQL 客户机共享库。在 `mysql.com` 上发布文件时进行的文件打包无法由 Build Forge 使用。运行安装程序的用户需要对安装了客户机驱动程序文件的目录（通常为 `/opt/mysql`）具有写访问权。安装程序将替换文件 `libmysqlclient.a` 和 `libmysqlclient.so.15`。如果安装程序无法将这些文件重新打包，那么会在安装期间通知您并继续安装。在这种情况下，您必须在运行 Build Forge 之前手动将文件重新打包。

注：如果您在运行使用此 MySQL 客户机的其他应用程序，那么您可能希望在单独目录中手动将文件重新打包。

要手动将文件打包，请执行以下操作。

1. 从安装了 MySQL 的目录中，创建一个新目录 `libbf`：

```
root@myaix:/opt/mysql/> mkdir libbf
```

2. 将 `lib/libmysqlclient.so.15` 文件复制到新目录：

```
root@myaix:/opt/mysql/> cp lib/libmysqlclient.so.15 libbf
```

3. 切换到新目录并构建新的归档文件：

```
root@myaix:/opt/mysql/> cd libbf
```

```
root@myaix:/opt/mysql/libbf/> ar -q libmysqlclient.a libmysqlclient.so.15
```

此目录现在可用作安装过程的客户机库路径（如果安装过程尚未运行）。如果使用了静默方法或命令行方法之一，那么请更新 `$BFROOT/rc/buildforge` 中的 `LIBPATH` 以使用此路径。

MySQL 配置

过程

1. 将数据库最大连接数增加到 200。如下编辑 `<mysql-installdir>/my.ini` (Windows) 或 `/etc/my.cnf` (UNIX/Linux) 中的 `[mysqld]` 部分：

```
max_connections=200
```

该值应至少等于**最大控制台进程数和运行队列大小**系统设置（在管理控制台的管理 → 系统页面中）的总计。

2. 确认您的 MySQL 实例正在使用 InnoDB 存储引擎。Rational Build Forge 数据库使用外键约束。MySQL MyISAM 存储引擎不会检查外键约束。如果使用 MyISAM，那么在创建新的数据库对象（尤其是通过 API）时会出现问题。

在安装期间所需的 MySQL 信息

关于此任务

在安装期间，将要求您在**数据库配置**面板中提供以下信息：

数据库配置

- **数据库主机：**安装 MySQL 的主机。
- **数据库端口：**对于 MySQL，Build Forge 将缺省端口 3306 置于此字段中。如果使用其他端口，请准备输入该端口号。

- **数据库名称:** 供 Build Forge 使用的数据库的名称。您在先前的设置步骤中创建了该数据库。
- **数据库用户名:** 供 Build Forge 在访问数据库时使用的用户名。您在先前的配置步骤中创建了该用户。
- **密码:** 该数据库用户名的密码。

测试数据库配置

- **MySQL 客户机库的路径** - MySQL 客户机库所在的目录。

要点: 在 UNIX 或 Linux 上安装 Build Forge 时, 该目录必须是包含 32 位客户机驱动程序库的目录。

- **JDBC 驱动程序位置** - JDBC 驱动程序所在的目录。该驱动程序由 Apache Tomcat 用来访问数据库。

将显示以下信息:

- **必需的驱动程序 JAR 文件** - 显示必需的驱动程序 JAR 文件。对于 MySQL, 该文件为 mysql-connector-java-5.*-bin.jar。
- **必需的 JDBC 驱动程序类** - 显示必需的 JDBC 驱动程序类。对于 MySQL, 该类为 com.mysql.jdbc.Driver。

MySQL 故障诊断

关于此任务

本任务描述针对以下情况的变通方法: 使用 MySQL 数据库的控制台失败并带有消息“Database extension 'mysql' not found.”在 Red Hat Linux V6.x 上使用以下文件时将发生该问题:

/opt/buildforge/server/apache/lib/php/extensions/no-debug-non-zts-20090626

过程

1. 检查缺失的库和许可权。使用 ldd 命令。示例响应:

```
ldd: warning: you do not have execution permission for './mysql.so'
linux-gate.so.1 => (0x00413000)
libmysqlclient.so.15 => not found
.....
libssl.so.6 => not found
libcrypto.so.6 => not found
```

2. 请执行以下其中一项任务以便能够找到 libmysqlclient.so.15 库。

- 将 libmysqlclient.so.15 的完整路径附加到 LD_LIBRARY_PATH。
- 将 libmysqlclient.so.15 复制到 /usr/lib。

3. 添加较旧库的链接。您需要手动添加较旧库的链接。示例:

```
ln -s /usr/lib/libcrypto.so.1.0.0 /usr/lib/libcrypto.so.6
ln -s /usr/lib/libssl.so.1.0.0 /usr/lib/libssl.so.6
```

4. 重新启动控制台。重新启动是必需的。

Oracle 设置

使用此过程可设置对 Oracle 数据库的支持。

开始之前

安装或配置以下项。使用以下各节中的指示信息。

Red Hat Linux 6 需求（对于 Oracle 10）

开始之前

在安装 Rational Build Forge 之前，需要对 Red Hat Linux 6 上的 Oracle 10 进行其他设置。

过程

安装兼容性共享库。必须安装以下软件包：

- `compat-libs-5.2-1.i386.rpm`
- `compat-libstdc++-33-3.2.3-68.i686`

Oracle 的数据库对象

关于此任务

在 Oracle 主机上创建一个本地用户：用户名为 **build**，密码为 **build**。

- 添加相应授权，包括 `CREATE SESSION` 和 `CREATE TABLE`。
- 在 `DEFAULT TABLESPACE` 中添加相应的 `QUOTA` 大小，从而为系统提供足够的数据存储空间。

```
create user build
  identified by password
  default tablespace users
  quota unlimited on users;
```

```
grant create session, create table
to build;
```

要点： 在安装时，将对数据库名称和 Oracle SID 使用相同值。由于 JDBC 驱动程序的局限，因此需要此约束。如果这两者不同，那么“快速报告”报告功能以及 Java 和 Perl 的服务层 API 将不起作用。

调整针对 Oracle 建议的参数

关于此任务

某些 Oracle 参数必须更改其缺省值，这样 Build Forge 才能正常运行。

注： 如果在已安装和运行 Build Forge 后更改这些参数，请在更改之前停止 Build Forge。重新启动数据库服务器之后，再重新启动 Build Forge。

过程

1. 设置调整参数。运行以下命令：

```
ALTER SYSTEM SET open_cursors=1000 SCOPE=BOTH
ALTER SYSTEM SET processes=500 SCOPE=BOTH
```

注： 如果在运行 `ALTER SYSTEM SET processes=500 SCOPE=BOTH` 时收到消息“SQL Error: ORA-02095: specified initialization parameter cannot be modified”，请改为使用 `ALTER SYSTEM SET processes=500 SCOPE=SPFile`。

请参阅 Oracle 文档以获取关于这些设置的效果的更多信息。

2. 重新启动数据库服务器。必须执行此步骤才能使参数生效。请先确保数据库上没有正在运行的会话。

在安装期间所需的 Oracle 信息

关于此任务

在安装期间，将要求您在数据库配置面板中提供以下信息：

数据库配置

- **数据库主机：**安装 Oracle 的主机。
- **数据库端口：**对于 Oracle，Build Forge 将缺省端口 1521 置此字段中。如果使用其他端口，请准备输入该端口号。
- **数据库名称：**供 Build Forge 使用的数据库的名称。您在先前的设置步骤中创建了该数据库。
- **数据库用户名：**供 Build Forge 在访问数据库时使用的用户名。您在先前的配置步骤中创建了该用户。
- **密码：**该数据库用户名的密码。

测试数据库配置

- **Oracle 客户机库的路径** - Oracle 客户机库所在的目录。

要点：在 UNIX 或 Linux 上安装 Build Forge 时，该目录必须是包含 32 位客户机驱动程序库的目录。

- **ORACLE_HOME 环境变量** - 安装 Oracle 的目录。
- **tnsnames.ora file (TNS_ADMIN) 的路径** - 包含 tnsnames.ora 文件的目录。请确保已设置了对 tnsnames.ora 文件的完整访问许可权。
- **JDBC 驱动程序位置** - JDBC 驱动程序所在的目录。该驱动程序由 Apache Tomcat 用来访问数据库。

将显示以下信息：

- **必需的驱动程序 JAR 文件** - 显示必需的驱动程序 JAR 文件。对于 Oracle，它是 ojdbc14.jar。
- **必需的 JDBC 驱动程序类** - 显示必需的 JDBC 驱动程序类。对于 Oracle，它是 oracle.jdbc.driver.OracleDriver。

Oracle 客户机配置

关于此任务

要为 Oracle 安装和配置客户机，请执行以下操作：

过程

1. 在管理控制台主机上安装 Oracle 即时客户机。无论您是在 32 位还是 64 位平台上运行，都必须安装 32 位版本。请对您的操作系统仅使用 V10.2 或 V11.2 的 **Instant Client Package - Basic**。
 - 请对 Oracle 10.2 使用版本 10.2 客户机。
 - 请对 Oracle 11g 使用版本 11.2 客户机。

请从 Oracle 下载客户机，网址是 <http://www.oracle.com/technology/software/tech/oci/instantclient/index.html>。

2. 在管理控制台主机上设置环境。 必须设置多个环境变量。

- LD_LIBRARY_PATH: 设置为包含客户机安装目录。

注: 可以在 Installation Manager 中的“数据库配置”页面上指定此值。

- NLS_LANG: 设置为国际语言支持的相应值。
 - UNIX 和 Linux 系统: 值必须包含 AL32UTF8。例如: AMERICAN_AMERICA.AL32UTF8。
 - Windows 系统: 值必须包含与 Windows 注册表中的活动代码页设置相对应的字符映射规范。请参阅『Windows 上的 Oracle 客户机示例配置』。

NLS_LANG 必须按说明显式设置。将安装客户机期间设置的默认字符集用于 *Build Forge* 是错误的。

- ORACLE_HOME: 设置为您的 Oracle 客户机安装目录的路径。

注: 可以在 Installation Manager 中的“数据库配置”页面上指定此值。

- ORA_NLS10: 设置为字符集数据在服务器上的路径。
- PATH: 设置为直接包含客户机安装。
- TNS_ADMIN: 设置为 tnsnames.ora 文件在服务器上的路径。请确保已设置了对 tnsnames.ora 文件的完整访问许可权。

注: 可以在 Installation Manager 中的“数据库配置”页面上指定此值。

要检查 Oracle 服务器上的当前语言设置，请登录 Oracle 并运行以下命令：

```
SQL> host echo $NLS_LANG
```

下一步做什么

要点: 在安装时，将对 tnsnames.ora 进行设置，从而对数据库名称和 Oracle SID 使用相同的值。由于 JDBC 驱动程序的局限，因此需要此约束。如果这两者不同，那么 Quick Report 报告功能以及 Java 和 Perl 的服务层 API 将不起作用。

Windows 上的 Oracle 客户机示例配置:

关于此任务

示例环境:

- Instant Client - Basic, 位于 C:\instantclient_11_2 中，使用美式英语
- Oracle 11.2, 位于 Windows 系统的 C:\oracle\product\11.2.0\db_1 中，为支持国际数据而安装

安装有客户机和 Build Forge 的系统上的变量设置:

- LD_LIBRARY_PATH 包括 C:\instantclient_11_2\
- NLS_LANG=AMERICAN_AMERICA.WE8MSWIN1252
- ORACLE_HOME=C:\instantclient_11_2\
- ORA_NLS10=C:\oracle\ocommon\nls\admin\data
- PATH 包括 C:\instantclient_11_2\
- TNS_ADMIN=C:\oracle\product\11.2\db_1\network\admin

关于 Windows 系统上的 NLS_LANG

Windows 上的 Oracle 客户机使用 Windows 中的设置来执行本地字符映射。

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\CodePage\ACP

客户机将数据发送到服务器时，将按照服务器的字符映射来发送字符。

以上的示例 NLS_LANG 设置对应于 ACP 设置 1252（美国英语版本的 Windows 的缺省值）。对于其他语言，请查阅 Oracle InstantClient 文档以获取 NLS_LANG 的正确语言、语言环境及字符映射部分。确保检查 Windows 系统上的实际 ACP 设置，并使设置的字符映射部分与其相匹配。

其他语言的字符映射如下所示。将这些字符映射与正确的语言和语言环境组合使用，以创建 NLS_LANG 设置。

```
1250 EE8MSWIN1250
1251 CL8MSWIN1251
1252 WE8MSWIN1252
1253 EL8MSWIN1253
1254 TR8MSWIN1254
1255 IW8MSWIN1255
1256 AR8MSWIN1256
1257 BLT8MSWIN1257
1258 VN8MSWIN1258
874 TH8TISASCII
932 JA16SJIS
936 ZHS16GBK
949 KO16MSWIN949
950 ZHT16MSWIN950
```

注：如果未正确设置字符映射，那么您不会收到警告，起初在行为或性能方面没有显著区别。客户机尝试在内存中进行字符转换。在其耗尽内存并且需要交换以执行转换时，客户机和服务器之间的通信性能将急剧下降。

UNIX 或 Linux 上的 Oracle 客户机示例配置:

关于此任务

示例环境:

- Instant Client - Basic, 位于 /usr/local/instantclient_11_2 中, 使用美式英语
- Oracle 11.2, 位于 UNIX 系统的 /usr/local/oracle/product/11.2.0/db_1 中, 为支持国际数据而安装

安装有客户机和 Build Forge 的系统上的变量设置:

- LD_LIBRARY_PATH (Solaris, Linux) 或 LIBPATH (AIX) 包含 /usr/local/instantclient_11_2
- NLS_LANG=AMERICAN_AMERICA.AL32UTF8
- ORACLE_HOME=/usr/local/instantclient_11_2
- ORA_NLS10=/usr/local/oracle/ocommon/nls/admin/data
- PATH 包括 /usr/local/instantclient_11_2
- TNS_ADMIN=/usr/local/oracle/product/11.2/db_1/network/admin

使用 Oracle Instant Client 11.2 的 UNIX 和 Linux 系统

Build Forge 应用程序依赖库 libclntsh.so.10.1，该库属于版本 10.2 客户机。在安装期间，安装程序会在 ORACLE_HOME 中检查此文件。如果该文件不存在，安装程序会假定您使用版本 11.2 客户机并且尝试创建一个从 libclntsh.so.10.1 指向相应版本 11.2 库 libclntsh.so.11.1 的符号链接。

```
ln -s libclntsh.so.11.1 libclntsh.so.10.1
```

要点：运行 Build Forge 安装程序的用户必须对 ORACLE_HOME 目录具有写访问权。如果您的站点具有此问题，请在安装 Build Forge 之前，让授权用户手动创建链接。在某些站点上，可以在以只读方式安装的文件系统上安装客户机（Solaris 系统的常用实践）。

使用 Oracle RAC 实施故障转移

使用该过程以在使用 Oracle RAC 时启用控制台故障转移

开始之前

Oracle RAC 的基本数据库设置与其他 Oracle 数据库的设置相同。使用这些设置安装控制台。安装控制台后，请使用以下过程实施故障转移。

关于此任务

要启用 Oracle RAC 故障转移以用于 Build Forge，需要进行其他配置。提供了两种方法：

- **SCAN 方法：**修改 buildforge.conf。SCAN（单客户机访问名称）是 Oracle 11g 中的设施。使用该方法时，集群中的所有节点都可以用于故障转移。至少应该提供三个节点并将其设置为按循环方式进行故障转移。
- **显式连接字符串方法：**修改 Oracle Instant Client 的 tnsnames.ora 文件，并修改 buildforge.conf。在该方法中，指定代表集群和每个所需集群节点的服务。至少应该提供三个节点并将其设置为按循环方式进行故障转移。必须设置 LOAD_BALANCE 和 FAILOVER 参数。

您可能需要 Oracle DBA 提供服务的名称以及要使用的节点的列表。配置 Oracle Instant Client 的 tnsnames.ora 文件，并设置 buildforge.conf 中的 JDBC 连接字符串。

这两种方法都要求您指定 SERVICE_NAME，这代表整个集群。这与非 RAC 配置不同，其中 SERVICE_NAME 通常与 SID 相同。

过程

- **SCAN 方法**

1. 编辑 buildforge.conf。将 db_url 参数添加到 buildforge.conf 并使用 JDBC 连接字符串作为参数。请注意，服务必须在 Oracle (mysvc.mycorp.com) 中设置并且具有多个节点。为了清楚起见，使用了换行符。请勿在 buildforge.conf 中换行。

```
db_hostname myoracledbhost.mycompany.com
db_url jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=myoracledbhost.mycompany.com)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=mysvc.mycorp.com)))
db_tcp_port 1521
db_database orclrac
db_username
db_password db_type odbc
db_provider oracle
```

2. 更新所有位置中的 buildforge.conf。

- Windows


```
<bfinstall>
<bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/classes/buildforge.conf
```
 - UNIX 或 Linux


```
<bfinstall>/Platform
<bfinstall>/server/tomcat/webapps/rbf-services/WEB-INF/classes/buildforge.conf
```
- 显式连接字符串方法 在示例中，使用了以下名称：
 - 服务名称: myorasvc.mycorp.com
 - 节点名称: clusnode-1vip, clusnode-2vip, clusnode-3vip, clusnode-4vip
1. 编辑 tsnames.ora。


```
myorasvc.mycorp.com=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-1vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-2vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-3vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-4vip)(PORT=1521))
  (LOAD_BALANCE=yes)(FAILOVER=ON)
  (CONNECT_DATA=(SERVER=DEDICATED)
  (SERVICE_NAME=myorasvc.mycorp.com)
  (failover_mode=(type=select)
  (method=basic)(retries=180)(delay=5)))
)
```
 2. 编辑 buildforge.conf 将 db_url 参数设置为 JDBC 连接字符串。为了清楚起见，使用了换行符。请勿在 buildforge.conf 中换行。


```
db_url jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on)
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-1vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-2vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-3vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-4vip)(PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=myorasvc.mycorp.com)))
```
 3. 更新所有位置中的 buildforge.conf。
 - Windows


```
<bfinstall>
<bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/classes/buildforge.conf
```
 - UNIX 或 Linux


```
<bfinstall>/Platform
<bfinstall>/server/tomcat/webapps/rbf-services/WEB-INF/classes/buildforge.conf
```

在安装期间所需的 Oracle 信息

关于此任务

在安装期间，将要求您在**数据库配置**面板中提供以下信息：

数据库配置

- **数据库主机：**安装 Oracle 的主机。
- **数据库端口：**对于 Oracle，Build Forge 将缺省端口 1521 置此字段中。如果使用其他端口，请准备输入该端口号。
- **数据库名称：**供 Build Forge 使用的数据库的名称。您在先前的设置步骤中创建了该数据库。
- **数据库用户名：**供 Build Forge 在访问数据库时使用的用户名。您在先前的配置步骤中创建了该用户。

- **密码:** 该数据库用户名的密码。

测试数据库配置

- **Oracle 客户机库的路径** - Oracle 客户机库所在的目录。

要点: 在 UNIX 或 Linux 上安装 Build Forge 时, 该目录必须是包含 32 位客户机驱动程序库的目录。

- **ORACLE_HOME 环境变量** - 安装 Oracle 的目录。
- **tnsnames.ora file (TNS_ADMIN) 的路径** - 包含 tnsnames.ora 文件的目录。请确保已设置了对 tnsnames.ora 文件的完整访问许可权。
- **JDBC 驱动程序位置** - JDBC 驱动程序所在的目录。该驱动程序由 Apache Tomcat 用来访问数据库。

将显示以下信息:

- **必需的驱动程序 JAR 文件** - 显示必需的驱动程序 JAR 文件。对于 Oracle, 它是 ojdbc14.jar。
- **必需的 JDBC 驱动程序类** - 显示必需的 JDBC 驱动程序类。对于 Oracle, 它是 oracle.jdbc.driver.OracleDriver。

安全性设置

安装期间, 系统会询问您有关您希望如何设置安全性的问题。

- **密钥库密码:** 您必须为密钥库提供一个密码。它既可用于启用安全登录 (凭证加密), 也可用作启用 HTTPS/SSL 的起点。
- **证书:** 您可以选择安装个人证书还是导入已有的证书。
- **安全 HTTP:** 将询问您是否希望安装为 HTTPS/SSL 启用的 Apache 服务器。将使用您选择的证书。如果需要使用除缺省值 443 以外的端口号, 那么需要在此时输入该端口号。

使用提供的个人证书

提供的证书具有以下属性集:

- **主题 DN:** “CN=hostname”, 其中 *hostname* 是要执行安装的主机的标准名称。
- **有效期:** 10 年 (表示为 3650 天)。您可以更改此值。典型的有效期为 1 到 2 年。如果有效期超过该期限, 将易受尝试猜测密钥的安全攻击。

您可以修改提供的证书。如果修改证书, 那么可以指定以下字段。

- 公共名称 (必填)
- 位置
- 省/直辖市
- 组织名称
- 国家/地区名称 (必填)
- 街道地址

将“公共名称”和“国家/地区名称”连接到要在证书创建期间指定的 X500Principal 类型的主题 DN。

系统将提示您输入对安装程序所创建的密钥库使用的密码。记录此密码。需要该密码才能完成 HTTPS/SSL 设置。

要点：可在以后更改该密码，但那是相当长的一个过程。使用满足本地复杂性需求的强密码。

使用自己的证书

如果您具有证书，可以导入该证书以供系统中使用 SSL 的所有组件和连接使用：

- 该证书必须在要安装 Build Forge 的主机上可用。将该证书复制到临时目录中。安装期间系统会提示您输入标准路径。
- 必须在安装期间指定密钥库密码。
- 该证书必须为 PKCS12 密钥库类型。如果证书为其他类型，那么可以使用 OpenSSL openssl 实用程序或 JDK keytool 实用程序将副本转换为 PKCS12。
- 系统将提示您输入对安装程序所创建的密钥库使用的密码。记录此密码。需要该密码才能完成 HTTPS/SSL 设置。

要点：可在以后更改该密码，但那是相当长的一个过程。使用满足本地复杂性需求的强密码。

第 7 章 安装管理控制台

本部分描述如何在 Windows、UNIX 和 Linux 平台上安装管理控制台。对 Installation Manager 所支持的任何安装场景使用以下过程。

1. 如 第 33 页的第 6 章,『安装前设置』中所述,执行安装前设置。任务包括以下内容。
 - 国际数据支持设置(必需)
 - 数据库设置(必需)。此步骤通常包括创建数据库对象、安装数据库客户机和收集在 Installation Manager 中执行安装步骤时需要的信息。
 - 安全性设置(可选,取决于您的需要)
2. 打开启动板以运行 Installation Manager。

注: 启动板在您运行 Installation Manager 的主机上对其进行搜索。如果它不存在,那么启动板会运行已打包的 Installation Manager,以在您的主机上安装 Installation Manager。然后它使用已安装的 Installation Manager 来安装 Build Forge。如果愿意,可以手动而不是从启动板安装 Installation Manager。

3. 在 Installation Manager 中执行安装步骤。
4. 执行必需的安装后检查。

请参阅第 63 页的第 8 章,『备选安装方法』以了解以下场景的安装过程:

- 使用您自己安装的以下必需应用程序(而不是 Build Forge 提供和安装的那些应用程序):
 - Apache HTTP server
 - PHP
 - Apache Tomcat
- 使用 IBM Installation Manager 对产品组件进行静默安装
- VMware 上的安装
- SUSE Linux on System z 上管理控制台的安装

使用启动板启动 Installation Manager

使用启动板启动 Installation Manager 并安装 Build Forge。

可以使用以下方法打开启动板:

- 从产品 DVD 启动启动板。
- 从下载的文件包启动启动板。

启动板检测是否在主机上安装了 Installation Manager:

- 如果 Installation Manager 存在,那么启动板会启动它。在 Installation Manager 中,选择要安装的 **Build Forge** 软件包。

要点: Installation Manager 的版本必须满足最低版本需求。请参阅第 21 页的『Installation Manager 需求』。

- 如果 **Installation Manager** 不存在，那么启动板使用已打包的 **Installation Manager**。在其中选择要安装的 **Installation Manager** 和 **Build Forge** 软件包。已打包的 **Installation Manager** 将安装 **Installation Manager**，然后启动它以安装 **Build Forge**。

启动板也在本地主机（如果是 Windows 主机）上安装了代理程序。在非 Windows 操作系统上，无法使用启动板来安装代理程序。有关代理程序安装指示信息，请参阅第 131 页的第 10 章，『安装代理程序』。

从产品 DVD 中启动启动板

使用这些指示信息来从产品 DVD 中启动启动板。

选择适用于您操作系统的指示信息。

• UNIX 或 Linux

1. 在安装 **Build Forge** 的主机上插入第一个 DVD。
2. 安装驱动器。
3. 在该驱动器的根目录中，运行 `launchpad.sh`。
4. 选择要在 **Installation Manager** 中安装的软件包。
 - 如果在主机上发现 **IBM Installation Manager**，那么启动板将启动该程序。

在第一个**安装软件包**页面上，选择 **Build Forge** 软件包，然后单击下一步。

- 如果找不到 **IBM Installation Manager**，已打包的 **Installation Manager** 会启动以安装 **Installation Manager**，然后使用它安装 **Build Forge**。

在第一个**安装软件包**页面上，选择 **Installation Manager** 和 **Build Forge** 软件包，然后单击下一步。

• Windows

1. 在安装 **Build Forge** 的主机上插入第一个 DVD。
2. 如果启用了自动运行，启动板将自动启动。如果未启用自动运行，请在驱动器的根目录中运行 `launchpad.exe`
3. 选择要在 **Installation Manager** 中安装的软件包。
 - 如果在主机上发现 **IBM Installation Manager**，那么启动板将启动该程序。

在第一个**安装软件包**页面上，选择 **Build Forge** 软件包，然后单击下一步。

- 如果找不到 **IBM Installation Manager**，已打包的 **Installation Manager** 会启动以安装 **Installation Manager**，然后使用它安装 **Build Forge**。

在第一个**安装软件包**页面上，选择 **Installation Manager** 和 **Build Forge** 软件包，然后单击下一步。

从已下载的软件包打开启动板

使用以下指示信息下载安装软件包并打开启动板。

1. 从 **IBM Passport Advantage**，将您的操作系统相应的安装软件包下载到要安装 **Build Forge** 的主机上的临时目录中。
2. 从已下载文件将安装映像解压到本地目录中。文件内容将解压到本地目录中。
3. 按如下所述从解压文件的目录打开启动板程序：

- Windows: 运行 launchpad.exe。
 - UNIX/Linux: 运行 launchpad.sh。
4. 选择要在 Installation Manager 中安装的软件包。
- 如果在主机上发现 IBM Installation Manager, 那么启动板将启动该程序。

在第一个**安装软件包**页面上, 选择 **Build Forge** 软件包, 然后单击**下一步**。

- 如果找不到 IBM Installation Manager, 已打包的 Installation Manager 会启动以安装 Installation Manager, 然后使用它安装 Build Forge。

在第一个**安装软件包**页面上, 选择 **Installation Manager** 和 **Build Forge** 软件包, 然后单击**下一步**。

Installation Manager 中的安装步骤

使用 IBM Installation Manager 可在大多数平台上安装产品组件。

开始之前

必须已启动 Installation Manager 并选择要安装的 **Build Forge** 软件包, 以便按照以下指示信息进行操作。

关于此任务

遵循提示以安装期望的软件包:

过程

1. 安装软件包 - 选中 **Build Forge** 和版本复选框。如果 Installation Manager 尚未安装, 请选中 **Installation Manager** 复选框。选中之后, 请单击**下一步**。
2. 安装软件包: 位置 - 共享资源 - 输入或选择要用于安装共享资源的目录, 然后单击**下一步**。
3. 安装软件包: 位置 - 软件包组 - 选择安装软件包将要安装到的目录, 然后单击**下一步**。缺省位置为: C:\Program Files\IBM\Build Forge。
4. 安装软件包: 功能 - 缺省情况下, 将安装所有这三个核心产品模块: Web 接口、进程引擎和服务层。单击**下一步**。
5. 安装软件包: 许可证服务器配置 - 选择“运行”用户以及控制台将使用的许可证类型。填写以下信息, 然后单击**下一步**。
 - (UNIX 和 Linux) 在 **Build Forge** 应作为哪个用户身份运行? 处, 接受缺省用户 (root 用户) 或指定其他用户。该用户将启动 Build Forge 引擎和提供的 Apache Tomcat。用户必须具备在 Installation Manager 的“数据库配置”页面中指定的数据库和 JDBC jar 文件的读和执行许可权。
 - 基于 **Rational License Server**: 输入 Rational License Server 的主机名。
 - 输入许可证服务器的有效主机名。如果计划稍后提供主机名, 请不要将该字段留空。请在此字段中输入字符或值。将该字段留空可能导致产品不完整且不可用。

安装完成后, 提供正确的主机名。有关指示信息, 请参阅第 22 页的『为 Build Forge 配置 Rational 许可证服务器』。

- 如果显示了许可证服务器但变灰，那么 FLEXlm 许可证客户机已为该主机注册了许可证服务器。

安装完成后，提供正确的主机名。有关指示信息，请参阅第 23 页的『更改管理控制台的许可证服务器』。

- **基于文件：**浏览至下载许可证文件的位置。

6. 数据库配置

根据要安装 Build Forge 的操作系统平台以及想要安装的数据库，必须指定某些信息。请参阅第 34 页的『数据库设置』中的数据库设置指示信息，以获取以下信息：

- 有关 DB2 Express，请参阅第 35 页的『DB2 Express 安装』。
- 有关 DB2，请参阅第 36 页的『DB2 设置』。
- 有关 Microsoft SQL Server，请参阅第 39 页的『Microsoft SQL Server 设置』。请注意：要指定的 JDBC 驱动程序取决于您选择的版本（SQL Server 2005 或 SQL Server 2008）。
- 有关 MySQL，请参阅第 43 页的『MySQL 设置』。
- 有关 Oracle，请参阅第 46 页的『Oracle 设置』。请注意：您必须选择与您已安装的 Oracle Instant Client 版本相对应的版本（Oracle 10 或 Oracle 11）。

注：在 UNIX 和 Linux 平台上，如果使用的是 Oracle、DB2 或 MySQL 数据库，必须安装并使用 32 位数据库客户机驱动程序。在 Installation Manager 的“数据库配置”页面上，对于特定的数据库类型，请确保在“数据库配置”页面上的指向 **[DB2|Oracle|MySQL] 客户机库**的路径字段中指定 32 位版本的驱动程序库。另外，对于 DB2，装入 db2profile。

7. 安装软件包：应用程序和 Web 服务器配置 – 填写所请求的信息，然后单击下一步。

a. Web 服务器/PHP 配置

- **提供您自己的 Web 服务器？** 如果想要提供您自己的 Web 服务器，请选择是。否是缺省值。
- **想要使用安全 HTTP 吗？** 如果要使用安全 HTTP，请选择是。否是缺省值。
- **Web 服务器应该使用哪个端口？** 如果不希望 Web 服务器使用端口 80，请输入其他端口号。80 是缺省值。
- **PHP 的内存限制：** 如果不想使用缺省值，请输入 PHP 的内存限制。256 MB 是缺省值。

b. 应用程序服务器配置

- **提供您自己的应用程序服务器？** 如果希望将 Build Forge 配置为使用已安装的应用程序服务器，请选择是。否是缺省值。

注：仅当将 Build Forge 配置为使用一个或多个已安装的组件时才如此设置。

通常 Build Forge 在安装期间安装这些组件。请参阅第 63 页的『使用您自己的组件安装』。

- c. **安全性配置** 在以下字段中输入并验证安装程序为 Build Forge 创建的密钥库密码。

注：如果这些字段不可见，请向下滚动来查找这些字段。

- 密钥库密码
- 验证密码

需要密码才能启用缺省安全登录（凭证加密）。如果要将 Build Forge 配置为使用 HTTPS/SSL，那么也需要密码。

要点：可在以后更改该密码，但那是相当长的一个过程。使用满足本地复杂性需求的强密码。

具有以下选项：

- 按原样使用安装程序创建的自签名证书。执行以下操作：
 - 1) 想要修改缺省值或上载定制证书吗？ 选择“否”。
 - 2) 具有现有的安全证书吗？ 选择“否”。
- 使用安装程序所创建的自签名证书，但修改其字段。执行以下操作：
 - 1) 想要修改缺省值或上载定制证书吗？ 选择“是”。此时将显示该证书的其他字段。请填写这些字段。
 - 2) 具有现有的安全证书吗？ 选择“否”。
- 提供您自己证书的位置。证书必须位于主机上，它必须是 pkcs12 格式，且您必须为该证书所在的密钥库提供现有的密码。执行以下操作：
 - 1) 想要修改缺省值或上载定制证书吗？ 选择“是”。
 - 2) 具有现有的安全证书吗？ 选择“是”。此时将显示其他字段。
 - 3) 请在类型为 **pkcs12** 的密钥库中指定已签署的证书。 输入文件名，或使用浏览来查找文件名。
 - 4) **密钥库密码** 输入含有证书的密钥库的密码。
 - 5) **验证密码** 重新输入含有证书的密钥库的密码。

注：浏览器通常显示访问具有自签名证书的安全站点的警告。用户通常可以选择继续操作，但是可能需要确认例外情况。

8. 服务配置 填写请求的信息，然后单击下一步。

- **服务层将运行在该机器上吗？** 如果不想要服务层运行在该机器上，请单击否。是为缺省值。

如果单击“否”，系统将提示您输入运行该服务层应用程序的应用程序服务器的主机名。

- **侦听端口：** 3966 为缺省值。必须选择一个或两个端口作为服务端口。
- **侦听安全端口：** 49150 为缺省值。必须选择一个或两个端口作为服务端口。

9. 安装软件包：控制台启动选项 – 填写所请求的信息，然后单击下一步。

a.

- **在桌面上创建快捷方式吗？** 缺省值是在桌面上创建快捷方式（对于 Windows）。

注：对于 Linux，将不创建桌面快捷方式。

b. 启动选项

- **请勿启动控制台：** 缺省情况下，该控制台将启动。如果不想控制台自动启动，请单击该单选按钮。

- 以服务方式启动控制台：缺省情况下，该控制台将以服务方式启动。
 - 以前台方式启动控制台：缺省情况下，该控制台不以前台方式启动。如果想要该控制台以前台方式启动，请单击单选按钮。
10. 安装软件包：摘要复审 – 复审该页面上的摘要信息以确认安装 Build Forge 组件的位置，然后单击安装。
- a. 目标位置
 - 软件包组名称：buildforge.console 是缺省的软件包名称。
 - 安装目录：缺省的安装目录为 C:\Program Files\IBM\Build Forge。
 - 共享资源目录：缺省的共享资源目录。缺省目录如下：
 - Windows: C:\Program Files\IBM\SDP70Shared
 - UNIX 和 Linux: /opt/IBM/SDP70Shared
 - b. 功能部件
 - **Build Forge 功能部件** 您可以复审将要安装哪些功能部件或模块。例如，核心产品模块有：Web 接口、进程引擎和服务层。
 - c. 环境
 - 英语是缺省环境。
 - d. 存储库信息
 - 将从以下位置中检索文件：使用该部分可复审并确认存储库位置。
11. 访问管理控制台。启动浏览器。转至管理控制台的 URL：
- 一般格式：http://<hostname>[:<portnumber>]。如果使用了 HTTP 缺省值（端口 80），那么端口号是可选的。
 - 本地：如果运行的浏览器与管理控制台位于同一主机上，那么可以使用 http://localhost。
- 注：如果安装后无法立即登录，请等待一分钟左右，然后重试。首次启动时，引擎（bfengine）必须装入数据库模式。
- 要点：不要在安装后立即停止 bfengine。这样做可能会损坏数据库模式。在这种情况下，您需要删除数据库中的所有 Build Forge 表并重新安装 Build Forge。
12. 登录。使用用户名 **root** 和密码 **root**。请立即更改 root 用户密码。

安装后核对表

该部分描述了安装 Build Forge 系统之后需要执行的操作。

- 检查 PATH 变量。
- 如果管理控制台必须通过代理服务器来访问数据库，请确定 PHP 要使用的代理服务器。
- 设置 JVM 内存。
- 日志消息迁移。

在 Windows 上检查 PATH 变量

Windows 上的 PATH 环境变量必须包含到数据库客户机或驱动程序 DLL 的路径。手动检查这些数据库的 PATH：

- DB2 - 包含 db2cli.dll 和 sqlar.dll 的目录
- MySQL - 包含 libmysql.dll 的目录
- Oracle - 包含 oci.dll 的目录

对于 Microsoft SQL Server, 设置 ODBC 连接将考虑该需求。

确定代理服务器

可选: 仅当管理控制台需要使用代理服务器访问其数据库时, 才需要此步骤。必须配置 PHP 以使用代理服务器。

- 编辑 php.ini 文件。它位于 *<bfinstall>*/Apache/php 中, 例如, C:\Program Files\IBM\Build Forge\Apache\php。

添加以下条目:

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

设置 JVM 内存 (对于快速报告必需)

可选: 仅当版本包含快速报告 (Standard Edition、Enterprise Edition 和 Enterprise Plus Edition) 时, 才需要该步骤。

- 将 JVM 的最大内存设置为 1 MB 或更大。运行报告需要最小 1 GB (1024 MB) 的堆大小。如果在运行报告时发生内存不足错误 (可能发生在大型报告上), 请增加此设置的值。这可能需要您向主机添加内存。

```
JAVA_OPTS -Xmx1024M
```

日志消息迁移

Build Forge 存储日志消息的方式不同于旧发行版。这样, 如果 Build Forge 安装是先前发行版的升级, 那么作业输出页面中的“消息”列可能暂时为空, 直至完成特定作业的迁移。因为在低优先级的小型批处理中发生此情况, 所以迁移可能花费一些时间; 如果任何作业具有空消息, 建议用户只需等待至作业的消息迁移完成即可。

增加 Linux 文件句柄的数量

开始之前

重要信息: 为了得到最佳结果, 在使用 Rational 产品之前, 请增加可用于 Rational Build Forge 的文件句柄的数量。系统管理员可能需要进行此更改。

当您在 Linux 上执行这些步骤来增加文件描述符时, 请谨慎操作。不遵循指示信息可能导致计算机无法正常启动。为了得到最佳结果, 请让系统管理员执行此过程。

要增加文件描述符, 请执行以下操作:

过程

1. 作为 root 用户登录。如果没有 root 用户访问权, 那么在继续前必须先获得该访问权。
2. 切换至 etc 目录。

3. 查找 `initscript` shell 脚本。打开该文件或使用 Linux 文本编辑器创建该文件。

重要信息：不要在计算机上将 `initscript` 文件保留为空。否则，下次打开或重新启动计算机时，计算机不会启动。

4. 在第一行上，将 `ulimit` 设置为远大于大多数 Linux 计算机上的缺省值 1024。

```
ulimit -n 4096
```

警告：将 `ulimit` 设置得过大可能会影响整个系统的性能。

5. 在第二行上，输入 `eval exec "$4"`。
6. 保存并关闭 shell 脚本。

结果

有关 `ulimit` 命令的更多信息，请参阅 `ulimit` 的联机帮助页。

第 8 章 备选安装方法

本部分描述安装管理控制台的备选方法。为以下方案提供了指示信息：

- 对以下应用程序使用您自己的安装（而不是由 Build Forge 提供和安装的那些）：
 - Apache HTTP Server 和 PHP
 - Apache Tomcat
- 使用 IBM Installation Manager 对产品组件进行静默安装
- VMware 上的安装
- SUSE Linux on System z 上的安装

使用您自己的组件安装

如果已经安装了这些组件并且希望使用已有（而不是由 Rational® Build Forge® 提供）的这些组件，请使用本部分设置所需的技术。

Build Forge 自动安装并配置所需的以下组件和技术：

- Apache HTTP Server 和 PHP
- Apache Tomcat

要点：您使用的每个组件的版本都必须与产品提供的组件版本相同。有关更多信息，请参阅第 21 页的『硬件、软件和数据库需求』。

您必须配置每个组件的安装，以满足 Build Forge 需求，然后运行 Installation Manager 以安装 Build Forge 组件。安装期间，您将有机会使用已设置的技术。

以下部分描述如何设置要与 Build Forge 一起使用的每种技术，以及随后如何安装 Build Forge 以使用这些技术。

先决条件

必须具备以下项才能在 UNIX 或 Linux 上执行安装：

- 因特网访问权。如果要安装 Build Forge 的计算机无法访问因特网，那么需要通过能够访问因特网的计算机下载文件，并将这些文件传送到 Build Forge 计算机，这样才能完成步骤。
- 有效并且可在您的平台上运行的 C 编译器（例如，Linux 上的 gcc 编译器）。
- 编译器制造商推荐的 make 设施（例如，用于 gcc 的 gnu-make）。
- root 用户身份的特权。
- 要使用 SSL，必须编译 OpenSSL。

国际数据支持

Build Forge 必须设置为在管理控制台中支持国际数据。

开始之前

- **Web 浏览器:**
 - 必须设置了语言
 - 必须已安装了用来显示数据的字体
- **代理程序**

Build Forge 建议在代理程序服务器上使用 UTF-8 字符集。

在 UNIX/Linux 上, 请使用以下命令来检查语言环境和字符集:

```
locale
```

您应当看到指示语言和字符集的值。以下示例来自使用美国英语和 UTF-8 字符集的 Solaris 系统:

```
LANG=en_US.UTF-8  
LC_CTYPE="en_US.UTF-8"
```

- **所有数据库:**

通常, 对国际数据的支持是在创建数据库时指定; 不能在创建数据库之后再配置国际数据支持。

要用于显示数据的字体必须安装在数据库主计算机上。

Build Forge 需要使用国际数据 (UTF-8 字符集)。

- **DB2:**
 1. 设置代码集和地域。示例: CREATE DATABASE USING CODESET UTF-8 TERRITORY US (或在控制中心中选择相应的代码集和地域)。
 2. 在管理控制台计算机上将 DB2CODEPAGE 环境变量设为 1208。

在 Windows 上, 请使用以下命令:

```
set DB2CODEPAGE=1208
```

在 UNIX 或 Linux 上, 请使用以下命令:

```
export DB2CODEPAE=1208
```

如果现有数据库中含有数据, 而您需要将它迁移到 UTF-8, 那么以下文档可能对您有所帮助: <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/t0024033.htm>

- **MySQL:** 设置服务器字符集和整理。如果您的 MySQL 安装当前不支持国际数据, 那么可以从代码源重新编译它, 并使用 `./configure --with-charset=utf8 --with-collation=utf8_bin`。如果未配置此支持, 那么 Build Forge 引擎不会启动。
- **Oracle:** 安装时, 请在实例上将字符集设置为 **UTF8 - Unicode 3.0**。在“数据库配置助手”中, 此设置是在“字符集”选项卡上的“初始化参数”步骤中进行的。

数据库安装和配置

通过本部分, 可以对使用您自己的组件进行的 Build Forge 管理控制台安装来安装您要使用的数据库。

开始之前

在安装和配置其他技术和 Build Forge 之前，必须安装和配置数据库以及数据库对象。需要执行以下操作：

通常，您必须完成以下任务：

- 确定要使用的数据库系统。验证该系统是否受 Build Forge 支持，并验证数据库主机和 Build Forge 主机之间是否存在必需的网络连接。如果需要代理服务器才能访问数据库，请获取代理服务器名称和路径。
- 如果要使用国际数据，请验证是否已将数据库配置为使用 UTF-8 字符集。
- 创建数据库对象和许可权，通常如下：
 - 数据库：它在示例中名为 **build**，但是可以使用其他名称。
 - 数据库用户：管理控制台使用此用户名来访问数据库。用户在示例中名为 **build**，但是可以使用其他用户名。
 - 数据库用户密码
 - 在 **build** 数据库中创建表空间的 **build** 用户的许可权。必须具有所有者许可权才能创建、修改和删除数据。

为每种数据库类型提供了具体的指示信息。

Apache HTTP Server 安装和配置

Installation Manager 安装并配置 Apache HTTP Server 作为 Build Forge 的 Web 服务器。使用提供的 Apache HTTP Server 是为 Build Forge 配置 Web 服务器的最快的方法。

作为标准配置的备选方法，可以配置现有的 Apache HTTP Server 代替 Build Forge 安装和配置的服务器。提供的指示信息假定您具有在操作系统上设置和配置 Apache HTTP Server 的经验。

要使用现有的 Apache HTTP Server，请按如下所示修改您的安装：

1. 修改 Apache HTTP Server 配置文件（httpd-vhosts.conf）以指向 Build Forge 应用程序。
2. 安装 PHP 并配置 Apache HTTP Server、Build Forge 数据库以及密码加密（如果希望使用此安全性功能）所需的 PHP 模块。
3. 为您的数据库配置 Apache。

使用 Installation Manager 安装 Build Forge

在 Installation Manager 中的“应用程序和 Web 服务器配置”页面的提供您自己的 Web 服务器提示处，选择是。

必备软件

- Apache HTTP Server 2.2.4
- PHP 5.2.4

编辑 Apache 服务器配置文件

1. 在服务器安装的 extras 目录中查找 Apache http-vhosts.conf 文件。

```
cd <apache-dir>/conf/extras/  
vi httpd-vhosts.conf
```

2. 编辑 Apache http-vhosts.conf 文件。要将关于 Build Forge 的信息添加到 httpd-vhosts.conf 中, 请添加以下行:

```
<VirtualHost *:80>  
    ServerAdmin build@yourdomain.com  
    DocumentRoot /opt/buildforge/webroot/public  
    ServerName ausbuild01.yourdomain.com  
    ServerAlias build.yourdomain.com mc.yourdomain.com  
    ErrorLog logs/ausbuild.error_log  
    CustomLog logs/ausbuild.access_log common  
</VirtualHost>
```

3. 修改 DocumentRoot 设置以指向 Build Forge Web 应用程序。在该示例中, Build Forge 安装目录为 /opt/buildforge。
4. 保留端口为 80 或将其更改为用于本地运行 Apache HTTP Server 的端口。

```
<VirtualHost *:80>
```

要点: 请勿使用端口 8080; 它是 Apache Tomcat 的缺省端口。

5. 根据 Apache HTTP Server 的需要, 修改 http-vhosts.conf 中的其他任何设置:
 - ServerAdmin: Build Forge 管理员的电子邮件地址
 - DocumentRoot: Build Forge 应用程序的条目页面的位置
 - ServerName: 安装有 Build Forge 应用程序的服务器
 - ServerAlias: Build Forge ServerName URL 的可选别名
 - ErrorLog: Build Forge 应用程序的 Apache 错误日志
 - CustomLog: 用于记录对 Build Forge 应用程序的访问的 Apache 错误日志

为 Apache HTTP Server 安装并配置 PHP

PHP 不是与 Apache HTTP Server 一起安装的。您必须安装 PHP 5.2.4 并将其配置为指向 Apache HTTP Server 的 httpd-vhosts.conf。

为 Build Forge 数据库安装并配置 PHP

PHP 安装期间, 为您正在作为 Build Forge 数据库使用的数据库类型选择和安装 PHP 扩展。

(可选) 配置 PHP OpenSSL 模块以支持密码加密

要支持 SSL, Build Forge 使用 PHP OpenSSL 模块。此支持是随 PHP 5.2.4 提供的; 不需要额外配置。

要支持密码加密, 需要一些额外配置。PHP 5.2.4 是支持此配置所必需的。您必须查找 OpenSSL 扩展的补丁文件, 将这些补丁文件安装到 OpenSSL 目录中, 并重新编译 PHP, 如下所示:

1. 在位于 Build Forge 安装目录的 misc 目录中查找 php_openssl.h 和 openssl.c 补丁文件, 例如:

Windows	C:\Program Files\IBM\Build Forge\misc
UNIX/Linux	/opt/buildforge/Platform/misc

2. 将补丁文件复制到位于 Build Forge 安装目录的 openssl 目录中。
3. 使用 `--with-openssl=<path_to_openssl>` 配置选项编译 PHP，其中 `<path_to_openssl>` 是 Build Forge openssl 目录。

为您的数据库配置 Apache

您需要向 httpd.conf 添加特定信息，具体取决于您的数据库。

DB2 的 Apache 配置

1. 向 Apache 启动脚本（通常是 /etc/init.d/httpd 或 /etc/init.d/apache2，具体取决于您的分发）的开头部分添加以下行。

```
source /home/db2bf/sql1lib/db2profile
```

2. 向 httpd.conf 添加以下行:

```
PassEnv LD_LIBRARY_PATH
PassEnv CLASSPATH
PassEnv LIBPATH
PassEnv VWS PATH
```

MySQL 的 Apache 配置

不需要任何额外配置。

Oracle 的 Apache 配置

1. 向 httpd.conf 添加以下行:

```
PassEnv LD_LIBRARY_PATH
PassEnv NLS_LANG
PassEnv ORACLE_HOME
PassEnv ORA_NLS
PassEnv ORA_NLS32
PassEnv TNS_ADMIN
```

2. 向在引导时启动 Apache 的脚本（通常是 /etc/init.d/httpd 或 /etc/init.d/apache2）添加以下行，并提供以下设置的值。

```
export LD_LIBRARY_PATH=<value>
export NLS_LANG=<value>
export ORACLE_HOME=<value>
export ORA_NLS=<value>
export ORA_NLS32=<value>
export TNS_ADMIN=<value>
```

启动 IBM HTTP Server

启动 Build Forge 引擎并启动管理控制台之前，启动您的 Apache HTTP Server。

PHP 安装和配置

使用以下过程可设置 PHP 以与管理控制台一起使用。

开始之前

需求:

- 版本: PHP 必须为 5.2.4 或更高版本
- 数据库驱动程序: 已安装的 Build Forge 数据库的 PHP 模块

关于此任务

遵循本部分中的指示信息为 Apache HTTP Server 或其他 Web 服务器配置 PHP。这些指示信息假定您已下载所需版本的 PHP。

- 下载 PHP
- 安装 PHP
- 配置 PHP
- 编辑 Apache 配置文件
- （可选）确定访问数据库要使用的代理服务器（仅当管理控制台主机通过代理服务器访问数据库时才需要）

安装 PHP

关于此任务

本部分描述如何从源代码开始编译并安装 PHP。如果您有 PHP 的现有安装，并且不希望重新编译，那么您只需要确定是否已安装了相应的数据库驱动程序。如果需要安装数据库驱动程序，请参考 PHP 文档以了解应使用的安装方法。安装要用于 Build Forge 的数据库的数据库驱动程序，如下所列：

- DB2: ibm_db2 驱动程序
- MySQL: mysqli 驱动程序
- Oracle Instant Client: oci8 驱动程序

注：目前，完整的 Oracle 客户机与 PHP oci8 不兼容。请仅使用 Oracle Instant Client。

过程

1. 配置 PHP 从而安装在刚创建的工作目录中。

```
$ ./configure --prefix=/usr/local/php-5.2.4 --with-<database>=shared \  
--with-apxs2 --with-ldap=shared --enable-mbstring --enable-shmop \  
--with-xml --with-zlib=shared
```

如下所示替换 `--with-<database>:`

- DB2: `--with-ibm_db2[=dir]`。如果未指定 `=dir`，将使用缺省值：`/home/db2inst1/sqllib`
- MySQL: `--with-mysqli[=file]`。可选文件参数为 `mysql_config` 的路径名。
- Microsoft SQL Server: `--with-mssql[=dir]`
- Oracle: 您必须独立安装 **Oracle Instant Client** 以使用 **PHP oci8**。使用 Oracle Instant Client 连接到数据库时，请使用 `--with-oci8=instantclient,lib`，其中 *lib* 是 Instant Client 库目录的路径。

请注意代码块中的行连续字符 `\`。此步骤指定 PHP 的安装位置以及安装时使用的选项。缺省情况下，它安装在 `/usr/local` 中。示例显示了如何将其放在 `/usr/local/php-5.2.4` 中。此位置将在稍后示例中会用到。

2. 编译 PHP。

```
$ make
```

此步骤在本地目录中编译可执行文件。

3. 安装 PHP（作为 root 用户执行）。

```
# make install
```

此步骤必须由对 Apache 安装目录（在本示例中为 /usr/local/apache-2.2.4）具有写特权的用户执行。这通常由 root 用户执行。您的本地管理设置可能有所不同。

配置 PHP

过程

1. 将扩展文件复制到扩展目录中。需要将数据库的扩展文件从存储库复制到活动的扩展目录中。以下示例假定 PHP 安装在 /usr/local/php-5.2.4 中。请注意：`<timestamp>` 是数字字符串。

```
$ cd /usr/local/php-5.2.4/lib/php/extensions/no-debug-non-zts-<timestamp>/
$ cp <db-extensions> ..
```

`<db-extensions>` 文件对应于您用于 Build Forge 的数据库，如下所示：

- DB2: `ibm_db2.so`
- MySQL: `mysql.so` 和 `mysqli.so`
- Oracle: `oci8.so`

2. 编辑 PHP 配置文件 `php.ini`。以下示例假定 PHP 安装在 /usr/local/php-5.2.4 中：

```
$ cd /usr/local/php-5.2.4/lib/
$ vi php.ini
```

添加以下条目：

```
extension_dir=/usr/local/php-5.2.4/lib/php/extensions
upload_tmp_dir=<directory>
extension=<db-extension-so>
```

使用针对您数据库的一个或多个 `<db-extensions-so>` 文件名，如下所示：

- DB2: `extension=ibm_db2.so`
- MySQL: 两个条目 -
`extension=mysql.so`
`extension=mysqli.so`
- Oracle: `extension=oci8.so`

注：运行 Apache Web 服务器的用户必须可以对用作 `upload_tmp_dir` 的目录进行写操作。通常，此用户是 `nobody`，但是您的本地管理实践可以有所不同。

编辑 Apache 配置文件

过程

编辑 Apache 配置文件。在 `httpd.conf` 中添加关于 PHP 的信息。

```
cd <apache-dir>
vi httpd.conf
```

添加以下行：

```
LoadModule php5_module modules/libphp5.so
AddHandler php5-script .php
AddType text/html .php
DirectoryIndex index.php
```

确定代理服务器

关于此任务

可选：仅当管理控制台需要使用代理服务器访问其数据库时，才需要此步骤。

过程

编辑 PHP 配置文件 `php.ini`。该文件位于 `<php-install>/lib` 中，例如 `/usr/local/php-5.2.4`。

添加以下条目：

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

Apache Tomcat 安装和配置

Installation Manager 将 Apache Tomcat 安装并配置为 Build Forge 的应用程序服务器。使用提供的 Apache Tomcat 应用程序服务器是为 Build Forge 配置应用程序服务器最快的方法。

作为标准配置的备选方法，还可以选择使用现有 Apache Tomcat 而不是 Build Forge 提供的 Apache Tomcat。本部分描述了这种备选方法的必备软件、安装前设置、安装以及安装后需求。提供的指示信息假定您具有设置和配置 Apache Tomcat 的经验。

必备软件

- Apache Tomcat Server:
 - 5.5.28（Solaris 平台）
 - 5.5.9（所有其他平台）
- J2SE 5: IBM 或 Sun
- Build Forge 数据库的 JDBC 数据库驱动程序：需要 Java 数据库连接（JDBC）驱动程序与 Apache Tomcat 一起使用。Sun 在以下地址提供了 JDBC 供应商列表：<http://developers.sun.com/product/jdbc/drivers>。

安装 JDBC 驱动程序的 jar 文件

下载并解压缩数据库的 JDBC 驱动程序。

要点：JDBC 驱动程序下载可能包含很多文件和子目录。找到 JDBC 驱动程序的 jar 文件，然后仅将该 jar 文件复制到 `$CATALINA_HOME/common/lib`

。`$CATALINA_HOME` 是 Tomcat 安装根目录，必须设置为环境变量。请参阅 JDBC 驱动程序的安装文档。

- DB2 - <http://www-306.ibm.com/software/data/db2/express/download.html>

单击 **JDBC** 和 **SQLJ** 的 **DB2 驱动程序** 旁边的下载链接；需要 IBM 帐户注册。还需要找到并安装许可 .jar，即 `db2jcc_license_cu.jar`。

- MySQL - <http://www.mysql.com/products/connector/j/>

单击 **MySQL Connector/J 5.0** 或 **5.1** 的链接。选择与 MySQL 版本对应的 JDBC 驱动程序版本。

- Oracle - http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

单击 Oracle 版本旁边的下载链接；需要帐户注册。

- Microsoft SQL Server - <http://msdn.microsoft.com/en-us/data/aa937724.aspx>

单击 **Download SQL Server JDBC Driver** 链接。

在 Installation Manager 中配置 Apache Tomcat 服务器

这些指示信息指示通过 Installation Manager 配置 Apache Tomcat 需要的信息。

1. 关闭 Apache Tomcat。

要点： 启动 Installation Manager 之前，必须停止 Apache Tomcat。

2. 启动 Installation Manager。
3. 在起始页上，单击**安装**。
4. 遵循 Installation Manager 向导中的指示信息来安装产品。
5. 在“应用程序服务器配置”页面上，单击**是**以配置您自己的应用程序服务器。

核对表：应用程序服务器配置

✓	字段	描述
	重定向 URL	输入应用程序服务器的主机名和端口号。必须将 rbf-services 指定为上下文路径。例如: <code>http: https://<app_server_host>:<app_server_port>/rbf-services</code> 。
	指定要用于安装 BF 服务插件的目录	指定应用程序服务器主机的本地目录。 Installation Manager 在此目录中安装 Build Forge 服务层应用程序插件扩展。运行应用程序服务器的用户必须具有对此目录的读、写和执行许可权。

注： 在先前版本的 Build Forge 中，必须指定 WAR 部署目录。WAR 部署目录现在自动设置为 `bfinstall/PrepForExternal`。

6. 通过 Installation Manager 完成安装。

Apache Tomcat 的安装后配置

在通过 Installation Manager 完成安装之后，完成以下安装后步骤。

1. 您必须将 `buildforge.conf` 文件手工添加到 `rbf-services.war` 文件中。

`rbf-services.war` 的位置取决于您的操作系统。UNIX 和 Linux 会将此文件置于 `/opt/buildforge/PrepForExternal` 下；而 Windows 则会将其置于 `C:\Program Files\IBM\Build Forge\PrepForExternal` 下。请参阅第 83 页的『更新 `buildforge.conf` 文件』并完成相关步骤。

2. 增加 Apache Tomcat 服务器的 JVM 堆大小。

将 JVM 最大堆大小选项 `-Xmx` 设置为 1024 M。

使用在 catalina.bat 或 catalina.sh 中的 CATALINA_OPTS 或 JAVA_OPTS 环境变量来设置此 JVM 选项。

3. 启动 Build Forge 之前，启动 Apache Tomcat:

```
$CATALINA_HOME/bin/catalina.sh start。
```

手动安装 Installation Manager

如果您使用启动板程序来启动产品安装，会自动安装或更新 IBM Installation Manager。请参阅第 55 页的『使用启动板启动 Installation Manager』。

对 Installation Manager 有使用经验或希望设置静默安装的用户可手动安装 Installation Manager。请执行以下步骤:

1. 通过从 Passport Advantage 下载或使用产品 DVD，获得产品安装软件包。
2. 查找适用于您的平台的 Installation Manager 文件:
 - InstallerImage_linux
 - InstallerImage_solaris
 - InstallerImage_win32
3. 输入以下某条命令以启动安装程序。
 - 要作为 Admin 用户运行安装，请运行以下命令:
install
 - 要作为非 Admin 用户运行安装，请运行以下命令:
userinst
4. 遵循安装指示信息以安装 Installation Manager。

安装之后，可以使用 Installation Manager 或 Installation Manager 安装程序以静默方式安装软件包。

启动 Installation Manager

在 Windows 或 UNIX/Linux 上启动 Installation Manager。

开始之前

如果使用启动板程序来启动产品安装，Installation Manager 会自动启动。如果已安装了 Installation Manager，那么可以使用以下某种方法来将它启动:

- Windows: 单击开始 > 所有程序 > IBM Installation Manager > IBM Installation Manager。
- 转至 <IM-installdir>，然后运行 ./IBMIM。

指定存储库 URL

IBM Installation Manager 使用每个产品软件包中的嵌入式 URL 通过因特网连接到存储库服务器，并搜索最新产品安装软件包。

开始之前

在 Installation Manager 中，可以在“首选项”窗口中的“存储库”页面上设置存储库位置。您的组织可能需要重定向存储库以使用内部网站点。

注：在启动安装过程之前，确保从您的管理员或 IBM 获得安装软件包存储库 URL。

要指定存储库，请完成以下步骤：

1. 启动 IBM Installation Manager。
2. 在“启动”页面上，单击**文件 > 首选项**。
3. 在“首选项”窗口中，单击**存储库**。此时会打开“存储库”页面，其中显示可用的存储库、它们的位置以及它们是否已连接。
4. 在“存储库”页面上，单击**添加存储库**。
5. 在“添加存储库”对话框中，输入存储库位置的 URL 或使用**浏览**以查找包含存储库的 .zip 或 JAR 文件、diskTag.inf 文件或已扩展存储库的 repository.config 文件，然后单击**确定**。

此时会列出新的存储库位置。如果未连接存储库，那么“连接”列中会显示一个红色 x。

注：要搜索已更新的软件包，请确保选中**安装期间搜索服务存储库和更新**。缺省情况下此选项是选中的。

6. 单击**确定**以关闭“首选项”窗口。

执行产品组件的静默安装

可通过以静默安装方式运行 Installation Manager 来静默安装 Rational Build Forge 产品组件。在静默方式下，用户界面不可用；将通过响应文件输入安装产品软件包所需的命令。

对于静默安装，以下任务是必需的：

1. 安装 Installation Manager。
2. 创建响应文件。
3. 以静默安装方式运行 Installation Manager。

注：不能在以下情况中使用静默安装：

- 在未安装 X11 的 Linux 服务器上安装

有关 Installation Manager 和以静默方式进行安装的更多信息，请参阅 Installation Manager 信息中心：<http://www.ibm.com/software/awdtools/installmanager/support/index.html>。

使用 Installation Manager 创建响应文件

通过记录您在使用 Installation Manager 安装产品软件包时的操作可以创建响应文件。记录响应文件时，在 Installation Manager UI 中进行的所有选择都将存储在 XML 文件中。以静默方式运行 Installation Manager 时，Installation Manager 会使用该 XML 响应文件来完成安装。

您可以创建响应文件并安装产品，或使用 `-skipInstall <agentDataLocation>` 自变量跳过产品安装，而只创建响应文件。以下指示信息提供了这两种选择的语法示例。

要为安装创建响应文件：

1. 在命令行下，切换至安装了 Installation Manager 的目录中的 eclipse 子目录。例如：

Windows	C:\Program Files\IBM\Installation Manager\eclipse
UNIX/Linux	/opt/IBM/InstallationManager/eclipse

2. 在命令行下，用您自己的文件名和位置替换响应文件和（可选）日志文件，使用以下某条命令来启动 Installation Manager。

请确保输入的文件路径存在；Installation Manager 不会创建响应文件和日志文件的目录。如果使用 -skipInstall 选项，那么 <agentDataLocation> 必须是可写的目录。

- 记录响应文件并安装产品：

IBMIM -record <响应文件路径和名称> -log <日志文件路径和名称>

- 记录响应文件而不安装产品：

IBMIM -record <响应文件路径和名称> -log <日志文件路径和名称> -skipInstall <agentDataLocation>

3. 遵循安装软件包向导中的指示信息以作出安装选择。
4. 单击**完成**，然后关闭 Installation Manager。

此时已创建 XML 响应文件，且该文件位于命令中指定的位置。

以静默方式安装和运行 Installation Manager

使用 Installation Manager 以静默方式从命令行安装产品软件包。

对于静默安装，以下任务是必需的：

要以静默方式运行 Installation Manager，请运行 eclipse 子目录中适用于您的平台的命令：

Windows	IBMIM.exe --launcher.ini silent-install.ini -input <响应文件路径和名称> -log <日志文件路径和名称> 例如：IBMIM.exe --launcher.ini silent-install.ini -input C:\mylog\responsefile.xml -log C:\mylog\silent_install_log.xml
UNIX/Linux	IBMIM --launcher.ini silent-install.ini -input <响应文件路径和名称> -log <日志文件路径和名称> 例如，IBMIM --launcher.ini silent-install.ini -input /root/mylog/responsefile.xml -log /root/mylog/silent_install_log.xml

Installation Manager 以静默安装方式运行时，它读取响应文件并将日志文件写入您指定的目录中。响应文件是必需的；日志文件是可选的。这一执行操作的结果应当是：状态 0 表示成功，非零数字表示失败。

下表描述了与静默安装命令一起使用的参数：

参数	描述
-vm	指定 Java 启动程序。在静默方式下，始终在 Windows 上使用 java.exe，而在其他平台上使用 java。
-nosplash	不显示启动屏幕。

参数	描述
--launcher.suppressErrors	不显示 JVM 错误对话框。
-silent	以静默方式运行 Installation Manager 安装程序。
-input	指定用作 Installation Manager 的输入的响应文件。响应文件包含安装程序或 Installation Manager 运行的命令。
-log	(可选) 创建记录静默安装结果的日志文件。该日志文件是 XML 文件。

执行产品组件的静默升级

可通过以静默安装方式运行 Installation Manager 来静默升级 Rational Build Forge 产品组件。

静默升级需要以下先决条件:

- 现有 Build Forge 控制台安装必须已使用基于 Installation Manager 的静默安装进行了安装。
- Installation Manager 必须与 Build Forge 控制台主机安装在同一主机上。

对于静默安装, 以下任务是必需的:

1. 创建升级响应文件。
2. 指定响应文件为输入后, 以静默方式运行 Installation Manager。

有关 Installation Manager 和以静默方式进行安装的更多信息, 请参阅 Installation Manager 信息中心: <http://www.ibm.com/software/awdtools/installmanager/support/index.html>。

使用 Installation Manager 创建更新响应文件

通过记录您在使用 Installation Manager 安装产品软件包时的操作创建响应文件。

记录响应文件时, 在 Installation Manager UI 中进行的所有选择都将存储在 XML 文件中。

要为更新安装创建响应文件:

1. 运行 Installation Manager。在“首选项”中, 将更新安装的产品存储库 URL 添加到 IM 存储库列表中, 并务必选中该 URL。
2. 退出 Installation Manager。
3. 在命令行下, 切换至安装了 Installation Manager 的目录中的 eclipse 子目录。例如:

Windows	C:\Program Files\IBM\Installation Manager\eclipse
UNIX/Linux	/opt/IBM/InstallationManager/eclipse

4. 开始记录安装 (但不实际安装产品)。

为 response_file 和 log_file 输入完整路径, 包括文件名。确保您输入的文件路径存在。Installation Manager 不会为响应文件和日志文件创建目录。agentDataLocation 必须是可写目录。

```
IBMIM -record response_file -log log_file -skipInstall agentDataLocation
```

- 5. Installation Manager 启动。在 Installation Manager 中，单击**更新**，然后响应提示。
- 6. Installation Manager 完成后，单击**完成**。
- 7. 退出 Installation Manager。

此时已创建 XML 响应文件，且该文件位于命令中指定的位置。

以静默方式运行更新安装

使用 Installation Manager 以静默方式从命令行安装产品软件包。

要以静默方式运行 Installation Manager，命令的一般形式如下：

IBMIMc.exe --launcher.ini silent-install.ini -input *response_file* -log *log_file*

对 *response_file* 和 *log_file* 使用完整路径和文件名。

- Windows 示例

IBMIMc.exe --launcher.ini silent-install.ini -input C:\mylog\responsefile.xml -log C:\mylog\silent_install_log.xml

- UNIX 或 Linux 示例

IBMIM --launcher.ini silent-install.ini -input /root/mylog/responsefile.xml -log /root/mylog/silent_install_log.xml

Installation Manager 以静默安装方式运行时，它读取响应文件并将日志文件写入您指定的目录中。响应文件是必需的；日志文件是可选的。这一执行操作的结果应当是：状态 0 表示成功，非零数字表示失败。

下表描述了与静默安装命令一起使用的参数：

参数	描述
-vm	指定 Java 启动程序。在静默方式下，始终在 Windows 上使用 java.exe，而在其他平台上使用 java。
-nosplash	不显示启动屏幕。
--launcher.suppressErrors	不显示 JVM 错误对话框。
-silent	以静默方式运行 Installation Manager 安装程序。
-input	指定用作 Installation Manager 的输入的响应文件。响应文件包含安装程序或 Installation Manager 运行的命令。
-log	（可选）创建记录静默安装结果的日志文件。该日志文件是 XML 文件。

在 VMware 上安装 Build Forge 系统

可以在 VMWare 上安装和运行 Build Forge。

请使用以下准则：

- 安装 Build Forge 在单个主机（首选物理主机而不是 VMWare 映像）上使用的数据库。
- 将 VMware 工作站用于运行虚拟机的内存设置为至少 1GB。在 VMWare 工作站中，单击**编辑 > 首选项 > 内存**以调整该值。
- 您可能需要确定其他系统资源参数以最大限度提高 VMWare 上的性能。

在 Linux on System z 上安装管理控制台

使用安装介质随附的 `mc-<version>-<build>.tar.gz` tar 文件，在 z/Linux 上安装和配置管理控制台。IBM Installation Manager 不用于此安装。

z/Linux 的控制台与 IBM HTTP Server（而非 Apache Web 服务器）打包在一起。

安装管理控制台后，请将代理程序 rpm 包 (`zlinux-bfagent-<version>.rpm`) 安装在 z/Linux 上，以便为 Build Forge 设置 z/Linux 服务器。有关安装指示信息，请参阅第 133 页的『在 UNIX 和 Linux 系统上安装代理程序』。

在安装期间所需的信息。

在安装期间，将询问您以下信息。

1. 安装目录

- 提供要安装 Build Forge 的位置的绝对路径。

2. 数据库信息

- Build Forge 将使用的数据库类型（DB2、Oracle 或 MySQL）
- 数据库服务器主机名
- 数据库端口号
- 要使用的数据库名称
- 供 Build Forge 连接到数据库时使用的数据库用户名
- 数据库用户名的密码
- 用于访问数据库的客户机库的位置
- JDBC 驱动程序 JAR 文件的位置

3. 应用程序服务器信息

- 要使用的应用程序服务器（提供的 Tomcat 或您已设置的 WebSphere Application Server 安装）

如果选择 WebSphere Application Server，请提供更多信息：

- 将 Build Forge 服务组件安装在 WAS 上时该服务组件的位置。rbf-services 的域、端口和路径是必需的。示例：
`http://mydomain.com:9080/rbf-services`
- 用于插件的目录，该目录是可读且可写的目录。当前，仅当与 Rational Team Concert 集成时才使用该目录。
- WAR 部署目录：Build Forge 服务 WAR 的临时位置。安装后，请将其部署到应用程序服务器。
- Java 可执行文件 (`java.jar`) 的路径
- 服务层的临时存储器目录。该目录必须可读且可写。服务组件使用该目录来存储临时信息。
- Build Forge 服务的 HTTP 端口（缺省值为 3966）
- Build Forge 服务的 SSL 端口（缺省值为 49150）

4. Web 服务器信息

- 要使用的 Web 服务器（提供的 IBM HTTP Server 或您已设置的 Web 服务器）

如果选择提供的 IHS 服务器，请提供关于 SSL 的更多信息，包括是否使用 SSL、要使用的端口和内存、是使用现有安全证书还是创建一个安全证书。

运行安装程序

1. 转至您解压缩 tar 文件中控制台的程序包时所在的目录。
2. 运行以下命令：

```
./cmdline-install.sh
```

示例

以下附注释的列表显示如何继续安装程序步骤。这是一个注释运行，它不会影响实际安装。其中缺省值是可用的，这些缺省值显示在方括号中，例如 [y]。按 Enter 键以接受缺省值。

安装目录 [/opt/buildforge]

```
What database will you be using?
Enter the # of the database you will use
1) DB2
2) Oracle
3) MySQL
```

注意：列表的其余部分假定选择 Oracle。

```
What is your database hostname? 127.0.0.1
What is your database port number? [1521]
What is your database name? build
What is your database user name? build
What is your database user password?
Confirm your database user password?
```

```
Would you like this installation to create the Build Forge database schema? (y|n) [y]
```

此时请指定客户机库和信息。根据您选择的数据库，将出现如下所示的提示。使用绝对路径。

- DB2

Where are your 32-bit DB2 client libraries (libdb2.so.1)?

注意：库必须是 32 位库。输入绝对路径。

Where is your DB2 (DB2_HOME) installed?

- MySQL

Where are your MySQL client libraries (libmysqlclient.so)?

注意：库必须是 32 位库。输入绝对路径。

- Oracle

Where are your Oracle client libraries (libclntsh.so)?

置于：库必须是 64 位库。输入绝对路径。

Where is your Oracle instant client (ORACLE_HOME) installed?

注意：这是即时客户机的根目录的绝对路径。

Where is your tns.names file located (TNS_ADMIN)?

注意：这是包含 tns.names 文件的目录。输入绝对路径。

Please enter the directory with your database JDBC jar file?

注意: 请输入 ojdbc14.jar 的绝对路径。

Will you be using the supplied Tomcat app server? (y|n) n

注: 该清单的其余部分假定您正在使用 WAS 作为应用程序服务器。

Enter the full URL used to contact the services layer on your application server:

http://mydomain.com:9080/rbf-services

Enter the directory to install the Build Forge Services plugins to:

注意: 该目录应该是应用程序服务器主机上可读且可写的目录。当 Rational Team Concert 与 Build Forge 进行集成时, 服务组件将使用该目录。

Specify the war deployment directory:

注意: 请指定本地主机上的一个目录。当安装完成时, 安装程序会将服务 .war 文件放在此处。然后请将其部署到应用程序服务器。

Enter the path to a jar executable (which should be included in any JDK):

注意: 请输入数据库驱动程序的 .jar 文件的路径。

Enter the temporary storage path for the Services Layer:

注意: 请输入服务组件可以使用的目录的路径。该目录必须是可读且可写的。

What http port will the Build Forge services layer use? [3966]

What ssl port will the Build Forge services layer use? [49150]

Will you be using the supplied Apache web server? (y|n)[y]

Note: The rest of the listing assumes you entered 'y'

Would you like Apache to use SSL? (y|n)[n] Enter 'y' to configure Apache for SSL

注意: 该列表的其余部分假定您输入了“y”。

What ssl port will Apache use? [443]

Please enter a memory limit for PHP (in MB): [256]

Would you like to modify or specify a custom SSL certificate? (y|n) [n]

注意: 列表的其余部分假定您输入了“y”并且要创建定制证书。

Do you have an existing secure certificate? (y|n)[n]

A validity period is required for this cert please enter in [number][period] format

Examples: 10Y = 10 years, 6M = 6 months, 350D = 350 daysEnter the validity period for this cert:

Enter the common name for the certificate (usually the name of the server) [linux142.rtp.raleigh.ibm.com]:

Please enter your Locality/City:

Please enter your State/Province:

Please enter your Organization Name:

Please enter your Organization Unit:

Please enter your Country from the list below:

France

Taiwan

Italy

Germany

Korea

United States

China

Brazil
Spain
Japan

Please enter your Street Address:

A keystore password is required, and must be at least 6 characters long

Please enter a keystore password

如果您使用 IBM HTTP Server（而不是 Apache）作为 Web 服务器，请参阅第 495 页的『使用 IBM HTTP Server 而不是 Apache HTTP Server』，以了解有关设置 IBM HTTP Server 并为其启用 SSL 的更多信息。

启动控制台

1. 启动管理控制台:

```
<bfinstall>/rc/buildforge start
```

2. 验证服务组件（Apache Tomcat 服务器）是否已启动；打开 catalina.out 并验证是否记录了启动消息。

```
<bfinstall>/server/tomcat/logs/catalina.out
```

3. 启动 Web 浏览器并输入 z/Linux 标准主机名。例如：http://myhost.mycompany.com。

管理控制台将启动并显示登录提示。以 **root/root** 身份登录。

安装许可证文件

z/Linux 的许可证文件位于 <bfinstall> 目录中。许可证文件名为 IRBF_license。安装之后，它将包含一个文本消息，用于指示您从 Passport Advantage 下载实际许可证文件。

下载许可证文件并将其置于 <bfinstall> 目录中后，请配置 Build Forge 以使用该许可证文件：

1. 启动管理控制台。
2. 以 **root/root** 身份登录。
3. 选择**管理>系统**。
4. 找到许可证服务器设置并将其值设置为许可证文件的标准路径。

例如：<bfinstall>/IRBF_license。

为管理控制台启用 SSL

可以启用 SSL 来对 Build Forge 组件之间传送的数据进行加密：

- Web 浏览器客户机和 Apache HTTP Server
- Apache Tomcat Server 和 Apache HTTP Server

当出现提示时，如果您回答“是”，那么安装程序会执行启用 SSL 所需的某些作业。请完成本部分中所述的任務以启用 SSL：

1. 复审个人证书和密钥库
2. 将 IBM HTTP Server 配置为使用 SSL
3. 在管理控制台 UI 中启用 SSL
4. 启用对 SSL 的调试

注：其他安全功能，如密码加密和单点登录（SSO），在本发行版的 z/Linux 的 Build Forge 上不支持。

复审个人证书和密钥库

安装程序将创建以下密钥库：

密钥库	描述
buildForgeKeyStore.p12	包含受密码保护的 keyEntry（具有公用/专用密钥对的个人证书）。
buildForgeTrustStore.p12	包含受密码保护的 trustedCertEntry（仅具有公用密钥的证书）。
buildForgeKey.pem	包含受密码保护的专用密钥。
buildForgeCert.pem	包含未受密码保护的证书（具有与 buildForgeKey.pem 中专用密钥对应的公用密钥）。
buildForgeCA.pem	最初包含与 buildForgeCert.pem 相同的信息；将添加其他对等证书以建立信任关系。
buildForgeKeyForApache.pem	为 Apache HTTP Server 启用 SSL 时需要此密钥库。与 buildForgeKey.pem 不同，它不受密码保护，这就允许启动 Apache HTTP Server 而不提示输入密码。

请参阅第 112 页的『管理证书』以获取有关转换现有 PEM 证书和管理证书的更多信息。

将 IBM HTTP Server 配置为使用 SSL

请参阅第 495 页的『使用 IBM HTTP Server 而不是 Apache HTTP Server』中的“将 SSL 配置为使用 IHS”。必须更改密钥库格式并向 httpd.conf 添加条目。

在管理控制台 UI 中启用 SSL

使用管理控制台 UI（管理 > 安全）设置，以在管理控制台中启用 SSL 并更新 Build Forge 数据库。然后，检查是否在 bfclient.conf 配置文件中更新了必需的属性值。

- 1. 启动 Build Forge。
- 2. 登录到 UI。
- 3. 转至管理 > 安全。
- 4. 将“启用 SSL”更改为：是。
- 5. 单击保存。
- 6. 单击更新主 BFClient.conf。

启用对 SSL 的调试

要在管理控制台中对 SSL 的问题进行调试，请使用以下指示信息来记录 SSL 所需的其
他信息。

- 1. 启用引擎中的调试。设置以下环境变量，然后启动 Build Forge 引擎：

```
export BFDEBUG_SECURITY=1
```

 - a. 重新启动 Build Forge 引擎。
 - b. 重新启动 IHS。重新启动 IHS 将使 PHP 能够使用此调试参数。
- 2. 启用 Tomcat 中的调试。在 <bfinstall>/server/tomcat/common/classes/
logging.properties 中进行以下更改：
 - a. 添加以下行：

```
com.buildforge.level = ALL
```

b. 在处理程序部分中，将所有其他级别从 FINE 更改为 ALL。

重新启动 Tomcat 以使更改生效。

第 9 章 在管理控制台中配置其他功能

本部分描述了配置 Build Forge® 以启用其他功能或提供缺省配置的备选方案的方法。

Build Forge 配置文件 (buildforge.conf)

buildforge.conf 文件是 Build Forge 产品配置文件。它包含不同 Build Forge 组件启动 Build Forge 数据库并与其通信所用的配置设置。

buildforge.conf 文件存储在两个位置，如果需要在安装后进行修改，那么必须在这两个位置都进行更新。请参阅『更新 buildforge.conf 文件』。

- 在应用程序服务器使用的 rbf-services.war 文件中，也称为服务层。
- 在安装根目录中。下表列出了产品的缺省或标准安装目录：

Windows	C:\Program Files\IBM\Build Forge
UNIX/Linux	/opt/buildforge/Platform

更新 buildforge.conf 文件

buildforge.conf 文件位于两个位置，如果需要在安装后进行更新，那么必须在这两个位置都进行更新。

如果将 Build Forge 数据库移动到其他主计算机上，您可能需要编辑 buildforge.conf 文件以更新数据库主机。编辑此文件的另一个常见原因是更新您的数据库密码，为遵守网络安全策略，必须定期更改该密码。

使用以下过程来更新 buildforge.conf 文件，然后使用 buildforge.conf 的已更新副本重新创建 rbf-services.war 文件。

注：如果仅需要将 buildforge.conf 添加到 rbf-services.war，请跳过编辑 buildforge.conf 的步骤。

1. 停止 Build Forge 引擎。
2. 在 Build Forge 安装根目录中找到 buildforge.conf 文件。
3. 使用文本编辑器打开该文件，修改配置设置，并保存该文件。

注：需要 root 用户或 Administrator 特权才能编辑此文件。

4. 浏览到包含 rbf-services.war 文件的目录，例如：

Apache Tomcat 服务器	<bfinstall>/Apache/tomcat/webapps \$CATALINA_HOME/webapps
-------------------	--

5. 创建 rbf-services.war 的备份副本，例如：

Windows	copy rbf-services.war rbf-services.war.bak
UNIX/Linux	cp rbf-services.war rbf-services.war.bak

6. 删除 rbf-services 目录，例如：

Windows	rmdir rbf-services /s /q
UNIX/Linux	rm -rf rbf-services

7. 在 Windows 上，完成以下命令：

a. 重新创建类目录，例如：

```
mkdir WEB-INF\classes
```

b. 将已更新的 buildforge.conf 文件复制到新的类目录，例如：

```
copy <path_to_bf_conf>\buildforge.conf WEB-INF\classes
```

c. 按如下所述更新 rbf-services.war 文件：

```
jar -uvf rbf-services.war WEB-INF
```

8. 在 UNIX 和 Linux 上，完成以下命令：

a. 使用 unzip 实用程序来创建和填充 rbf-services 目录：

```
unzip -d rbf-services rbf-services.war
```

b. 将已更新的 buildforge.conf 文件复制到新的类目录，例如：

```
cp <path_to_bf_conf>/buildforge.conf rbf-services/WEB-INF/classes
```

c. 设置 rbf-services 中可执行文件的许可权。

```
chmod 755 <bfinstall>/server/tomcat/rbf-services/bin/*
```

9. 重新启动 Build Forge 引擎。

在 Windows 中，rbf-services 文件会重新部署，并使用已更新的 buildforge.conf 重新自动创建 rbf-services 目录。

Buildforge.conf 引用

buildforge.conf 文件存储关于 Build Forge 管理控制台运行方式的设置。

该文件在安装目录中。它是由安装程序自动构建的。如果需要编辑，可以将该文件另存为 ASCII 文本文件或 XML 文件。语法如下所述：

- 在一行中输入关键字及其值。
- 关键字和值之间用等号隔开（无空格）。例如，输入关键字：bf_file_storage=C:\Program Files\IBM\BuildForge\files

关键字	值
bf_file_storage	临时 Build Forge 文件所在的目录。示例： C:\Program Files\IBM\BuildForge\temp
bf_plugin_dir	连接到管理控制台的 IDE 插件所在的目录。
birt_home	Eclipse 报告工具（BIRT）的文件位置。
db_database	为了供控制台使用而创建的数据库的名称。
db_hostname	运行数据库的计算机的主机名/IP 地址。输入 db_hostname 的值时，请使用实际名称或 IP 地址。不要使用缺省值 localhost。
db_password	您为数据库用户名创建的密码。

关键字	值
db_provider	您选择以用于安装 Build Forge 的数据库。不要编辑此值。
db_schema	数据库的模式名称（通常与 db_username 相同，但是您可能已选择了其他模式名称）。
db_tcp_port	您正在使用的数据库连接端口。
db_type	安装 Build Forge 的数据库的类型。缺省值为 <code>odbc</code> 。不要编辑此值。
db_username	数据库的用户名。这是在运行安装程序之前设置的。 对于 DB2 和 DB2 Express，请在 Windows 而不是 DB2 中创建用户。 对于所有其他数据库类型，请创建数据库用户名。 请参阅第 34 页的『数据库设置』。 此值对于所有控制台类型都是必需的。
services_hostname	运行 Build Forge 服务层的计算机的主机名/IP 地址。它是服务配置中的标准域名（FQDN）。
services_ssl_port	用于安全地连接至 Build Forge 服务的 SSL 端口。
services_tcp_port	未指定 SSL 时用于连接至 Build Forge 服务的 TCP 端口。
services_url	指定用于服务层的端口的 URL。示例： <code>services_url http://mybfhost.com:8080</code>

将管理控制台配置为使用备用端口

可以在缺省端口 80 以外的其他端口上运行管理控制台。

关于此任务

可以通过以下两种方法将管理控制台配置为在备用端口上运行：

- 在安装期间，将该端口设置为期望的值。
- 如果已经安装管理控制台，请完成以下步骤：
 1. 更改 `httpd.conf`（对于 Windows 安装，位于 `<bfinstall>/Apache/conf/` 中；对于 *nix 安装，位于 `<bfinstall>/server/apache/conf/`）中的两项设置。例如，如果 `myHost` 是本地计算机，且您想使用端口 81，请指定以下设置：


```
Listen 81
ServerName myHost:81
```
 2. 启动控制台，并作为 `root` 用户或用具有管理特权的用户名登录。
 3. 选择**管理 > 系统**，将“控制台 URL”系统配置设置更改为运行管理控制台的 URL 和端口。示例：`http://myHost:81`（不使用结尾的斜杠）。
 4. 停止然后重新启动引擎。
 - Windows: 单击**开始 > 程序 > IBM Rational Build Forge 管理控制台 > 停止引擎服务**，然后单击**启动引擎服务**。

如果 Build Forge 正在前台中运行，请转至其正在其中运行的 Windows 控制台，然后输入 `Ctrl-C`。

- UNIX 或 Linux: 使用为 `rc` 文件所提供的脚本。

```
$ /opt/buildforge/rc/buildforge start$ /opt/buildforge/rc/buildforge stop
```

您也可以使用手动命令。

- a. 要停止，请找到进程标识并停止进程。

```
$ ps aux | grep buildforge
$ kill ${<PID>}
```

- b. 要启动，请使用以下命令，其中 *<bfinstall>* 是安装目录的路径：

```
<bfinstall>/Platform/buildforge
```

配置冗余

可以设置多台计算机以运行 Build Forge，所有计算机都与相同的 Build Forge 数据库通信。此设置称为冗余。

关于冗余

冗余有助于平衡作业处理，并在一个安装失败时增强可用性。

要点： 冗余不提供故障转移功能或其他高可用性功能。它仅增加作业处理容量。如果一个冗余安装失败，那么该安装所管理的所有在执行的作业都将丢失，但其余安装将继续处理其执行作业和接受新作业。

用户启动作业时，数据库中将创建作业的相应条目。进程引擎轮询数据库以查看是否有新作业。存在多个进程引擎时，将自然进行负载均衡，因为每个引擎在其非繁忙周期中都独立执行轮询。

设置冗余时，请执行管理控制台的正常安装，然后在其他主机上安装管理控制台。所有安装都配置为访问相同的 Build Forge 数据库。

要点： 每个安装都必须在自己的主机上进行。不能在相同主机上安装多个管理控制台。

安装冗余系统

关于此任务

这些指示信息假设您已设置数据库并对管理控制台进行了首次安装以使用该控制台。要在其他主机上创建附加安装，请执行以下操作：

过程

1. 根据数据库的指示信息在主机上执行安装前设置。根据数据库，此设置可能需要在主机上安装数据库客户机并执行其他配置。请参阅第 33 页的第 6 章，『安装前设置』。
2. 请确保数据库服务器配置为接受外部连接（TCP）。
3. 请遵循安装指示信息。在第 55 页的第 7 章，『安装管理控制台』处开始，执行以下操作：
 - 安装 Installation Manager（如有必要）
 - 启动 Installation Manager
 - 运行安装

以下步骤中仅描述需要特定输入的面板。

4. 在**安装程序包**：功能面板中，请确保选择了所有功能。
5. 在**数据库配置**面板中，请执行以下操作：
 - 提供数据库名称和模式名称。这些名称必须与第一个控制台上指定的名称相同。
 - 提供数据库用户名和密码。使用与在第一个控制台上指定的相同的用户名和密码。
 - 对是否要在安装时填充此数据库？，请选择否。

注意：

数据丢失风险：如果选择“是”，将覆盖数据库中的 **Build Forge** 数据库模式。首次安装的任何 **Build Forge** 操作中的所有数据都将丢失。

 - 单击**执行测试**。测试通过时，可以单击下一步以继续。

注：选中否时，会执行的唯一测试就是检查 JDBC 驱动程序的路径是否正确。
6. 在**控制台启动选项**面板中，已选中**不启动 Build Forge** 选项。该选项还将变灰，以便无法对其进行更改。需要在安装后手动启动该控制台。
7. 执行任何必需的**安装后配置**。请参阅第 171 页的第 11 章，『安装后任务』。如有必要，请对数据库进行编目，以便数据库客户机能够与之连接。

使用冗余

在设置冗余之后，可按如下方式使用：

- 将用户指向首次安装的 URL。
- 如果不希望用户访问它们，可以在附加安装上停止 Apache 服务器。
- 如果要增加用于处理 HTTP 请求的容量，请在所有安装上运行 Apache。安装负载均衡器以在各个安装中分发请求。

启用 IPv6 网络支持

可以对 IPv6 和混合 IPv6/IPv4 网络配置管理控制台。

关于此任务

对 IPv6 配置管理控制台时，需要执行以下操作：

1. 更改 httpd.conf 中的服务器条目。
2. 设置用于 IPv6 网络所需的 FLEXlm 许可证客户机。

复审将 Build Forge 用于 IPv6 的需求和配置步骤。请参阅第 25 页的『实现 IPv6 支持的联网需求』。

修改 httpd.conf

关于此任务

需要正确配置计算机和网络才能支持 IPv6。网络配置问题会使在 Build Forge 系统中指定的主机名和地址无法正确得以解析。

必须手动配置 Build Forge 用于 IPv6。为此，请修改主 Apache 配置文件 httpd.conf 中的条目：

1. 浏览到 httpd.conf 文件（buildforge/server/apache/conf/httpd.conf）。

- 2. 添加引用标准域名（FQDN）的 ServerName。例如，ServerName qlnx500-v6.ipv6.lexma.ibm.com
- 3. 如下修改来自 0.0.0.0:80 的 Listen 伪指令。
 - Windows: [::]:80
 - UNIX 或 Linux: 80

设置 FLEXlm 客户机

关于此任务

此任务仅适用于 Build Forge V7.1 至 7.1.3.3。在 IPv6 网络上运行 Build Forge 需要不同版本的 FLEXlm 许可证客户机。

随产品安装和配置的 FLEXlm 客户机支持 Rational License Server V7.0 和 V7.1。

要点：除非明确要求对 Rational License Server 7.1 使用 IPv6 地址，否则不要完成此任务。

要配置 FLEXlm 客户机的 IPv6 版本：

- 1. 停止 Build Forge 引擎。
- 2. 浏览到安装了 FLEXlm 客户机软件的目录：

Windows	C:\<bfinstall>\Apache\tomcat\webapps\rbf-services\bin, 其中 <bfinstall> 是 Build Forge 安装目录。 缺省安装目录为 C:\Program Files\IBM\Build Forge。
UNIX/Linux	<bfinstall>/server/tomcat/webapps/rbf-services/bin, 其中 <bfinstall> 是 Build Forge 安装目录。 在 UNIX/Linux 上, 缺省安装目录为 /opt/buildforge。

- 3. 重命名操作系统当前的 FLEXlm 客户机许可证文件以备份该文件，例如：

Windows	将 flexhelper.exe 重命名为 flexhelp.exe.bak。
UNIX/Linux	将 flexhelper-Linux-i386 重命名为 flexhelper-Linux-i386.bak。

- 4. 重命名 IPv6 的 FLEXlm 客户机许可证文件，以便产品将它用作当前 FLEXlm 客户机许可证文件：

Windows	将 flexhelper-IPv6.exe 重命名为 flexhelper.exe。
Linux	将 flexhelper-Linux-i386-IPV6 重命名为 flexhelper-Linux-i386。

- 5. 启动 Build Forge 引擎。

安全性功能

本部分描述在 Build Forge® 中启用安全性功能的方法：

- 安全登录，在安装期间缺省启用
- 单点登录（SSO）
- 启用 HTTPS 和 SSL

- 启用密码加密
- 用于安全性配置的 `bfclient.conf` 文件

通过在管理控制台中选择**管理 > 安全性**以及手工设置配置文件的组合操作来启用这些功能。本部分包括 `bfclient.conf`（用于启用安全性功能的配置文件）中的引用部分。

注：本部分不针对在 z/Linux 上运行 Build Forge 的用户。关于 z/Linux 上的 Build Forge 可用的安全功能的信息，请参阅第 77 页的『在 Linux on System z 上安装管理控制台』。

实施单点登录

随 Build Forge 提供了单点登录框架。

单点登录是允许用户访问应用程序而不必每次都输入用户名和密码的认证方案。Build Forge 提供了可以与第三方 HTTP 拦截器结合使用以实施单点登录的框架。

关于单点登录框架

Build Forge SSO 框架提供与市场上多个 SSO 解决方案集成的能力。该 SSO 框架基于拦截器，表示该框架拦截 HTTP 请求并提供处理该请求的方法。可以编写定制拦截器以接收并验证 HTTP 请求中的安全性工件。特别是拦截器可以在 HTTP 响应中设置令牌，然后在连续的请求中查找这些令牌。

Build Forge 随附提供以下两个 SSO 解决方案：

- 用于 SPNEGO（简单且受保护协商协议）的拦截器。请参阅第 93 页的『使用 Active Directory 域中的 SPNEGO 实施单点登录』。
- 用于 WebSphere SSO 集成的拦截器。请参阅第 98 页的『使用定制拦截器与 WAS 安全性集成』。

SSO 框架方法： SSO 拦截器是一个 Java 类，用于实施 Build Forge SSO 框架所用的接口：

```
com.buildforge.services.server.sso.ISSOInterceptor
```

它位于服务层组件中：

```
<bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/classes
```

该接口提供以下方法。

initInterceptor

装入拦截器时调用该方法。配置属性的映射将传递至 `initInterceptor()` 方法。配置属性在 Build Forge 控制台中创建（**管理 > 安全性 > SSO**）。

isTargetInterceptor

复审入站请求中的属性以确定该拦截器是否需要对那些属性执行操作。如果需要，该拦截器将负责以 `authenticateRequest()` 方法对请求进行认证。否则，将跳过该拦截器。拦截器选择假设配置了多个正在运行的拦截器。将按顺序使用这些拦截器。

authenticateRequest

使用请求中的数据认证该请求。它使用响应属性将数据发回客户机。

logoutRequest

在处理请求后，清除所有用户相关的安全性信息。

拦截器配置和排序: 拦截器配置在**管理 > 安全性 > SSO**中定义。以下配置随 Build Forge 提供:

- 表单 SSO 拦截器 - 缺省情况下处于活动状态，实施简单登录表单。
- SPNEGO SSO 拦截器 - 缺省情况下处于不活动状态，实施 SPNEGO 以执行认证。

实施拦截器类并将该类置于 Build Forge Apache Tomcat 应用程序服务器中之后，在此处配置新的 SSO 配置。该类是 SSO 配置的一个属性。

该列表的顺序确定了查阅拦截器以处理请求的顺序。可以配置多个拦截器来处理请求。在登录期间，将按顺序查阅每个拦截器。处理请求的拦截器是其属性对应于请求中的属性的第一个活动的拦截器。只有一个拦截器处理请求。始终是对 isTargetInterceptor 作出 true 响应的第一个拦截器。

注: 表单 SSO 拦截器应该保持为活动状态，以在出错时提供回退操作。定制拦截器在列表中应该位于表单 SSO 拦截器的前面。

添加定制的 SSO 拦截器: 要在 Build Forge 中创建定制拦截器，请执行以下操作:

1. 创建定制的 Java 类。

该类必须实施 ISSOInterceptor 接口。

2. 将定制类部署至服务层组件 WAR。

- a. 创建 JAR 文件，其中包含已编译的定制 SSO 拦截器类。
- b. 将该 JAR 文件复制到以下位置的 Build Forge 服务层组件中:
`<bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/classes`
- c. 在该目录中解压 JAR 文件。SSOManager 查找要在此处装入的类。
- d. 重新启动 Build Forge。

3. 可选: 定义环境。该环境可以作为属性对象传递至 initInterceptor() 方法。

- a. 在管理控制台中，转至**环境**。
- b. 单击**添加环境**。
- c. 定义 SSO 拦截器所需的所有属性以进行初始化。

4. 将 SSO 拦截器添加至 Build Forge:

- a. 在管理控制台中，转至**管理 > 安全性 > SSO**。
- b. 单击**添加 SSO 配置**。填充其属性:
 - **名称** - 输入 SSO 配置的名称。
 - **活动** - 设置为“是”。在认证请求过程中访问所有的活动配置。这些活动配置将按其在该面板中出现的顺序进行访问。
 - **Java 类** - 输入该类的完整软件包名称。给定的类只能指定给一个 SSO 拦截器。
 - **环境** - 如果定义了用于该 SSO 拦截器的环境，请选择该环境。
- c. 单击**保存**。

您的 SSO 拦截器此时将显示在列表中。

5. 对 SSO 配置进行排序。单击 SSO 拦截器左侧的图标，然后选择**移至顶部**。

在请求过程中，将按活动的 SSO 配置显示在该面板中的顺序对它们进行访问。您的配置必须置于**表单 SSO** 配置之前，因为缺省情况下表单 SSO 配置处于活动状态，且在访问时始终返回 true。缺省情况下，**SPNEGO SSO** 配置处于不活动状态。

示例 *authenticateRequest* 实施： 以下示例取自 WebSphere SSO 拦截器，该拦截器用于将 WebSphere 安全性与 Build Forge 集成。

该拦截器使用反射来查找 WebSphere 类 WSSubject。该类使用 `getCallerPrincipal` 方法返回用于登录至 `AuthServlet` 的主体。需要先保护 `AuthServlet`，然后 WAS 将向它进行认证。

有一些其他方法能够返回更多信息。有一些类似方法可用于任何应用程序服务器。

```
public Result authenticateRequest
    (Request requestAttributes, Response responseAttributes)
    throws SSOException {

    Result result = null;

    try {
        Class<?> cl =
            Class.forName("com.ibm.websphere.security.auth.WSSubject");
        Method theMethod = cl.getMethod("getCallerPrincipal",
            (Class[])null);
        String principal = (String)theMethod.invoke((Object[])null,
            (Object[])null);

        if (principal != null
            && principal.length() > 0
            && !principal.equals("UNAUTHENTICATED")) {
            result = new Result(Result.UseridOnlyOID, domain, principal);
            responseAttributes.setStatus(HttpServletResponse.SC_OK);
        } catch (Exception e) {
            throw new SSOException(e);
        }

        return result;
    }
}
```

在实施 `authenticateRequest` 期间，必须在返回之前设置响应状态：

- 如果无需执行任何重定向，且所得的信息完好，可以返回以下代码：
`responseAttributes.setStatus(HttpServletResponse.SC_OK);`
- 如果请求中没有足够的信息来继续进行有效的登录，请返回以下代码：
`responseAttributes.setStatus(HttpServletResponse.SC_FORBIDDEN);`
- 如果需要执行重定向以收集更多信息，请返回以下代码：
`responseAttributes.setStatus(HttpServletResponse.SC_MOVED_TEMPORARILY);`
`responseAttributes.sendRedirect(url);`

可以使用其他状态值。关于 `HttpServletResponse`，请参阅 `JavaDoc`。

从登录错误中恢复： 如果定制的拦截器在测试时未正常运行，那么最有可能的问题是认证。您将看到一个错误页面，其中显示以下信息：

Build Forge Error

Access is denied to the Build Forge console

"Error authenticating:
com.buildforge.services.common.api.APIException - API:

Authentication Error."

Please click [here](#) to try the same type of login again
or click [here](#) to force a form login (user ID/password).

有两个用于恢复的选项:

- 重试该登录。它将以相同方式再次浏览已配置的拦截器列表。
- 强制执行表单登录。该选择绕过了定制拦截器并使用表单登录页面。

方法源列表: 以下注释和源列表提供了 ISSOInterceptor 接口中有关方法的更多信息。

initInterceptor

```
/**
 * This method is called when the interceptor is loaded. A map of the
 * configuration properties is passed into the init method. You can create
 * the configuration properties from a BuildForge Environment and associate
 * it with the SSO configuration.
 *
 * @param initializationProps used to configure the implementation
 * @return true if successful, false if an error should be reported.
 * @throws SSOException if the initialization fails
 */
public boolean initInterceptor (Properties initializationProps) throws SSOException;
```

isTargetInterceptor

```
/**
 * This methods will review the attributes in the requestAttributes Map
 * to determine if there is something that this interceptor should
 * act on. If the interceptor return is "true", then the interceptor will
 * be responsible for authenticating the request and the authenticateRequest
 * method is invoked. If the interceptor return is "false", then this
 * interceptor is skipped and the next isTargetInterceptor in the list will
 * be called. Ordering of the interceptors during the configuration will
 * return which interceptor has the first shot at authenticating a request.
 *
 * @param requestAttributes attributes found in the inbound request
 * @return true if this interceptor will authenticate the request,
 *         false if it will not.
 * @throws SSOException
 */
public boolean isTargetInterceptor(Request requestAttributes) throws SSOException;
```

authenticateRequest

```
/**
 * This method is called on an interceptor that returns true for the
 * isTargetInterceptor method. The Request will contain data used
 * to perform the authentication. The Response is for the interceptor
 * to send information back to the client. The Result returned will contain
 * the following information if the status code is 200:
 *
 * * OID: an object identifier of the SecurityContext that can process token
 *       information stored in this map when going to an Agent.
 * * Domain: a valid BF domain name or <default> if not known
 *           (the username must be valid in the configured realm).
 * * Username: a valid BF username. This will be used to lookup BFUser attributes
 *             that are used in checking authorization policy.
 * * @see com.buildforge.services.common.security.context.Result
 *
 * @param requestAttributes attributes found in the inbound request
 * @param responseAttributes sent back in the outbound response
 * @return com.buildforge.services.common.security.context.Result - result
 *         information that tells BF how to handle the authentication request.
```

```

    * @throws com.buildforge.services.server.sso.SSOException
    **/
public Result authenticateRequest(
    Request requestAttributes,
    Response responseAttributes)
    throws SSOException;

LogoutRequest

/**
 * This method is called to logout a request. The first interceptor that
 * returns true for the isTargetInterceptor method will perform the logout.
 * The main point is to clean up any user-related security information that
 * should not be kept. The interceptor can inspect the request and response
 * objects to determine what needs to be removed.
 *
 *
 * @param requestAttributes attributes found in the inbound request
 * @param responseAttributes sent back in the outbound response
 * @return boolean - true if request redirect to exit page,
 *                  false if redirect to login page.
 * @throws com.buildforge.services.server.sso.SSOException
 **/
public boolean logoutRequest(
    Request requestAttributes,
    Response responseAttributes)
    throws SSOException;

```

使用 Active Directory 域中的 SPNEGO 实施单点登录

提供了简单且受保护的 GSS-API 协商 (SPNEGO) 机制，以在 Active Directory 域中实施单点登录

开始之前

该任务需要网络中的以下元素：

- Active Directory 域
- 目录服务器主机名
- Kerberos 密钥分发中心 (KDC) 主机名
- Active Directory 域中主机上的 Build Forge 安装
- Active Directory 域中的客户机主机
- 每个客户机上的 Kerberos 配置文件
- 安装在目录服务器主机上的 Windows Server 2003 SP2 资源工具箱
- 受支持的 Web 浏览器

注：不支持将 Internet Explorer 6 与 SPNEGO 一起使用。请使用受支持的浏览器。

以下过程包含基于以下设置的示例：

- mycompany.com 是用于域中所有主机的 TCP/IP 域的名称。
- ITDEV.COM 是 Active Directory 域的名称。
- it_directory.mycompany.com 是运行目录服务器的主机。它还运行 Kerberos KDC。
- it_domain.mycompany.com 是运行 Active Directory 域控制器的主机。
- it_buildforge.mycompany.com 是安装 Build Forge 的主机。
- bfuser 是 Build Forge 系统的域用户名。

- happy_user 是示例用户的域用户名，该示例用户将在 Web 浏览器中使用 SSO 来访问 Build Forge。

关于此任务

执行以下任务以在 Active Directory 域和 KDC 中实施 SPNEGO SSO。为每个任务提供了具有详细过程的部分。

注：SPNEGO 拦截器可用于 Active Directory 除外的任何 KDC。

过程

1. 设置 Active Directory 用户和服务主体。
2. 设置 Kerberos 文件。
3. 将 Build Forge 配置为使用 Active Directory 和 SPNEGO。
4. 配置浏览器客户机以进行安全访问。
5. 通过 SSO 访问 Build Forge。

设置 **Active Directory** 用户和服务主体:

Build Forge 服务器和 Build Forge 客户机必须在 Active Directory 域中进行设置。

开始之前

以下过程需要 Windows 2003 SP2 的支持工具。它们包含 setspn 命令，该命令对于在 Active Directory 中设置服务主体是必需的。从 Windows Server 2003 产品 CD 或 Microsoft Download Center 中安装支持工具。

关于此任务

Build Forge 客户机和服务器处于 Active Directory 域时，当登录至 Windows 主机时用户将生成 Kerberos 凭证令牌。然后当该用户尝试访问 Build Forge 服务器时，SPNEGO 拦截器接收用户令牌并验证该令牌。通过验证的标识通过配置的 Microsoft Active Directory LDAP 服务器传递至 Build Forge 来执行登录。

过程

1. 登录至域控制器主机。 在本示例中，主机为 it_example.mycompany.com。
2. 如果 Build Forge 主机尚不是成员，请将其添加至 Active Directory 域。 在本示例中，将主机 it_buildforge 添加至 ITDEV.COM 域。此时该主机在域中具有标准名称 it_buildforge.ITDEV.COM
3. 将 Build Forge 用户添加至 Active Directory 域。 在本示例中，创建用户 bfuser。

要点:

- 选择**密码从不到期**。可以选择其他密码管理。但是，将需要在每次密码到期时为 Build Forge 服务器输入新的密码。
 - 在**帐户**选项卡中，选择**帐户是信任委派**
4. 如果用户帐户不存在，请在 Microsoft Active Directory 中为所有的客户机创建用户帐户。 在本示例中，要创建一个名为 happy_user 的用户。

5. 为 Build Forge 创建服务主体名称 (SPN)。在本示例中, Active Directory 用户 bfuser 与服务名称 HTTP/it_buildforge.mycompany.com 相关联以创建 Build Forge 服务器的 SPN it_buildforge。

```
setspn -A HTTP/it_buildforge.mycompany.com bfuser
```

HTTP 是 Build Forge 服务的服务名称。

设置 **Kerberos** 认证的文件:

需要在 Build Forge 主机上设置启动文件 (Kerberos 客户机配置文件) 和密钥表文件。

过程

1. 在运行 Build Forge 的主机上设置启动文件。

- 对于 Windows 系统:
 - 将文件命名为 krb.ini, 并将其置于 C:\winnt。如果 C:\winnt 不存在, 请创建该目录。
 - 将 default_keytab_name 设置为 FILE:C:\winnt\krb5.keytab。
- UNIX 和 Linux 系统:
 - 将文件命名为 krb.conf, 并将其置于 C:\winnt。
 - 将 default_keytab_name 设置为 FILE:/etc/krb5.keytab。

以下示例文件是使用示例系统的域和区域设置针对 Windows 设置的。

```
[libdefaults]
default_realm = ITDEV.COM
default_keytab_name = FILE:C:\winnt\krb5.keytab
default_tkt_encypes = rc4_hmac
default_tgs_encypes = rc4_hmac
# kdc_default_options = 0x40800000
forwardable = true
renewable = true
noaddresses = true
clockskew = 300
[realms]
ITDEV.COM = {
    kdc = it_directory.itdev.com:88
    default_domain = mycompany.com
}
[domain_realm]
.mycompany.com = ITDEV.COM
```

注: 如果客户机主机和 Build Forge 服务器主机之间的时钟偏差大于 300 秒, 那么令牌将无效。将客户机和服务器主机上的时间、日期和时区设置在偏差限制内。

2. 设置 Kerberos 密钥表文件。当客户机尝试访问 Build Forge 服务器 URL 时, Build Forge 服务器将使用密钥表文件验证 Kerberos 令牌。在域控制器主机上使用 ktpass 命令来创建文件。ktpass 命令包含在必备 Windows 资源工具箱中。以下示例使用在示例场景中对 Build Forge 设置的 Build Forge 服务主体名称和 Active Directory 用户名。使用您自己的密码替换 -pass Rational。出于清晰性, 在示例中显示了换行符。请勿在 ktpass 命令中使用这些换行符。

```
ktpass -out C:\it_buildforge.keytab
-princ HTTP/it_buildforge.mycompany.com@ITDEV.COM
-mapuser bfuser -mapop set
-pass Rational /crypto RC4-HMAC-NT /rndpass /ptype KRB5_NT_SRV_HST
```

将 `it_buildforge.keytab` 重命名为 `krb5.keytab`，并将其置于 Build Forge 主机上包含 Kerberos 启动文件的目录中。

- 对于 Windows: `C:\winnt\`
- 对于 UNIX 和 Linux: `/etc`

将 **Build Forge** 配置为使用 **Active Directory** 和 **SPNEGO**:

过程

1. 在 Build Forge 中，将 LDAP 设置为指向 Active Directory 域控制器。
 - a. 在 Build Forge 中，单击**管理** > **LDAP**。
 - b. 按如下所示创建新的 LDAP 配置并设置属性来设置对域控制器的访问。
 - 名称: 设置为 Active Directory 域的名称。在示例环境中，这将是 `itdev`。
 - 管理员 DN: 设置为域中的管理员用户。
 - 映射访问组: 否
 - 主机: 设置为域控制器主机的 IP 地址。
 - 绑定用户帐户: 是
 - 协议: LDAP
 - 显示名称: `displayname`
 - 专有名称: `distinguishedname`
 - 组名: `memberof`
 - 邮件名称: `displayname`
 - 搜索条件: `on=users,do=domainname,do=domainextension`。在示例环境中，这将是 `on=users,do=itdev,do=.com`
 - 唯一标识: `sAMAccountNames=%`
 - c. 单击**设为缺省值**。该配置必须作为缺省 LDAP 配置。
2. 设置 SPNEGO 的 Build Forge 环境变量。
 - a. 在 Build Forge 中，转至**环境** > **SPNEGO SSO** 的环境。
 - b. 将 `bf_spnego_service_name` 设置为 `HTTP` 这将与服务主体名称相匹配。
 - c. 将 `bf_spnego_server_name` 设置为 Build Forge 服务器主机的标准主机名 `it_buildforge.mycompany.com`。如果未设置该变量，那么 `INetAddress` API 将尝试查找主机名。
 - d. 将 `bf_spnego_realm` 设置为 Kerberos 域名称 `ITDEV.COM`。如果未设置该变量，那么将使用 Kerberos 启动文件中的值。
3. 启用 SPNEGO 拦截器。
 - a. 在 Build Forge 中，转至**管理** > **安全性** > **SSO** > **SPNEGO SSO 拦截器**。
 - b. 将“活动”属性设置为“是”，然后单击**保存**。
 - c. 在**管理** > **安全性** > **SSO**中，将 **SPNEGO SSO 拦截器**移至列表顶部。使用 SPNEGO SSO 拦截器的“SSO 选项”菜单中的**移动至顶部**选择，然后单击**保存**。

配置 **SSO** 的客户机浏览器:

客户机浏览器必须将安全性设置设置为使用 SPNEGO。

关于此任务

对于访问 Build Forge 的浏览器（Microsoft Internet Explorer 或 Mozilla Firefox）使用客户机设置指示信息。

过程

- 对于 Internet Explorer，请执行以下操作。
 1. 登录至 Active Directory 域。在示例配置中，将登录至 itdev.com。
 2. 打开 Internet Explorer。
 3. 单击工具 > **Internet 选项** > 安全性。
 4. 选择本地内部网，然后单击站点。
 5. 在本地内部网对话框中，选中包含未在其他专区中列出的所有本地（内部网）站点，然后单击高级。
 6. 将运行 Build Forge 的主机添加至 **Web** 站点列表，然后单击确定。
 7. 单击确定以关闭本地内部网对话框。
 8. 在 **Internet 选项** 窗口中，单击高级选项卡。
 9. 滚动至“安全性”组，如果尚未选择启用已集成的 **Windows** 认证（需要重新启动），请选择该选项。
 10. 单击确定。
 11. 重新启动 Internet Explorer。
- 对于 Mozilla Firefox，请执行以下操作。
 1. 登录至 Active Directory 域。在示例配置中，将登录至 itdev.com。
 2. 打开 Firefox。
 3. 在地址字段中输入 **about:config**。
 4. 在“过滤器”框中，输入 **network.n**。该列表将自行更新。
 5. 双击 **network.negotiate-auth.trusted-uris**。输入受信域的列表。它应该包含目录服务器主机和 Build Forge 服务器主机（在示例中为 it_directory.mycompany.com 和 it_buildforge.mycompany.com）。单击确定。
 6. 设置授权。双击 **network.negotiate-auth.delegation-uris**，然后输入浏览器可以委托用户认证的站点列表。

通过 SSO 访问 Build Forge:

输入服务器 URL 以通过 SSO 测试登录。

过程

1. 使用处于 Active Directory 用户列表中的用户名登录至处于 Active Directory 域的主机。
2. 打开浏览器。
3. 输入 Build Forge 服务器主机的 URL。使用示例配置时，它将是 http://it_buildforge.mycompany.com。如果 SSO 配置正确，将看见 Build Forge 管理控制台。
4. 请验证显示在 Build Forge 控制台右上方的用户名是否与客户机的 Windows 登录名相匹配。

使用定制拦截器与 WAS 安全性集成

该部分描述了如何创建 SSO 拦截器以与 WebSphere Application Server (WAS) 安全性集成。

开始之前

所提供的表单 SSO 拦截器通过基于表单的登录页面认证用户。以下是如何创建定制 SSO 拦截器的示例。定制拦截器使用定制拦截器类。

拦截器类访问 WAS 以获取已认证的用户凭证。获取了这些凭证之后，将对这些凭证进行高速缓存。后续登录使用高速缓存后的凭证。

先决条件：必须使用 LDAP 用户凭证配置 WAS 中的用户。

注：必须建立包含 WAS 用户的 Build Forge LDAP 域（“缺省的”LDAP 服务器）。要执行该操作，请转至该 Build Forge LDAP 域，并选择“创建缺省值”。

保护授权服务 (**AuthServlet**)：

Build Forge 通常将其服务作为应用程序运行在所提供的 Apache Tomcat 应用程序服务器中。

关于此任务

以下指示信息将 Build Forge 配置为使用 WAS 而非 Tomcat。请按照第 491 页的『使用 WebSphere Application Server 而非 Apache Tomcat』部分中的指示信息进行操作，但有一个例外：rbf-services.war 文件包含名为 web.xml 的文件，您需要将其解压并修改以添加安全性约束。在 WAS 中安装该应用程序之前，需要在修改 war 文件后重新生成该文件，以便可以使用该文件的受保护版本。

要执行此操作，请遵循以下指示信息：

过程

1. 浏览至包含 rbf-services.war 文件的目录（Tomcat 服务器根中的 webapps 目录）。将该文件复制到临时位置，例如 C:\rbf。
2. 扩展 WAR 文件。从命令行中，运行命令：%IBM_JAVA_HOME%\jar -xvf rbf-services.war 以扩展 war 文件的内容。

注：Java 必须可用，且必须已创建 IBM_JAVA_HOME 环境变量。

3. 保存 rbf-services.war 文件以便稍后进行检索：
 - a. 对于 Windows: copy rbf-services.war rbf-services.war.bak
 - b. 对于 UNIX 或 Linux: cp rbf-services.war rbf-services.war.bak
4. 在 WEB-INF 目录中（从 war 文件中扩展的文件中）查找 web.xml 文件。编辑该文件以添加安全上下文。例如，在文件结束处，在 </web-app> 标记之前，添加以下各行：

```
<security-constraint id="SecurityConstraint_1">
  <web-resource-collection id="WebResourceCollection_1">
    <web-resource-name>*/</web-resource-name>
    <url-pattern>/AuthServlet/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>PUT</http-method>
```

```

        <http-method>DELETE</http-method>
    </web-resource-collection>
    <auth-constraint id="AuthConstraint_1">
        <description>myconstraint:+:</description>
        <role-name>User</role-name>
    </auth-constraint>
    <user-data-constraint id="UserDataConstraint_1">
        <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
</security-constraint>
<login-config id="LoginConfig_1">
    <auth-method>BASIC</auth-method>
    <realm-name>full-qualified-domain</realm-name>
</login-config>
<security-role id="SecurityRole_1">
    <role-name>User</role-name>
</security-role>

```

注: <auth-method> 可以是 WAS 支持的任何 J2EE 认证方法。最常见的认证方法是 FORM, 该方法需要附加的配置参数。请参阅 WebSphere 文档, 以获取有关在应用程序中配置 FORM 的指示信息。

5. 从命令行中, 通过运行以下命令重新生成 war 文件 (从抽取到的同一目录中调用):

```
%IBM_JAVA_HOME%\jar -cvf rbf-services.war
```

此时应该具有已修改的新版本 rbf-services, 以通过 J2EE 约束保护 AuthServlet。完成安装指示信息以在 WAS 中运行, 并通过[应用程序->安装新应用程序](#)来安装该版本的 rbf-services.war。请确保在配置 WAS 时, Build Forge 未运行。

这安装完后, 请转至[应用程序->企业应用程序](#)。单击应用程序名称以配置该应用程序。在[详细属性](#)下, 单击名为[安全角色到用户/组的映射](#)的链接。选择用户的所有[已认证](#)复选框。作出该更改后, 请确保保存至主配置。

还需要在 WAS 下启用应用程序安全性。要执行该操作, 请转至[安全性->安全管理、应用程序和基础结构](#)。请确保选中了[启用应用程序安全性](#)。

此时, 重新启动 WAS 服务器, 然后重新启动 Build Forge 服务器。

注: rbf-services 此时应该作为 WAS 启动进程的一部分启动, 因此 WAS 需要在 Build Forge 之前启动。

新建 SSO 配置:

新建 SSO 配置以使用拦截器。

过程

1. 在 Build Forge 控制台中, 转至[管理->安全性->SSO](#)。

2. 单击添加 SSO 配置。

3. 设置该配置的属性。

- **名称** - 输入该配置的名称。
- **Java** 类 - 输入 com.buildforge.services.server.sso.was.WebSphereSSOInterceptor
- **活动** - 选择“是”。

4. 单击保存。

5. 将该配置移动至列表的最前位置。 在配置名称左侧的菜单中, 选择**移动至顶部**。

将 **LDAP** 用户或 **LDAP** 组映射到本地用户:

将 LDAP 用户或 LDAP 组映射到本地用户, 如本地 root 用户。

关于此任务

创建 WebSphere SSO 拦截器后, 可以配置拦截器进行以下操作:

- 将 LDAP 用户映射到本地用户, 包括 root 用户
- 将 LDAP 组映射到本地用户, 如 root 用户

要设置映射, 请定义环境, 然后参考拦截器中的环境。

定义环境:

过程

1. 在 Build Forge 控制台中, 单击**环境**。
2. 为环境指定名称, 然后单击**保存环境**。
3. 要将 LDAP 用户映射到本地用户, 请设置并保存以下变量:
 - a. 将 `ldap_user_list_mapping` 设置为 `LDAP_user1|local_user1; LDAP_user2|local_user2; LDAP_user3|local_user3`
 - b. 将 `ldap_realm_name` 设置为 `LDAP_domain_name`

例如, 假设以下环境设置为:

```
ldap_user_list_mapping = user1|root;user2|root;user3|root
ldap_realm_name = bluepages.ibm.com:389
```

在这种情况下, LDAP user1、user2 和 user3 将均映射到本地 root 用户。

4. 要将特定的 LDAP 组成员映射到本地用户, 请设置并保存以下变量:
 - a. 将 `ldap_group_name_mapping` 设置为 `LDAP_group|local_user`
 - b. 将 `ldap_realm_name` 设置为 `LDAP_domain_name`

注: 如果使用两种映射类型, 用户映射优先级高于组映射。

如果为组映射, 假设以下环境设置为:

```
ldap_group_name_mapping = cn=bf_admin,ou=memberlist,ou=ibmggroups,o=ibm.com|root
ldap_realm_name = bluepages.ibm.com:389
```

在这种情况下, 为 LDAP 组成员的用户 `cn=bf_admin、ou=memberlist、ou=ibmggroups、o=ibm.com` 将都映射到本地 root 用户。

参考环境:

过程

1. 在 Build Forge 控制台中, 单击**管理->安全性->SSO**。
2. 单击 WebSphere SSO 拦截器的名称。
3. 设置**环境**属性以配置刚定义的环境。
4. 单击**保存**。
5. 将该配置移动至列表的最前位置。 在配置名称左侧的菜单中, 选择**移动至顶部**。

运行 SSO 定制拦截器:

现在可以使用新配置登录。

关于此任务

该定制的 SSO 拦截器此时可使用 WAS 安全性技术认证用户（通过 AuthServlet 请求作为用户传递至 Build Forge）。

过程

1. 打开 Web 浏览器。输入地址 `http://localhost`。
2. 此时将看到认证页面，而不是 Build Forge 登录表单。输入用户凭证，并单击 **Enter** 键。
3. 认证之后，请注意登录应该自动发生。
4. 注销之后，将显示缺省的 jsp 页面，而不是登录表单。如果该用户仍然处于已认证状态，后续的登录将自动进行。

还原至基于表单的 SSO:

您可以还原为使用 SSO 登录格式。

关于此任务

为了将系统重新配置为使用表单登录，您必须在 WAS 下卸载 rbf-services，并重新安装原始的 rbf-services.war 文件。在 Build Forge 下，请确保基于表单的 SSO 拦截器已启用并列为顶端项。禁用定制的 WAS 拦截器。WAS 和 Build Forge 需要重新启动，以便这些更改生效。

过程

1. 卸载 rbf-services 并重新安装原始的 rbf-services.war 文件。
2. 在 Build Forge 下，确保表单 SSO 拦截器已启用，并列为第一项（请参阅上述“Build Forge SSO 安全性配置”）。
3. 禁用定制的 WAS 拦截器。
4. 重新启动 WAS。
5. 重新启动 Build Forge。

启用 SSL 和 HTTPS

配置 Build Forge 系统以使用 SSL 和 HTTPS 可提高系统的安全性。SSL 包括端点认证和数据加密。

缺省情况下，Build Forge 系统仅针对登录表单使用 SSL，登录表单在 Apache Tomcat 上使用认证 servlet。在整个 Build Forge 系统上启用附加 SSL 保护需要以下设置：

1. 在 Apache 服务器上启用 SSL。仅当在安装期间未指定要启用 SSL 时，才需要该步骤。
2. 为客户机通信和内部通信启用 SSL
3. 为代理程序启用 SSL

注：如果要与 WebSphere 组件集成，请确保满足 SSL 支持的先决条件。

- 要了解有关与 WebSphere Application Server 集成的信息，请参阅第 491 页的『使用 WebSphere Application Server 而非 Apache Tomcat』。

关于缺省登录安全性

缺省情况下，Build Forge 包含登录安全性。当用户登录时，请求将重定向至认证 servlet。将对输入的用户名和密码进行加密，以供 servlet 使用。如果登录成功，那么将显示控制台用户界面主页。客户机与控制台之间的后续会话通信可能是通过 http（缺省情况）或通过 https 进行的。使用 https 需要对系统进行其他配置。请参阅第 101 页的『启用 SSL 和 HTTPS』。

在安装期间，您可提供用于加密的密钥库的密码。还可以选择安装自签名证书。

关于自签名证书的证书消息： 如果使 Build Forge 安装自签名证书，那么通过启用了安全性的浏览器访问系统的用户将得到有关证书的警告消息。

要防止显示这些警告，请将证书分发给用户以在其浏览器中安装。安装证书的具体细节因浏览器而异。请查阅浏览器文档。

该证书位于 `<bfinstall>/keystore` 中。

禁用缺省的登录安全性： 如果禁用登录安全性，那么在登录期间，将以未加密的明文形式将用户凭证传达给控制台。禁用登录安全性不会影响控制台使用 HTTPS/SSL（如果控制台配置为使用 HTTPS/SSL）。

要禁用认证 servlet，请执行以下操作：

1. 如果 Build Forge 正在运行，请停止该 Build Forge。
2. 编辑 `<bfinstall>/buildforge.conf` 以指定 HTTP 和端口 8080，用于和服务层通信。

更改该行：

```
services_url https://hostname:8443/rbf-services
```

将该行更改为以下格式：

```
services_url http://hostname:8080/rbf-services
```

3. 编辑服务层配置文件以关闭强制实施的 SSL。编辑 `<bfinstall>/Apache/tomcat/webapps/rbf-services/WEB-INF/web.xml`。将 ForceHttps 设置更改为 false。如果文件中不存在此设置，请按粗体所示添加。

```
<display-name>A Services Layer Bootstrap Servlet</display-name>
<servlet>
  <servlet-name>ServicesBootstrap</servlet-name>
  <servlet-class>com.buildforge.services.server.web.BootstrapServlet</servlet-class>
  <init-param>
    <param-name>port</param-name>
    <param-value>3966</param-value>
  </init-param>
  <init-param>
    <param-name>sslPort</param-name>
    <param-value>49150</param-value>
  </init-param>
  <init-param>
    <param-name>ForceHttps</param-name>
```

```
<param-value>false</param-value>
</init-param>
<load-on-startup>0</load-on-startup>
</servlet>
```

4. 启动 Build Forge。

注：如果禁用认证 servlet，那么用户凭证将通过网络以明文形式进行通信。这样会存在安全性风险。

关于 SSL 和 Build Forge 组件

缺省情况下，Build Forge 组件设置为在启用 SSL 时使用某些端口和安全性设置。

缺省 SSL 设置： 使用缺省证书时，启用 SSL 相对比较简单。本部分中的过程均基于该情况。

但是，通常建议不要在每个系统上使用相同的证书（专用密钥）。如果一个系统上的专用密钥泄密，那么可能危及整个基础结构的安全。通过强制实施物理安全性可以减少泄密的机会。

更安全的系统对于每个进程都使用一个证书。在 Build Forge 中，意味着您将执行以下操作：

- 为每个代理程序创建证书。
- 为每个引擎创建证书。设置冗余时，这是适用的。请参阅第 86 页的『配置冗余』。

此设置需要更多的证书管理。可以选择：

- 可以使用 CA（认证中心）来生成证书。这样做可以减少签署者交换的次数。
- 可以确保每个信任库或 CA 库具有建立连接所需的签署者。

以下部分说明 Build Forge 系统中强制实施 SSL 安全性的接口。

客户机接口： 用户通过客户机接口访问 Build Forge 系统。

Web 客户机访问 Build Forge

Web 客户机通过其 Apache Web 服务器访问 Build Forge。当启用 SSL 并使用启用安全性的 Web 浏览器时，将使用以下接口。

- **Apache Web 服务器端口 443**

Web 客户机通过其 URL 访问 Build Forge。启用 SSL 时，该 URL 为以下格式：

`https://host/`

host 是 Build Forge 运行的主机。如果设置 443 以外的端口用于对 Apache 进行安全访问，那么用户还必须指定端口：

`https://host:port/`

Web 客户机将重定向至在 Apache Tomcat 服务器上运行的认证 servlet。

- **Apache Tomcat 应用程序服务器端口 8443**

认证 servlet 接受登录凭证并认证用户。servlet 对凭证加密，以便凭证不会通过线路以明文格式显示。

Apache Tomcat servlet 所使用的侦听器端口的配置通过配置文件进行管理。配置位于 `<bfinstall>/Apache/tomcat/conf/server.xml` 中。查找以下连接器配置。

```
<Connector port="8443" maxHttpHeaderSize="8192" algorithm="IbmX509"
  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
  enableLookups="false" disableUploadTimeout="true"
  acceptCount="100" scheme="https" secure="true"
  clientAuth="false" sslProtocol="SSL_TLS"
  keystoreFile="C:\BuildForge71.536\keystore\buildForgeKeyStore.p12"
  keystorePass="password"
  keystoreType="PKCS12"
  truststoreFile="C:\BuildForge71.536\keystore\buildForgeTrustStore.p12"
  truststorePass="password"
  truststoreType="PKCS12"/>
```

API 程序客户机访问 Build Forge

• Apache Tomcat 应用程序服务器端口 49150

API 客户机通过其服务层组件（运行在 Apache Tomcat 应用程序服务器上的应用程序）访问 Build Forge。API 客户机需要具有有效的 `bfcclient.conf` 文件。

服务层组件使用 SSL 配置以进行入站通信。该配置在 Build Forge 控制台中定义（管理 > 安全性 > **SSL**）。使用的缺省值为缺省 **JSSE 入站 SSL**。

内部接口： Build Forge 由 Web 接口组件（Apache Web 服务器和 PHP）、服务层组件和引擎组件构成。Web 接口和引擎组件是服务层组件的客户机。API 程序客户机也是服务层组件的客户机。

服务层入站

Apache Tomcat 应用程序服务器端口 49150

服务层组件使用 SSL 配置以进行入站通信。该配置在 Build Forge 控制台中定义（管理 > 安全性 > **SSL**）。使用的缺省值为缺省 **JSSE 入站 SSL**。

服务层客户机出站

Apache Tomcat 应用程序服务器端口 49150 Web 接口组件（通过 PHP）和引擎组件都使用 SSL 配置，该 SSL 配置专用于至服务层组件的出站通信。该配置在 Build Forge 控制台中定义（管理 > 安全性 > **SSL**）。所使用的缺省值为缺省 **JSSE 出站 SSL**。

要使 SSL 握手能够成功，客户机出站配置和服务层入站配置的 SSL 属性必须兼容。管理 > 安全性 > **SSL**、类型和握手协议属性必须匹配。

每个 SSL 配置具有引用密钥库配置：

- 密钥库配置：定义含有专用证书的密钥库的属性。
- 信任库配置：定义含有受信签署者的密钥库的属性。

按名称指定配置。可以在管理 > 安全性 > **SSL** 中定义这些配置。提供了若干缺省值。

外部接口： 外部接口是 Build Forge 用于与外部系统对话的接口。

- Build Forge 引擎与代理程序进行通信。
- Build Forge 服务层组件与数据库进行通信。

Build Forge 引擎与代理程序通信

为该接口启用 SSL 需要以下操作:

- 配置代理程序。它需要在代理程序配置文件中进行一次更改,并将证书置于代理程序主机上。
- 对使用代理程序的每个服务器资源启用 SSL 通信。在控制台上的 **服务器**面板中执行该操作。

请参阅第 107 页的『为代理程序通信启用 SSL』。

Build Forge 服务层组件与数据库的通信

该接口的 SSL 配置在数据库的设备驱动程序中进行定义。

在 Apache 服务器上启用 SSL

缺省情况下, Build Forge 组件设置为在启用 SSL 时使用某些端口和安全性设置。

安装期间,在“Web 和应用程序服务器”面板中会询问您以下问题: **是否希望使用安全的 HTTP?**

如果回答“是”,那么会在 Apache 服务器上启用 SSL。跳过此部分。

如果回答“否”,但是您确实希望启用 SSL,请登录到 Build Forge 主机并按如下所述配置 Apache 服务器:

- httpd.conf 文件

编辑 `<bfinstall>/Apache/conf/httpd.conf` 以使用以下设置:

```
Listen 0.0.0.0:443
ServerName localhost:443
```

- ssl.conf 文件

编辑 `<bfinstall>/Apache/conf/ssl/ssl.conf` 以使用以下设置:

```
<VirtualHost *:443>
ServerName localhost:443
SSLEngine on
```

可以选择指定该主机的标准域名。

为客户机和内部连接启用 SSL

缺省情况下, Build Forge 组件设置为在启用 SSL 时使用某些端口和安全性设置。

要启用从客户机到 Build Forge 以及 Build Forge 内部组件之间的 SSL 连接,请严格按照指定的顺序执行以下操作:

- 在控制台中,转至**管理 > 安全性**。
- 将**启用 SSL** 设置为“是”。此时会显示几个其他属性。将它们保留为缺省值。如果需要,可在以后对它们进行定制。
- 单击**保存**。将在 Build Forge 数据库中保存 SSL 配置。
- 单击**更新主 BFClient.conf**。SSL 配置用于更新 BFClient.conf 文件。要允许 Build Forge 使用这些设置,它们必须在此文件中。
- 停止并重新启动 Build Forge。这是必需的。

一旦证明系统在使用缺省设置,您就可以编辑影响 SSL 的属性。

注: 如果希望客户机使用 SSL 但是不希望在内部客户机 (Web 接口组件和引擎组件) 与服务层组件之间使用 SSL, 请执行以下操作:

- 手动编辑 BFCClient.conf 文件。该文件位于 `<bfinstall>` 中。
- 更改 `bf_services_preferred_protocol` 设置 `tcp`。
`bf_services_preferred_protocol to tcp`

如果已在物理上对 Build Forge 主机进行了保护, 那么此配置将提高性能而使安全风险最小。

在已锁定的系统上重新启用 TCP 通信

如果 SSL 中存在错误配置, 那么系统会拒绝您访问。

要获取对已锁定系统的访问权, 请执行以下操作:

- 停止 Build Forge。
- 在 `<bfinstall>` 目录中, 使用文本编辑器打开 `bfclient.conf` 文件。
- 按如下所述更改协议属性:
`bf_services_preferred_protocol=tcp`
- 启动 Build Forge。

您应该能够登录。

注: 更改协议并不会禁用安全登录认证, 它在缺省情况下是启用的。用户被重定向到安全连接, 该安全连接允许登录凭证到 Build Forge 的安全通信。

为 API 客户机 (Perl 或 Java) 启用 SSL

使用 `bfclient.conf`, 为以 Perl 或 Java 编写的 API 客户机配置 SSL 连接。

关于此任务

要将 SSL 与 API 客户机程序连接, 请设置 `bfclient.conf`。此文件包含 SSL 配置属性。要简化此过程, 下面的步骤假设您运行第一个步骤中提到的客户机目录中的客户机。

过程

1. 创建要运行脚本的客户机目录。

Windows: `mkdir c:\client`

UNIX 或 Linux: `mkdir /temp/client`

2. 将 `bfclient.conf` 复制到客户机目录中。

Windows: `copy c:\BuildForge\bfclient.conf c:\client`

UNIX 或 Linux: `cp /opt/BuildForge/Platform/bfclient.conf /temp/client`

3. 在客户机目录中创建密钥库子目录。

Windows: `mkdir c:\client\keystore`

UNIX 或 Linux: `mkdir /temp/client/keystore`

4. 将密钥库从 BuildForge 安装复制到客户机密钥库目录中。

Windows: copy \BuildForge\keystore*.pem \client\keystore

UNIX 或 Linux: cp /opt/BuildForge/Platform/keystore/*.pem /temp/client/keystore

5. 对于 Perl, 请设置以下 OS 环境变量以找到 bfclient.conf 位置。

Windows: set BFCLIENT_CONF=c:\client\bfclient.conf

UNIX 或 Linux: export BFCLIENT_CONF="/temp/client/bfclient.conf"

6. 对于 Java, 请完成以下步骤:

- 使用代码中的 SecureAPIClientConnection 而不是 APIClientConnection 来使用 bfclient.conf 进行 SSL 连接。
- 运行脚本时, 在 Java 命令行中社自豪以下系统属性。

Windows: -Dcom.buildforge.client.config=c:\client\bfclient.conf

UNIX 或 Linux: -Dcom.buildforge.client.config=/temp/client/bfclient.conf

7. 编辑 bfclient.conf 并检查以下属性:

- bf_services_preferred_protocol=ssl

请确保值为 SSL 而不是 TCP。

- bf_services_ssl_port=49150

请确保 49150 是服务层 SSL 端口。

- bf_keystore_location=./keystore/buildForgeKey.pem

存在多个密钥库位置。如果在除客户机目录以外的目录中运行脚本, 请更改每个密钥库位置以使用标准路径。

8. 如果您希望确保脚本正在正确使用 bfclient.conf, 请在环境中设置以下调试属性。运行脚本时, 应参阅更多有关 SSL 连接属性的输出。

Windows: set BFDEBUG_SECURITY=1

UNIX 或 Linux: export BFDEBUG_SECURITY=1

为代理程序通信启用 SSL

缺省情况下, Build Forge 组件设置为在启用 SSL 时使用某些端口和安全性设置。

要启用 Build Forge 和代理程序之间的 SSL 通信, 需要执行以下操作:

- 在 UNIX 计算机上, 确保托管代理程序的计算机上安装了 GCC 库。
- 先决条件: 启用 SSL 以进行客户机通信和内部通信。请参阅第 105 页的『为客户机和内部连接启用 SSL』。
- 配置每个代理程序。该任务包括:
 - 将证书添加至代理程序主机
 - 编辑代理程序的 bfagent.conf 文件。
- 在控制台中, 在连接至代理程序的每个服务器定义中启用 SSL。

配置代理程序使用 SSL

1. 如果该代理程序正在运行，请停止该代理程序。
2. 将证书的 .PEM 文件置于该代理程序的根安装目录。

要快速实施和测试 SSL，可以从 Build Forge 安装中复制 .PEM 文件。这些文件位于 `<bfinstall>/keystore`。

SSL 的最佳实践是对每个代理程序使用不同的证书：

- a. 对以下每一项创建不同的密钥库文件（.PEM）：
 - 专用密钥（密钥）
 - 专用密钥的公共证书（证书）
 - 受信签署者（CA 或认证中心）
 - b. 如果要针对代理程序使用唯一的证书（而不是 Build Forge 引擎证书的副本），请将针对代理程序的证书添加至 Build Forge 的认证中心密钥库 `<bfinstall>/keystore/buildForgeCA.pem` 中。如果正在运行多个引擎（冗余引擎），请将证书添加至每个引擎的认证中心密钥库。
 - c. 如果您要在 System z 上使用 kdb，请使用 `gskkeyman` 创建 kdb。
 - d. 如果您要使用系统授权工具 (SAF) 密钥环，请使用 `RACF` 创建密钥环，并将证书连接到该密钥环。
3. 编辑 `BFAgent.conf`。该文件中将注释掉以下各行：除去注释前缀。

```
ssl_key_location buildForgeKey.pem
ssl_key_password password
ssl_cert_location buildForgeCert.pem
ssl_ca_location buildForgeCA.pem
ssl_protocol TLSv1
ssl_cipher_group ALL
```

`password` 代表 `buildForgeKey.pem` 密钥库。如果想要对该密钥库加密，请参阅第 123 页的『将 `buildforge.conf` 和 `bfagent.conf` 中的密码加密』。

如果需要在建立与代理程序的连接时进行客户机认证，请取消下行的注释：

```
ssl_client_authentication true
```

此设置要求引擎证书添加至代理程序的认证中心密钥库 `buildForgeCA.pem`。

如果想要使用特定的密码，请取消该行的注释，并添加密码列表：

```
ssl_cipher_override cipher_list
```

4. 对于在 System z 上运行的代理程序，请编辑 `BFAgent.conf` 并取消以下行的注释：

```
gsk_ssl_key_location <root/cert SAF>
gsk_ssl_key_location /etc/key.kdb
gsk_ssl_kdb_password <password>
gsk_keyring_label <certificate label>
gsk_ssl_protocol ALL
gsk_ssl_cipher_v2 6321
gsk_ssl_cipher_v3 0906030201
gsk_ssl_client_authentication true
gsk_password_encrypt false
```

仅使用一个 `gsk_ssl_key_location`。您可以选择 SAF 密钥环或 kdb。SAF 密钥环由 `<userid>/<keyring>` 指定。

5. 启动代理程序。代理程序必须正在运行，以测试控制台发出的连接。

在服务器定义中启用 SSL

该控制台使用服务器定义连接至代理程序。

对于连接至启用了 SSL 的代理程序的每个服务器定义，执行以下操作：

- 在控制台中，转至**服务器**面板。
- 单击服务器定义名称。
- 在服务器定义的**详细信息**选项卡中：
 - 将**启用 SSL** 设置为“是”。
- 单击**保存**。
- 单击**测试连接**。

对与代理程序的 SSL 通信进行故障诊断

以下核对表描述了启用 SSL 时的常见问题：

- 代理程序
 - 尽管在安全性和服务器定义中启用了 SSL，但是未配置代理程序 SSL。
 - Build Forge 引擎不信任代理程序的证书。该代理程序证书需要添加至该引擎的 CA 密钥库：
`<bfinstall>/keystore/buildForgeCA.pem`
 - BFAgent.conf 中指定的密钥库密码不正确。
 - 在 BFAgent.conf 中指定了客户机认证，但引擎的证书尚未添加至代理程序的认证中心 buildForgeCA.pem。
- Build Forge 控制台
 - 在控制台上启用了 SSL。它尚未保存，或在保存之前单击了**更新主 Bfclient.conf**，或在保存之后从未单击**更新主 Bfclient.conf**。
- 服务器定义
 - 未成功启用 SSL。必须单击**保存**，然后单击**测试连接**。
- 引擎和代理程序设置相匹配
 - 握手协议不匹配。引擎 SSL 配置和代理程序 SSL 配置的握手协议必须设置为相同的值：TLSv1 或 SSLv3。缺省值为 TLSv1。
 - 密码套件不匹配。在引擎 SSL 配置和代理程序 SSL 配置中指定的密码套件必须具有公共的密码。缺省的密码套件组为 ALL。

启用调试消息

可以在引擎和代理程序中启用调试。启用调试后，将产生更多的详细输出，这些输出有助于识别配置中的问题。

- 启用引擎调试：
 1. 在启用了 SSL 的服务器定义所用的环境中，添加以下变量：
`BFDEBUG_SECURITY=1`
 2. 停止引擎，并重新启动该引擎。
 - 在 Windows 上，请在前台启动引擎。输出将显示在命令窗口中。
 - 在 UNIX 或 Linux 上，请在打开调试的前提下启动引擎：

```
cd <bfinstall>/rc
./buildforge start
```

引擎的输出将写入 <bfinstall>/log 中的引擎日志文件。

- 启用代理程序调试:
 1. 停止代理程序。
 2. 将下行添加至 BFAgent.conf:

```
activity_log bfaagent.log
```

在本示例中，代理程序将输出写入 bfaagent.log 中。可以指定不同的文件名。

注：如果代理程序作为服务运行，指定绝对路径。

3. 启动代理程序。

SSL 连接成功时的示例引擎调试输出

引擎成功连接到代理程序时会产生以下输出。

```
SSL_ca_file: ./keystore/buildForgeCA.pem
SSL_cert_file: ./keystore/buildForgeCert.pem
SSL_key_file: ./keystore/buildForgeKey.pem
SSL_verify_mode: 0x01
SSL_version: TLSv1
SSL_cipher_list: ALL
SSL_use_cert: 1
Making as SSL connection using socket IO::Socket::INET=GLOB(0x1e8f0f4).
SSL connection to agent.
DEBUG: .../IO/Socket/SSL.pm:1387: new ctx 80662848
DEBUG: .../IO/Socket/SSL.pm:880: dont start handshake: IO::Socket::SSL=GLOB(0x1e8f0f4)
DEBUG: .../IO/Socket/SSL.pm:284: ssl handshake not started
DEBUG: .../IO/Socket/SSL.pm:327: Net::SSL::connect -> 1
DEBUG: .../IO/Socket/SSL.pm:382: ssl handshake done
Socket is of type: ref(IO::Socket::SSL=GLOB(0x1e8f0f4))
ReadyLine: 202 HELLO TLS - BuildForge Agent v_VERSION_
.
Storing Agent Version [999.999.999.999-999-9999] for [08974C8E-6C3B-1014-972D-D9B2901D9F42]
cmd ping
username pbirk
encpass c1713f4a31af3f1300f7b2414a24559c4d6097e07310cf9c412e
goSending agent request...
```

SSL 连接成功时的示例代理程序调试输出

正常运行的代理程序建立 SSL 连接时会生成以下输出。

```
[2256] main.c          : 409: === NEW AGENT ===
[2256] io.c             : 264: In start_SSL
[2256] io.c             : 89: Key location: buildForgeKey.pem
[2256] bfpwdlocloader.c: 134: Looking for password locator: ssl_key_password_locator
[2256] bfpwdlocloader.c: 244: Looking for password for prop
                        ssl_key_password from bfaagent.conf.
[2256] bfcryptloader.c : 202: Loading password encryption module.
[2256] bfcryptloader.c : 276: Password encryption property
                        password_encrypt_module is not configured.
[2256] bfcryptloader.c : 539: Password decoded.
[2256] io.c             : 98: Cert location: buildForgeCert.pem
[2256] bfpwdlocloader.c: 134: Looking for password locator:
                        ssl_cert_password_locator
[2256] bfpwdlocloader.c: 244: Looking for password for prop
                        ssl_cert_password from bfaagent.conf.
[2256] io.c             : 153: Setting key password in default userdata.
[2256] io.c             : 160: Getting private key from PEM.
```

```

[2256] io.c : 166: Checking private key from PEM.
[2256] io.c : 172: Getting CA store information.
[2256] bfpwdlocloader.c: 134: Looking for password locator:
      ssl_ca_password_locator
[2256] bfpwdlocloader.c: 244: Looking for password for prop
      ssl_ca_password from bfagent.conf.
[2256] io.c : 178: CA location: buildForgeCert.pem
[2256] io.c : 184: Checking the CA store.
[2256] io.c : 230: Returning from init_CTX.
[2256] io.c : 281: Calling SSL_new
[2256] io.c : 294: Calling SSL_accept.
[2256] io.c : 346: Cipher chosen: AES256-SHA
[2256] io.c : 367: ssl_state = SS_CERTIFIED

```

代理程序上密钥库密码出错时的示例输出

如果代理程序端上配置的密钥库密码错误，那么该密码将同时显示在引擎输出和代理程序输出中。

引擎输出（摘录）：

```

SSL_use_cert: 1
Making an SSL connection using socket IO::Socket::INET=GLOBAL(0x1e8f0f4).
SSL connection to agent.
DEBUG: .../IO/Socket/SSL.pm:1387: new ctx 80662848
DEBUG: .../IO/Socket/SSL.pm:880: dont start handshake: IO::Socket::SSL=GLOBAL(0x1e8f0f4)
DEBUG: .../IO/Socket/SSL.pm:284: ssl handshake not started
DEBUG: .../IO/Socket/SSL.pm:327: Net::SSL::connect -> -1
DEBUG: .../IO/Socket/SSL.pm:1135: SSL connect attempt failed with unknown error
error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number

DEBUG: .../IO/Socket/SSL.pm:333: fatal SSL error: SSL connect attempt failed with unknown error
error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number
DEBUG: .../IO/Socket/SSL.pm:1422: free ctx 80662848 open=80662848 80566656
DEBUG: .../IO/Socket/SSL.pm:1425: OK free ctx 80662848

```

代理程序输出（摘录）：

```

[ 5272] io.c : 98: Cert location: buildForgeCert.pem
[ 5272] bfpwdlocloader.c: 134: Looking for password locator: ssl_cert_password_locator
[ 5272] bfpwdlocloader.c: 244: Looking for password for prop ssl_cert_password from bfagent.conf.
[ 5272] io.c : 153: Setting key password in default userdata.
[ 5272] io.c : 160: Getting private key from PEM.
[ 5272] io.c : 218: Failure reason: SSL_ERROR_BAD_KEY_FILE
[ 5272] io.c : 221: OpenSSL error string: error:00000000:lib(0):func(0):reason(0)
[ 5272] io.c : 281: Calling SSL_new
[ 5272] platform.c :2693: platform_release_credentials
[ 5272] main.c : 412: --- EXITING ---

```

代理程序输出中的错误代码

该列表包含可能遇到的某些其他错误代码及导致这些错误代码的原因：

- **SSL_ERROR_BAD_CA**: 在 buildForgeCA.pem 中装入签署者时出现问题。该问题可能是由于添加签署者的方式而导致的文件格式的问题。
- **SSL_ERROR_BAD_CERT**: 在 buildForgeCert.pem 中装入证书时出现问题。原因可能是该证书与专用密钥不匹配或 PEM 中的该证书已损坏。
- **SSL_ERROR_BAD_KEY_FILE**: 为 BFAgent.conf 中的 ssl_key_password 属性指定的 buildForgeKey.pem 密码出现问题。
- **SSL_ERROR_BAD_KEY**: SSL_CTX_check_private_key 返回的值不为 1。专用密钥格式无效或该格式与证书不匹配。

- **SSLERrorFIPSEnablement**: 启用 FIPS 时发生了错误。这通常由于在 FIPS 自检中出现问题。该错误可能是内部错误。
- **SSLERrorInvalidCipher**: 密码规范与 OpenSSL 允许的规范不匹配。检查 BFAgent.conf 中的属性 ssl_cipher_group 或 ssl_cipher_override 上指定的密码。
- **SSLERrorNoCtx**: 新建 SSL CTX 对象时出现问题。该错误可能是内部错误。

管理证书

Build Forge 使用的证书和密钥库可以在安装后修改。

安装期间，您可以指定要使用的证书（自己的或 Build forge 生成的）和密钥库密码。本部分描述以下过程：

- 将 PEM 按键转换为 Build Forge 所需的 OpenSSL 和 JSSE 按键
- 更改按键密码并修改 Build Forge 以使用新的密码。
- 创建新的自签名证书
- 使用客户机证书登录
- 下载不受限制的策略文件，用于高强度加密的证书。

将 **PEM** 密钥库转换为 **Build Forge** 密钥库:

从认证中心接收到的 PEM 密钥库可以转换为用于 Build Forge 的密钥库。

开始之前

为 SDK 下载不受限的策略文件。仅当密钥大小对于受限策略文件而言太大时，此必备文件才适用。从以下网址下载文件：[//www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=jcesdk&lang=en_US&S_PKG=142ww](http://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=jcesdk&lang=en_US&S_PKG=142ww)

注：必须使用 IBM 提供的 keytool 实用程序。

关于此任务

如果您有来自认证中心的一组 PEM 文件，必须用它们为 Build Forge 创建一组 OpenSSL 和 JSSE 按键。

过程

1. 在您的 PATH 中包括 Build Forge 工具目录。
 - `<bfinstall>/openssl`
 - `<bfinstall>/ibmjdk/bin` (Windows)
 - `<bfinstall>server/ibmjdk/bin` (UNIX 或 Linux)

对于 UNIX 和 Linux，在 LD_LIBRARY_PATH 中包括以下目录：

`<bfinstall>/openssl`

2. 将 PEM 文件转换为 PKCS12 密钥库。

请使用以下命令：

```
openssl pkcs12
    -export
    -name "buildforge"
    -out buildForgeKeyStore.p12
```

```
-inkey <key.pem>
-passin pass:<pempassword>
-in <crt.pem>
-password pass:<bfpasword>
```

3. 验证证书已添加且可读。

```
keytool -v
-list
-keystore buildForgeKeyStore.p12
-storepass <bfpasword>
-storetype pkcs12
```

如果您收到有关密钥大小无效的错误，请下载未受限制的策略文件。使用本部分开始处的指示信息。

4. 导出公用证书。

在命令窗口中，转至 *<bfinstall>/keystore*，然后运行以下命令：

```
keytool -export
-alias buildforge
-file cert.der
-keystore buildForgeKeyStore.p12
-storepass <bfpasword>
-storetype pkcs12
```

- 证书存储在文件 *cert.der* 中。
- 使用在安装期间为密钥库指定的同一 *<bfpasword>*。否则，您需要更改配置。

5. 创建信任库并导入公用证书。

在命令窗口中，转至 *<bfinstall>/keystore*，然后运行以下命令：

```
keytool -import
-noprompt -trustcacerts
-alias buildforge
-file cert.der
-keystore buildForgeTrustStore.p12
-storepass <bfpasword>
-storetype pkcs12
```

6. 将公用客户机证书放在 *buildForgeCert.pem* 中。

在命令窗口中，转至 *<bfinstall>/keystore*，然后运行以下命令：

```
openssl pkcs12 -clcerts -nokeys
-in buildForgeKeyStore.p12
-passin: pass:<bfpasword>
-out buildForgeCert.pem
```

7. 将证书和密钥放在 *buildForgeKey.pem* 中

在命令窗口中，转至 *<bfinstall>/keystore*，然后运行以下命令：

```
openssl pkcs12
-in buildForgeKeyStore.p12
-passin pass:<bfpasword>
-passout pass:<bfpasword>
-out buildForgeKey.pem
```

8. 创建 PEM 认证中心 *buildForgeCA.pem*。

- 将 CA 根证书下载到 *<bfinstall>/keystore* 中。该证书名为 *CARootCert.crt*。需要将其添加到 PEM 密钥库中，并且可以导入 *buildForgeTrustStore.p12* 中。
- 在命令窗口中，转至 *<bfinstall>/keystore*，然后运行以下命令：

```
cat CARootCert.crt > buildForgeCA.pem
keytool -import -noprompt -v -trustcacerts
        -alias "CA Root"
        -file CARootCert.crt
        -keystore buildForgeTrustStore.p12
        -storepass <bfpassword>
        -storetype pkcs12
```

结果

Build Forge 使用受密码保护的 PEM 密钥库 buildForgeKey.pem。Apache 服务器启动时将提示输入该密码。

如果不希望在启动时看到输入密码的提示，那么生成不受密码保护的 PEM 密钥库，从而允许 Apache 服务器使用该密钥库。以下命令是一个示例。

```
openssl rsa -in buildForgeKey.pem
        -passin pass:<password>
        -out buildForgeKeyForApache.pem
```

请确保需要访问运行 Build Forge 的进程标识的任何用户都可以读取不受保护的 PEM 密钥库。

更改密钥库密码： 安装期间，您可以指定密钥库密码。如果希望更改该密码，需要执行以下操作：

1. 修改 Build Forge 密钥库以使用新的密码。
2. 修改 Build Forge 配置以使用新的密码。

修改密钥库密码：

关于此任务

Build Forge 具有三个由密码保护的缺省密钥库，全部和 Build Forge 引擎安装在同一主机上，位于 *<bfinstall>/keystore* 目录中：

- buildForgeKey.pem - 用于 OpenSSL，需要 openssl 工具来更改密码。
- buildForgeKeyStore.p12 - 用于 JSSE，需要 ibmjdk 工具来更改密码。
- buildForgeTrustStore.p12 - 用于 JSSE，需要 ibmjdk 工具来更改密码。

Build Forge 软件中包含了这些工具。

注： 为了清楚起见，在示例命令中使用了换行符。请勿在命令中使用换行符。将其作为一个字符串输入，或使用连行字符（对于 Windows 是 ^，对于 UNIX 或 Linux 是 \）。

要点： 对所有密钥库使用相同密码。它在示例中显示为 *newpassword*。

过程

1. 登录至安装了 Build Forge 引擎的主机。
2. 将工具目录放入 PATH 中。
 - *<bfinstall>/openssl*
 - *<bfinstall>/ibmjdk/bin*
3. 禁用 SSL。 在控制台中，转至**管理 > 安全性**。将启用 **SSL** 设置为“否”。
4. 单击**保存**。

5. 单击更新主 **BFCClient.conf**。

6. 停止 Build Forge 引擎。

7. 备份现有密钥库。将现有 Build Forge 密钥库复制到临时目录中。如果修改后的文件被破坏，可以使用备份的密钥库。

8. 修改 buildForgeKey.pem。在目录 *<bfinstall>/keystore* 中，运行以下命令：

```
openssl rsa
-in buildForgeKey.pem
-passin pass:oldpassword
-out buildForgeKey.pem
-passout pass:newpassword -aes128
```

9. 修改 buildForgeKeyStore.p12。在目录 *<bfinstall>/keystore* 中，运行以下命令：

```
keytool -storepasswd -all
-new newpassword
-keystore buildForgeKeyStore.p12
-storepass oldpassword
-storetype pkcs12
```

10. 修改 buildForgeTrustStore.p12。在目录 *<bfinstall>/keystore* 中，运行以下命令：

```
keytool -storepasswd -all
-new newpassword
-keystore buildForgeTrustStore.p12
-storepass oldpassword
-storetype pkcs12
```

结果

更改密码之后，您需要修改 Build Forge 配置以使用新密码。

修改 Build Forge 配置以使用新密码：

Build Forge 配置必须更改为使用已更改的密钥库密码。

开始之前

先决条件：

- 关闭 SSL、停止 Build Forge 引擎并修改密钥库密码以来，尚未启动 Build Forge 引擎。

关于此任务

Apache Tomcat 应用程序服务器在 *server.xml* 配置文件中包含了密钥库密码。这些密码以明文形式存储。在此设置中 Apache Tomcat 不支持编码或加密密码。在本过程中，将修改 *server.xml* 以及 Build Forge 控制台中的安全性属性。

过程

1. 在 Tomcat 配置中输入新密码。编辑 *<bfinstall>/Apache/tomcat/conf/server.xml*。针对 SSL 的 Connector 语句就在注释 *<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->* 的下面。

```
<Connector port="8443" maxHttpHeaderSize="8192" algorithm="IbmX509"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
```

```
clientAuth="false" sslProtocol="TLS"
keystoreFile="C:\Program Files\IBM\Build Forge\keystore\buildForgeTrustStore.p12"
keystorePass="newpassword"
keystoreType="PKCS12"
truststoreFile="C:\Program Files\IBM\Build Forge\keystore\buildForgeTrustStore.p12"
truststorePass="newpassword"
truststoreType="PKCS12"/>
```

2. 启动 Build Forge。
3. 登录至控制台。 使用 root 用户或具有安全性访问角色的登录名。
4. 输入密钥库的新密码。 在**管理 > 安全性 > 密钥库**中，编辑以下条目以使用“密码”属性中的 *newpassword*。
 - 缺省 JSSE 密钥库
 - 缺省 JSSE 信任库
 - 缺省 OpenSSL 密钥库
5. 启用 SSL。
 - a. 在控制台中，转至**管理 > 安全性**。
 - b. 将**启用 SSL** 设置为“是”。
 - c. 单击**保存**。
6. 将更改导出至 BFCClient.conf。 单击**更新主 BFCClient.conf**。
7. 启动 Build Forge 引擎。

创建新的自签名证书:

使用提供的工具创建新的自签名证书。

开始之前

您需要的是在安装期间为密钥库指定的密码。如果您不知道该密码，请在 `bfinstall/Apache/tomcat/conf/server.xml` 中查找。 SSL/HTTP 连接器上的 `keystorePass` 属性包含密码。

关于此任务

此过程描述了如何替换在 Build Forge 安装期间自动创建的证书。它创建具有以下属性的证书:

- 密钥库: `buildForgeKeyStore.p12`
- 期限: 15 年 (设置为 5475 天)
- 主题 DN: `CN=hostname`, 其中 *hostname* 是标准主机名。

使用 `openssl` 和 `ibmjdk` 工具创建证书。Build Forge 软件中包含了这些工具。

需要以下 5 个密钥库:

- `buildForgeKeyStore.p12` - 密钥库, 证书和密钥的容器
- `buildForgeTrustStore.p12` - 信任库, 证书和密钥的容器
- `buildForgeKey.pem` - PEM 密钥库
- `buildForgeCert.pem` - 公用证书
- `buildForgeCA.pem` - PEM 认证中心 (CA)

注：为了清楚起见，在示例命令中使用了换行符。请勿在命令中使用换行符。将其作为一个字符串输入，或使用连行字符（对于 Windows 是 ^，对于 UNIX 或 Linux 是 \）。

要点：对所有密钥库使用相同密码。该密码在示例中显示为 *password*。

过程

1. 登录至安装了 Build Forge 引擎的主机。
2. 将工具目录放入 PATH 中。
 - `<bfinstall>/openssl`
 - `<bfinstall>/ibmjdk/bin`
3. 将 openssl 目录置于 LD_LIBRARY_PATH 中。
 - `<bfinstall>/openssl`
4. 创建密钥库 buildForgeKeyStore.p12、证书和公用/专用密钥对。
 - a. 在临时目录中，运行 keytool 以创建密钥库：

```
keytool -genkey -alias buildforge
-keyalg RSA -keysize 1024 -validity 5475 -dname "CN=hostname"
-keystore buildForgeKeyStore.p12
-storepass password
-storetype pkcs12
```
 - b. 将密钥库文件 (buildForgeKeyStore.p12) 复制到 `<bfinstall>/keystore`。它会覆盖现有文件。
5. 导出公用证书。在目录 `<bfinstall>/keystore` 中，运行以下命令：

```
keytool -export -alias buildforge
-file cert.der -keystore buildForgeKeyStore.p12
-storepass password
-storetype pkcs12
```
6. 创建信任库。
 - a. 在临时目录中，运行 keytool 以创建信任库：

```
cd /temp
keytool -import -noprompt -trustcacerts -alias buildforge
-file cert.der -keystore buildForgeTrustStore.p12
-storepass password
-storetype pkcs12
```
 - b. 将信任库文件 (buildForgeTrustStore.p12) 复制到 `<bfinstall>/keystore`。它会覆盖现有文件。
7. 将公用客户机证书放在 buildForgeCert.pem 中 在目录 `<bfinstall>/keystore` 中，运行以下命令：

```
openssl pkcs12 -clcerts -nokeys
-in buildForgeKeyStore.p12 -passin pass:password
-out buildForgeCert.pem
```
8. 将证书和密钥放在 buildForgeKey.pem 中 在目录 `<bfinstall>/keystore` 中，运行以下命令：

```
openssl pkcs12
-in buildForgeKeyStore.p12 -passin pass:password
-passout pass:password -out buildForgeKey.pem
```
9. 创建 PEM 认证中心 buildForgeCA.pem。它是 buildForgeKey.pem 的副本。在目录 `<bfinstall>/keystore` 中，运行以下命令：

```
cat buildForgeCert.pem > buildForgeCA.pem
```

下一步做什么

BuildForgeKey.pem 是受密码保护的 PEM 密钥库。Apache 服务器启动时将提示输入该密码。如果不希望在启动时看到输入此密码的提示，请生成用于 Apache 服务器的不受密码保护的 PEM 密钥库。

要从专用密钥除去密码，可以运行以下步骤。确保需要访问运行 Build Forge 的进程标识的那些人员可以读取 buildForgeKeyForApache.pem 文件。

```
openssl rsa -in buildForgeKey.pem -passin pass:password  
-out buildForgeKeyForApache.pem
```

使用客户机证书登录:

配置使用客户机证书登录的功能时有两个选择。

使用 WebSphere Application Server 托管 Build Forge 时登录:

关于此任务

使用 WebSphere Application Server 托管 Build Forge 时，您可以选择使用某些修改。此选项需要 WebSphereSSOInterceptor，它包含 WebSphere 执行认证后认证的主体。

在以下过程中进行某些修改以使用此选项。

过程

1. 保护 Build Forge WAR 文件使 WAS 容器管理认证将认证 rbf-services。有关如何设置此保护的信息，请参阅第 98 页的『使用定制拦截器与 WAS 安全性集成』。
2. 配置 WebSphere Application Server 以支持用于 rbf-services Web 应用程序的客户机证书认证映射。通过配置以下项来设置此支持：
 - 用于客户机证书的 SSL
 - 用于客户机证书的 rbf-services webapp
 - 用于客户机证书的 Web 服务器
 - 用于客户机证书的 LDAP 服务器映射

有关配置可在配置 WebSphere Application Server 中进行指导的 WebSphere Portal 的客户机证书认证，请参阅 <http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.ent.doc/wpf/certauth.html>。请遵循步骤 1 至 3。该过程参阅在第 98 页的『使用定制拦截器与 WAS 安全性集成』中修改的 web.xml 文件。修改 \${WAS_INSTALL_ROOT}/profiles/\${PROFILE_NAME}/installedApps/\${CELL_NAME}/rbf-services_war.ear/rbf-services.war/WEB-INF/web.xml 中的文件。

根据定制的 Build Forge SSO 拦截器登录:

关于此任务

此选项需要您添加定制的 SSO 拦截器，如第 89 页的『关于单点登录框架』中所述。

过程

1. 请确保定制的 SSO 拦截器从以下请求的属性接收 X509 证书:

```
X509Certificate[]  
certs = (X509Certificate[])request.getAttribute("javax.net.ssl.peer_certificates");
```

2. 请确保定制的 SSO 拦截器将 Subject DN 的部分映射到 LDAP 属性中。通常，将常见名称 (CN) 从证书映射到 LDAP 目录的用户标识中。在 Build Forge SSO 拦截器的 `authenticateRequest` 方法中返回结果。

```
responseAttributes.setStatus(HttpServletResponse.SC_OK);
principal = mapCert(certs); /* 定制从证书映射
                             到 Build Forge LDAP 主体的方法。*/
result = new Result(Result.UseridOnlyOID, domain, principal); /* 指定
                        主体所在的 LDAP 服务器的域名。*/
return result;
```

下载不受限制的策略文件:

使用高强度加密的证书需要您下载 JCE 的不受限制的策略文件

由于导出和导入法规，安装管理器和 Build Forge 随附的缺省 JCE 管辖区域策略文件支持您使用高强度、但是受限的密码术。如果您要使用的证书的加密超过导出所批准的加密强度，那么需要下载不受限制的 JCE 策略文件，并针对安装管理器和 Rational Build Forge 更新 JVM。

如果在安装期间尝试提供证书以供使用时遇到错误，那么可能需要更新 JCE 策略文件。无法验证该证书，在不使用该证书的情况下进行安装，安装后尝试添加该证书。

警告: 包含对软件开发包 (SDK) 的更新的 Build Forge 升级可能覆盖策略文件。在此情况下，升级后请将不受限制的策略文件重新应用到 SDK。

要点: 您的产地国对加密软件的进口、拥有、使用或再次出口到其他国家可能有一些限制。下载或使用未限制的策略文件之前，必须检查您的国家或地区的法律、规章以及对加密软件进行进口、拥有、使用和再次出口的相关政策，从而确定是否可以使用该文件。

1. 针对 SDK V1.4.2 及以上版本下载不受限制的 JCE 策略文件。
2. 解压缩文件。
3. 将解压缩的文件 `local_policy.jar` 复制到适当的 Build Forge 安装目录。覆盖现有文件。使用针对您的操作系统的目录。

- Windows 系统

```
<bfinstall>\ibmjdk\jre\lib\security
```

- UNIX 和 Linux 系统

```
<bfinstall>/Platform/ibmjdk/jre/lib/security
```

4. 要支持在将来安装 Build Forge 时验证该证书，请更新安装管理器。将解压缩的文件 `local_policy.jar` 复制到使用安装管理器安装的 IBM JRE。覆盖现有文件。

- Windows 系统

```
<IInstall>\eclipse\jre_version\jre\lib\security
```

- UNIX 和 Linux 系统

```
<IInstall>/eclipse/jre_version/jre/lib/security
```

启用密码加密

将 Build Forge 系统配置为使用密码加密可增强系统的安全性。

实施物理安全性以防止对系统的未授权访问也是很重要的。

注：如果要与 WebSphere 组件集成，请确保在配置密码加密之前满足密码加密的先决条件。

- 要了解有关与 WebSphere Application Server 集成的信息，请参阅第 491 页的『使用 WebSphere Application Server 而非 Apache Tomcat』。
- 关于与 IBM HTTP Server (IHS) 的集成，请参阅第 495 页的『使用 IBM HTTP Server 而不是 Apache HTTP Server』。

关于 Build Forge 中的密码安全性

缺省情况下 Build Forge 系统使用经过编码的密码，但是可使用加密密码以使安全性更高。

当密码加密被启用时，它将作为对称密钥密码模式进行启用。使用密码的客户机和所访问的服务都必须使用同一密钥。

- Build Forge 引擎和 Build Forge 代理程序
- Build Forge 服务层和 Build Forge 使用的数据库

在 Build Forge 系统中，密钥保存在 `bfpwcrypt.conf` 文件中。该文件位于 Build Forge（对于引擎）和代理程序的安装目录中。

密码加密使用对称密钥。需要解密公共数据库密码的所有系统都需要相同的密钥。而且，从引擎接收加密密码的所有代理程序都需要该引擎的密钥。如果多个引擎都在运行（冗余配置），那么代理程序需要每个引擎的密钥。

对于一个主机上有一个 Build Forge 管理控制台并且另一个主机上有一个代理程序的简单安装，启用密码加密需要以下过程：

- 在 Build Forge 控制台中启用密码加密（**管理 > 安全性**）。
- 将当前密钥导出至文件。（该密钥供代理程序使用，并且还供其他引擎使用。）
- 生成代理程序的新密钥。将其导出至文件。（代理程序使用该密钥来加密其密钥库密码。）
- 使用这两个密钥更新代理程序的 `bfpwcrypt.conf`。将新密钥放在最后。
- 在 Build Forge 控制台上，对于使用该代理程序的所有服务器定义启用密码加密。（**服务器面板**）
- 在 Build Forge 主机上，使用 `bfpwencrypt` 实用程序来加密 Build Forge 用于访问数据库的密码。将当前密码（已编码）替换为 `buildforge.conf` 中的已加密密码。
- 更新服务层的 `buildforge.conf` 副本。请参阅第 83 页的『Build Forge 配置文件（`buildforge.conf`）』。

为管理控制台启用密码加密

在管理控制台中启用密码加密是在所有其他组件中启用密码加密的先决条件：

- 冗余管理控制台需要从要包含在其 `bfpwcrypt.conf` 文件中的第一个管理控制台导出密码密钥。所有管理控制台都必须使用相同的密钥。通常它们还必须使用该密钥将数据库密码加密。

- 如果引擎正在将加密后的密码发送给代理程序，那么代理程序需要导出密码密钥。引擎和代理程序应该使用相同的密钥。从管理控制台导出的密钥允许它们将加密的服务器认证密码进行解密。

如果代理程序未在接收已加密的密码，但需要生成已加密的密码以在其 `BFAgent.conf` 文件中使用，那么每个代理程序都应该使用不同的密码密钥。在管理控制台中单独生成每个密钥。

要在管理控制台中启用密码加密，请执行以下操作：

1. 在控制台中，转至**管理 > 安全性**。
2. 将**启用密码加密**设置为“是”。
3. 单击**保存**。此步骤将配置保存在 Build Forge 数据库中。
4. 单击**更新主 BFClient.conf**。此步骤将配置保存在 Build Forge `bfclient.conf` 文件中。
5. 重新启动 Build Forge。需要执行此步骤，这样正在运行的 Build Forge 进程才能使用 `bfclient.conf` 文件中的新设置。

当**启用密码加密**为“是”并且在配置中时，可以执行以下操作：

- 将密码密钥导出到文件
- 生成新的密码密钥
- 运行 `bfencrypt` 和 `bfagent` 命令以创建要包含在控制台和代理程序配置文件中的已加密密码

管理密码密钥

缺省情况下 Build Forge 系统使用经过编码的密码，但是可使用加密密码以使安全性更高。

密码加密密钥文件：

`bfencrypt.conf` 文件包含密码加密配置属性。

第一次启动 Build Forge 时，会在与 `bfclient.conf` 文件相同的位置自动生成 `bfencrypt.conf` 文件。

- Windows: `<bfinstall>`
- UNIX 或 Linux: `<bfinstall>/Platform`

要点：不要重命名此文件。其名称必须始终为 `bfencrypt.conf`。

该文件包含以下属性：

bfencrypt_key_alias

加密密钥的别名。别名是使用此密钥加密的任何密码的一部分。系统使用它来确定要使用的密钥。可以对此属性进行多个定义，每个密钥一个定义。最后一个定义用于将密码加密。所有其他定义都用于在读取已加密的密码时将其解密。已加密的密码可能出现在配置文件、数据库或代理程序与引擎之间的通信中。

bfencrypt_key

已进行 128 位 AES 加密处理的主密钥。该密钥用于将密码加密。

bfpwcrypt_key_password

将 bfpwcrypt_key 解密所需的密码。已对此密码进行了编码。

bfpwcrypt.conf 文件示例:

```
***** Password Encryption Configuration Properties *****
bf_pwcrypt_key=MKuoiwD+MsWBFgl/2xeG0TEtpY+hAzXQu21fBcofM0M=
bf_pwcrypt_key_alias=8a679d430c401000b55e00007d1a7d1a
bf_pwcrypt_key_password=Tq0eDXc4G/bdaWeatKTYUx6Sw4S3i6wX
# Creation date=Thu Nov 20 03:44:48 CST 2008
# Origination host=myhost.mycompany.com
```

导出密码密钥:

从 Build Forge 控制台导出密码密钥，以便将其存放在其他位置。

导出在 Build Forge 引擎上使用的密码密钥，以将其存放在以下位置:

- 访问相同数据库的其他 Build Forge 控制台
- 托管代理程序（在启用了密码加密的服务器定义中确定）的计算机

要导出密码密钥，请执行以下操作:

1. 在控制台中，转至**管理 > 安全性**。

注: 已启用密码加密必须已设置为“是”，并已在主 BFClient.conf 文件中保存和更新此设置。

2. 单击**导出密钥文件**。系统会询问您将文件保存到何处。
3. 指定一个位置，然后单击**保存**。

将文件存放到需要密钥来将密码进行解密的主机（其他冗余管理控制台和代理程序）上或将密码进行加密的主机（代理程序）上。将文件的内容（所有三个属性设置）添加到该主机上的 bfpwcrypt.conf 文件中。下表描述内容的存放位置如何决定使用密钥的情况。

放置在 bfpwcrypt.conf 文件中	在以下情况下使用密码密钥
文件末尾	<ul style="list-style-type: none">• 解密包含其别名的密码• 加密新密码
任何其他位置	<ul style="list-style-type: none">• 解密包含其别名的密码

注: 生成新密钥时，在 bfpwcrypt.conf 文件的末尾将再存放一个密钥。导出密钥时，仅导出最新的密钥。

生成新密码密钥: 可以从管理控制台生成新的密钥，以更改系统级密钥。为保持良好的系统安全性，应该定期执行此操作。必须在所有引擎和代理程序上更新该新密钥。

生成新密码密钥时会将新密钥添加到 bfpwcrypt.conf 文件中:

- 如果已使用旧密钥生成或保存了用户或控制台密码，那么它们将仍然有效。
- **重要信息:** 如果已导出以前的密钥并将其合并到代理程序的 bfpwcrypt.conf 文件中，那么在添加新生成的密钥之前，与代理程序的通信会失败。
- 保存在 Build Forge 中或使用 bfpwencrypt 生成的任何新密码都使用新密钥。
- 导出密钥文件将仅导出最新的密钥。

要生成新密码密钥，请执行以下操作：

1. 在控制台中，转至**管理 > 安全性**。

注：已启用密码加密必须已设置为“是”，并已在主 `BFCClient.conf` 文件中保存和更新此设置。

2. 单击**生成新的密钥**。系统会要求您确认。
3. 单击**是**。

启用代理程序的密码加密

在代理程序的配置文件中启用代理程序的密码加密。

要启用代理程序的密码加密，请执行以下操作：

1. 如果代理程序正在运行，请停止该代理程序。
2. 请转至安装了代理程序的目录：
3. 编辑 `bfagent.conf` 并打开加密设置：

```
password_encrypt_module ./bfcrypt.dll;./bfpwcrypt.conf
```

如果从该代理程序安装目录除外的其他目录中启动该代理程序，请更改该路径以直接引用这些文件。

4. 获取管理控制台加密密钥。将该密钥导出至文件。代理程序需要该密钥来解密已加密的服务器认证密码。该密钥还用于加密本地密钥库密码。
5. 将该密钥置于代理程序的 `bfpwcrypt.conf` 文件中。将代理程序的已生成密钥置于文件最后。手动加密密码时，将使用该文件中的最后一个条目。
6. 启动代理程序。
7. 在管理控制台中，转至**服务器**。对于使用该代理程序的每个服务器定义，请将**密码加密配置**属性设置为“已启用”。
8. 单击**测试连接**以确保加密密码可使该连接成功。

将 `buildforge.conf` 和 `bfagent.conf` 中的密码加密

使用导出的密码密钥可构建要在 `buildforge.conf` 和 `bfagent.conf` 中使用的已加密密码。

`buildforge.conf` 文件包含 Build Forge 用于访问数据库的用户名和密码（`db_password`）。该密码通常已编码但是可以加密。要为管理控制台将密码加密，请执行以下操作：

1. 转至管理控制台根目录。
 - Windows: `<bfinstall>`
 - UNIX 或 Linux: `<bfinstall>/Platform`
2. 运行以下命令：

```
bfpwencrypt -e password
```

使用您想加密的明文密码来替换 `password`。

加密后的密码将发送到 `stdout`。

`bfagent.conf` 文件包含代理程序用于访问密钥库的密码密钥（`ssl_key_password`）。该密码通常为明文，但是可以加密。要为代理程序加密密码，请执行以下操作：

1. 转至代理程序根目录。
2. 运行以下命令:

```
bfagent -e password
```

使用您想加密的明文密码来替换 *password*。加密后的密码将发送到 `stdout`。

注: 如果使用 AIX 且 GCC 库不在 `/lib` 或 `/usr/lib` 中, 那么您可能会收到错误消息, 表明“无法装入模块 `/usr/local/bin/bfcrypt.dll`。”您可以通过更新 `inetd.conf` 来解决此问题。

要更正错误, 请执行以下操作:

- a. 找出以下行:

```
bfagent stream tcp nowait root /usr/local/bin/bfagent bfagent
```

- b. 将该行更改为以下行:

```
bfagent stream tcp nowait root /usr/bin/env env LIBPATH=path /usr/local/bin/bfagent
```

path 是 GCC 库的位置。

- c. 使用 `refresh -s inetd` 命令重新装入 `inetd.conf`。

加密后的密码以字符串 `bfcrypt:` 和花括号中包含的密码密钥别名开头, 后跟已加密 (AES 128 位) 并编码 (Base63) 的密码。已编码和加密的密码的示例:

Encoded:

```
dd8b42eed5cc051500f5bffe2b82b1aa6a67baee028a85d0cefa
```

Encrypted:

```
{bfcrypt:7427ab360c4010008f9d000049664966}drAIT1zLDGX/xRcvw65+B8aFpTqvmAdbmnh6FpwkHjU=
```

对密码加密的问题进行调试

如果已加密的密码有问题, 以下各节描述了进行调试的方法。

调试控制台中的密码加密问题:

服务层、Web 接口和引擎共享同一个密钥文件。

当所有三个组件都安装在同一个主机上时, 它们将使用同一个密钥文件:

- 对于 Windows:

```
<bfinstall>/bfpwcrypt.conf
```

- 对于 UNIX 或 Linux:

```
<bfinstall>/Platform/bfpwcrypt.conf
```

如果启用密码加密后出现问题, 请检查以下事项:

- 请确保在启用密码加密之后重新启动 Build Forge。请确保所有进程都已正确地停止并重新启动 (Apache、Apache Tomcat 以及引擎)。
- 冗余控制台: 如果使用同一数据库进行管理控制台的多次安装, 那么这些安装必须全部使用同一个 `bfpwcrypt.conf` 文件。最安全的方法是手动 (而不是通过网络) 分发该文件。
- 服务器定义: 如果控制台中的**测试连接**失败, 请确保已导出密钥, 并将其正确地置于 `bfagent.conf` 文件中。要确定存在密码问题, 请禁用密码加密并尝试**测试连接**。

- 登录: 如果在启用密码加密后无法登录, 请确保 Build Forge 在 `bfclient.conf` 和 `buildforge.conf` 中使用了正确的 `bfpassword.conf` 密钥。 `buildforge.conf` 必须在 `<bfinstall>` 目录中及其服务层的副本中进行更新。请参阅第 83 页的『Build Forge 配置文件 (`buildforge.conf`)』。

如果执行了所有这些检查, 但是问题仍然存在, 请尝试启用跟踪并检查输出日志。

- Web 接口 (UI): 设置环境变量 `BFDEBUG_SECURITY=1`。

Web 接口: 输出显示在文件中。

- 对于 Windows:

`<bfinstall>/Apache/logs/php_error.log`

- 对于 UNIX 或 Linux:

– `<bfinstall>/server/apache/logs/php_error.log`

- 引擎: 在前台启动引擎。在安装目录中运行 `bfengine -d`。对于 UNIX 或 Linux, 可以使用 `bfengine -d 2>&1 | tee out.txt` 将其以管道方式传输至文件。对于 Windows, 如果有 `tee` 实用程序, 那么可以执行相同的操作。

- 服务: 执行以下操作:

1. 停止 Build Forge。

2. 在编辑器中打开日志文件。

- 对于 Windows:

`<bfinstall>/Apache/tomcat/common/classes/logging.properties`

- 对于 UNIX 或 Linux:

`<bfinstall>/server/apache/tomcat/common/classes/logging.properties`

3. 将下行添加至文件末尾。

`com.buildforge.services.common.security.level=ALL`

4. 启动 Build Forge。

5. 检查输出。

- 对于 Windows:

`<bfinstall>/Apache/tomcat/logs/catalina.out`

- 对于 UNIX 或 Linux:

`<bfinstall>/server/apache/tomcat/logs/catalina.out`

调试代理程序中的密码加密问题:

调试代理程序通信包含代理程序和引擎组件。

如果启用密码加密后出现问题, 请检查以下事项:

- 检查 `bfagent.conf`。应对以下行取消注释:

`password_encrypt_module ./bfcrypt.dll;./bfpassword.conf`

- 检查 `bfpassword.conf` 是否存在于启动代理程序的目录中。它必须包含至少一个密钥条目。

- 检查引擎的 `bfpassword.conf` 中的最后一个条目是否存在于代理程序的 `bfpassword.conf` 中的某处。

如果执行了所有这些检查，但是问题仍然存在，请尝试启用跟踪并检查输出日志。要启用跟踪，请执行以下操作：

1. 打开 `bfagent.conf`。

2. 取消下行的注释：

```
activity_log bfagent.log
```

可以指定另一个路径来代替 `bfagent.log`。

bfcrypt.dll 的路径问题

在 `bfagent.conf` 中，`password_encrypt_module` 属性必须指向 `bfcrypt.dll` 的正确路径。示例：

```
password_encrypt_module /opt/buildforge/bfcrypt.dll
```

打开跟踪后，该路径的问题将通过类似以下的输出指示：

```
[ 8928] bfcryptloader.c : 208: Loading password encryption module.
[ 8928] bfcryptloader.c : 223: module: bfcrypt
[ 8928] bfcryptloader.c : 232: Loading module: C:/BuildForge71.181.Agent/bfcrypt.dll
[ 8928] bfcryptloader.c : 262: Failed loading DLL, error code = 0
```

如果装入成功，将产生类似以下的输出：

```
[ 12248] bfpwdlocloader.c: 134: Looking for password locator: ssl_key_password_locator
[ 12248] bfpwdlocloader.c: 244: Looking for password for prop ssl_key_password from bfagent.conf.
[ 12248] bfcryptloader.c : 208: Loading password encryption module.
[ 12248] bfcryptloader.c : 223: module: bfcrypt
[ 12248] bfcryptloader.c : 232: Loading module: ./bfcrypt.dll
[ 12248] bfcryptloader.c : 269: Loading procedure bfcrypt_init.
```

密码解密失败

当密码由于密钥错误或某些其他原因而未能解密时，日志将包含类似于以下内容的行：

```
[ 4912] agent.c : 237: AUTH failed
```

如果确定密码是正确的，那么可以进一步诊断问题。启用对 `bfcrypt.dll` 模块的调试。要启用调试，请设置以下环境变量：

```
BFDEBUG_SECURITY=1
```

如果代理程序作为服务运行，那么该环境变量需要全局设置。

调试输出置于启动代理程序的目录中的 `bfcrypt.txt` 内。

以下输出表示代理程序上的 `bfpwcrypt.conf` 中没有正确的密钥：

```
load_keys_from_file: Parsed 1 key configurations.decrypt:
Looking for key matching info: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Comparing against: 4d553f110c401000ac08000051f651f6, length=32
decrypt: Warning! No matching key found.
```

以下输出表示密钥匹配正确：

```
load_keys_from_file: Parsed 2 key configurations.decrypt:
Looking for key matching info: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Comparing against: 7427ab360c4010008f9d000049664966, length=32
decrypt: Comparing against: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Found match: 922492fe0c4010008304c3670e1e0e1e, length=32
```

bfclient.conf 引用

bfclient.conf 文件存储有关 Build Forge 安全性的设置。该文件位于 Build Forge 安装根目录中。

bfclient.conf 文件包含用于启用安全通信（SSL）和密码加密的设置。它包含以下部分:

- 连接属性
- 登录属性
- 用于 OpenSSL 和 JSSE 的 SSL 属性
- 仅用于 OpenSSL 的 SSL 属性
- 仅用于 JSSE 的 SSL 属性
- 密钥库属性（bf_keystore_*）
- 密码属性

表 1. 连接属性

属性名	缺省	可能值	必需	描述
bf_services_hostname	在安装期间指定	不适用	是	Build Forge 服务层所在的主机名
bf_services_tcp_port	在安装期间指定	不适用	是	用于连接 Build Forge 服务的 TCP 端口。当未指定 SSL 时将使用此端口。
bf_services_ssl_port	在安装期间指定	不适用	是	用于安全地连接至 Build Forge 服务的 SSL 端口
bf_services_preferred_protocol	tcp	ssl, tcp	是	对于 Perl 或 PHP 客户机，指定 SSL 或 TCP 以建立连接。对于 Java 客户机，SecureAPIClientConnection 对象指定 SSL，而 APIClientConnection 指定 TCP。

表 2. 登录属性

属性名	缺省	可能值	必需	描述
bf_login_user	无	Build Forge 用户列表中的用户标识	否	用作登录标识。还可以在客户机程序中指定登录标识。
bf_login_password	无	bf_login_user 的密码	是（如果使用 bf_login_user）	bf_login_user 的密码
bf_login_realm	无	LDAP 域名	否	如果用户未处于用户表中，那么是要查询的 LDAP 域。

表 3. 用于 openssl 和 JSSE 的 SSL 属性

属性名	缺省	可能值	必需	描述
bf_ssl_usage	无	jsse, openssl	是	选择 SSL 实施。根据该选择，将提供各种不同的属性。

表 3. 用于 openSSL 和 JSSE 的 SSL 属性 (续)

属性名	缺省	可能值	必需	描述
bf_ssl_cipher_group	ALL	ALL、HIGH、MEDIUM 和 LOW	否	指定在 SSL 握手期间要提供的一组密码。HIGH 最安全，LOW 提供最佳的性能，ALL 提供了最好的互操作性。
bf_ssl_cipher_override	无	所提供的密码套件	否	覆盖 bf_ssl_cipher_group。可用于选择要在 SSL 握手期间使用的一组数量更少的密码。
bf_ssl_protocol	TLSv1	TLSv1, SSLv3。 可因实施而异。	否	用于 SSL 的握手协议。TLSv1 是首选协议。
bf_ssl_cert_alias	无	所配置密钥库中的有效证书别名	否	指定要使用的证书。同一密钥库中可能存在多个证书。

表 4. 仅用于 openSSL 的 SSL 属性

属性名	缺省	可能值	必需	描述
bf_ssl_key_ref	openssl_key	包含专用密钥的任何有效 PEM 密钥库引用	否	对密钥库配置的引用，其中包含客户机在连接至服务器时要使用的专用密钥。使用时，还必须在 bf_ssl_cert_ref 中指定该专用密钥的有效证书。仅当服务器设置为请求个人证书时才使用。
bf_ssl_cert_ref	openssl_cert	包含所指定专用密钥的证书的任何有效 PEM 密钥库引用	否	对密钥库配置的引用，其中包含以上专用密钥的证书。仅当服务器设置为请求个人证书时才使用。
bf_ssl_ca_ref	openssl_ca	任何有效的 PEM 密钥库引用，其中包含一个或多个证书，用于验证该客户机连接至的服务器证书	是	对密钥库配置的引用，其中包含在 SSL 握手期间用于验证服务器证书的一个或多个签署者证书。证书可以是 CA 根证书、中间证书或自签署证书。

表 5. 仅用于 JSSE 的 SSL 属性

属性名	缺省	可能值	必需	描述
bf_ssl_keystore_ref	jsse_keystore	包含 keyEntry（专用密钥和证书）的任何有效 PKCS12、JKS 或 JCEKS 密钥库引用	否	对密钥库配置的引用，其中包含客户机连接至服务器时要使用的个人证书（专用密钥和相关证书）。仅当服务器请求客户机认证的个人证书时，这才是必要的。
bf_ssl_truststore_ref	jsse_truststore	包含 keyEntry（专用密钥和证书）的任何有效 PKCS12、JKS 或 JCEKS 密钥库引用	是	对密钥库配置的引用，其中包含在 SSL 握手期间用于验证服务器证书的签署者证书。密钥库包含一个或多个 trustedCertEntry，它们是用于验证其他证书签名的证书。

表 6. 密钥库属性

属性名	缺省	可能值	必需	描述
bf_keystore_alias	各不相同	字符串	是	这是 SSL 配置引用密钥库配置所用的名称。
bf_keystore_location	各不相同	所指定类型的密钥库的相对或标准路径。	是	这是所指定类型的密钥库的路径和位置。该路径可以是相对路径，但对于启动目录必须是正确的。

表 6. 密钥库属性 (续)

属性名	缺省	可能值	必需	描述
bf_keystore_type	PEM (对于 openssl), PKCS12 (对于 jsse)	PEM (对于 openssl)。PKCS12, JCEKS 或 JKS (对于 “jsse”)	是	密钥库的类型。必须与 bf_keystore_location 属性所引用的实际密钥库类型相匹配。
bf_keystore_password	在安装期间指定	受密钥库类型支持的字符串。某些密钥库不支持非 ASCII 字符串。	否	用于访问密钥库的密码。对于 OpenSSL, 仅包含公用密钥的 Cert 和 CA 密钥库通常不需要密码。

表 7. 密码属性

属性名	缺省	可能值	必需	描述
bf_pw_crypt_enabled	false	true, false	否	指定密码是已编码 (false) 或是已加密 (true)。启用时, 密码加密实施使用 bfclient.conf 所在目录中名为 bfpwcrypt.conf 的文件。

第 10 章 安装代理程序

本部分描述如何安装、运行、配置代理程序以及对其进行故障诊断。

在您希望用作管理控制台中服务器资源的每台主机上安装代理程序。代理程序是一种服务，它负责接收来自管理控制台的请求以运行项目和步骤。

在 Windows 平台上安装代理程序

要在 Windows 平台上安装代理程序，请执行以下步骤：

1. 在安装介质中找到并启动代理程序安装程序。安装程序的文件名是 `win-bfagent-version.exe`。

提示：当您选择 **Rational Build Forge 代理程序安装时**，启动板将启动此安装过程。

2. 如果安装程序检测到现有版本的代理程序，会提示您确认是否要将它覆盖。单击 **确定**。缺省值：确定。
3. “欢迎”消息打开之后，单击 **下一步**。
4. 如果同意许可协议的条款，请单击 **我同意**。
5. 在“选择安装位置”窗口中，设置 **目标文件夹**，然后单击 **下一步**。使用缺省位置 `C:\Program Files\IBM\Build Forge\Agent`，这样可以轻松地找到该文件。
6. 在“配置”窗口中，选择您想要的 **代理程序选项**，然后单击 **安装**。
7. 选择以下某种安装方法：
 - 作为服务安装
 - 安装用户方式代理程序

仅当代理程序必须能够运行 GUI 应用程序时，才选择用户方式代理程序。

8. 可选：单击启用 **Cygwin 支持**

提示：如果使用 Cygwin Linux[®] 仿真环境，那么安装代理程序时可以选择安装 Cygwin 支持。如果安装 Cygwin 支持，请执行以下步骤。

- a. 在安装 Cygwin 时，选择 **DOS/text** 行结尾。
- b. 在项目中，对命令使用 UNIX[®] 样式的语法。

要点：

Cygwin 只能使用 US ASCII。它不支持 UTF-8，因此无法用于其他任何系统。

9. 指定代理程序用于与管理控制台通信的 **端口**。缺省端口为 5555。
10. 在“正在完成设置”面板中，单击 **完成**。

注：安装期间不要关闭任何弹出窗口。允许它们在安装运行时出现和消失。

用户方式代理程序

安装为用户方式的代理程序允许用户与项目启动的应用程序交互。

用户方式是仅对 Windows 代理程序提供的选项。该选项是在安装期间设置的。在安装后无法对其进行配置。用户方式具有以下应用：

- 在作业运行时通过 GUI 应用程序手动收集输入。这使得任务依赖于人工输入。
- 对项目 and 步骤进行故障诊断。在用户方式下可以看到在服务方式操作期间隐藏的输
出。每个步骤运行时都会产生一个控制台窗口。

要点： Windows 7 系统。如果您在 Windows 7 系统上安装或重新安装用户方式代理程序，那么必须重新引导操作系统。除非重新引导操作系统，否则该代理程序无法运行。

设置项目以在用户方式下使用计算机时，请记住以下几点差异：

- 用户方式代理程序以系统中当前登录用户身份运行。代理程序仅在此用户登录计算机这段时间内才处于活动状态。如果用户注销，那么运行用户方式代理程序的服务器就无法使用。
- 以用户方式在计算机上运行的步骤对于该计算机的任何用户都是可见的。
 - 每个步骤会在运行代理程序的 Windows 计算机上打开一个控制台窗口。它显示管理控制台的命令活动。
 - 如果从步骤启动 GUI 应用程序，应用程序窗口会出现在运行代理程序的 Windows 计算机上。管理控制台会等待应用程序退出，然后继续步骤。

备选方法：如果希望不等待继续执行作业，请使用启动命令。

- 不要使用 `_USE_BFCREDS` 变量。任何使用该变量的步骤都会失败。
- 用户必须具有以下特权。缺省情况下，这些特权通常不可用。必须显示地添加这些特权。

```
SeInteractiveLogonRight
SeAssignPrimaryTokenPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeTcbPrivilege
```

在 Windows 操作系统上执行静默代理程序安装

要在 Windows 上执行代理程序的自动和静默安装，请使用 `/S`（大写 S）选项。例如，在命令提示符处输入以下命令。该选项是区分大小写的：

```
win-bfagent-7.0.1.2305.exe /S
```

静默安装使用以下设置。这些设置无法修改。

- 覆盖现有安装：yes
- 安装位置：C:\Program Files\IBM\Build Forge\Agent
- 作为服务安装：yes
- Cygwin 支持：no
- 端口：5555

在 UNIX 和 Linux 系统上安装代理程序

请遵循对应于您平台的安装指示信息:

• AIX

1. 使用 `aix5-bfagent-<version>.tar.gz` 文件或 `aix5np-bfagent-<version>.tar.gz` 文件。

`aix5np` 文件不包含对 PAM 认证的支持。

要点:

如果安装 `aixnp` 代理程序以作为 `root` 用户运行, 那么在运行时代理程序将通过您指定的服务器认证凭证, 使用 AIX 认证调用来进行认证。如果未将代理程序安装为作为 `root` 用户运行, 那么还必须使用 `bfagent.conf` 文件中的 `magic_login` 设置以限制对代理程序的访问。

2. 通过输入以下命令抽取文件:

```
gzip -d gzipfilename.gz
```

3. 通过输入以下命令抽取文件:

```
tar xvf tarfilename.tar
```

4. 通过输入以下命令安装代理程序:

```
cd extracted-agent-directory
./install.sh
```

要点: 如果针对 AIX 使用配置选项 `--without-pam` 对代理程序进行编译, 那么将关闭该代理程序的认证。

如果随后使用 `root` 用户特权安装了代理程序, 那么将允许人们作为任何有效用户进行连接, 而不考虑他们所指定的密码。

如果需要编译代理程序以在未使用 PAM 的 AIX 系统上运行, 请确保对该代理程序使用专用帐户, 安装该代理程序以使用该用户身份运行, 并使用 `bfagent.conf` 文件中的 `magic_login` 设置来限制对该代理程序的访问。

• HP-UX

1. 确保 Rational® Build Forge® 代理程序能够找到您在项目中指定的所有操作系统命令。可以使用以下任一选项来完成此任务:

- 编辑系统的 `/etc/PATH` 文件以包含所需的全部目录。

- 在 HP-UX 系统上, 在 `bfagent.conf` 文件中指定 `nologonshell` 设置。该设置会阻止 Build Forge® 处理 `/etc/profile`。必须为在应用于每个项目的 Build Forge® 环境中的 `PATH` 和 `/etc/profile` 中的其他必需设置指定值。

2. 从安装介质获取 `hpux11-bfagent-<version>.tar.gz` 文件。将其放在想要安装代理程序的位置。

3. 将此该文件解压。

```
gzip -d gzipfilename.gz
```

4. 将此 `tar` 文件解压。

```
tar xvf tarfilename.tar
```

5. 安装代理程序。

```
cd extracted-agent-directory
./install.sh
```

6. 修改 `/etc/profile` 中的以下行，使代理程序能在非交互的登录 shell 程序中运行命令。

```
if [ ! $VUE]
```

将该行更改为以下格式:

```
if [ -z "$VUE" -a -n "$PS1" ]
```

- **Mac OS**

1. 从安装介质获取 `mac-bfagent-<version>.dmg` 文件，并将其放在您期望的位置。
2. 双击该文件以抽取其组件。

- **Red Hat Linux 和 SuSE Linux**

1. 从安装介质获取 `rh9-bfagent-<version>.rpm` (Red Hat Enterprise Linux 4 或 SuSE Linux 10) 或 `rhel5-bfagent-<version>.rpm` file (Red Hat Enterprise Linux 5)
2. 使用 RPM 实用程序来安装文件。

```
rpm -iUvh rh9-bfagent-version.rpm
```

- **Solaris**

1. 对于 SPARC 上的 Solaris 9 或 Solaris 10, 请使用 `sol9-bfagent-<version>-sparc-opt.gz` 文件。
2. 解压缩包:

```
gzip -d solN-bfagent-<version>-platform-opt.gz
```

3. 在命令提示符处, 输入以下命令:

```
pkgadd -d ./unzipped-package
```

- **其他平台 - 从源代码进行编译**

如果需要用于其他平台的代理程序, 请使用 `src-bfagent-<version>.tar.gz` 文件, 从源代码编译代理程序:

1. 抽取已下载的 tar 文件。
2. 运行位于 `src` 目录中的配置脚本。

要配置 SSL, 选项为:

```
./configure --with-ssl=/usr/include/
```

要配置 SSL 和密码加密, 选项为:

```
./configure --with-ssl=/usr/include/ --enable-bfcrypt-dll
```

3. 运行 `src` 目录中的 **make** 命令。

源代码包需要对应于您系统的 GNU C 编译器或 C 编译器。如果源代码包和预构建的代理程序不包含针对本地计算机的安装程序, 那么在计算机的系统 `inetd/xinetd` 配置文件中包含了一个用于安装该代理程序的安装程序。

在 System i 平台上安装代理程序

使用以下指示信息在 System i 上手动安装代理程序。

步骤 7 中的命令脚本创建在启动时运行的作业描述, 并作为具有 *ALLOBJ 特权的 BFAGENT 用户启动代理程序。

- 通过使用在管理控制台中指定的服务器认证凭据，可以对具有 *ALLOBJ 特权的任何用户或 QSECOFR 用户进行认证。
- 要对不具有这些特权的用户进行认证，必须在 bfaagent.conf 文件中配置 magic_login 设置。请参阅第 159 页的『bfaagent.conf 引用』以了解详细信息。

要在 System i 平台上安装代理程序，请执行以下步骤：

1. 使用产品安装介质或下载映像，查找 iseries-bfaagent-<version>.tar.gz 文件。
2. 通过输入以下命令，从归档抽取 tar 文件：
gzip -d iseries-bfaagent-<version>.tar.gz
3. 从 tar 文件抽取文件。
tar xvf iseries-bfaagent-<version>.tar
4. 在 iSeries 服务器上，将 bfaagent 可执行文件放在代理程序安装目录（例如：/bin）中。
5. 在 iSeries 服务器上，将 bfaagent.conf 文件放在 /etc 中。
6. 在 bfaagent.conf 文件中，取消注释 shell 选项，指定用于 PASE 的缺省 shell 程序（如以下示例中所示），或指定您的首选 shell 程序。
shell /bin/sh
7. 配置 System i 以在启动时作为 BFAAGENT 用户运行。

输入以下命令以创建具有 *ALLOBJ 特权的 BFAAGENT 用户，并创建在启动时作为 BFAAGENT 用户运行的作业描述。在以下示例中，bfaagent 可执行文件安装在 /bin 中。

```
CRTLIB BFAAGENT
```

```
CRTSBSD SBSD(BFAAGENT/BFAAGENT) POOLS((1 *BASE)) TEXT('Build Forge Agent subsystem')
```

```
CRTJOBQ JOBQ(BFAAGENT/BFAJOBQ) TEXT('Build Forge Agent job queue')
```

```
CRTUSRPRF USRPRF(BFAAGENT) PASSWORD(*NONE) INLMNU(*SIGNOFF) LMTCPB(*YES)
  SPCAUT(*ALLOBJ) TEXT('Build Forge Agent user profile')
```

```
CRTJOBQ JOBQ(BFAAGENT/BFAJOBQ) JOBQ(BFAAGENT/BFAJOBQ)
  TEXT('Build Forge Agent autostart')USER(BFAAGENT) RQSDTA('CALL PGM(QP2SHELL)
  PARM('/bin/bfaagent' ' ' -s'))
```

```
CRTCLS CLS(BFAAGENT/BFACLS) TEXT('Build Forge Agent job class')
```

```
ADDRTGE SBSD(BFAAGENT/BFAAGENT) SEQNBR(1) CMPVAL(*ANY) PGM(QCMD) CLS(BFAAGENT/BFACLS)
```

```
ADDJOBQE SBSD(BFAAGENT/BFAAGENT) JOBQ(BFAAGENT/BFAJOBQ) MAXACT(*NOMAX) SEQNBR(10)
```

```
ADDAJE SBSD(BFAAGENT/BFAAGENT) JOB(BFAAGENT) JOBQ(BFAAGENT/BFAJOBQ)
```

在 System z 平台上安装并运行代理程序

遵循这些指示信息在 System z 上手动解压和编译 Build Forge 代理程序源代码。仅作为未编译源提供了用于 z/OS 的代理程序源代码。未提供二进制分发。

以下软件和程序是必需的：

- c89 编译器和 Unix 头文件。在 z/OS 系统上，代理程序在 Unix 系统服务（USS）环境中运行。

- z/OS UNIX shell 接口。安装过程中，在 z/OS 上的 z/OS UNIX shell 中运行所有命令。
- gzip 实用程序。

注：如果 gzip 在 z/OS 系统上可用，那么在将源代码包传送到 z/OS 之后，可以在 z/OS 系统上解压 tar 文件。否则，您必须首先在非 z/OS 计算机上将文件解压，然后传送到 z/OS 系统。

- 用于 z/OS 的 Build Forge 代理程序源代码包：src-bfagent-<version>.tar.gz。

先决条件

- 针对客户机和内部通信启用 SSL。请参阅第 105 页的『为客户机和内部连接启用 SSL』。
- 必须安装系统 SSL 库。
- 在编译代理程序代码时使用最新版本的 GSKit。

要在 System z 平台上安装代理程序：

1. 使用产品安装介质或下载产品映像，查找代理程序源代码包文件：src-bfagent-<version>.tar.gz。

将源代码包复制或下载到非 z/OS 计算机上的目录中。

2. 在非 z/OS 计算机上的 shell 提示符处，通过输入以下命令从代理程序源代码包解压 tar 文件：

```
gzip -d src-bfagent-<version>.tar.gz
```

3. 使用 ftp 或其他传输方法，将 tar 文件作为二进制映像传输到 z/OS 系统中，并将它放在专用的 HFS 子目录中，通常是用户帐户的 USS 主目录中。
4. 在 z/OS 系统上，运行以下命令以构建代理程序源代码：

- a. 第一步：.configure-zos。请注意 -with-system-ssl 的使用。您提供到系统 SSL 的路径。使用最新版本的 GSKit。

```
pax -rf src-bfagent-<version>.tar -ofrom=ISO8859-1,to=IBM-1047
cd bfagent-<version>/src
./configure-zos --with-system-ssl=<path>
```

- b. 在 ./configure-zos 脚本完成之后，运行以下命令：

```
./build-zos
```

注：如果您在 ./build-zos 脚本运行之后接收到错误，请参阅第 137 页的『对 z/OS 上的代理程序安装进行故障诊断』。

5. 在 z/OS 系统上，将 bfagent.conf 文件放在 /etc 中。

如果 bfagent.conf 不在 /etc 中，那么必须使用 -f 选项启动代理程序。请参阅第 156 页的『bfagent 引用』。

6. 在 z/OS 系统上，将 bfagent 可执行文件放在合适的位置（例如，/usr/bin 或 /usr/local/bin）。
7. 在 z/OS 系统上，作为 root 用户运行以下命令：

```
# extattr +p -s bfagent
```

8. 如果您正在使用系统 SSL，请将系统 SSL 库添加到 PROGRAM CONTROL。使用以下命令：

```
ralter program * addmem('SYS1.SIEALNKE'//nopadchk) uacc(read)
```

9. 在 z/OS 系统上, 作为 root 用户登录, 用 -s 选项手动启动代理程序:

```
bfagent -s
```

如果安全策略不允许您作为 root 用户登录, 请参阅 第 159 页的『bfagent.conf 引用』, 并查看有关 bfagent.conf 中的 magic_login 设置的指示信息。

代理程序作为独立守护程序运行, 并使用缺省代理程序端口 5555。要更改缺省端口, 请使用 bfagent.conf 中的端口设置。请参阅第 156 页的『bfagent 引用』。

注: 如果 Unix TCP/IP 守护程序 (inetd 或 xinetd) 安装在 z/OS 系统上并处于活动状态, 您可以将 Build Forge 代理程序设置为作为服务运行并自动启动。请参阅第 154 页的『在 UNIX、Linux 和 MacOS 上运行代理程序』。

10. 在 z/OS 系统上, 使用 telnet 命令测试连接。请参阅第 167 页的『测试连接』。

对 z/OS 上的代理程序安装进行故障诊断

在 z/OS 上构建代理程序源代码后, 可能接收到错误消息。此主题描述对某些常见错误的修正。

configure-zos 脚本设置某些常见值, 并执行某些基本检查以确定系统上可用的头和函数。

由于 z/OS 系统配置中的变化, ./configure-zos 脚本运行时可能不会出现错误, 但是运行 ./build-zos 脚本时, 可能会看到以下错误。

CEE3501S The module CCNDRVR was not found.

FSUM3066 COMPILE 步骤结束, 返回码如下:

```
-1: EDC5083I An error occurred attempting to load a module into storage.
```

此错误表示所需动态库无法由编译器装入。

运行命令: % export STEPLIB="SYS1.SCCNCMP"

重新运行 ./build-zos 命令。如果命令再次失败, 请联系系统管理员, 以帮助查找需要的库。

IKJ56228I DATA SET CEE.SCEE0BJ NOT IN CATALOG OR CATALOG CAN NOT BE ACCESSED

FSUM3066 COMPILE 步骤结束, 返回码如下:

FSUM3052 无法解析数据定义名 C8961。找不到数据集。请确保正确指定了数据集名称 CEE.SCEE0BJ。

此错误表示链接程序无法查找完成编译所需的系统库。运行命令:

```
% export _C89_LSYSLIB=SYS1.SCEELKED:SYS1.SCEELKEX
```

```
% export _C89_PSYSLIB=SYS1.SCEE0BJ
```

重新运行 ./build-zos 命令。如果命令再次失败, 请联系系统管理员, 以帮助查找需要的库。

IEW2456E 9207 SYMBOL xxx UNRESOLVED

无法解析的符号错误表示构建需要系统 C 库定义实际上不存在的符号。在大多数情况下, 这是其他系统中通常也会缺少的符号, 在 config.h 中将有对此问题提供变通方法的设置。

例如，系统可能未定义 `unsetenv` 函数。`configure-zos` 脚本通常应当检测到这一情况；否则，编辑代理程序源代码包随附的 `config.h` 文件，如下所示：

```
Change #define HAVE_UNSETENV 1 to #undef HAVE_UNSETENV.
```

重新运行 `./build-zos` 命令以更正此问题。

注：其他函数存在类似的 `#define` 语句。

代理程序更新和部署实用程序

代理程序更新和部署实用程序 **agentupdate** 简化了在多台计算机上安装和更新 IBM Rational Build Forge 代理程序的过程。

在 Rational Build Forge 自动执行任务的所有计算机上需要 Rational Build Forge 代理程序。在许多计算机上安装或更新代理程序可能非常耗时。该实用程序可以同时所有目标计算机上安装或更新代理程序。您可以在命令行上或通过 Rational Build Forge 提供的项目启动该实用程序。该实用程序使用您提供的 XML 文件以确定目标计算机、这些计算机的登录凭证、连接方法以及相关信息。

要点： 该工具支持单向安装或升级操作。

- 该工具不支持从安装或升级回滚。
- 该工具不会针对要安装的版本来检查现有代理程序的版本。如果您指定要安装的版本低于已安装的版本，也会安装较低版本。

启动该实用程序时，会为每台目标计算机完成以下操作：

1. 登录到计算机
2. 确定操作系统
3. 停止任何正在运行的代理程序

通过在用于实用程序的 XML 文件中使用 **StopCommand** 参数，您可以使用用于停止 Rational Build Forge 代理程序的定制命令。

4. 确定计算机是否具有足够的可用临时空间
5. 将适用于操作系统的代理程序安装程序复制到目标计算机
6. 运行安装程序以更新或安装计算机上的代理程序

如果代理程序位于缺省位置，该过程会保存您的 `BFAgent.conf` 文件。如果代理程序位于您通过在用于实用程序的 XML 文件中使用 **ExistingAgentPath** 参数指定的其他位置，该过程也会保存该文件。

7. 从临时空间除去安装程序

注：实用程序还创建了一个日志文件以捕获其操作的历史记录。该日志文件存储在运行实用程序的计算机上。

受支持的操作系统

代理程序更新和部署实用程序适用于两种操作系统。

实用程序在以下操作系统上运行：

- Windows

- Linux

注释:

- 运行实用程序的计算机必须安装 IBM Rational Build Forge 所包含的 IBM Java。
- 实用程序需要 Rational Build Forge 所包含的 IBM Tivoli Remote Execution and Access。

实用程序可以在以下操作系统上安装和更新 Rational Build Forge 代理程序:

- Windows
- Linux
- IBM AIX
- HP-UX
- Solaris

安装

IBM Rational Build Forge 安装程序可在以下文件中安装代理程序更新和部署实用程序:

- Windows 系统
 - <bfinstall>\java\agentupdate.jar
 - <bfinstall>\agentupdate.bat
- UNIX 和 Linux 系统
 - <bfinstall>/Platform/java/agentupdate.jar

运行 UNIX 或 Linux 的目标计算机的连接设置

必须配置运行 UNIX 或 Linux 的目标计算机, 以便可以使用 IBM Tivoli Remote Execution and Access 访问这些计算机。

为确保可通过 Tivoli Remote Execution and Access 进行访问, 请确保目标计算机满足以下需求。

SSH: 确保在代理程序更新和部署实用程序将访问的任何 UNIX 或 Linux 目标计算机上启用 SSH。

运行 Windows 的目标计算机的连接设置

必须配置运行 Windows 的目标计算机, 以便可以使用 IBM Tivoli Remote Execution and Access 发现这些计算机。

为确保可通过 Tivoli Remote Execution and Access 进行访问, 您有以下两个选项可供选择。

- **SSH/RSH:** 确保在代理程序更新和部署实用程序将访问的目标计算机上启用 SSH/RSH。

如果运行 Windows 操作系统的目标计算机具有 SSH 或 RSH, 那么可以跳过其余的配置步骤。

- 配置运行 Windows 操作系统的目标计算机以满足以下部分中的 Tivoli Remote Execution and Access 需求。第一部分针对所有 Windows 操作系统。后续部分针对所指示的 Windows 操作系统。

注：本主题中的 Windows 信息取自以下链接中提供的信息。有关最新信息，请参阅 Windows 操作系统上 IBM Tivoli Remote Execution and Access 配置的以下链接：

- http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/index.jsp?topic=/com.ibm.tivoli.tpm.scenario.doc/sdi/rept_prereq_win.html
- http://publib.boulder.ibm.com/infocenter/rafwhelp/v7r1/index.jsp?topic=/com.ibm.rafw.install.doc/topics/r_network_connection_reqs_Windows.html

所有 Windows 操作系统

- 必须启用目标计算机上的远程注册表管理（它是缺省配置）。打开服务，如下所示：
 - Windows XP 和 Windows Vista: 选择**开始 > 控制面板 > 管理工具 > 服务**
 - Windows Server 2003 和 Windows Server 2008: 选择**开始 > 设置 > 控制面板 > 管理工具 > 服务**
- 在 **Remote Registry** 服务中，确保该服务的状态为“已启动”。
- 必须启用管理共享以使用 Tivoli Remote Execution and Access 连接至运行 Windows 的目标计算机。缺省管理磁盘共享的示例是 C\$ 和 D\$。如果您禁用共享，那么 Tivoli Remote Execution and Access 会将位于驱动器内的目录视为隐藏。在这种情况下，将显示以下消息：

```
XCIM0009E: Error connecting to remote target <host_name>. Exception: java.io.FileNotFoundException:
CTGRI0003E The remote path name specified cannot be found: <file_or_directory_path>.
Cause: com.starla.smb.SMBException: The network name is incorrect.
```

按照以下步骤启用管理共享：

1. 单击**我的电脑**。
 2. 右键单击要启用管理共享的磁盘驱动器。
 3. 单击**共享和安全**。
 4. 选择**共享此文件夹**。
 5. 指定共享名（如 C\$ 或 D\$），然后单击**确定**。
- 如果禁用了目标上的 Windows 脚本编制主机 (WSH) 或 WMI 服务，或者因其他原因禁用了 VBScript，那么某些 Windows 协议方法将无效。

Windows XP Professional

- 运行 Windows XP 操作系统的目标计算机必须禁用简单文件共享，Tivoli Remote Execution and Access 才能工作。简单网络会强制所有登录以访客身份进行认证。访客登录不具有 Tivoli Remote Execution and Access 运作所需的权限。要禁用简单文件共享：
 1. 在 Windows 资源管理器窗口中，单击**工具 > 文件夹选项**，然后单击**查看**选项卡。
 2. 在“高级设置”列表中，清除**使用简单文件共享**复选框。
 3. 单击**应用**，然后单击**确定**。
- 必须启用目标计算机上的端口 135 (RPC) 和 445 (TCP) 以确保使用 Tivoli Remote Execution and Access 成功进行通信。如果 Tivoli Remote Execution and Access 发现端口 445 被禁用，那么它会使用端口 139 (NetBIOS over TCP/IP)。

- 防火墙：Windows XP 操作系统包括称为“Internet 连接防火墙”的内置防火墙（缺省情况下是禁用的）。另外，Windows XP Service Pack 2 (SP2) 随附 Windows 防火墙（缺省情况下是启用的）。如果在运行 Windows XP 的目标计算机上启用了任何防火墙，那么防火墙将阻止 Tivoli Remote Execution and Access 进行访问。

在 Windows XP SP2 上，您可以选中 Windows 防火墙配置的“例外”页面上的**文件和打印机共享**复选框以允许访问。

Windows Vista

- 必须为 Guest 或 Everyone 帐户启用文件共享，并禁用密码保护的共享。

要禁用密码保护的共享：

1. 单击**开始** > **控制面板**。
 2. 单击**网络和 Internet**，然后单击**共享和发现**。
 3. 单击**密码保护的共享**旁边的向下箭头。
 4. 单击**关闭密码保护的共享**。
 5. 单击**应用**，并退出控制面板。
- 您可能需要修改注册表项：
 1. 在左下角的**开始** > **启动搜索**字段中，输入 regedit，然后按 Enter 键。
 2. 在左窗格中，浏览至 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system 文件夹。
 3. 右键单击右窗格中的空白区域。
 4. 单击**新建**。
 5. 单击 **DWORD 值**。
 6. 输入 LocalAccountTokenFilterPolicy。
 7. 双击您创建的项。
 8. 在框中输入 1。
 9. 单击**确定**。
 10. 重新启动计算机。
 - 防火墙：Windows Vista 包含称为“Internet 连接防火墙”的内置防火墙（缺省情况下是禁用的）。如果在运行 Windows Vista 操作系统的目标计算机上启用了任何防火墙，那么防火墙将阻止 Tivoli Remote Execution and Access 进行访问。

Windows Server 2003

- 必须启用目标计算机上的端口 135 (RPC) 和 445 (TCP) 以确保使用 Tivoli Remote Execution and Access 成功进行通信。如果 Tivoli Remote Execution and Access 发现端口 445 被禁用，那么它会使用端口 139 (NetBIOS over TCP/IP)。

Windows Server 2008

- 必须为 Guest 或 Everyone 帐户启用文件共享，并禁用密码保护的共享。

要禁用密码保护的共享：

1. 单击**开始** > **控制面板**。
2. 单击**网络和 Internet**，然后单击**共享和发现**。
3. 单击**密码保护的共享**旁边的向下箭头。

4. 单击**关闭密码保护的共享**。
 5. 单击**应用**，并退出控制面板。
- 您可能需要修改注册表项：
 1. 单击**开始 > 运行**。
 2. 输入 `regedit`，然后按 **Enter** 键。
 3. 在左窗格中，浏览至 `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system` 文件夹。
 4. 右键单击右窗格中的空白区域。
 5. 单击**新建**。
 6. 单击 **DWORD 32 位值**。
 7. 输入 `LocalAccountTokenFilterPolicy`。
 8. 双击您创建的项。
 9. 在框中输入 `1`。
 10. 单击**确定**。
 11. 重新启动计算机。
 - 防火墙：Windows Server 2008 包含称为“Internet 连接防火墙”的内置防火墙（缺省情况下是禁用的）。如果在运行 Windows Server 2008 操作系统的目标计算机上启用了防火墙，那么防火墙将阻止 Tivoli Remote Execution and Access 进行访问。

用法概述

要使用代理程序更新和部署 (**agentupdate**) 实用程序，需要三个步骤。

要在多台计算机上安装或更新 IBM Rational Build Forge 代理程序：

1. 将目录中的代理程序安装程序置于将运行 **agentupdate** 实用程序的计算机上。

您可以从安装介质以及从位于 Jazz.net 的 Jazz 社区站点上的 Rational Build Forge 项目区域获取安装程序。

2. 创建一个 XML 文件以指定用于安装或更新 Rational Build Forge 代理程序的计算机。

注：将登录凭证包含在 XML 文件中时，以明文形式指定这些凭证。有关对密码进行加密的信息，请参阅第 146 页的『运行 **agentupdate** 实用程序』。

3. 从命令行或通过提供的 Rational Build Forge 项目运行 **agentupdate** 命令。

有关 XML 文件的更多信息，请参阅『在 XML 文件中指定目标系统』。

有关如何运行实用程序的更多信息，请参阅第 146 页的『运行 **agentupdate** 实用程序』。

在 XML 文件中指定目标系统

使用 XML 文件指定用于更新或安装代理程序的目标系统。

XML 文件概述

代理程序更新和部署实用程序需要 XML 文件以确定用于更新或安装代理程序的计算机。该文件还必须指定登录凭证和实用程序需要的其他项。

文件具有以下需求:

- 一个 <AgentDeploymentAndUpdate> 元素, 用于包含其他元素
- 一个或多个 <Target> 元素, 用于指定每个目标计算机
- 如果使用 <Globals> 元素, 那么只能使用一个。将其用于适用于所有目标计算机的项

以下示例显示了单个 <Target> 元素和一个 <Globals> 元素。<Globals> 元素在只有单个 <Target> 元素的情况下是无用的。<Globals> 元素在具有多个 <Target> 元素的情况下有用。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Target id="9.184.112.152">
    <Parameter Name="ConnectionType" Value="ssh"/>
    <Parameter Name="UserName" Value="root"/>
    <Parameter Name="Password" Value="passWord"/>
    <Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="GZIPPath" Value="/usr/contrib/bin"/>
  </Target>

  <Globals>
    <Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
  </Globals>

</AgentDeploymentAndUpdate>
```

XML 文件语法

每个 <Parameter> 元素在 <Target> 和 <Globals> 元素中都有效。有了这种灵活性, 如果参数适用于所有目标计算机, 您可以使用 <Target> 元素以仅指定 id 属性并将每个 <Parameter> 元素放置在 <Globals> 元素中。

指定 Value="" 的参数相当于不指定该参数。

以下各表描述了文件语法。

目标属性	值
id	目标计算机的名称或 IP 地址。

参数	值
AgentInstallLocation	仅针对 IBM AIX、HP-UX 和 Windows 系统 - 对其他平台没有任何影响 放置新代理程序的位置。如果未指定, 将使用系统缺省临时目录。
BFAgentPort	仅针对 IBM AIX 和 HP-UX 系统 代理程序要使用的端口。生成响应文件时使用。
BFASolarisAdminFile	仅针对 Solaris 系统 目标系统上存储用于静默安装的响应文件的位置。如果未指定, 将使用系统缺省临时目录。
BFASolarisResponseFile	仅针对 Solaris 系统 目标系统上存储响应文件的位置。如果未指定, 将使用系统缺省临时目录。

参数	值
ConnectionType	UNIX 或 Linux: SSH 或 RSH Windows: SSH、RSH、SMB 或 CIFS (RSH 是远程 shell。SMB 是服务器消息块。CIFS 是通用因特网文件系统。)
ConnectionPort	(可选) 如果目标未使用缺省端口, 那么为用于指定的连接类型(通过 ConnectionType 参数)的端口号。如果目标使用缺省端口, 那么您无需指定该参数。
ExistingAgentPath	IBM AIX、HP-UX 和 Windows 系统所需。 目标计算机上现有代理程序的位置。实用程序会在指定的位置中创建 BFAgent.conf 文件的备份, 然后在完成安装或升级后将其恢复。对于其他系统, 将使用缺省系统安装位置。
GZIPPath	IBM AIX、HP-UX 和 Solaris 系统。 代理程序安装程序可在其中找到 gzip 命令(用于解压缩安装程序)的目录。
IsSudoEnabled	如果在目标系统上启用了 sudo, 并且您要使用 sudo, 那么设置为 yes。将 SudoPassword 设置为正确的 sudo 密码。如果未设置, 那么实用程序会尝试使用 Password 以获取 sudo 登录。
KeyFile	运行实用程序的计算机上, 存储了用于密码加密和解密的密钥的文件。
LocalTempLocation	仅针对 IBM AIX、HP-UX 和 Windows。 目标系统上存储响应文件的位置。如果未指定, 将使用系统缺省临时目录。对于 Solaris 系统, 使用 BFASolarisResponseFile 或 BFASolarisAdminFile。
ManagedScriptsDir	设置为运行实用程序的目录。该目录位于运行 agentupdate 实用程序的计算机上。
SourceDirectory	存储目标计算机的安装程序(使用 .exe、.rpm 和 .gz 格式)的目录。该目录位于运行 agentupdate 实用程序的计算机上。
StartCommand	(可选) 目标计算机上用于启动 Rational Build Forge 代理程序的定制命令。 确保该命令包含在 PATH/path 变量中, 或者在 XML 文件中提供完整路径。
StopCommand	(可选) 目标计算机上用于停止 Rational Build Forge 代理程序的定制命令。 确保该命令包含在 PATH/path 变量中, 或者在 XML 文件中提供完整路径。
SudoPassword	用于 sudo 的密码。请参阅 IsSudoEnabled 。
Password	登录到目标计算机时用于 UserName 的密码。 要点: 最初在 XML 文件中以明文形式定义 Password 参数。如第 146 页的『运行 agentupdate 实用程序』中所示, 使用 -encrypt 命令行选项对密码进行加密。
TemporaryLocation	运行实用程序之前, 目标计算机上用于存储安装程序的位置。
UseManagedScripts	设置为 yes 或 no。如果设置为 yes, 那么实用程序会尝试在 ManagedScriptsDir 目录中找到 StartCommand 和 StopCommand 。
UserName	登录到目标计算机时使用的用户名。

样本 XML 文件

这些样本演示了如何使用 XML 文件。

- 样本 1

该样本只有一个目标。目标 IP 地址为 9.164.259.143。<Target> 元素不包含可选参数 **SourceDirectory** 或 **TemporaryLocation**。但是, <Globals> 元素具有 **SourceDirectory** 参数。该实用程序将该参数的值用于此目标。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

<Globals>
<Parameter Name="SourceDirectory" Value="..\test"/>
<Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
</Globals>

<Target id="9.164.259.143">
<Parameter Name="ConnectionType" Value="ssh"/>
<Parameter Name="UserName" Value="root"/>
<Parameter Name="Password" Value="toor"/>
<Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
<Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
</Target>

</AgentDeploymentAndUpdate>
```

- 样本 2

该样本具有四个目标。但是，任何目标都不需要特定于该目标的参数。因此，所有参数都在 <Globals> 部分中指定，包括 **UserName**、**Password** 和 **ConnectionType**。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

<Globals>
<Parameter Name="SourceDirectory" Value="..\test"/>
<Parameter Name="ConnectionType" Value="ssh"/>
<Parameter Name="UserName" Value="root"/>
<Parameter Name="Password" Value="toor"/>
<Parameter Name="TemporaryLocation" Value="/tmp"/>
<Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
<Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
<Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
</Globals>

<Target id="9.164.259.143">
</Target>

<Target id="9.164.102.169">
</Target>

<Target id="9.164.102.196">
</Target>

<Target id="9.106.189.157">
</Target>

</AgentDeploymentAndUpdate>
```

- 样本 3

您可以为目标系统指定 IP 地址的范围。以下示例适用于 IP 地址 9.184.112.152 到 9.184.112.160（在每个系统上都使用相同的登录凭证：登录名 root，密码 password）。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>
<Target id="9.184.112.152-160">
<Parameter Name="ConnectionType" Value="ssh">
<Parameter Name="ConnectionPort" Value=""/>
<Parameter Name="UserName" Value="root">
<Parameter Name="Password" Value="passWord">
</Target>
</AgentDeploymentAndUpdate>
```

- 样本 4

如果所有目标都具有相同属性（连接类型、连接端口、用户名和密码），那么可在文件的 **Globals** 部分指定这些属性。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>
  <Globals>
    <Parameter Name="ConnectionType" Value="ssh">
    <Parameter Name="ConnectionPort" Value="">
    <Parameter Name="UserName" Value="root">
    <Parameter Name="Password" Value="passWord">
  </Globals>

  <Target id="9.184.112.152-160">
</Target>

  <Target id="9.184.80.12-16">
</Target>
</AgentDeploymentAndUpdate>
```

运行 agentupdate 实用程序

创建 XML 文件后，可以在命令行上或通过提供的 **BuildForgeAgentUpdateUtilityProject** 运行代理程序更新和部署实用程序。

安装注意事项

该工具支持单向安装或升级操作。

- 该工具不支持从安装或升级回滚。
- 该工具不会针对要安装的版本来检查现有代理程序的版本。如果您指定要安装的版本低于已安装的版本，也会安装较低版本。

在命令行上运行实用程序

以下示例演示了如何在命令行上运行实用程序。有关实用程序的语法，请参阅 第 147 页的『agentupdate 实用程序引用』。

以下命令使用 **-preview** 选项。通过该选项，实用程序将连接至目标计算机并打印将在这些目标上运行的命令（实际并不运行命令）。

```
java -jar agentupdate.jar -preview -targets agent_deploy_config.xml -keyfile mykeyfile.txt
```

下一条命令对输入 XML 文件中的所有明文密码进行加密。该命令未连接至任何远程计算机。

```
java -jar agentupdate.jar -encrypt -targets agent_deploy_config.xml -keyfile mykeyfile.txt -logFile LogFile.txt
```

此命令会在目标计算机上更新或安装 IBM Rational Build Forge 代理程序。**-sourceDir** 选项指示哪个目录包含所有目标的代理安装程序。如果在运行实用程序时未包含该选项，请在 XML 文件中包含相应的参数。如果安装程序位于 XML 文件中针对所有目标的公共目录，请在 XML 文件的 **<Globals>** 元素中指定 **SourceDirectory** 参数。但是，如果任何一个目标的源目录不同，请在该计算机的 **<Target>** 元素中定义 **SourceDirectory** 参数。例如，请参阅第 144 页的『样本 XML 文件』中的样本 2。

```
java -jar agentupdate.jar -sourceDir .\resources -targets agent_deploy_config.xml -logFile LogFile.txt
```

通过 BuildForgeAgentUpdateUtilityProject 运行实用程序

IBM Rational Build Forge 可提供 bfaagentupdate.xml 样本项目，以演示如何使用该实用程序。

要使用样本项目，请通过使用管理控制台中的导入设施或使用 **bfimport** 命令导入该项目。从 C:\Program Files\IBM\Build Forge\samples\agentupdate\BFproject\ (Windows) 或 /opt/buildforge/samples/agentupdate/BFproject/ (Linux) 导入该项目。

项目需要三个步骤：预览、加密和运行。预览和加密步骤不包含在样本项目中。

- 预览步骤：构建系统将连接至目标计算机并显示将在每个目标上运行的用于更新或安装代理程序的命令（实际并不运行命令）。
- 加密步骤：构建系统将对所有密码进行加密，而不更新或安装目标计算机上的代理程序。该步骤会将输入 XML 文件中的明文密码替换为加密的密码。
- 运行步骤：构建系统将连接至目标计算机，标识其操作系统，停止所有正在运行的代理程序，检查安装程序的临时空间，将安装程序复制到目标计算机，完成更新或安装，然后从临时空间除去安装程序。

根据您的需求启用相应步骤，然后运行项目。

agentupdate 实用程序引用

实用程序提供一些命令行选项和环境变量以控制其行为。

下表按字母顺序列出了命令行选项。

选项	必需	描述
-debug	否	启用扩展的调试输出。 您还可以通过将 DEBUG 环境变量设置为任意值来启用调试。有关优先顺序的信息，请参阅 第 148 页的『XML 文件参数、命令行选项和环境变量的优先顺序』。
-encrypt	否	将 XML 文件中的明文密码替换为密码的加密形式。
-genkey <key_file>	否	针对密码的加密和解密生成密钥。 指定 -genkey 选项时，还必须指定 -keyfile <key_file> 选项以将密钥存储在指定文件中。
-keyfile <key_file>	否	保存用于密码的加密和解密的密钥。 您还可以通过设置 KeyFile XML 文件参数来指定该文件。有关优先顺序的信息，请参阅 第 148 页的『XML 文件参数、命令行选项和环境变量的优先顺序』。
-logFile <log_file>	否	指定实用程序记录消息的文件。 缺省情况下，名称为 RemoteAgentDeployerUpdater_Log.txt。
-maxThreads	否	设置运行时实用程序使用的最大线程数。缺省情况下，它设置为主机硬件中处理器核心数的两倍。
-preview	否	列出（而不运行）实用程序将针对每个目标运行的命令。

选项	必需	描述
-sourceDir <path>	否	指定各种平台的代理程序安装程序（使用 .exe、.rpm 和 .gz 格式）将保存到的目录。 agentupdate 实用程序确定每个目标的操作系统，然后使用该目录中对应的代理程序安装程序。 注： 该实用程序使用代理程序安装程序。它不使用代理程序源代码。 您还可以通过设置 SourceDirectory XML 文件参数或 SOURCEDIR 环境变量来指定该目录。有关优先顺序的信息，请参阅『XML 文件参数、命令行选项和环境变量的优先顺序』。
-targets <XML_file>	是	针对您创建用于指示更新和安装 IBM Rational Build Forge 代理程序的目标计算机的 XML 文件，指定其路径和文件名。

下表描述了环境变量。

环境变量	必需	描述
DEBUG	否	启用扩展的调试输出。 您还可以通过指定 -debug 命令行选项来启用调试。有关优先顺序的信息，请参阅『XML 文件参数、命令行选项和环境变量的优先顺序』。
SOURCEDIR	否	指定各种平台的代理程序安装程序（使用 .exe、.rpm 和 .gz 格式）将保存到的目录。 agentupdate 实用程序确定每个目标的操作系统，然后使用该目录中对应的代理程序安装程序。 注： 该实用程序使用代理程序安装程序。它不使用代理程序源代码。 您还可以通过设置 SourceDirectory XML 文件参数或 -sourceDir 命令行选项来指定该目录。有关优先顺序的信息，请参阅『XML 文件参数、命令行选项和环境变量的优先顺序』。

XML 文件参数、命令行选项和环境变量的优先顺序

您可以通过多种方式控制实用程序。优先顺序确定哪种方法实际影响实用程序。

代理程序更新和部署实用程序按优先顺序提供以下用于控制其行为的以下方法：

1. <Target> 元素中的参数
2. 命令行选项
3. <Globals> 元素中的参数
4. 环境变量

实用程序按顺序考虑每种方法。如果某个方法是使用有效值定义的，那么实用程序将使用该值并忽略以下所有方法。

示例

该示例描述实用程序如何找到安装程序。<Target> 和 <Globals> 元素已定义 **SourceDirectory** 参数。这些参数指示安装程序的可能位置。但是，实用程序具有一些要检查的位置。一旦实用程序找到安装程序，实用程序会使用该安装程序。

1. 实用程序先检查 `<Target>` 元素中的参数。如果 `..\test\wind` 目录存在，那么实用程序会检查该目录中是否存在安装程序。如果安装程序位于该目录中，那么实用程序将使用该安装程序。
2. 如果目录不存在或未包含安装程序，那么实用程序会检查是否指定了 `-sourceDir` 命令行选项。如果该选项中指定了有效目录，那么实用程序会检查该目录并使用该目录中的安装程序。
3. 如果未指定 `-sourceDir`、目录无效或目录未包含安装程序，那么实用程序会检查 `<Globals>` 元素中的参数。现在，实用程序将检查 `..\test` 目录。
4. 如果 `..\test` 目录未生成安装程序，那么实用程序将检查 `SOURCEDIR` 环境变量。如果该目录不存在或未包含安装程序，那么实用程序会生成错误。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

<Globals>
<Parameter Name="SourceDirectory" Value="..\test"/>
<Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
</Globals>

<Target id="9.164.102.169">
<Parameter Name="SourceDirectory" Value="..\test\wind"/>
<Parameter Name="ConnectionType" Value="ssh"/>
<Parameter Name="UserName" Value="root"/>
<Parameter Name="Password" Value="toor123"/>
<Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
<Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
</Target>

</AgentDeploymentAndUpdate>
```

通过日志文件查看实用程序操作

日志文件提供实用程序操作的历史记录。

您可以使用 `-logFile` 选项指定用于记录消息的文件。如果没有指定 `-logFile` 选项，那么实用程序会将消息记录到 `RemoteAgentDeployerUpdater_Log.txt` 中。该文件是在存储 **agentupdate** 实用程序的目录中创建的。

日志文件的格式为 `[MSGTYPE] [IP address/MachineName] [Msg]` 。

所有目标的消息都存储在同一日志文件中。

故障诊断

可通过 **agentupdate** 实用程序提供的多种方式来显示信息以帮助对问题进行故障诊断。

缺省情况下，实用程序会提供足够的信息以了解常规问题，如连接问题、磁盘空间不足等。但是，如果您需要更多调试信息，请使用 `-debug` 命令行选项。或者，您可以定义具有任意值的 `DEBUG` 环境变量以通过工具启用额外的日志消息（或更多调试信息）。

agentupdate 的消息引用

agentupdate 实用程序提供操作期间的错误和参考消息。

以下主题提供了对所有 **agentupdate** 消息的引用。

CRRBU0001I 已成功启动代理程序更新和部署实用程序。

CRRBU0002I 调试方式已开启。

CRRBU0003I 已完成密码加密。

CRRBU0004I 已解析 XML 配置文件。正在处理目标...

CRRBU0005I 正在处理目标 [{0}]

CRRBU0006I [{0}] 协议 = {1} 用户 = {2}

CRRBU0007I [{0}] 已加密的密码 = {1}

CRRBU0008I 已成功创建受管脚本目录: {0}。

CRRBU0009I I[{0}] 已解密的密码 = #{1}#

CRRBU0010I 参数 keyfile = {0}。

CRRBU0011I 子代数: {0}。

CRRBU0012I 参数 genkey = {0}。

CRRBU0013I [{0}] sudo 加密的密码 = {1}。

CRRBU0014I 文本节点

CRRBU0015I 注释节点

CRRBU0016I 属性节点

CRRBU0017I 文档节点

CRRBU0018I 文档类型节点

CRRBU0019I 元素节点

CRRBU0020I 未知节点

CRRBU0021I 目标总数: {0}

CRRBU0022I 元素计数 = {0}。

CRRBU0023I 正在针对用户标识为 {1} 的 {0} 尝试 windows 协议。

CRRBU0024I 正在使用 RXA 界面检查目标机器上的可用空间。

CRRBU0025I 目标系统上的可用空间 = {0} KB。

CRRBU0026I 正在检查操作系统 {0} 上安装位置 {1} 中的代理程序。

CRRBU0027I 在以下位置中找到代理程序安装: {0}。

CRRBU0028I [正在停止进程]

CRRBU0029I 进程标识 = {0}, 程序名 = {1}

CRRBU0030I 正在解析阶段: 元素: {0} = {1}

CRRBU0031I 无法停止目标上的代理进程 (bfagent)。

CRRBU0032I 正在停止目标上运行的代理程序: {0}。

CRRBU0033I 正在尝试停止 Windows 上的服务 {0}。

CRRBU0034I 正在停止目标上的代理服务 - 使用 RXA 界面。

CRRBU0035I 正在停止目标上的代理服务: {0}

CRRBU0036I 正在使用目标上的临时目录 {0}。

CRRBU0037I 正在检查目录中的前缀匹配项。{0}

CRRBU0038I 正在 <{0}> 下查找安装程序。

CRRBU0039I 正在将安装程序 <{0}> 复制到目标上 - 使用 RXA 界面。

CRRBU0040I 已创建密钥文件: {0}

CRRBU0041I 用户配置文件的备份: {0}

CRRBU0042I 正在清除目标上的 {0}。

CRRBU0043I 正在除去目录 [{0}]

CRRBU0044I 代理程序安装位置 {0}

CRRBU0045I 安装正在 {0} 处继续进行

CRRBU0046I 安装/升级目标: {0}

CRRBU0047I 预览方式已启用。

CRRBU0048I 在目录 {0} 中已创建 **response.txt**。

CRRBU0049I 正在恢复用户配置文件: {0}。

CRRBU0050I 正在运行命令: {0}

CRRBU0051I 当前的工作目录是 [{0}] 。

CRRBU0052I 正在复制到 {0}。

CRRBU0053I 正在杀死目标上运行的 **bfdispatch** 进程: {0}

CRRBU0054I 已针对目标启动新线程。

CRRBU0055I 现在, 代理程序已安装到目标上。

CRRBU0056I 远程访问问题: 预览方式不能继续。

CRRBU0057I 正在使用 {0} 停止目标上运行的代理程序

CRRBU0058I 已成功停止目标上的代理程序。

CRRBU0059I 代理程序未安装到目标上。

CRRBU0060I 目标上的临时存储位置是 {0}。

CRRBU0061I 对所有密码进行加密。

CRRBU0062I 安装程序 {0} 将用于 {2} 中的 {1}。

CRRBU0063I 已成功将 {0} 复制到临时位置。

CRRBU0064I 已成功备份配置文件。

CRRBU0065I 现有的代理程序安装位置: {0}。

CRRBU0066I 使用 {0} 启动目标上运行的代理程序

CRRBU0067I 正在退出线程。

CRRBU0068I 已创建名为 {0} 的管理文件

CRRBU0069I 节点: {0}

CRRBU0070I 节点属性: {0}

CRRBU0071I 参数 **logFile** = {0}

CRRBU0072I 参数 **sourceDir** = {0}

CRRBU0073I 参数 **sourceFile** = {0}

CRRBU0074I 参数 **targets** = {0}

CRRBU0075I **getOS**: [{0}][{1}]

CRRBU0084I 包含目标信息的文件的路径 (必需)。

CRRBU0085I 已从以下位置创建密钥: {0}。

CRRBU0088I 最大线程数已设置为: {0}

CRRBU1000W [{0}] 协议未指定。

CRRBU1001W [{0}] 目标协议不受支持: {1}

CRRBU1002W XML 文件中没有 **Globals** 部分。

CRRBU1003W XML 文件中没有 **Targets** 部分。

CRRBU1004W 无效的目标条目：在 XML 文件中找不到某个目标部分的标识。

CRRBU1005W [stderr] {0}

CRRBU1006W [stdout] {0}

CRRBU1007W 目标上的临时位置已设置为 **null**。

CRRBU1008W 不受支持：Sun Solaris V{0}

CRRBU1009W 在为 XML 文件的某个目标部分指定的源目录中找不到安装程序。

CRRBU1010W 在命令行中未指定安装程序位置。

CRRBU1011W 在 XML 文件的 **Globals** 部分中未指定安装程序位置。

CRRBU1012W 在环境中未指定安装程序位置。

CRRBU1013W 目标错误：无法生成临时管理文件。

CRRBU1014W 目标错误：无法创建临时响应文件。

CRRBU1015W [返回码] {0}

CRRBU2000E 捕获到异常。

CRRBU2001E 本地或远程位置中不存在启动脚本。

CRRBU2002E 无法创建受管脚本目录：{0}

CRRBU2003E 缺少必需的参数：{0}

CRRBU2004E **TargetInfoTable** 数据结构的用法不正确。

CRRBU2005E 在 XML 文件中找到多个 **Globals** 标记。请仅指定一个。

CRRBU2006E 参数 **ManagedScriptsDir** 未定义。

CRRBU2007E 运行启动命令 <{0}> 时出现问题

CRRBU2008E [{0}] 未指定明文密码，未指定 **KeyFile**

CRRBU2009E 为启动/停止脚本创建文件对象时出错。

CRRBU2010E 无效的目标条目：XML 文件中某个目标的主机名或地址为 **null**。

CRRBU2011E 在本地或远程位置中不存在停止脚本。

CRRBU2012E 写入 XML 文件时出错。

CRRBU2013E 将忽略重复的目标条目：XML 文件中的目标 {0}。

CRRBU2014E 没有提供足够的空间。在 {2} 上，提供的可用空间 = {0}，所需可用空间 = {1}

CRRBU2015E 捕获到连接异常：{0}

CRRBU2016E 捕获到“找不到文件”异常：{0}

CRRBU2017E 捕获到 I/O 异常：{0}

CRRBU2018E 停止目标上的代理进程 (**bfagent**) 时出错。

CRRBU2019E 停止目标上的代理服务时出错。

CRRBU2020E 目录无效或者安装程序不存在：{0}

CRRBU2021E 目录中有太多的文件匹配前缀：{0}

CRRBU2022E 目录中没有文件匹配前缀：{0}

CRRBU2023E 未指定安装程序源目录或者在指定的目录中找不到安装程序。

CRRBU2024E 错误：无法杀死目标上的 **bfdispach** 进程。

CRRBU2025E 运行停止命令 `</0>` 时出现问题

CRRBU2026E 无法完成停止命令。不在 `{0}` 上进一步处理

CRRBU2027E 停止目标上的代理程序时出现问题。

CRRBU2028E 无法完成停止命令。不进一步处理。

CRRBU2029E 目标上的磁盘空间不足： `{0}`

CRRBU2030E 找不到安装程序。

CRRBU2031E 将 `{0}` 复制到临时位置失败。

CRRBU2032E 备份配置文件失败。

CRRBU2033E 恢复配置文件失败。

CRRBU2034E 由于 `{0}` 的认证错误，导致远程访问失败

CRRBU2035E 操作系统不受支持。

CRRBU2036E 无法完成启动命令。不在 `{0}` 上进一步处理

CRRBU2037E 参数 **ManagedScriptsDir** 未定义。

CRRBU2038E 平台不受支持。

CRRBU2039E **StopScript** 没有本地文件，或存在多个具有前缀 **StopScript_** 的条目。

CRRBU2040E **StartScript** 没有本地文件，或存在多个具有前缀 **StartScript_** 的条目。

CRRBU2041E 无法向服务器进行认证。

CRRBU2042E 无法创建密钥文件 `{0}`。

CRRBU2043E 加密算法错误。确保在您的环境中提供了 **DES**。无法创建密钥文件。

CRRBU2044E 密钥文件 `{0}` 不是有效的密钥文件。无法设置密钥文件。

CRRBU2045E 加密算法错误。确保在您的环境中提供了 **DES**。无法设置密钥文件。

CRRBU2046E 无法访问密钥文件 `{0}`。无法设置密钥文件。

CRRBU2047E 加密算法错误。确保在您的环境中提供了 **DES**。加密失败。

CRRBU2048E 使用 **DES** 的加密的填充在环境中不可用。加密失败。

CRRBU2049E 从密钥文件生成的密钥无效。确保您使用的是正确的密钥文件。解密失败。

CRRBU2050E 不支持针对您的密码的字符编码。加密失败。

CRRBU2051E 为密码提供的数据的长度不正确。加密失败。

CRRBU2052E 预期的数据填充不存在。加密失败。

CRRBU2053E 加密算法错误。确保在您的环境中提供了 **DES**。解密失败。

CRRBU2054E 使用 **DES** 的加密的填充在您的环境中不可用。解密失败。

CRRBU2055E 从密钥文件生成的密钥无效。确保您使用的是正确的密钥文件。解密失败。

CRRBU2056E 为密码提供的数据的长度不正确。解密失败。

CRRBU2057E 预期的数据填充不存在。解密失败。

CRRBU2059E 未指定任何密钥文件。

CRRBU2058E 不支持针对您的密码的字符编码。解密失败。

CRRBU2060E {0} 不是有效答案。

运行代理程序

本部分描述如何设置以运行代理程序。它通常作为自动启动的服务或守护程序运行。

在 Windows 上运行代理程序

通常代理程序会作为服务安装并设置为“自动”，这样它会在打开计算机时就启动。您必须登录安装有代理程序的计算机才能启动和停止该代理程序。

要启动和停止代理程序，可以使用开始菜单：

- 要启动代理程序，请单击 **启动 > 程序 > IBM Rational Build Forge > 启动代理程序服务**。
- 要停止代理程序，请单击 **开始 > 程序 > IBM Rational Build Forge > 停止代理程序服务**。

还可以在命令提示符中使用以下命令：

- `net start bfagent`
- `net stop bfagent`

在 UNIX、Linux 和 MacOS 上运行代理程序

代理程序将作为服务来运行，并且在系统重新启动时需要自动重新启动。

相应地将 `bfagent` 条目添加到 `inetd` 或 `xinetd` 的配置中。以下示例是 Linux 系统上 `xinetd.d` 中的 `bfagent` 条目，其中代理程序安装在 `/usr/local/bin` 中：

```
# description: The IBM Rational Build Forge Agent serves build requests
#   from the IBM Rational Build Forge Management Console.
service bfagent
{
    disable           = no
    flags             = REUSE
    socket_type       = stream
    wait              = no
    user              = root
    server            = /usr/local/bin/bfagent
    log_on_failure    += USERID
}
```

如有必要，代理程序可以在 `inetd/xinetd` 环境外部运行。要将其作为独立守护程序运行，请使用 `-s` 选项。

`bfagent -s`

使用该选项时，代理程序会转到后台并开始侦听连接。将此命令放在启动脚本中，这样代理程序会在计算机启动时自动启动。

在 System i 上运行代理程序

如果计划在 System i 平台上运行代理程序，请查看此主题中的信息。

验证代理程序端口号是否唯一

端口 5555 是标准的 Build Forge 代理程序端口，可以预分配给 System i 服务器上的其他代理程序。在此情况下，应在启动代理程序之前，将 Build Forge 代理程序端口更改为未分配的端口。为此，请直接编辑 `bfagent.conf` 文件。有关详细信息，请参阅第 157 页的『更改代理程序端口』。

手动启动代理程序

如果您已完成了安装指示信息第 134 页的『在 System i 平台上安装代理程序』中的步骤 7，那么 System i 启动时代理程序将作为 `BFAGENT` 用户启动。

或者，可以通过使用以下命令在 System i 上手动启动代理程序。

```
bfagent -s
```

注：如果 `bfagent.conf` 文件未安装在 `/etc`（缺省位置）中，请使用 `-f` 选项指定 `bfagent.conf` 位置。

当您发出 `bfagent` 命令并手动启动代理程序时，代理程序将作为启动该代理程序的用户启动。

- 如果 `QSECOFR` 用户或具有 `*ALLOBJ` 特权的用户启动代理程序，那么通过在管理控制台中指定的服务器认证对该用户进行认证。
- 如果其他用户启动代理程序，那么通过配置 `bfagent.conf` 文件中的 `magic_login` 设置对该用户进行认证。请参阅第 159 页的『`bfagent.conf` 引用』。

验证是否已安装 i5/OS PASE 程序

代理程序作为 i5/OS 可移植应用程序解决方案环境（PASE）程序运行。PASE 包含在 i5/OS 中，它使得 AIX 二进制文件和命令能够运行。PASE 通常已缺省安装。

要确定是否已安装 PASE 程序，请在命令行运行 `DSPSFWRSC`。

如果未安装 PASE 程序，请从安装 CD 装入该程序。

使用 PASE 中的代理程序

在 i5/OS 上构建应用程序所必需的大多数任务都可从 PASE 环境访问。当规划和定义用于 iSeries 平台的过程的自动化时，记住这一点很重要。

步骤中的命令由 PASE shell 解释。还可以使用以下语法运行本机命令：

```
system -bi0E "<本机命令>"
```

要点：步骤中的每条系统命令都运行它们自己的进程。这对只在自己进程中运行的命令有影响。

例如，如果想要为一组步骤设置库列表：

- 不能使用 `CHGSYSLIBL` 或 `ADDLIBLE` 作为步骤命令，因为它们是本机命令（PASE 无法识别）。
- 在步骤中不能使用受支持的本机命令语法（例如，`system -bi0E "ADDLIBLE FLGHT400"`），因为它只为命令自己的进程更改库列表。后续命令和步骤不受此更改的影响。

虽然不能仅为单个步骤、一组步骤或单个项目设置库列表，但是可以在 BFAGENT 用户的启动命令脚本中对它们进行设置。请参阅第 134 页的『在 System i 平台上安装代理程序』中的示例启动脚本。如果在启动命令脚本中设置库列表，那么会为示例中作为 BFAGENT 用户运行的所有项目和步骤设置库列表。运行项目和步骤的用户必须可以访问所需库。

要设置库列表，请为列出了所需库的代理程序添加作业描述。以下示例作业描述包括库 FLGHT400 和 FLGHT400M。

```
10    UTLIB
20    QGPL
30    QTEMP
40    FLGHT400
50    FLGHT400M
```

代理程序在其启动例程中指定此作业描述。例如，如果作业描述为 BFAJOB，那么系统启动例程中的行将如下所示：

```
ADDAJE SBSDB(BFAGENT/BFAGENT) JOB(BFAGENT) JOBD(BFAGENT/BFAJOB)
```

此解决方法会影响与该代理程序关联的 System i 服务器上运行的所有命令（任何步骤和项目）。

bfagent 引用

可执行文件 bfagent 启动 Build Forge 代理程序。它从同一目录中的 BFAgent.conf 文件读取配置。

命令的语法为：

```
bfagent [-f configfile | -s]
```

选项

-f configfile

使用 configfile 中的配置文件（而不是 BFAgent.conf）运行。在 UNIX 或 Linux 上，这是一个运行时选项。在 Windows 上手动运行代理程序时，这是一个调试选项。在 Windows 上，它无法用于启动服务。

-s 作为单机服务启动。您只能在 UNIX 或 Linux 上使用此选项。这种运行方式是用 **inetd** 或 **xinetd** 启动 bfagent 的另一种途径。

配置代理程序

本部分描述安装后如何配置代理程序。

查找代理程序配置文件

代理程序配置文件 BFAgent.conf 提供了代理程序操作的运行时配置。它包含解释所有可能选项的注释。该文件位于代理程序安装目录中：

- Windows 缺省值：C:\Program Files\IBM\Build Forge\Agent\BFAgent.conf
- UNIX 和 Linux 缺省值：/etc/bfagent.conf

要点：如果对安装目录中的 BFAgent.conf 文件进行了更改，就必须在以后对代理程序进行重新安装或升级后重复作出这些更改。每次安装时都会覆盖配置文件。

可以指定备用的配置文件:

- 在 UNIX 或 Linux 系统中, 可以使用代理程序安装目录外部的配置文件来保存代理程序配置。在执行此操作时, 请在启动代理程序的命令中使用 `-f` 命令行选项。示例:

```
bfagent -f /opt/bfagent.conf
```

- 在 Windows 系统上, 无法使用此选项启动服务。只能在手动运行代理程序时使用服务。它是调试工具。

更改代理程序端口

如果在端口 5555 已被占据的服务器上安装代理程序, 那么在安装后可以更改代理程序端口。

要在 Windows 操作系统上更改端口:

1. 打开注册表编辑器: 单击**开始** > **运行**, 然后输入 `regedit`。
2. 转至 `HKEY_LOCAL_MACHINE\SOFTWARE\BuildForge Agent`。
3. 将 `AgentPort` 的值更改为您期望的端口号。

在 UNIX、Linux 和 Macintosh 操作系统上:

1. 打开 `/etc/services` 文件。
2. 将 `BuildForge Agent Port` 的值更改为您期望的端口号。

配置不同的 shell

通过编辑 `BFAgent.conf` 文件中的参数, 可以将代理程序配置为使用缺省 shell 程序以外的 shell。

例如, 要将 Windows 系统更改为使用 MKSTools 提供的 Korn shell 程序, 可使用以下命令更改 `shell` 参数:

```
shell C:\MKSTools\mksnt\ksh.exe -L -c "%s\"
```

当系统向服务器发送命令时, 该命令中的 `%` 替换为步骤命令。在这种情况下, 使用反斜杠转义字符, 以在命令中包括引号作为文字。

在网络共享文件系统 (Windows) 上运行代理程序命令

Build Forge 代理程序最初将以 Windows 系统帐户凭证启动。要运行命令, 代理程序稍后将通过 Build Forge 服务器认证凭证向 Windows 进行认证。

服务器认证凭证对于本地命令是可接受的, 但是对于代理程序必须在外部网络共享驱动器上运行的某些命令可能失败。例如, 要在 ClearCase 动态视图中修改文件, 代理程序必须访问网络共享驱动器上的 ClearCase 文件。

命令失败, 因为外部文件系统忽略了代理程序服务器认证凭证; 它仅识别代理程序的初始系统帐户凭证。

如果在网络共享驱动器上运行命令时遇到问题, 尝试以下操作:

使用服务器认证凭证运行命令

要使用对网络共享具有访问权的 Build Forge 服务器认证凭证来运行命令, 请将 `win_reexec_after_auth` 设置添加至 `BFAgent.conf` 文件中。

如果想要使用 Build Forge 服务器认证凭证建立对网络共享的访问，添加此设置是先决条件。

win_reexec_after_auth 设置会使代理程序在向 Windows 认证后启动新进程。新进程将强制共享文件系统确认代理程序已更改了用户凭证。

如果设置了 win_reexec_after_auth，那么代理程序将作为服务运行，并且不区别访问网络共享的命令和不访问网络共享的命令，因此您可能要注意性能影响。

以单用户方式运行代理程序

在代理程序安装期间，将代理程序设置为在无 Build Forge 服务器认证凭证的情况下以单用户方式运行命令。选择**安装用户方式代理程序**选项。

如果指定的用户是“管理员”组的成员，那么必须使用服务器认证凭证指定用户凭证。

如果用户不是管理员，那么使用 BFagent.conf 中的 magic_login 设置可防止对代理程序进行未授权访问。

当您登录至管理控制台时，代理程序将使用您提供的用户名启动并运行，这将通过使用该用户的凭证立即授予对网络共享的访问权。

代理程序使用专用用户帐户作为服务运行

将代理程序设置为使用专用用户帐户作为 Windows 服务运行。该选项将您限制为作为单个用户帐户运行代理程序，但是不需要代理程序启动新进程进行重新认证，因此对性能无影响。

要使用专用用户帐户作为服务运行代理程序：

1. 在 Build Forge 服务器上，单击**管理工具 > 服务**以打开 Windows 控制面板。此时会打开服务列表。
2. 打开 IBM Rational Build Forge Agent 服务。
3. 为想要用于运行代理程序命令的用户提供用户帐户信息。例如，为 ClearCase 管理员用户或有权访问 ClearCase 动态视图和 VOB 的其他用户提供信息。

配置代理程序之间的直接文件传输

当 _XSTREAM_PROTOCOL 触发器变量出现在作业的环境中时，它支持在代理程序之间进行直接文件传输。发送方代理程序、接收方代理程序和引擎都必须支持直接文件传输。一些参数控制编码、网络 and 文件传输特性。

某些操作系统上的代理程序对于直接文件传输的支持存在限制：

- System z: 不支持直接文件传输。
- System i: 仅限 _XSTREAM_PROTOCOL 值为 PRNG 或 PLAIN 时才支持直接文件传输。不支持将 SSL 用于文件传输（_XSTREAM_PROTOCOL 值为 AES-CBC）。

请参阅第 270 页的『触发器变量引用』和第 159 页的『bfagent.conf 引用』参考资料中的 xstream_* 参数。

触发器变量和代理程序性能

以下两个触发器变量可通过减少为步骤日志生成的消息数，影响代理程序的整体性能：

- `_SUPPRESS_ENV_OUTPUT`: 如果设置该变量, 那么会禁止在步骤日志中打印 ENV 消息。
- `_SUPPRESS_LOG_OUTPUT`: 如果设置该变量, 那么会禁止在步骤日志中打印几乎所有消息。

请参阅“处理环境”中的“触发器变量参考信息”。如果触发器变量包含在项目环境或步骤环境中, 那么这些变量会影响作业。

bfagent.conf 引用

`bfagent.conf` 文件存储关于 Build Forge 代理程序运行方式的设置。该文件与 `bfagent` 可执行文件位于同一目录中。

该文件列出了所有设置和内部缺省值。不活动的设置已被注释掉。

设置

activity_log *path*

打开活动日志记录。将信息附加到 *path* 指定的文件中。路径必须存在, 并且代理程序用户必须对其具有写许可权。

注: 如果路径不存在或者不能对文件进行写操作, 代理程序不会报告错误。

要点: 文件大小没有限制。必须手动删除文件。此设置旨在临时用于调试代理程序。它不应用作工作代理程序的永久日志。

allow IP-address-or-range [...]

此设置仅适用于以下情况:

- 在 Windows 中运行的代理程序
- 以独立方式在 UNIX 或 Linux 上运行的代理程序 (启动时使用 `-s` 选项)

此设置限制到代理程序的连接。只允许来自与 `IP-address-or-range` 匹配的 IP 地址的连接。缺省情况下, 允许来自所有地址的连接。

指定以下一项或两项:

- **IP 地址:** 标准 IPv4 或 IPv6 地址。例如, 对于 IPv4, 标准地址为 255.192.192.003。允许特定的 IP 地址。
- **IP 地址范围:** 部分限定 IPv4 或 IPv6 地址。这些示例 (192.168 或 192.168.63) 对于 IPv4 是正确的。允许与此限定匹配的所有 IP 地址。

注: 如果正在超级服务器 (如 `inetd` 或 `xinetd`) 上运行代理程序, 请使用其他方法来控制访问。您可能希望使用防火墙、TCP 包装器 (`hosts.allow` 和 `hosts.deny`) 或 `xinetd` 的内置过滤功能。

bind

此设置允许用户为代理程序指定显式绑定地址。此设置与“port”设置共同确定代理程序在使用 `-s` 命令行选项启动时将如何侦听连接。`bfagent.conf` 文件中给出的值将强制代理程序绑定到 IPv4 本地主机地址; 这样, 代理程序只会接收到来自相同计算机上的控制台的连接。示例: `bind 255.192.192.003`

注: 它对通过系统的服务体系结构启动的 Windows 或 UNIX 代理程序 (如 `inetd`、`xinetd` 或 `launchd`) 没有影响。

ccviewroot root-path

此设置为此主机指定缺省视图根。请参阅关于 `init` 的 ClearCase 文档以了解更多信息。内部缺省值如下所示：

- Windows: `ccviewroot M:`
- UNIX 或 Linux: `ccviewroot/view`

cc_suppress_server_root

如果设置，那么查看路径为 `ccviewroot` 设置的路径。如果不设置，那么服务器定义中设置的路径将附加到 `ccviewroot` 设置的路径。此设置不需要值。如果 `bfagent.conf` 中存在此设置，将设置该值。

command_output_cache_size

此设置使代理程序高速缓存输出，直到达到指定的大小（以字节为单位）。内部缺省值不会存入高速缓存。使用高速缓存可以显著改善代理程序性能并减少网络开销。高速缓存大小取决于命令产生的输出量。

最小值：2048。当设置低于该值时，内部将使用值 2048。

cygwin

此设置仅用于 Windows 上的代理程序。

此设置使得代理程序能在使用 Cygwin、Linux 之类环境的 Windows 主机上工作。使用 Cygwin 时，对代理程序提供了若干 Linux 工具。

使用此设置时，可能还需要设置 `cygwin_script_magic` 和 `shell` 设置。示例显示了配置这些设置的一种方法：

```
cygwin
shell C:\cygwin\bin\bash.exe --login -c "%s"
cygwin_script_magic #!/bin/bash
```

`shell` 设置必须与您的 Cygwin 安装相匹配。

cygwin_script_magic

此设置仅用于 Windows 上的代理程序（如果设置了 **cygwin**）。

此设置指定在执行步骤时要使用的 `#!` 行。缺省为 `#!/bin/bash`。

default_logon_domain

用于指定在认证请求不包含域的情况下要使用的域。如果未指定，将使用代理程序计算机的域。

disable_telnet_support

为了得到最佳结果，可使用 `telnet` 测试代理程序连接。

对于该代理程序，存在与处理和正确处理 Telnet 控制序列相关联的某些内置处理开销。

使用此设置可禁用代理程序对特殊 `telnet` 字符代码的处理，因此可以稍微提高性能。在生产环境中，使用此设置可提高性能，从而使您受益。

disable_transcode

当操作系统未在使用 UTF-8 编码时，将关闭代理程序为了转换国际数据而执行的处理。要避免混合编码和数据损坏，请对代理程序操作系统使用 UTF-8。

如果操作系统不使用 UTF-8 编码，那么代理程序必须将数据转换为适于操作系统的语言环境设置的正确编码。

如果操作不使用 UTF-8，那么为了得到最佳结果并提高代理程序的性能，请使用此设置。

enable_agent_dll

此设置启用 DLL 进程跟踪（一种调试工具）。

env_recursion_limit number-of-recursions

设置变量替换递归限制以用于预解析。如果未设置，限值为 32。

extensions

此设置指定函数外部库的路径。可以将函数用作步骤中的点命令。如果未指定此设置，将不会装入外部库。

在进行解析时，步骤命令中的首个标记将作为函数名。第二个标记是字符串，而第三个标记是整数超时值（以秒为单位）。

需求：操作系统中支持动态装入程序。例如，在 UNIX 或 Linux 中，您需要 /usr/include/dlfcn.h。在内部使用这些缺省值。

- UNIX 或 Linux: /usr/local/bin/bfextensions.so
- Windows: C:\program files\ibm\build forge\agent\bfextensions.dll

getaddrinfo_using_addrconfig

此设置仅用于将代理程序作为独立的服务在 Unix 或 Linux 操作系统上运行（bfagent -s）。调用 getaddrinfo() 来选择侦听接口时，此设置可使代理程序使用 AI_ADDRCONFIG。缺省情况下，不使用 AI_ADDRCONFIG。

如果使用此设置，代理程序将忽略未正确配置地址的接口。它仅侦听地址配置正确的接口。

gsk_ssl_key_location [<kdb_path> | <SAF_specification>]

指定 kdb 文件的完整路径或 SAF 密钥环规范。

gsk_ssl_kdb_password <password>

kdb 文件的密码。可以是纯文本或加密文本。如果使用 SAF 密钥环，请使用 NULL。使用 bfagent -e <plaintext>，从纯文本创建加密的密码。

gsk_ssl_protocol <protocol>

要使用的协议，是 ALL（缺省值）、SSLV2、SSLV3、TLSV1 或 TLSV1_1 之一。

gsk_ssl_cipher_v2 <seed>

用于系统 SSL V2 (SSLV2) 的密码套件。缺省值为 6321，适用于大多数应用程序。请参阅 System z 文档以了解更多信息。

gsk_ssl_cipher_v3 <seed>

用于系统 SSL V3 (SSLV3) 的密码套件。缺省值为 0906030201，适用于大多数应用程序。请参阅 System z 文档以了解更多信息。

gsk_keyring_label <label>

kdb 文件中的密钥标签。

gsk_password_encrypt [true | false]

用于引用加密密码。如果设置为 true，请使用 bfagent -e <plaintext> 创建加密值并设置 gsk_ssl_kdb_password。缺省情况下，设置为 false。

gsk_ssl_client_authentication [true | false]

指定是否验证客户机证书。缺省是 false

lang *lang-code*

仅当管理控制台未提供有效语言时使用此设置。

此设置指定代理程序用于写入消息和命令输出的语言。通常，这不是显式设置的，因为代理程序使用管理控制台指定的语言。但是，如果计算机中没有所需的语言环境，那么设置语言可能很有用。如果管理控制台未能以某种语言进行通信，或者以无效语言进行通信，那么此设置作为一种备份也是有用的。

内部缺省值为 **en**，就象以下列方式对其进行显式设置一样：

```
lang en
```

leave_tmp_file

此设置仅用于故障诊断。

此设置使得用于存放步骤命令的临时文件得以保留，而不是在命令执行后就删除。故障诊断时，该文件可与出现在管理控制台中的步骤进行比较。

注： 请不要对典型操作使用此设置。

locale **locale-code.charset-code**

此设置仅用于 UNIX 和 Linux 操作系统。Windows 以不同的方式处理语言环境。

此设置指定本地化应用程序使用的语言和多字节字符集。此设置通过设置代理程序上下文的 **LANG** 环境变量起作用。

要将代理程序设置为将命令输出视为美国英语 UTF-8，请使用操作系统的 UTF-8 语言环境。例如，对于 Linux 使用以下表示法。

```
locale en_US.UTF-8
```

要为操作系统确定 UTF-8 语言环境的正确表示法，请运行 **locale -a** 命令。

如果未指定此设置，代理程序将使用操作系统的语言环境。此设置作为方便之用。如果操作系统的缺省语言环境不是代理程序要使用的语言环境，那么此设置将特别有用。如果更改系统语言环境以符合代理程序要求不太实际，那么此设置将特别有用。

magic_login **user:encoded-password**

代理程序通常使用管理特权（例如 **root** 或 **admin**）登录至操作系统。**magic_login** 设置是对标准系统认证的替代。通过此设置，系统可以使用单一用户名和密码来认证登录。

如果代理程序作为 **root** 或 **admin** 用户运行，那么将忽略此设置并尝试标准认证。

代理程序使用启动代理程序的用户（而非用于登录的用户名）的许可权来运行所有命令。

此设置仅用于以下情形：

- 无法使用管理特权运行代理程序时。例如，将此设置用于无法处理 PAM 的 UNIX 系统。
- 由于安全策略而不允许使用管理特权运行代理程序时。

要配置代理程序的登录，请执行以下操作：

1. 创建使用用户名和密码的服务器认证。在管理控制台中，单击**服务器 > 服务器认证**。

2. 对于此示例，用户名为 `build`，密码为 `MySecretPassword`。
3. 创建使用代理程序的服务器。在**认证**字段中将服务器认证与此服务器相关联。
4. 为代理程序生成经编码的密码。在代理程序的安装目录中，以选择的密码运行 **bfagent -P**。

将返回以 SMD5 散列编码的密码，如下所示：

```
bfagent -P "MySecretPassword"  
eca0b7f2f4fbf110f7df570c70df844e1658744a4871934a
```

5. 在 `BFAgent.conf` 中，设置 `magic_login` 以使用期望的用户名和经编码的密码。
6. 启动代理程序。
7. 测试服务器连接。在**服务器**中选择服务器，然后单击**测试服务器**。

map drive-and-user-spec[; ...]

此设置指定映射的驱动器。有些系统可能需要驱动器映射。例如，由于 `shell` 是从共享驱动器运行的，因此可能需要驱动器映射。在指定由管理控制台中的环境变量 `_MAP` 所指定的映射之前，将执行在代理程序上指定的映射。此示例说明了两个驱动器映射：

```
map X:=//host1/share;Z:=//host2/share(username,password)
```

map_drive_is_failure

如果已指定此设置，那么在步骤执行前指定了未映射的驱动器将导致该步骤失败。如果未指定此设置，步骤将忽略驱动器故障并尝试执行步骤。在此情况下，请确保故障生成有意义的错误消息。

no_preparse_command

此设置禁用变量扩展解析（代理程序通常在将命令传递给 `shell` 之前对命令执行此解析）。另请参阅环境变量 `_NO_PREPARSE_COMMAND`，它用于单个项目或步骤。

no_pty

此设置仅用于在 `UNIX` 或 `Linux` 系统上运行的代理程序。

此设置可用于帮助防止系统 `shell` 在与代理程序的伪终端交互时被锁定。此设置通常用于 `HP-UX` 和 `z/OS`。您还可以使用其他两种方法以帮助防止此种锁定：

- 使用备选 `shell`
- 使用 `nologonshell` 设置

no_pty 设置禁用伪终端分配。

注：使用 **no_pty** 会影响某些命令。例如，**ls** 命令以单列而不是以三列返回输出。如果使用此设置，请在将作业部署到生产环境中之前进行彻底的测试。

nologonshell

此设置仅用于在 `UNIX` 或 `Linux` 上运行的代理程序。

此设置使代理程序运行的 `shell` 成为普通 `shell`，而不是登录 `shell`。此设置通常在以下情况下很有用：

- 登录 `shell` 提供详细的输出

- 登录 shell 以意外方式更改环境设置
- 登录 shell 尝试与用户以交互方式通信

设置后，将使用标准方法请求 shell 应为普通 shell 而不是登录 shell。这可能并非在所有平台上都有效，在这些情况下，也许可以使用 shellflag 设置将标志传递至 shell 以修改其行为。

代理程序不需要这些行为，因为它以用户身份运行，而不是以交互用户身份运行。

注：Mac OS X 10.5 系统使用 /bin/bash，它不会对 nologonshell 作出响应。使用 shellflag -l。

注：z/OS 操作系统始终将 /etc/profile 脚本用于登录 shell 和非登录 shell。如果 shell 行为不适用于代理程序，那么您可能需要更改脚本的内容或使用其他 shell。

另请参阅 **shellflag** 设置。标志可用于更改登录脚本行为。

password_encrypt_module *dll_path;conf_path*

这是在代理程序上启用 SSL 所必需的。它指定了 DLL 和配置文件的路径。

- *dll_path* 是 bfcrypt.dll（通常为 ./bfcrypt.dll）的路径。
- *conf_path* 是 bfpwccrypt.conf（通常为 ./bfcrypt.conf）的路径。

port *port-number-or-range* [...] **[-s]**

此设置仅用于以独立方式在 UNIX 或 Linux 上运行的代理程序（启动时发出 **-s** 选项）。

此设置指定代理程序用于侦听与管理控制台的连接的端口。

指定代理程序用于侦听与管理控制台的连接的端口。

注：缺省情况下，该端口被设置为 5555。对于 UNIX 或 Linux，此设置在 /etc/services 中。

read_timeout

代理程序在断开连接之前，等待请求的时间（秒）。缺省值为 1800 秒（30 分钟）。将值设置为 0 则禁用超时。

此伪指令有助于在未收到合法的引擎请求时，防止客户机连接联系人将端口保持打开状态。某些网络端口扫描软件就是这样工作的。

请勿为该伪指令设置太小的值。正常引擎行为在请求之间可能会有几分钟的间隔。

shell *shell_name* [*options*]

此设置指定缺省 shell 程序。内部缺省值如下：

- Windows: shell cmd.exe /q /c "%s", 除非使用了以下设置：
 - 如果使用 cygwin 设置，缺省为 shell C:\cygwin\bin\bash.exe --login -c "%s"
 - 如果不使用 cygwin 设置，缺省为 shell cmd.exe /u /q /c "%s"
- UNIX 或 Linux: 为用户帐户设置的 shell，或 /bin/sh（如果无法确定用户的 shell）。请注意，虽然您无法指定此设置中的参数，但是可使用 shellflag 设置传入这些参数。代理程序会通过插入连字号，自动将缺省值强制设置为

登录 shell。例如，将 `/bin/ksh` 设置为 `-ksh`。如果显式设置了 `shell`，那么会隐式设置 `nologonshell`。请参阅 `nologonshell`。

- *System i*: 将 `shell` 值设置为 `/bin/sh`

可以在步骤内覆盖此设置。用包含 `#!` 的行开始的步骤会覆盖此 `shell` 设置，而 `nologonshell` 设置用于运行步骤命令。

shell_compatible_undef_vars

此设置将未定义的变量强行表示为空字符串。如果未设置，将表示为变量的名称，格式为 `$VAR`、`${VAR}` 或 `%VAR%`，而 `$(VAR)` 用于表示空字符串。

shellarg

此设置仅用于在 UNIX 或 Linux 上运行的代理程序。

如果命令出现乱码，请使用此设置。Red Hat Linux Enterprise 上的某些 shell 需要此设置。

此设置将更改命令脚本传给 shell 的方式。通常，通过标准输入传递脚本：

```
/bin/sh < /tmp/bfshellscript.sh
```

此设置会将脚本作为参数来传递，从而运行脚本：

```
/bin/sh /tmp/bfshellscript.sh
```

shellflag flag

此设置仅用于在 UNIX 或 Linux 上运行的代理程序。

此设置在 shell 正在运行时添加标志。只能指定一个标志。它通常用于禁用 `rc` 脚本处理以减少输出或不需要的处理。示例：

- `csh` 和派生：使用 `shellflag -f` 来禁用 `rc` 脚本处理。
- `bash`：使用 `shellflag --noprofile` 来禁用概要脚本处理。

ssl_ca_location path

指定包含认证中心的密钥库文件。如果代理程序作为服务运行，请使用绝对路径。

ssl_cert_location path

指定包含专用证书的密钥库。如果代理程序作为服务运行，请使用绝对路径。

ssl_client_authentication [true | false]

如果设置为 `true`，那么连接到代理程序时要求进行客户机认证。如果为 `true`，那么必须将 Build Forge 引擎的证书添加到代理程序的认证中心密钥库中。

ssl_cipher_group [grouplist | ALL]

指定要使用的单独密码组。可以设置为 `ALL`。

ssl_cipher_override cyphers

覆盖密码组。指定要使用的密码。

ssl_key_location path

指定包含密钥的密钥库文件。如果代理程序作为服务运行，请使用绝对路径。

ssl_key_password password

密钥的密码。缺省情况下，此属性以明文存储。您可以将代理程序配置为使用其自己的密钥或 Build Forge 服务器的密钥加密此密码。

ssl_protocol *protocol*

要使用的 SSL 握手协议: SSL、SSLv2、SSLv3、SSL_TLS、TLS 或 TLSv1 中的一个。该协议必须与 Build Forge 服务器使用的协议匹配。

update_path *path*

此设置确定 Build Forge 代理程序可执行文件的完整路径。此设置是在安装期间自动建立的。该目录是操作系统的缺省目录或您指定的安装目录。

注: Windows 代理程序上忽略此设置。更新路径从注册表键中获取。这些键是在代理程序安装期间设置的。

win_reexec_after_auth

如果需要使用 Build Forge 服务器认证凭证在网络共享文件系统中运行代理程序命令, 请添加此设置。例如, 要在 ClearCase 动态视图中修改文件, 代理程序必须在网络共享文件系统上访问 ClearCase 文件。

Build Forge 代理程序最初将以 Windows 系统帐户凭证启动。要运行命令, 代理程序稍后将通过 Build Forge 服务器认证凭证向 Windows 进行认证。

如果没有此设置, 网络共享将仅识别初始 Windows 系统帐户凭证, 并忽略访问并写入网络共享文件系统上的文件所需的后续服务器认证凭证。

win_reexec_after_auth 将在使用服务器认证凭证再次向 Windows 认证后启动新的进程, 并强制共享文件系统识别更改后的凭证。

如果使用 win_reexec_after_auth 设置, 那么代理程序将作为服务运行, 并且不区别访问网络共享文件的命令和不访问网络共享文件的命令, 因此您可能要注意性能影响。

xstream_allow_ssl_mismatch

如果在使用 OpenSSL 编译的代理程序和不是使用 OpenSSL 编译的代理程序之间需要文件传输, 那么此设置是必需的。缺省情况下, 使用 OpenSSL 编译的代理程序需要由 AES_CBC 加密的文件传输。除非使用此设置, 否则它们会拒绝使用 PLAIN 或 PRNG 编码请求的任何文件传输。

xstream_bind *ip_address*

指定仅用于直接文件传输的 IP 地址。接收文件的代理程序必须可访问此地址。缺省情况下, 代理程序侦听所有网络接口。另请参阅 **bind**。

xstream_conn_timeout *seconds*

代理程序等待连接的时间 (秒)。引擎必须将连接请求转发给接收方代理程序, 而接收方代理程序在这段时间内必须与发送方代理程序建立连接。缺省情况下, 设置为 20 秒。

xstream_listen_range *port-range*

代理程序用于侦听连接的端口范围。如果连接主机之间有防火墙, 那么此设置就很有用。防火墙管理员可以配置防火墙, 以允许某些端口 (例如 22880-22889) 可用于连接。缺省端口范围是 16384-32767。但是, 如果使用 xstream_bind, 但未使用 xstream_listen_randomize, 那么代理程序不会指定范围, 而是由操作系统确定使用哪些端口。

xstream_listen_randomize

在 xstream_port_range 内随机选择一个端口。如果未指定, 那么代理程序开始检查最小端口号。强烈建议将此设置作为一种安全措施。

xstream_recv_timeout *seconds*

等待文件传输的时间。如果在文件传输的任何时候，在该时间段过后接收方代理程序未能从发送方代理程序获得数据，那么传输失败，连接会关闭。缺省值为 20 秒。

xstream_send_timeout *seconds*

等待文件传输的时间。如果在文件传输的任何时候，在该时间段过后发送方代理程序未能向接收方代理程序写数据，那么传输失败，连接会关闭。缺省值为 20 秒。

对代理程序进行故障诊断

本部分描述可用于对无法正常运行的代理程序进行故障诊断的过程。请按顺序执行这些过程。如果无法通过使用这些过程使代理程序运行，请与支持人员联系。

测试主机名解析

验证是否可以从管理控制台主机访问代理程序主机。请使用管理控制台主机的 `ping` 实用程序来测试代理程序主机：

```
ping hostname
```

此示例会话在安装了管理控制台和代理程序的 Windows 上运行。

```
C:\> ping localhost
```

```
Pinging somehost.city.company.com [127.0.0.1] with 32 bytes of data:
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

与以下消息类似的消息指示出现了问题：

```
Unknown host
```

问题出在管理控制台主机的网络配置。请联系网络管理员。

测试连接

可以使用 `telnet` 或管理控制台的测试来对代理程序的连接进行测试。

要从命令行测试连接：

1. 使用 `telnet` 命令连接到代理程序。如果已登录运行代理程序的主机，那么可以使用 `localhost` 作为主机名。

```
telnet hostname 5555
```

此响应指示连接成功：

```
200 HELLO - BuildForge Agent v7.0.1.buildnumber
```

2. 使用登录凭证，通过发出以下命令检查认证：

```
telnet localhost 5555
username user name
password password
cmd ping
go
```

以下消息指示成功:

```
AUTH: set user account to <user name>
```

如果以上测试成功而作业失败, 且服务器的测试显示用户认证错误, 请检查可插拔认证模块 (PAM) 配置。如果看到如下消息, 请继续下一步骤。

```
AUTH: unable to set user account to <user name>: unknown account (1)
```

3. 输入以下命令:

```
cmd ping go
```

以下是 telnet 会话的典型输出。尤其要在输出末尾查找 RESULT 0, 这是成功的指示。该测试针对在 Windows 上运行的代理程序。

```
300 DATA s 67
AUTH: Running as: [SYSTEM] in domain [NT AUTHORITY] SID Type [User]
300 DATA s 52
AUTH: Running with Privilege: [Lock pages in memory]
300 DATA s 66
AUTH: Running with Privilege: [Adjust memory quotas for a process]
300 DATA s 63
AUTH: Running with Privilege: [Create permanent shared objects]
300 DATA s 46
AUTH: Running with Privilege: [Debug programs]
300 DATA s 56
AUTH: Running with Privilege: [Bypass traverse checking]
300 DATA s 61
AUTH: Running with Privilege: [Back up files and directories]
300 DATA s 54
AUTH: Running with Privilege: [Change the system time]
300 DATA s 68
AUTH: Running with Privilege: [Remove computer from docking station]
300 DATA s 73
AUTH: Running with Privilege: [Impersonate a client after authentication]
300 HEARTBEAT 1
300 DATA s 16
PLAT: Windows XP
250 RESULT 0
PING: internal loopback test complete
260 EOR
```

要从管理控制台测试连接:

注: 仅当已设置了使用该代理程序的服务器时, 才应使用此方法。

1. 转到**服务器**。
2. 在服务器列表中, 单击要测试的服务器。
3. 单击**测试连接**。

在测试后, 结果会显示在**测试结果**选项卡中。

与以下输出类似的消息指示出现了问题。

```
Could not open a connection to host on port 5555
```

这是由于管理控制台无法与主机连接, 或者代理程序未运行。

对 Windows 上的代理程序进行故障诊断

要对 Windows 上的代理程序进行故障诊断，请执行以下操作：

1. 检查已安装的可执行文件。验证这些文件是否在代理程序安装目录中：
 - `bfagent.exe`
 - `bfdispatch.exe`

注：继续之前，请确定是否有定制配置。如果有定制配置，请将 `BFAgent.conf` 保存在安装目录之外，并在安装后将其恢复。

2. 重新安装代理程序。通过重新安装代理程序，可以解决大多数 Windows 代理程序问题。这是因为该操作会更新可执行文件并恢复注册表键。

对 UNIX、Linux 或 MacOS 上的代理程序进行故障诊断

要对 UNIX、Linux 或 MacOS 上的代理程序进行故障诊断，请尝试以下过程：

- 从 shell 运行 **bfagent**。正确的响应类似于此消息：

```
200 HELLO - Build Forge Agent v7.0.1.122
```

如果接收到类似于示例的消息，并存在共享库问题，那么将接收到关于这些问题的消息。您可以通过正确设置路径来解决大多数共享库问题。

- 检查代理程序正在进行侦听。请使用以下命令（假定端口为缺省端口 5555）：

```
telnet localhost 5555
```

200 HELLO 响应指示代理程序正在侦听。如果未得到此响应，请检查您的系统网络配置。验证 **inetd** 配置是否正确，或向 Linux 或 UNIX 系统管理员核实。

- 检查认证。使用您的登录凭证，发出以下命令：

```
telnet localhost 5555
username <user name>
password <password>
cmd ping
go
```

与以下消息类似的一条消息指示正在进行正确认证：

```
AUTH: set user account to <user name>
```

如果先前测试成功而构建失败，且服务器的测试显示用户认证错误，请检查可插拔认证模块 (PAM) 配置。如果看到类似于以下消息的消息，请继续到下一个过程。

```
AUTH: unable to set user account to user name: unknown account (1)
```

- 检查 PAM 配置。PAM 配置的问题是 AIX 平台上的常见问题。会按以下两种方式中的一种来配置 PAM（取决于您的操作系统）：通过 `pam.conf` 文件中的行或通过 `pam.d` 目录中的文件。

提示：以下过程不适用于 Solaris 10：删除所有指定 `pam_dial_auth` 模块的行（例如 `pam_dial_auth.so.1`）。如果包含了该模块，那么代理程序认证无法工作。

1. 验证 `/etc/pam.conf` 是否存在。如果不存在，请转至此主题中稍后针对 `pam.d` 的指示信息。如果该文件存在，请继续下一步。
2. 在文件中，为 `bfagent` 创建条目。
3. 复制其他应用程序（例如，`sshd` 或 `login`）的行并将 `[application]` 字段替换为 `bfagent`。

[application] [when] [mode] [module]

字段如下:

- [application] 是需要认证用户的应用程序的名称
- [when] 是认证请求的类型
- [mode] 是认证请求的方式
- [module] 是要调用的认证模块。以下示例显示从 login 复制到 bfagent 的条目。请注意, 模块名称可能因系统而异。

bfagent auth requisite	pam_authtok_get.so.1
bfagent auth required	pam_dhkeys.so.1
bfagent auth required	pam_unix_cred.so.1
bfagent auth required	pam_unix_auth.so.1

4. 设置 PAM 条目后, 按步骤 3 中所述尝试重新登录。

5. 有关更多信息, 请参阅位于以下位置的 PAM 文档: <http://www.sun.com/software/solaris/pam>。

- 要对在 pam.d 中配置的 PAM 进行故障诊断:

1. 查找 /etc/pam.d 目录, 并注意它包含若干部分, 每个部分针对一个应用程序命名。在每个文件中, 每行已按下列方式进行格式化:

[when] [mode] [module]

2. 复制其他应用程序 (如 sshd 或 login) 中的文件, 并将其重命名为 bfagent。

3. 设置 PAM 条目后, 按步骤 3 中所述尝试重新登录。

4. 有关更多信息, 请参阅位于以下位置的 PAM 文档: <http://www.sun.com/software/solaris/pam/>。

第 11 章 安装后任务

该部分描述了成功安装后要执行的任务。

将讨论以下主题:

- 启动和停止引擎
- 设置用户
- 验证安装
- 对常见问题进行故障诊断

启动和停止引擎

以下各节描述了如何启动和停止引擎。

在 Windows 上启动和停止引擎

在 Windows 上:

- 在开始 > 程序 > **IBM Rational Build Forge 管理控制台**, 选择以下一项:
 - 启动引擎服务
 - 停止引擎服务

控制面板: 也可使用**管理工具 > 服务控制面板**来启动或停止 **IBM Rational Build Forge 管理控制台**服务。

在前台运行: 如果在运行引擎时遇到任何问题, 请在前台运行引擎以便查看它生成的状态和错误消息: 开始 > 程序 > **IBM Rational Build Forge 管理控制台 > 启动引擎 (前台)**。当管理控制台运行时, 日志输出将显示在控制台窗口中。要停止以此方式运行引擎, 请在控制台窗口输入 Ctrl-C。

在 UNIX 或 Linux 系统上启动和停止引擎

如果已安装 rc 文件, 请使用以下命令启动和停止该产品:

```
$ /opt/buildforge/rc/buildforge start  
$ /opt/buildforge/rc/buildforge stop
```

如果未安装 rc 文件, 请使用以下命令启动该产品:

```
$ /<bfinstall>/Platform/buildforge &
```

通过确定其进程标识然后发出 kill 命令来停止它:

```
$ ps aux | grep buildforge  
$ kill ${<PID>}
```

设置用户

本部分描述如何在 Build Forge® 系统中设置用户帐户。

将讨论以下主题:

- Root 用户
- 添加用户帐户
- 报告的只读用户

Root 用户

root 用户, 是登录名为“root”的用户, 该用户具有特殊特征:

- **安装时创建:** root 用户是安装程序创建的唯一缺省用户。缺省密码为“root”。(安装之后立即更改密码)。
- **不需要许可证:** root 用户不使用用户许可证。无论已登录的用户有多少, 您总是可以作为 root 用户登录。(当某人作为 root 用户登录时, 将会注销已作为 root 用户登录的其他用户。)
- **系统时区:** root 用户的时区是管理控制台的缺省时区。其他用户(系统中用户和 LDAP 用户)的时区缺省情况下取自 root 用户的时区。用户可在登录后设置其自己的时区。系统中报告的所有时间和日志都是以用户的时区表示的。
- **所有许可权:** root 用户具有所有可用的许可权, 并可以编辑其他用户的属性。您不能删除 root 用户的任何访问权限。虽然 root 用户不是任何访问组的成员, 但是 root 用户仍然可以查看、编辑或使用系统中的任何数据对象。
- **优先级:** root 用户始终是具有优先权的用户。
- **注销当前用户:** root 用户可以通过单击**注销用户**来注销用户。(单击**管理 > 用户**, 然后单击用户的名称。)

创建和编辑用户

您可以通过使用**管理 > 用户**来创建用户并对用户指定属性。还可以将系统连接到 LDAP/Active Directory 数据库以获取用户信息。您是通过将用户分配到组来管理用户安全许可权的, 因此必须创建一些用户来测试安全功能。

单击**管理 > 用户**以显示当前用户列表。用户面板显示在列表的后面。系统将显示每个用户的名称、登录、电子邮件、限制、活动(自上次用户活动以来的耗用时间)以及时区。

- 要编辑用户, 请单击用户的名称, 并在用户面板中编辑属性, 然后单击**保存**。
- 要创建新用户, 如果未选择任何用户, 请在用户面板中指定属性。如果选择了用户, 请单击**添加用户**来清除面板。完成对用户信息的编辑之后, 单击**保存**。
- 要注销用户, 单击该用户的名称, 然后单击**注销用户**。
- 要使得固定许可证席位可用, 请首先作为 root 用户登录。单击用户的名称, 然后单击**清除席位**。控制台将从标识列表中除去占用固定许可证名额的用户。如果用户已经登录, 控制台还将注销该用户。对于固定许可证, 控制台将统计已登录的用户的数量。达到限制时, 新用户将无法获取许可证。必须删除或清除现有用户, 这样另一用户才能使用固定许可证。清除席位不会从控制台中删除用户。如果用户重新登录, 固定许可证计数将增加。如果对浮动许可证用户使用**清除席位**, 那么该操作与单击**注销用户**效果相同。

- 要删除用户帐户，请单击用户名，然后单击**删除**。

如果**删除**已禁用，那么说明该用户帐户拥有调度的作业。要删除具有调度作业的用户帐户，请首先删除这些调度作业。

用户记录为用户可与系统交互的方式设置缺省属性，并控制用户的登录名、密码和密码到期时间。可以通过管理控制台向系统提供用户记录的数据，也可以从 LDAP/Active Directory 数据库中派生出该数据。

注：如果所编辑用户的记录是从 LDAP 数据库获取的，那么“用户”页面上的许多字段处于禁用状态。在源数据库中更改这些属性。

要添加新用户，请单击**添加用户**，编辑面板，然后单击**更新**。

显示用户记录时，有三个选项卡可用：

- **详细信息：**显示可以编辑的用户属性。该主题的后部分描述了可用属性。
- **当前组：**显示用户所属的访问组（直接属于某个组或通过属于其他组的组）。
- **更改组：**显示用户直接隶属的组，并允许您将用户添加至组或从组中除去用户。

对于每个用户，可以在**详细信息**选项卡中设置以下属性：

名称 指定用户的显示名称和标签。

电子邮件

指定系统向该用户发送电子邮件通知时使用的目标电子邮件地址。

注：仅向显式选择来通知的用户发送电子邮件。

用户名 指定用户输入以登录至管理控制台的名称。

密码 输入用户提供以登录至管理控制台的密码。对于已登录的用户，不会显示该字段。使用该字段可输入新密码或更改密码。应在**验证**字段输入相同的密码。

限制 指定用户可以在一天中启动的最大作业数。达到限制时，系统显示消息，指示已经超出用户的运行限额。

时区 指定用户的时区。系统使用 root 用户的时区作为所有公布时间的缺省时区。

缺省情况下，为系统中用户和 LDAP 用户指定的时区与 root 用户的时区相同。用户可以编辑为其指定的时区。

注：从先前的版本升级至 Build Forge 7.1 时，将需要手动复位 root 用户的时区。

验证密码

再次输入密码以验证是否正确输入密码。

优先级登录

指定该用户是否为优先级用户。优先级用户可以始终登录至系统；如果没有更多可用的用户许可证，系统将注销具有最早会话的用户，以便优先级用户可以登录。root 用户始终是优先级用户。

日期格式

设置用户的首选日期格式。

语言 设置用户的首选语言。

密码到期

指定用户的密码将到期。如果选中该选项，那么在经过一定的天数之后用户的密码会到期，该天数在**密码到期天数**系统设置中指定。

使用屏幕朗读者

启用该界面以支持屏幕朗读者功能，例如动态突出显示和聚焦。

日历起始日

选择调度日历首先显示星期几。缺省值为星期日。

验证安装

本部分描述了如何测试已安装和配置的 Build Forge® 系统。

将讨论以下主题:

- 配置服务器
- 创建测试项目
- 运行项目

服务器认证

服务器认证用于将登录凭证关联到服务器。通过管理一组服务器认证，可以对许多服务器使用相同的凭证以及全局更新凭证。

服务器认证将登录名和密码存储为单一的已命名对象，您可以将该对象与一个或多个服务器相关联。“服务器认证”页面可用于创建和编辑服务器认证。

配置服务器

为了充分利用系统的动态服务器选择，必须按特定顺序设置多个数据对象。本主题概述了使用服务器的最低要求。

1. 创建服务器认证。

服务器认证为服务器提供登录名和密码。您可以将一个服务器认证应用于多个服务器，这样就不必为每个服务器指定唯一的登录凭证；并且当您更改登录凭证时，可以为一组系统进行更改。

2. 为服务器组创建收集器。

收集器能够从服务器收集属性，也能向服务器分配属性。可以使用没有收集器的服务器。您必须根据 **BF_NAME** 之类的缺省属性来选择服务器。

3. 创建选择器。

请考虑创建以下类型的选择器:

- 基于名称: 为每个服务器创建一个选择器，它会根据服务器的主机名来选择服务器。这使您接下来能够按名称选择服务器。
- 基于操作系统: 为环境中的每种操作系统创建一个选择器，这样项目可根据操作系统选择服务器。
- 基于容量: 可以根据可用 RAM 或硬盘空间更具体地选择服务器。

4. 在要用作 Build Forge 服务器的每台计算机上安装一个代理程序。

5. 在要与 Build Forge 一起使用的每台计算机的管理控制台中创建一个服务器。
6. 测试服务器。

单击**测试连接**来测试与服务器的连接。查看它们的清单以确保它们具有期望的属性。

创建测试项目

要验证 Build Forge® 系统是否正常运行，请按照第 10 页的『创建 hello world 项目』中所述，创建并运行简单项目。


运行项目

有多种不同的方式可以启动项目。

开始之前

该任务假定您已经创建了选择器、服务器和项目。

过程

- 查看项目列表时，单击任意项目前面的**快速启动**  图标以立即启动项目。如果项目没有步骤，或者它具有操作为 **Must Change On Project** 的任何环境变量，那么您不能使用此方法。通过该方式运行项目将使用它的选择器、类、标记和环境变量的缺省值。
- 查看项目的步骤时，单击**启动项目**。该方法将显示项目的“启动项目”页面，在其中您可以更改项目参数、环境变量值和选择要从运行中排除的步骤：
 - 选择项目参数的新值。
 - 编辑项目标记变量值。
 - 编辑项目环境变量值。如果希望将更改保存为这些变量的新缺省值，请单击**保存环境**复选框。
 - 选择“作业步骤”选项卡来显示项目步骤的列表。您可以选择个别的步骤，仅将它们从该运行中排除。

进行选择后，单击**执行**启动项目。

- 选择**作业 > 启动**，然后单击项目名称。使用**启动项目**时，该方法将显示“启动项目”页面。

作业详细信息

作业步骤

项目参数

项目环境

保存环境 ☐

运行链接: ☐

快照: 基础快照

选择器: Selector1

类: Production

标记格式: BUILD_\$B

标记示例: BUILD_1

项目标记 ☐ 可编辑的标记

B1

结果

项目运行时，查看作业 > 正在运行页面以检查项目状态。

要查看作业结果，请选择作业 > 已完成以显示已完成的作业。单击“标记名称”可访问用于查看作业结果的选项。

日志的位置和设置

Rational Build Forge 由多个组件构成。每个组件保留其自己的日志。

Build Forge
<bfinstall>/db.log

Apache
<bfinstall>/Apache/logs

Apache Tomcat
<bfinstall>/Apache/tomcat/logs。Tomcat 是一种应用程序服务器，用于运行服务层 (rbf-services.war) 和帮助 (BuildForgeHelp.war)。

PHP <bfinstall>/db.log。PHP 用于运行用户界面。

日志循环

在一些情况下，组件将写大量数据。Tomcat 生成的 catalina 日志文件就属于此类情况。根据设计，不会循环使用 Tomcat out 日志。在罕见情况下，out 日志的增长可能会超出预期。

- 有两种方法可以解决此问题:
- 在服务器上下文中使用 swallowOutput 属性。这样输出会重定向到循环使用的 Web 应用程序日志。请参阅 Apache Tomcat Configuration Reference，以了解详细信息。
 - 修改 Catalina 启动脚本。具体步骤取决于您的首选项，但以下步骤对 Catalina 输出进行管道处理，并使用 cronolog 设施进行日志循环。修改 <bfinstall>/Apache/tomcat/bin/catalina.sh 以包含以下命令或类似命令：
start >> catalina.out 2>&1 & --replaced with
start |cronolog /tomcat/logs/%Y-%M-%D.catalina.out >> /dev/null 2>&1 &

对常见问题进行故障诊断

升级 Solaris 上的代理程序需要运行 **pkgrm** 命令。

运行 **pkgadd** 命令前，使用 **pkgrm BFAgent** 命令除去现有的 Solaris BuildForge 代理程序。

7.0 通知模板的 URL 在更高版本中可能无效

单击通知电子邮件中的 URL 链接时，通知模板 URL 将打开 Build Forge 作业报告。

在 V7.0.1 和 V7.1 中，通知模板中的 URL 已更改；因此，当从 7.0 升级至更高版本时，URL 可能无效。

如果遇到链接错误，请完成这些步骤以手动编辑通知模板：

- 1. 选择项目 > 模板。
- 2. 单击通知模板名称以在“详细信息”选项卡上显示其属性。
- 3. 在“正文”字段中，找到模板的 URL。URL 应该类似于以下示例中的 URL：

```
http://${CONSOLEHOST}:${CONSOLEPORT}/fullcontrol/index.php?mod=projectruns&action=edit&bfid=${PID}&bfid=${BID}&bfid=${UID}
```

- 4. 将以下 URL 元素替换为相应的 7.0.1 及更高版本的 URL 元素：

URL 元素	7.0.1（及更高版本）URL 元素
projectruns	jobs
&	&
action=edit	action=build.view
&bfid=\${PID} &bfid=\${BID} &bfid=\${UID}	&bf_id=\${BID}

产品无响应

如果 Rational Build Forge 无响应，请查看 *<bfinstall>/Apache/tomcat/logs/*（UNIX 或 Linux）或 *<bfinstall>\Apache\tomcat\logs*（Windows）中的日志以获取类似如下的消息：

```
Services: 20075: CRRBF1381I: Established connection to Build Forge Services.
DBD::DB2::st execute failed: [IBM][CLI Driver] SQL30081N  A communication error has been detected.
  Communication protocol being used: "TCP/IP".
  Communication API being used: "SOCKETS".
  Location where the error was detected: "XXX.XXX.XXX.XXX".
  Communication function detecting the error: "recv".
  Protocol specific error code(s): "131", "*", "0".  SQLSTATE=08001
  August 17, 2010 7:18:14 AM EDT
Database: 20075: CRRBFEEEE: DBD::DB2::st execute failed:
[IBM][CLI Driver] SQL30081N  A communication error has been detected.
  Communication protocol being used: "TCP/IP".
  Communication API being used: "SOCKETS".
  Location where the error was detected: "XXX.XXX.XXX.XXX".
  Communication function detecting the error: "recv".
  Protocol specific error code(s): "131", "*", "0".  SQLSTATE=08001

Database: 20075: CRRBF0551I: StackTrace from Process id [20075] called from [
  BuildForge::DB::db2(./PerlApp/BuildForge/DB/db2.pm:65)
  BuildForge::Utilities::SysParams(./PerlApp/BuildForge/Utilities/SysParams.pm:57)
  main(.buildforge.pl:305)
  main(.buildforge.pl:213)
```

```

]
Database: 20075: CRRBF0556I: Trying to call [execute] on a non-existent database handle
DBD::DB2::st fetchrow_hashref failed: no statement executing at /PerlApp/BuildForge/DB/db2.pm
line 78, <$sock> line 1054.
20075: CRRBF0555E: Problem performing Database Operation [fetchrow_hashref] : DBD::DB2::st
fetchrow_hashref failed: no statement executing at /PerlApp/BuildForge/DB/db2.pm
line 78, <$sock> line 1054.

Database: 20075: CRRBFEEEE: DBD::DB2::st fetchrow_hashref failed: no statement executing
at /PerlApp/BuildForge/DB/db2.pm line 78, <$sock> line 1054.

Database: 20075: CRRBF0551I: StackTrace from Process id [20075] called from [
BuildForge::Utilities::SysParams(/PerlApp/BuildForge/Utilities/SysParams.pm:59)
main(.buildforge.pl:305)
main(.buildforge.pl:213)
]
[IBM][CLI Driver] CLI0106E Connection is closed. SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm
line 385, <$sock> line 1054.
Database: 20075: CRRBF0555E: Problem performing Database Operation [prepare] :
query [SELECT * FROM bf_jobcount WHERE bf_engine_id=?]
DBD::DB2::db prepare failed: [IBM][CLI Driver] CLI0106E Connection is closed.
SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm line 385, <$sock> line 1054.

Database: 20075: CRRBFEEEE: DBD::DB2::db prepare failed: [IBM][CLI Driver] CLI0106E
Connection is closed. SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm
line 385, <$sock> line 1054.

```

其中, XXX.XXX.XXX.XXX 是托管数据库的服务器。

该消息指示 Rational Build Forge 和数据库之间无连接。

要重新建立连接:

1. 如 第 229 页的『启动和停止引擎』中所述, 关闭 Rational Build Forge。
2. 确保数据库正在运行。
3. 如 第 229 页的『启动和停止引擎』中所述, 启动 Rational Build Forge。

第 12 章 从先前版本升级

以下主题描述如何将组件从先前版本升级到当前版本:

- 升级的先决条件。

要点: 如果不遵守先决条件, 可能损坏并禁用系统。

- 升级管理控制台
 - 升级 V7.1.x 控制台
 - 升级 V7.0.2.x 控制台
- 升级代理程序

升级的先决条件

执行升级安装前应先执行以下操作。

- **备份数据库。** 另请参阅第 34 页的『数据库备份 (必需)』
- 如果要升级数据库和数据库客户机的版本, 请首先执行升级。无法更改已安装的系统以使用新的版本。系统使用您在安装程序中指定的数据库和数据库客户机文件。
- 备份任何 Apache 与 tomcat 的定制配置文件。

```
<bfinsastall>/Apache/conf  
<bfinsastall>/Apache/tomcat/conf
```

httpd.conf 和 php.ini 文件通常是针对 Apache 定制的。

- 如果系统配置为使用 HTTPS 和 SSL, 那么请备份您的证书目录。升级不会对其产生影响。
- 如果您的当前系统安装有 Rational Automation Framework for Websphere (RAFW), 那么必须迁移至 Rational Automation Framework (RAF)。升级安装 Build Forge 期间, RAFW 数据置于压缩文件内, 并移至 <bfinstall>/Manager/rafw。请参阅 RAF 文档以获取有关将数据移动至 RAF 的信息。

升级 V7.1 控制台

如果要从任何 V7.1.x 升级到当前版本, 请使用本部分。

开始升级前请查看『升级的先决条件』。

要升级, 请执行更新安装。

注: 在升级 V7.1 控制台时, 还需执行其他任务以支持性能增强。请参阅 jazz.net 上的 release 页面以了解详细信息。

执行更新安装 过程

要执行更新安装:

1. 如果 Build Forge 正在运行，请停止该 Build Forge。然后，如果在 Windows 上运行，请在安装目录中查找名为 `bfengine.pid` 的文件。如果存在该文件，请将其删除。如果存在该文件，Installation Manager 就不能执行更新。如果您是在前台运行管理控制台，那么很可能存在该文件。
2. 启动 IBM Installation Manager。
3. 更新之前，必须将存储库 URL 设置为已更新的存储库。请参阅第 72 页的『指定存储库 URL』以获取指示信息。
4. 在 Installation Manager 中，单击**更新**。
5. 在更新软件包中，选择要查找更新的软件包组，然后单击**下一步**。
6. 选择要安装的更新或修订。从列表选择安装，然后单击**下一步**。
7. 将向您显示可以更新到的版本列表。从列表选择安装，然后单击**下一步**。
8. 选中我接受许可条款复选框，然后单击**下一步**。
9. 将显示“更新软件包”页面。将为您选择您要的功能。单击**下一步**。
10. 配置数据库并单击屏幕底部附近的**测试连接**以测试配置。在测试通过后，单击**下一步**。
11. 在下一个页面上，将显示您在 `buildforge.conf` 文件中编辑的信息。将询问您以下问题：
 - a. 是否想要安装程序进行所需的数据库修改？选择此复选框。它将使安装程序对模式进行所需更改。如果此复选框未选中，必须在安装之后手动运行 `bfschema -u` 以应用模式更改。
 - b. 是否要在升级后启动控制台？选择此复选框以使控制台在安装完成之后自动启动。
12. 单击**下一步**。
13. 此时将列出您选择的要安装的功能。
14. 单击**更新**开始安装。
15. 更新成功执行后，单击**完成**。
16. 关闭 Installation Manager。启动 Build Forge。如果要从 V7.1.1.x 升级，在字符大对象 (CLOB) 迁移完成之前，您将不会从构建中看到步骤日志。在启动 Build Forge 时，迁移便会开始。在迁移完成时，Build Forge 将通知您。

在升级到 V7.1.3.1 后复制适配器

从 7.1.1.x 升级到 7.1.3.1 及更高版本后，复制适配器时出现问题。针对 V7.1.3.1 更新了一些适配器。如果升级到 V7.1.3.1，然后尝试复制在升级之前创建的适配器，那么会报告错误：

CRRBF0789I：必须指定有效的适配器模板标识。

变通方法：在尝试复制适配器之前，执行以下操作：打开此适配器，对其重命名，然后保存。也可以打开此适配器，复制适配器的内容，然后创建一个新适配器。

升级 V7.0.2.x 控制台

如果要将在 V7.0.2.x 系统升级到当前版本，请使用本部分。

要升级：

1. 请查看第 179 页的『升级的先决条件』。
2. 从 V7.0.2.x 升级到 V7.1.1.3。

从 IBM 支持站点下载 V7.1.1.3 软件包，然后遵循 *IBM Rational Build Forge version 7.1.1.3 Installation Guide* 中的指示信息。由于数据库模式更改，从 V7.0.2.x 升级到 V7.1.1.3 要求迁移配置数据（项目）以及选择性地迁移历史数据（来自于作业的日志）。

要点：不会迁移所有项，特别是定制通知模板（项目 > 模板中的项）

3. 在 V7.1.1.3 上执行更新安装以将其更新为当前版本。

注：在升级 V7.1 控制台时，还需执行其他任务以支持性能增强。请参阅 jazz.net 上的 [release](#) 页面以了解详细信息。

升级代理程序

通过在旧代理程序的基础上安装新代理程序来升级代理程序。

可以通过在旧代理程序的基础上安装新代理程序来升级代理程序。请遵循安装代理程序的指示信息。

第 13 章 卸载产品组件

使用 IBM Installation Manager 可卸载您使用 Installation Manager 安装的产品组件。
要卸载 Build Forge 代理程序软件，请使用本部分中描述的操作系统工具和命令。

使用 Installation Manager 卸载产品

使用此场景可卸载随 Installation Manager 一起安装的 Build Forge 产品组件。

注：该过程不会卸载已针对 Build Forge 创建的数据库或者数据库对象。您必须使用针对您数据库的工具单独将其除去。针对 Windows 上安装的数据库，您可以使用**控制面板 > 添加或删除程序**。请咨询数据库文档以获取指示信息。在除去 Build Forge 产品组件之后再除去数据库或数据库对象。

要卸载产品组件，请完成以下步骤：

1. 使用安装产品软件包时所用的同一用户帐户登录至操作系统。

关闭随 Installation Manager 安装的任何正在运行的程序。

2. 如果 Build Forge 引擎和服务正在运行，请如下停止该引擎和服务：

Windows	<ul style="list-style-type: none">• 如果引擎正在前台运行：按住 Ctrl + c 以停止引擎和 Build Forge 服务。 <p>要点：停止引擎和服务的最可靠的方法是使用 Ctrl + c。如果卸载时引擎或任何 Build Forge 服务仍在运行，那么卸载将失败。</p> <ul style="list-style-type: none">• 如果引擎正在后台运行：打开命令窗口并输入以下命令以停止引擎和服务： <pre>net stop bfengine71</pre> <p>这些命令停止 Build Forge 引擎、Apache HTTP Server 和 Apache Tomcat 应用程序服务器。</p>
UNIX/Linux	<ol style="list-style-type: none">1. 在命令窗口中，切换至 Build Forge rc 目录。 <pre>cd /opt/buildforge/rc</pre>2. 停止引擎。 <pre>./buildforge stop</pre> <p>该命令停止 Build Forge 引擎、Apache HTTP server 和 Apache Tomcat 应用程序服务器。</p>

3. 启动 IBM Installation Manager。
4. 在“启动”页面，单击**卸载**。
5. 在“卸载软件包”页面，从“安装软件包”列表中，选择您想卸载的产品软件包，并单击**下一步**。
6. 在“摘要”页面上，查看要卸载的软件包的列表，并单击**卸载**。除去软件包后，将显示“完成”页面。

- 单击**完成**。
- 关闭 Installation Manager 窗口，并退出 Installation Manager。

必须先退出 Installation Manager，然后才能清除 Build Forge 文件、除去 DB2 Express 数据库（可选）或重新安装产品。

如果 Installation Manager 安装失败，请手动卸载该产品

如果在停止 Build Forge 引擎和服务之前启动 IBM Installation Manager，那么卸载将失败。

要在 Installation Manager 失败之后手动卸载产品组件，请执行以下操作：

- 在命令窗口中，切换至 Build Forge 安装目录中的管理器目录，例如：

Windows	C:\Program Files\IBM\Build Forge\Manager
UNIX/Linux	/opt/buildforge/Manager

- 在命令提示符下，输入 `main.exe uninstall main.res`。
- 命令运行完后，请删除 Build Forge 安装目录。
- 使用 Installation Manager 重新安装该产品。

卸载后清除 Build Forge 文件

卸载后需要手动执行一些清除操作。

通过 Installation Manager 卸载产品组件后，手动删除以下目录。如果这些目录存在，那么您不能使用 Installation Manager 重新安装 Build Forge。

目录	缺省位置
Build Forge 安装目录	Windows: C:\Program Files\IBM\Build Forge UNIX/Linux: /opt/buildforge
共享文件目录	Windows: C:\Program Files\IBM\SDPShared UNIX/Linux: /opt/IBM/SDPShared

使用 Installation Manager 重新安装产品并使用现有的 DB2 Express

如果在卸载其他产品组件时未除去该产品，可以重新安装产品并使用现有的 DB2 Express 数据库。在场景中，您可访问现有 Build Forge 项目和作业日志。

启动 Installation Manager 之前，请使用以下步骤中的核对表来获取必需的 DB2 Express 信息。

要点：如果您已将系统配置为使用 HTTPS 和 SSL，那么需要在 `<bfinstall>/keystore` 和 `<bfinstall>/bfclient.conf`（存储证书密码的位置）中备份证书。在 UNIX 和 Linux 系统上，该目录位于 `<bfinstall>/Platform`。

要重新安装并使用现有的 DB2 Express 数据库，请执行以下操作：

- 启动 Installation Manager。

- 2. 在“启动”页面上，单击**安装**。
- 3. 遵循 Installation Manager 向导中的指示信息以重新安装产品组件。
- 4. 在“数据库配置”页面上，作出以下选择：
 - a. 对于安装 *DB2 Express*，请选择**否**。
 - b. 对于是否要在安装时填充此数据库？，请选择**否**。
 - c. 要完成其余字段，请使用下表中的信息：

核对表： **DB2 Express** 配置信息

✓	字段	描述
	数据库主机	所提供的 DB2 Express 数据库将安装在本地主机名（127.0.0.1）上。
	数据库名称	数据库名称为 BUILD（全大写）。
	数据库模式名称	数据库模式的名称为 BUILD（全大写）。
	数据库用户名	您提供来创建 DB2 Express 数据库的数据库用户名。
	数据库密码	您为该数据库用户名提供的密码。
	DB2 客户机库的路径	DB2 Express 的 DB2 客户机库（db2cli.dll）的路径为 C:\Program Files\IBM\SQLLIB\bin。
	JDBC 驱动程序位置	DB2 Express 的 JDBC 驱动程序（db2jcc.jar）的路径为 C:\Program Files\IBM\SQLLIB\java。

- 5. 单击**测试连接**。
- 6. 单击**下一步**以继续并完成安装。

卸载 Windows Build Forge 代理程序

使用 Windows 的“添加或删除程序”工具除去 Build Forge 代理程序软件。

注：如果正在重新安装，可以覆盖相同版本的代理程序软件。

- 1. 在 Windows 中，找到“添加或删除程序”。例如，选择**开始 → 所有程序 → 控制面板 → 添加或删除程序**。
- 2. 在当前安装的程序的列表中查找 IBM Rational Build Forge 代理程序。
- 3. 单击**更改/删除**。
- 4. 请遵循指示信息以除去代理程序。

卸载 UNIX 或 Linux Build Forge 代理程序

使用以下指示信息从 UNIX 或 Linux 平台中卸载代理程序软件。

Linux 代理程序

要使用 rpm 软件包除去已安装的代理程序软件，请执行以下操作：

- 1. 查找代理程序软件并列出软件包名称和版本：

```
rpm -qa | grep bfagent
```
- 2. 删除代理程序软件：

```
rpm -e bfagent-<version_number>
```

Solaris 代理程序

要使用 `pkgadd` 程序除去已安装的代理程序软件，请运行以下命令：

```
pkgrm BFAgent
```

其他代理程序

对于其他平台，卸载过程是手动执行的，且因平台不同而有所区别。遵循适用于您的平台和超级服务器实施的指示信息。

注：要运行最多的命令，需要 `root` 用户访问权，且 `/sbin` 和 `/usr/sbin` 目录必须设置在当前 `PATH` 环境变量中。

1. 除去代理程序服务守护程序 `bfagent`。使用适用于您的平台的超级服务器实施（`inetd`、`xinetd`、`launchd` 或 `SMF`）的指示信息。

超级服务器	过程
inetd, 常用于较早的 UNIX 系统	<ol style="list-style-type: none">1. 编辑 <code>/etc/inetd.conf</code> 文件并除去 <code>bfagent</code> 的对应行。2. 查找 <code>inetd</code> 的进程标识。 <pre>ps -ef grep [i]netd</pre><p>对于 BSD 派生的系统，例如 FreeBSD 和 Mac OS/X 10.4 以及更早版本，请用 <code>ps auwwwx</code> 替换 <code>ps -ef</code>。</p>3. 阅读已更新的 <code>inetd.conf</code> 并启动 <code>inetd</code>。 <pre>kill -HUP <PID></pre>
xinetd, 常用于较新的 UNIX 系统	<ol style="list-style-type: none">1. 要除去代理程序服务，请运行以下命令： <pre>rm /etc/xinetd.d/bfagent</pre>2. 查找 <code>inetd</code> 的进程标识。 <pre>ps -ef grep [i]netd</pre><p>对于 BSD 派生的系统，例如 FreeBSD 和 Mac OS/X 10.4 以及更早版本，请用 <code>ps auwwwx</code> 替换 <code>ps -ef</code>。</p>3. 阅读已更新的 <code>inetd.conf</code> 并启动 <code>inetd</code>。 <pre>kill -HUP <PID></pre>
用于 Mac OS/X 和 OpenBSD 系统的 launchd	<ol style="list-style-type: none">1. 运行 <code>launchctl</code>。2. 输入 <code>stop com.ibm.rational.bfagent</code>。3. 输入以下命令：<code>unload /Library/LaunchDaemons/com.ibm.rational.bfagent.plist</code>4. 输入 <code>quit</code>。5. 运行以下命令：<code>rm /Library/LaunchDaemons/com.ibm.rational.bfagent.plist</code>
Solaris 10 的 Solaris 系统管理设施 (SMF)	<ol style="list-style-type: none">1. 运行 <code>inetadm -d network /bfagent/tcp</code>2. 运行 <code>svccfg delete -f network/bfagent/tcp</code>

2. 从 PAM 接口除去代理程序服务。
 - a. 编辑 `/etc/pam.conf` 并除去以 `bfagent` 开头的所有行。
 - b. 运行 `rm /etc/pam.d/bfagent`
3. 从 `etc/services` 文件中除去协议条目。

编辑 `/etc/services`, 并除去 `bfagent` 的相应行。
4. 除去由代理程序安装的以下文件:
 - `/etc/bfagent.conf`
 - `/etc/bfagent.conf-example`
 - `/usr/local/bin/bfagent/usr/local/bin/bfcrypt.dll`

第 14 章 发行说明

本部分包含该产品的当前发行版的发行说明。

发行说明 - *IBM Rational Build Forge V7.1.3.5*

Rational Build Forge V7.1.3.5 已正式发布。涉及兼容性、安装和其他入门问题。

© Copyright International Business Machines Corporation 2003, 2013. All rights reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

内容

- 『描述』
- 『新增和更改的功能』
- 『本发行版中已修订的 APAR』
- 『声明』
- 第 190 页的『与早期版本的兼容性问题』
- 第 190 页的『系统要求』
- 第 190 页的『安装』
- 第 190 页的『已知问题』

描述

IBM® Rational® Build Forge® 是一种适应性流程执行框架，它可以自动化、协调、管理和跟踪软件开发组装流水线中各次交付间的所有流程，从而创建自动化软件工厂。*IBM Rational Build Forge* 可以集成到您当前的环境中，并支持主要的开发语言、脚本、工具和平台。通过它，您可以在增加流程自动化、加速、通知以及调度的宝贵功能的同时继续利用现有投资。

新增和更改的功能

下表指示本发行版中的新增和更改的功能。有关最新的功能列表，请参阅 [Jazz.net](#) 上 GA 发行版的 [New & Noteworthy](#) 页面（需要帐户）。

本发行版中已修订的 APAR

有关此发行版中修订的 APAR 的最新列表，请参阅 [Jazz.net](#) 上 GA 发行版的 [Release Notes](#) 页面（需要帐户）。

声明

该版本的发行声明可从 www.ibm.com/common/ssi/index.wss 处获取。请参阅该声明获取以下信息：

- 详细产品描述，包括新增功能描述
- 硬件和软件要求
- 软件服务和支持可用性
- 打包和订购详细信息
- 国际兼容性信息

与早期版本的兼容性问题

V7.1 中的数据库表体系结构已更改。需要从旧的数据库表体系结构迁移 Build Forge 配置数据和历史数据。

升级到 V7.1.3.5 的需求取决于您的当前版本。

- 7.1.0.x、7.1.1.x 和 7.1.2.x: 如果正在运行其中的任何版本，可以使用 Installation Manager 的 Update 功能进行升级。
- 7.0.2.x: 如果正在运行其中的任何版本，必须先升级到 V7.1.1.3，然后再升级到 V7.1.3.5。

系统要求

特定产品的详细系统需求站点的查询报告中提供了硬件和软件需求。您提供所使用的产品、产品修订版、版本和操作系统。所述需求针对硬件、操作系统、数据库、代理主机和 Web 客户机。

安装

要了解有关逐步安装的指示信息，请参阅信息中心内的“安装”部分。

已知问题

已知问题以单独的技术说明的形式记录在以下网址处的支持知识库中：IBM Rational Build Forge 支持站点。发现并解决问题后，IBM 支持团队将更新知识库。通过搜索知识库可以快速查找变通方法或解决方案。

您还可以运行自己的查询，请求更多类型的信息：

- 下载
- Flash（警报）
- 新闻
- 产品信息和出版物
- 故障诊断

相关信息：

 [联系 IBM 支持](#)

第 15 章 “仪表板”选项卡

“仪表板”选项卡为您提供了系统状态和活动的概述。

作业成功

过去四周内通过或失败的作业数（按周显示）。您可以选择查看所有项目或单个项目的趋势。

调度 调度在将来运行的作业（按其调度运行时间列出）。您可以选择如何呈现列表（所有或有限数量）并在列表中进行浏览。

服务器状态

服务器及其状态：“可用”或“关闭”。您可以选择查看所有服务器的状态，或仅查看与您所选收集器相关联的服务器的状态。您可以选择如何呈现列表（所有或有限数量）并在列表中进行浏览。

作业队列

正在运行和正在等待的作业数。图表随时间的推移自动更新。“最大同时作业数”系统设置在图表中显示为阈值。

清除队列

正在运行和正在等待的清除数。图表随时间的推移自动更新。“最大同时清除数”系统设置在图表中显示为阈值。

得到许可的用户数

已使用的得到许可的用户数。它显示在指示器拨号盘和文字说明中：“已使用 (X) 个得到许可的用户数（共 (Total) 个）”。

注： 由于浏览器限制，使用 Internet Explorer 时，不会显示拨号盘上的数字。文字说明中的数字是正确的。

第 16 章 管理

本主题描述 Build Forge® 系统的管理操作。

关于管理

“管理”面板用于管理配置和首选项。

您可以在“管理”面板内处理用户、安全性特权、通知设置以及系统设置，以配置管理控制台。

要访问“管理”面板，请在左侧菜单中单击**管理**。



访问组

访问组是系统用于控制许可权的用户集合。

使用**管理 > 访问组**面板以新建访问组、添加或除去用户以及修改组属性。该面板显示现有访问组的列表。单击组名以将其选中，并在面板下半部分显示其属性。

[UI 配置](#) | [控制台](#) | [报告](#) | [注销](#) Root User

访问组

添加组

[帮助](#)

[过滤器](#)

正在显示 1 - 4 (共 6 项)

[全部显示](#)

访问组名称	新用户缺省值	LDAP 组 DNS
构建工程师	否	
开发人员	是	*
操作员	是	*
访客	否	

新建组

[详细信息](#)

☐ 缺省

名称:

所有者:

LDAP 组 DNS:

- 要新建组，请单击**添加组**以清除面板下半部分中的字段（如果需要）。然后为组提供**名称**并选择**所有者**组。（**所有者**组控制新组的访问权。）要编辑组，用户必须是**所有者**组的成员。）如果使用的是 LDAP 认证，请填写“LDAP Group DN”字段，以便通知系统要将哪些 LDAP 组映射到您的组。例如，可以将 Developer 组映射到名为 SoftwareEngineers 的 LDAP 组，然后为 Developer 组指定许可权，以便提供希望软件工程师拥有的访问权类型。
 - 在“LDAP 组 DN”字段中，列出所有 LDAP 组（其成员应接收与该访问组关联的管理控制台安全性特权）的“专有名称”。
 - 可以将多个 LDAP 组映射到任何访问组。可以在该属性中使用星号（*）字符来提供该访问组中所有 LDAP 用户的成员资格。可以列出多个 LDAP 组并用分号分隔这些组。
- 要删除访问组，请将其选中，然后单击**删除**。

注：但是，您不能删除指定为**访问**属性的访问组。例如，如果创建了访问组并设置了项目的**访问**属性以使用该访问组，那么不能删除该组。您必须先编辑项目以使用其它访问组。

- 要添加或除去用户，请选中组，然后单击**用户**选项卡。系统将在左侧显示非成员列表，在右侧显示成员。选择用户并使用**添加**和**除去**按钮将用户从一个列表移到另一个列表。

注：已将“映射访问组映射”设置为“是”的 LDAP 用户不会出现在**用户**选项卡中。此行为可确保“映射访问组映射”设置实际执行组映射，而不会与 Build Forge 的手工设置产生冲突。

- 要嵌套组，请将一个组添加为另一个组的子组。执行此操作后，应用于包含组的所有许可权也将应用于成员组中的所有用户。要使一个组成为另一个组的子组，请选

194 Rational Build Forge: 信息中心

中期望的父组，然后单击面板下半部分中的**子组**选项卡。选择要设置为子组的组，然后单击**添加**。您可以递归地嵌套组，例如，向子组添加父组，以便 A 组包含 B 组，而 B 组也包含 A 组。如果执行此操作，系统会将 A 组的所有成员视为 B 组的成员，反之亦然。

- 要管理某个组的许可权，请选中该组并单击**许可权**选项卡。您可以查看该组的当前许可权，也可以添加或删除许可权。

访问概述

系统在其数据库中管理用户。您可通过分配给用户的组控制用户特权。您将特权分配给组，然后使每个用户成为相应组的成员。

这是基于角色的系统：每个组代表某个用户在组织中可担任的一个角色。角色有特权。用户的特权是该用户所属组的权限总和。您不能将特权直接分配给各个用户，只能分配给组。

系统还使用访问组来进行通知。在将系统配置为发送通知消息时，消息的目标必须是访问组。请参阅第 297 页的『设置通知』。

安全特权或许可权，定义了组可执行哪些操作以及可看见哪些内容。它们可充当系统的组体验的过滤器。例如，属于“来宾”组（且不属于其他组）成员的某个用户只能看到将“来宾”组指定为“访问”属性的项目。该用户只能启动“来宾”可访问的项目。如果该用户同时又是“开发人员”组的成员，他将看到“访问”属性为“来宾”或“开发人员”的所有项目。

注：可使用现有的 LDAP 数据库代替系统的数据库来进行用户认证。当您使用 LDAP 代替在系统中定义用户时，即允许 LDAP 数据库中的部分或全部用户访问系统。您还可以将访问组映射到 LDAP 组。有关设置 LDAP 的详细信息，请参阅第 202 页的『关于 LDAP 集成』。

可利用访问组进行控制的活动和资源为许可权、服务器、项目、步骤和访问组。

- 要将对资源（服务器、项目或步骤）的访问权扩展到某个特定组，请选择该资源，然后将其“访问”字段更改为该组的名称。例如，要将“开发者”组的访问权分配给名为 Win234 的特定服务器，请将该服务器的“访问”属性设置为“开发者”。

注：如果用户不是服务器或项目的访问组的成员，那么该用户看不到“服务器”或“项目”列表页面上列出的对象。如果某个用户不是步骤的访问组的成员，那么该用户可以在列表中看到针对该项目的步骤，但是不能编辑或运行该步骤；如果用户运行包含该步骤的项目，系统将跳过该用户无权访问的步骤。

- 要允许一个访问组的成员编辑另一个访问组，请将一个组设置为另一个组的控制组。例如，要允许 A 组的成员向 B 组添加成员，请将 A 组设为 B 组的控制组。
- 要将全局特权扩展到某个组，请使用**管理 > 许可权**页面，为该组启用特定的许可权。

这一灵活的模型可使您安全地向某些类型的用户赋予一种特权（例如运行构建的能力），同时又限制了其他特权（例如编辑项目或使用某些服务器的权限）。

访问示例：对组授予运行作业的能力

可以使用安全性功能将运行某些作业的能力扩展到您的某个访问组。例如，您可能想要允许一组设备驱动程序程序员运行与其工作有关的作业，同时不让其他作业打乱他们的系统视图。但是，不想允许这些程序员编辑这些作业。要创建这一场景：

1. 在您的组织中为此角色创建一个访问组（例如，DeviceDriverDevs）。
2. 将这个新的访问组指定为您希望这些用户能够运行的所有项目的访问属性。
3. 同时确保项目的步骤具有相应的访问属性。运行作业时，将跳过这些用户无权访问的所有步骤。
4. 将“执行构建”许可权分配给该组。
5. 将需要启动这些构建的所有用户添加到新 DeviceDriverDevs 组。您可能需要将系统管理员也设为该组的成员。当您更改了项目的访问属性后，不是 DeviceDriverDevs 组成员的用户将无法再查看、运行或编辑项目。

请注意：用户可能是很多组的成员，并且许可权是累加的。您可能代表另一个项目团队的组（例如，PlatformDevs），如果项目的“访问”属性设置为这两组中的任何一组，同为这两个组成员的用户将能够查看和启动这些项目。

团队和项目安全性规划

如果您有很多用户从事着不同的项目，那么您可使用下面的一般规划来管理这些用户，以使各个用户获得他们所需要的许可权，但又只能看到他们需要与之交互的项目和其他资源：

- 为人员执行的各种活动创建基于角色的访问组。例如，创建“构建管理人”和“开发人员”组。为这些组分配与其作业相称的权限。“构建管理人”可能拥有大部分可用权限，而“开发人员”可能只有与执行作业相关的权限。
- 为组织中的每一个跨职能团队创建额外的组。您可能会有 IDE 团队、打印机驱动程序团队等等。
- 将项目、服务器和其他资源的“访问”属性设置为这些团队组。所有与 PrinterDriver 团队相关的项目都应将 PrinterDriver 访问组作为其访问属性。
- 在向系统添加用户时，请将用户分配到所有相应的访问组。所有用户必须分配给至少一个角色组和至少一个团队组。

如果遵循这些准则，用户将只能看到与之相关的项目，并拥有与其在那些项目中的角色相称的权限。此外，当他们在组织中的工作更改时，您可以轻松更改其用户权限。

管理访问属性

用户根据其访问组设置访问属性。用户必须是访问组的成员，才能向该访问组分配数据对象，如项目、服务或步骤。

例如，如果您不是 Admin 组的成员，那么不能向该组分配项目。

访问组列表仅包含您所属的那些访问组。

步骤从其父项目中继承访问组属性。步骤创建程序可更改步骤的访问组属性，以便该步骤具有与项目不同的访问组属性。如果为步骤指定了不属于访问组的用户，那么这些用户不能运行该步骤。这使您可以阻止用户运行项目中的特定步骤。

用户

这些主题描述了如何管理系统中的用户。

用户属性

您可以通过使用**管理 > 用户**来创建用户和对用户指定属性。还可以将系统连接到 LDAP/Active Directory 数据库以获取用户信息。您是通过将用户分配到组来管理用户安全许可的，因此必须创建一些用户来测试安全功能。

显示用户记录时，有三个选项卡可用：

- **详细信息**：使用此选项卡可编辑大多数用户属性。下面描述了可用的属性。
- **当前组**：显示用户所属的访问组（直接属于其他组或者是某个直接组的成员）。
- **更改组**：显示用户直接隶属的组，并允许您将用户添加至组或从组中除去用户。

The screenshot shows a web-based user management interface. At the top, there are several action buttons: (New User), Save, Copy, Switch User, Password Expiry, Register User, Clear Cache, Unlock, and Delete. Below these is a tabbed interface with 'Details' selected. The 'Details' tab contains a form with the following fields and values:

Field	Value
用户名 (Username)	
电子邮件 (Email)	
名称 (Name)	
密码 (Password)	
时区 (Timezone)	US/Central
日期格式 (Date Format)	03-1-1 下午2:30
使用终端 (Use Terminal)	否 (No)
用户类型 (User Type)	普通 (Normal)
已验证 (Verified)	
语言 (Language)	简体中文 (Simplified Chinese)
每星期的日历起始日 (Weekly Start Day)	星期日 (Sunday)
限制 (Limit)	无限制 (Unlimited)
优先级登录 (Priority Login)	否 (No)
密码到期 (Password Expiry)	是 (Yes)
截断 (Lockout)	缺省值 (Default)
搜索 (Search)	区分大小写 (Case Sensitive)
步骤日志初始视图 (Step Log Initial View)	第一页 (First Page)

对于每个用户，可以在**详细信息**选项卡中设置以下属性：

名称 用户的显示名称和标签。

电子邮件

系统向该用户发送电子邮件通知时使用的目标电子邮件地址。

注： 仅向显式选择来通知的用户发送电子邮件。支持以下电子邮件地址格式。
Username 是电子邮件用户名。*Name* 是任意字符串，通常是某种形式的给定名称。

```
username@host.com
name username@host.com
name "<username@host.com>"
name <username@host.com>
name "username@host.com"
```

用户名 用于登录管理控制台的名称。

密码 用于登录管理控制台的密码。对于当前已登录的用户，不会显示该字段。使用该字段可输入新密码或更改现有密码。应在**验证**字段输入相同的密码。

限制 设置用户可以在一天中启动的最大作业数。当达到限制时，系统会显示消息，表明已经超出用户的运行限额。如果该值为 0，那么系统允许用户运行任意数量的构建。

时区 用户的时区。系统使用 root 用户的时区作为所有公布时间的缺省时区。公布时间使用服务器主机时钟。

缺省情况下，为系统中用户和 LDAP 用户指定的时区与 root 用户的时区相同。用户可以编辑分配给自己的时区。

验证密码

在此处再次输入密码以对其进行验证。

优先级登录

指定该选项使用户成为优先级用户。优先级用户始终可以登录系统。如果没有更多可用用户许可证，系统会注销具有最早会话的用户，以便为优先级用户腾出名额。root 用户可始终登录并且不占用许可证。

日期格式

选择用户的首选显示格式。

语言 选择用户的语言。

密码到期

如果选中该选项，那么在经过一定的天数之后用户的密码会到期，该天数在**密码到期天数**系统设置中指定。

使用屏幕阅读器

如果设置为“是”，那么界面能够支持针对视力不佳用户的屏幕阅读器功能，例如动态突出显示和聚焦。

用户类型

用户可以属于以下三种类型之一：

- 正常：可以使用 Web 客户机访问控制台。他们可以根据其访问组成员资格和许可权对系统进行更改。对每个用户只允许一个会话。
- 只读：可以使用 Web 客户机访问控制台。他们不能对其个人信息（例如，其密码和首选项）之外的任何信息进行更改。对每个用户只允许一个会话。
- API：不能使用 Web 客户机访问控制台。此类型用户使用以 Java API 或 Perl API 开发的客户机来访问系统。同一 API 用户可以有多个登录会话。会话信息将自动共享。

日历起始日

选择调度日历首先显示星期几。缺省值为星期日。

截断 控制列表和下拉菜单中显示的字符数。例如，如果设置为 20，对于项目名称只能显示其前 20 个字符。

搜索 确定“过滤”字段如何在整个产品中执行搜索。值如下：

- 区分大小写（缺省值） - 搜索会考虑到每个字母的大小写。
- 不区分大小写 - 搜索不考虑大小写。

步骤日志初始视图

指定在作业结果中打开步骤日志时对其进行定位（以供查看）的方式。如果设置为“第一页”（缺省值），那么会将视图定位于日志的第一页。如果设置为“最后一页”，那么会将视图定位于日志的最后一页。

管理用户

这些主题描述了对用户的基本管理操作。

选择**管理 > 用户**以显示当前用户的列表，在该列表下方有一个用户面板。系统将显示每个用户的名称、登录、电子邮件、限制、活动（自上次用户登录以来的耗用时间，使用服务器主机时钟计算）以及时区。

- 要创建新用户，请在未选择任何用户的情况下在用户面板中输入属性。如果选择了用户，请单击**添加用户**来清除面板。完成对用户信息的编辑之后，单击**保存**。
- 要编辑用户，请单击用户名并在用户面板中编辑属性，然后单击**保存**。
- 要注销用户，单击该用户的名称，然后单击**注销用户**。
- 要以不使用密码的用户登录，请首先作为 root 登录。单击用户名，然后单击**切换用户**。root 用户可以任何用户身份并无需使用密码。如果您当前以非 root 用户身份登录，并希望以另一非 root 用户身份登录，那么首先必须以 root 用户身份登录。
- 要释放固定许可证席位，首先作为 root 登录。单击用户的名称，然后单击**清除席位**。控制台将从标识列表中除去占用固定许可证名额的该用户。如果用户已经登录，控制台还将注销该用户。对于固定许可证，控制台将统计已登录的用户数量。达到限制时，新用户将无法获取许可证。必须删除或清除现有用户才能为其他用户留出名额。清除席位不会从控制台中删除用户。如果用户重新登录，固定许可证计数将增加。如果使用浮动许可证用户，**清除席位**与**注销用户**效果相同。
- 要复制用户，请单击该用户的名称，然后单击**复制**。新用户名称显示在列表中，其后将附加单词“Copy”。

注：复制的用户的密码将被重新设置为 password。要更改密码，请作为 root 用户或复制的用户登录。

- 要删除用户帐户，请单击用户名，然后单击**删除**。

如果“删除”被禁用，那么说明已调度作业由该用户帐户所有，且无法删除该用户帐户。如果想要删除具有调度作业的用户帐户，必须首先删除调度的作业。

用户记录为用户的系统体验设置缺省属性，并控制用户的登录名、密码和密码到期设置。用户记录数据可以通过管理控制台输入到系统，也可以从 LDAP/Active Directory 数据库获取。

注：如果所编辑用户的记录是从 LDAP 数据库获取的，那么“用户”页面上的许多字段处于禁用状态。必须在源数据库中更改这些属性。

使用用户类型控制访问

“用户类型”属性的设置确定用户拥有的访问类型。

- **正常：**可以使用 Web 客户机访问控制台。他们可以根据其访问组成员资格和许可权对系统进行更改。
- **只读：**可以使用 Web 客户机访问控制台。他们不能对其个人信息（例如，其密码和首选项）之外的任何信息进行更改。
- **API：**不能使用 Web 客户机访问控制台。此类型用户通过使用 Java API 或 Perl API 开发的客户机进行工作。同一 API 用户可以有多个登录会话。会话信息将自动共享。

通过 LDAP 登录的用户可能会确定为用户类型“正常”或“只读”，具体取决于写访问组 DN 属性的设置。请参阅第 202 页的『LDAP 集成』。

帐户锁定

可以使用系统设置来实施锁定策略。

锁定策略确定多次失败登录尝试将阻止用户登录的条件。条件设置为连续失败登录尝试的最大次数。计时器确定是否将一组登录视为连续登录尝试。系统可根据计时器“忘记”失败的登录。

如果用户帐户自上一次成功登录后曾登录失败，那么下一次成功登录时会显示一个弹出窗口，以显示登录尝试失败的次数。仅当未满足锁定条件时才会显示该弹出窗口。

满足锁定条件时，用户会进入锁定状态。用户可能需要执行以下两个操作之一：

- 等待预先确定的时间量，然后才会接受其他登录尝试。
- 与系统管理员联系以将帐户解锁。

系统设置确定锁定策略：

- **帐户锁定最大尝试次数：** 如果为 0，将禁用帐户锁定。它可以设置为触发锁定所需的连续失败尝试次数。缺省值为 3。
- **帐户锁定重置计时器：** 如果设置为负值，那么用户必须请管理员将其帐户解锁，才能再次登录。它可以设置为用户能够再次登录之前所需的分钟数。缺省值：120 分钟。
- **帐户锁定递减：** 如果设置为负值，那么绝不会忽略失败的尝试。它可以设置为忽略上次失败登录尝试之前所需的分钟数，并且不会针对最大失败尝试次数进行计数。缺省值：60 分钟。

请参阅**管理 > 系统**。

要重置锁定的帐户，root 用户需要使用 `COMMAND_USER_RESET_LOCKOUT` 命令。

Root 用户

root 用户（登录名为 **root** 的用户）在系统中有特殊的性质：

- **安装时创建：** root 用户是安装程序创建的唯一缺省用户。缺省密码为 **root**（请在安装后立即更改密码）。
- **不需要许可证：** root 用户不使用用户许可证。无论已登录的用户有多少，您总是可以作为 root 用户登录。（当有人作为 root 登录时，其他已作为 root 登录的任何用户将注销。）
- **系统时区：** root 用户的时区是管理控制台的缺省时区。其他用户（系统中用户和 LDAP 用户）的时区缺省情况下取自 root 用户的时区。用户可在登录后设置其自己的时区。系统中报告的所有时间和日志都是以用户的时区表示的。
- **所有许可权：** root 用户具有所有可用的许可权，并且可以编辑其他用户的属性。您不能删除 root 用户的任何访问权限。虽然 root 用户不是任何访问组的成员，但是 root 用户仍然可以查看、编辑或使用系统中的任何数据对象。
- **优先级：** root 用户始终是具有优先权的用户。
- **作为任何用户登录：** root 用户可以通过单击**管理 > 用户 > <UserName>** 页面上的**切换用户**，在不使用密码的情况下以某个用户身份登录。
- **注销当前用户：** root 用户可以通过单击**管理 > 用户 > <UserName>** 页面上的**注销用户**来注销用户。
- 缺省情况下，分配给 LDAP 用户的时区与 root 用户相同。但是，他们可在登录后编辑分配给他们的时区。系统会记住新的首选项。

API 用户

通过使用 Java API 或 Perl API 开发的客户机访问控制台的用户应设置为用户类型 = API。此类型的用户不能使用 Web 客户机访问控制台。但是，此类型的用户可以有多个登录会话。会话信息在会话间自动共享。

同一 API 用户打开的多个会话仅耗用一个许可证。所有会话必须全部注销才能释放该许可证。您还可以使用 root 帐户释放许可证。请参阅第 198 页的『管理用户』。

如果您的用户需要通过 Web 客户机和 API 客户机访问控制台，那么请为其设置单独的用户标识。访问 Web 控制台的用户的用户类型为“正常”或“只读”。这些用户类型不具有多个会话。如果某用户尝试再次登录，将终止第一个登录会话。

许可权

许可权定义用户可在系统内执行的操作。请将许可权指定给访问组；切勿将其直接指定给用户。

要处理许可权，请选择**管理 > 许可权**。



要为组指定许可权，请选择许可权并确保相应组列出且具有该许可权。选择**管理 > 许可权**，从列表中选择许可权，然后处理面板的下半部分。左侧的组缺少许可权，而右侧的组则具有许可权。要为组提供许可权，请在左侧的组中选中该许可权，然后单击**添加**。

许可权练习

在此示例中，您对用户 Jane Doe 授予添加和编辑服务器的专有权，方法是将这些许可权授予新的访问组，并将她分配到该组。

1. 作为 root 用户登录。
2. 创建名为 Server Admin 的新访问组。

3. 将 Jane Doe 添加到该新访问组中。
4. 转至**管理 > 许可权**。
5. 滚动至**添加新服务器**许可权并单击它。
6. 在**详细信息**选项卡中，使用**添加**和**除去**使 Server Admin 成为具有此许可权的唯一访问组。
7. 单击**更新**。现在，只有 Jane Doe 有权向系统中添加服务器。

注： root 用户始终具有所有许可权。您不能删除 root 用户的任何访问权限。

请注意：只有 Jane 可以添加服务器。但她不具有输入服务器登录信息的能力。这可以通过将 Server Admin 组添加到**编辑服务器认证**许可权中来实现。

LDAP 集成

您可以将 Build Forge 设置为使用 LDAP 服务器登录用户。通过使用 LDAP，用户就可以使用他们在组织的其他地方使用的相同登录名和密码登录 Build Forge。使用 LDAP 时，不必在 Build Forge 中手动创建用户。每个用户首次登录 Build Forge 时，Build Forge 中都会创建该用户。

也可以将 LDAP 组映射到 Build Forge 访问组。这种集成使您能够管理 LDAP 中的组，并且可以在用户登录时自动更新用户许可权。

仍然可以在 Build Forge 系统中手动创建并维护用户。这些用户的访问权必须手动进行管理。

要启用此集成，请在**管理 > LDAP** 中创建条目。

注： 仅 root 用户可以在 Build Forge 用户界面中创建和编辑 LDAP 域条目。

关于 LDAP 集成

用户首次使用 LDAP 凭证登录到 Build Forge 时，该用户将在 Build Forge 中进行认证和设置，如下所示。

要点： 如果您要使用组映射，请在用户登录之前 启用 LDAP 组映射。

如果组映射被禁用，您在用户登录后再启用组映射，那么不会对现有用户执行该映射。如果在用户已登录后启用 LDAP 组映射，请从 Build Forge 用户列表中删除这些用户，然后让他们再次登录。这样，用户的 Build Forge 访问组成员资格就基于 LDAP 组映射，而不是您所做的任何手工更改。

1. 用户在登录面板上会看到**域**字段。如果配置了多个域，该字段为下拉列表。该用户选择域并登录。

注： 如果您配置多个域，那么各个唯一用户标识在这些域中必须是唯一的。该系统对每个用户只允许一次登录。如果一个用户已登录，随后另一个用户使用相同的唯一用户标识登录，那么第一个用户会话将关闭。请参阅第 7 页的『访问和使用控制台』以获取有关用户会话的更多信息。

2. Build Forge 会在 LDAP 服务器上检查该帐户。您可以将 Build Forge 配置为使用普通用户或管理用户执行检查。

3. 如果找到了该用户名，那么 Build Forge 会尝试使用用户在 Build Forge 登录面板上提供的凭证（或是使用 API 客户机的程序登录时所用的凭证）登录到 LDAP。
- 如果凭证不匹配或找不到该用户名，登录失败。
 - 如果凭证匹配，登录继续。
4. 如果该用户之前没有登录过，那么 Build Forge 会自动在其用户列表中创建该用户。对于通过 LDAP 登录的用户，用户名、密码、登录、确认和电子邮件字段都已禁用，因为这些信息由 LDAP 提供。

注：系统会在 LDAP 用户首次登录时向其分配 root 用户的时区，因为系统不会从 LDAP 获得时区信息。您也可以在以后手动设置时区。

5. Build Forge 对用户应用访问组。
- 如果启用了 LDAP 组映射，那么将应用指定的访问组。同时还将应用缺省的 Build Forge 访问组。启用组映射需要在 Build Forge LDAP 域属性中进行配置。

注：每次用户登录时，都将执行组映射。这可使 Build Forge 与 LDAP 中的组成员资格更改保持同步。

- 如果未启用 LDAP 组映射，将应用缺省的 Build Forge 访问组。然后，可以手动管理访问组成员资格。

LDAP 域属性

要编辑已创建 LDAP 域的属性，请执行以下操作：

1. 选择管理 > LDAP<域名>
2. 选择要编辑的域。属性将显示在 LDAP 域属性面板中。

新建 LDAP 域 保存 复制 删除 设置为缺省值 测试 LDAP 域

详细信息

名称: 管理 DN: 映射访问组:

主机:	<input type="text" value="ldap.example.com:389"/>	显示名称:	<input type="text" value="displayname"/>	已授权组 DN:	<input type="text"/>
密码:	<input type="text"/>	专有名称:	<input type="text" value="distinguishedname"/>	写访问组 DN:	<input type="text"/>
已验证:	<input type="text"/>	组名:	<input type="text" value="memberof"/>	搜索条件:	<input type="text" value="cn=users,dc=example,dc=com"/>
绑定用户帐户:	<input type="text" value="是"/>	邮件名称:	<input type="text" value="mail"/>	组搜索条件:	<input type="text"/>
协议:	<input type="text" value="LDAP"/>	唯一标识:	<input type="text" value="SAMAccountName=%"/>	组唯一标识:	<input type="text"/>

3. 编辑任何字段的值，然后单击保存。以下字段是必填字段：
- 名称
 - 主机
 - 绑定用户帐户
 - 协议
 - 显示名称
 - 专有名称
 - 邮件名称

- 唯一标识

4. 如果您希望缺省使用该域, 请单击**设为缺省值**。

名称 必需。Build Forge 中的 LDAP 域名称。如果配置了至少一个 LDAP 域, Build Forge 登录表单将按名称列出域。

管理 DN

用于提供对 LDAP 服务器数据库的搜索访问的帐户。如果服务器允许匿名绑定以进行数据库搜索, 请将此字段保留为空。

一些 LDAP 服务器要求进行管理绑定才能进行数据库搜索。该设置使您可以指定管理员帐户的 DN, 如以下示例所示。

`cn=Administrator,cn=users,dc=example,dc=com`

在**密码**和**验证密码**字段中, 指定“管理 DN”帐户的密码。

映射访问组

确定是否将组信息从 LDAP 服务器映射到管理控制台中的访问组。缺省值为“否”。Build Forge 中每个访问组的 **LDAP 组 DN** 属性都必须设置为 LDAP 中的正确组名。

- 如果为**否**, LDAP 组不会映射到 Build Forge 访问组。只要用户登录一次, 您就可以将用户分配到 Build Forge 访问组。使用此选项意味着您在 Build Forge 中管理用户访问组。用户首次登录并在 Build Forge 中创建了用户名时, 将应用缺省访问组。
- 如果为**是**, 用户每次登录到 Build Forge 时, Build Forge 都会在 LDAP 服务器中刷新该用户的组成员资格信息。自上次登录以来在 Build Forge 中对用户的访问组成员资格所作的任何更改均会被覆盖。使用此选项意味着您在 LDAP 中管理所有组成员资格。LDAP 组成员资格会自动映射 (添加或删除) 到 Build Forge 访问组。组属性如下用于确定用户的组成员资格:
 1. 如果**组名称**不为空, 查询指定关键字的值。将返回的值用作用户的组。
 2. 如果**组名称**为空或**查询未返回值**, 请使用**组搜索条件**和**组唯一标识**来查询用户所属的 LDAP 组。
 3. 如果在 (1) 和 (2) 中都未返回组信息, 那么将允许用户登录, 并对其分配指定为新用户缺省访问组的组中的成员资格。

主机 必需。LDAP 服务器的主机名和端口。示例:

`ldapservers.mycompanyname.comldap.mycompany.com:9000`

密码 管理 DN 帐户的密码。如果指定了**管理 DN**, 此字段是必填字段。

验证密码

重复输入管理 DN 密码。

绑定用户帐户

必需。确定 Build Forge 在登录时是否尝试针对 LDAP 验证用户凭证。缺省值为“是”。

- 如果为**是**, 那么 Build Forge 在登录时将检查对 LDAP 服务器提供的用户名和密码。
- 如果为**否**, 那么 Build Forge 无需验证就接受用户名。此设置在对 Build Forge 实施外部密码验证 (如单点登录 (SSO)) 时使用。

协议 必需。确定 Build Forge 从目录服务读写数据以认证 Build Forge 用户时所使

用的协议。缺省值为 LDAP。如果使用 LDAP over SSL (LDAPS)，请输入 LDAPS。此选项还需要其他设置。请参阅启用安全的 LDAP (LDAPS)。

显示名称

必需。输入指定用户全名的键名。

专有名称

必需。输入指定用户帐户的专有名称的键名。

邮件名称

必需。输入指定用户电子邮件地址的键名。

组名 输入 LDAP 模式中保存用户所属组的组列表的键名。此字段仅在映射访问组为“是”或使用已授权组 **DN** 时才使用。

已授权组 DN

LDAP 组的专有名称。如果已设置，那么只允许指定组的成员登录。如果为空白，那么任何有效的 LDAP 用户均可登录到控制台。

写访问组 DN

确定用户具有正常还是只读访问权。值可以是以下选项之一：

- 空 - 对于新登录，用户类型将设置为“正常”。现有用户保持其分配的用户类型（正常、只读或 API）。类型在**管理 > 用户**中设置。
- *（星号） - 所有登录的用户类型都将为“正常”。
- LDAP 组名 - 如果用户属于该组，那么该用户的类型将设置为“正常”。如果用户不属于该组，那么该用户的类型将设置为“只读”。
- 其他 - 使用其他任何值将强制所有用户为“只读”。示例：RO。

搜索条件

必需。用于在 LDAP 记录中查询用户的搜索字符串。示例：

```
cn=users,dc=buildforge,dc=com
```

唯一标识

必需。确定 LDAP 数据库中用于与用户登录时输入的用户名进行比较的字段。使用 % 字符代表用户输入的登录名。示例：

```
(sAMAccountName=%)
```

组搜索条件

需要**组唯一标识**。此字段仅在映射访问组为“是”或使用已授权组 **DN** 时才使用。它是用于在 LDAP 记录中查询组数据的搜索字符串。如果 LDAP 数据库用来存储组成员资格的数据库独立于用来存储用户记录的数据库，那么需要此选项。示例：

```
cn=groups,dc=buildforge,dc=com
```

组唯一标识

需要**组搜索条件**。此字段仅在映射访问组为“是”或使用已授权组 **DN** 时才使用。确定 LDAP 用户数据库中用于获取组成员资格信息的字段。该过滤器可以使用用户帐户的任意数据字段作为进入组表的键。使用 *%fieldname%* 语法可以标识此字段。如果组表使用 sAMAccountname 字段作为用户的键，那么可使用以下示例。

```
sAMAccountName=%sAMAccountname%
```

任务

这些主题介绍有关使用 LDAP 域的任务。

创建 LDAP 域条目

可以根据需要创建任意数量的 LDAP 域条目。用户试图登录时，必须指定域。对于 API 客户机登录，必须在登录调用中指定域，否则必须在 `bfclient.conf` 中配置要使用的域。

要添加域条目，请执行以下操作：

1. 选择**管理 > LDAP**。
2. 单击添加 **LDAP 域**。
3. 填充或更改域属性。**名称**属性对 Build Forge 而言是内部属性。缺省提供的值旨在用于 LDAP 或标准 Microsoft® Active Directory 服务器。
4. 单击**保存**。

测试 LDAP 域条目

要验证 LDAP 域是否设置正确，请执行以下操作：

1. 选择**管理 > LDAP**。
2. 从列表中选择域。
3. 单击**测试 LDAP 域**。

系统将使用 LDAP 域条目中的属性来查询 LDAP 服务器。

启用安全 LDAP (LDAPS)

如果 LDAP 服务器支持 LDAP over SSL (LDAPS)，可以将 Build Forge LDAP 域条目也配置为使用 LDAPS。缺省情况下已配置了严格的 SSL。严格的 SSL 要求服务器认证。

1. 在 Build Forge 中创建 LDAP 域条目。
2. 将**协议**属性设置为 LDAPS。这样将启用 LDAPS 的仅加密方法。
3. 将**主机**设置为 LDAPS 服务器的标准域名和 SSL 端口。端口 636 是为严格安全 LDAP 定义的缺省端口。示例: `myldap.mycompany.com:636`。
4. 从 LDAP 服务器获取签署者证书并将其添加到 Build Forge 信任库。缺省情况下，出站 LDAP 配置为使用**管理 > 安全**中的以下设置：
 - SSL 面板：缺省 JSSE 出站 SSL
 - 密钥库面板：缺省 JSSE 信任库。缺省情况下，该信任库设置为使用 `<bfinstall>/keystore/buildForgeTrustStore.p12`。将签署者证书放在此处。
5. 重新启动 Build Forge。
6. 转至**管理 > 安全**，选择您的安全 LDAP 配置。
7. 单击**测试连接**。

注：缺省情况下，在 Build Forge 中设置了严格的 LDAPS SSL。严格的配置要求验证服务器证书。如果不想使用严格的 LDAP，请执行以下操作：

1. 在 `JAVA_OPTS` 环境变量中设置 `Tomcat` 系统属性 `-Dcom.buildforge.services.server.ldap.strict=false`。Tomcat 脚本读取此变量并将指定的所有系统属性应用于 Tomcat 进程。

2. 重新启动 Build Forge。

在此配置中，无需向 Build Forge 信任库添加 LDAP 服务器证书。但是，此配置是 SSL 协议设计的一个并不强健的实施。Build Forge 与 LDAP 服务器进行通信时不会验证该服务器的身份。

更改 LDAPS SSL 配置

出站 LDAP 请求使用的 SSL 配置在缺省情况下已经设置。您可以更改两方面的设置：

- SSL 配置。如果 LDAP 服务器无法通过使用缺省的协议或握手与 Build Forge 通信，那么需要执行此操作。
- 密钥库配置。严格的 SSL 要求将签署者证书放在客户机（Build Forge）使用的信任库中，以实现与 LDAP 服务器的安全通信。如果想使用其他信任库或将其放在其他位置，需要在 Build Forge 中为信任库新建密钥库配置。

以下指示信息假设已经为 Build Forge 启用了安全 LDAPS，但还没有为 Build Forge 组件启用 SSL。

要更改 LDAPS SSL 配置，请执行以下操作：

1. 如果要更改信任库的位置或名称，请将信任库放在 Build Forge 主机上的所需位置。将 LDAP 服务器的签署者证书添加到库中。
2. 在**管理 > 安全 > 密钥库**中创建信任库配置（如果需要）。信任库配置包含了信任库的位置和名称属性。
3. 在**管理 > 安全 > SSL**中创建 SSL 配置（如果需要）。将它配置为使用新的信任库配置（如已创建）。根据需要对配置进行其他调整。
4. 在**管理 > 安全**中，将**启用 SSL**设置为“是”（如果尚未设置）。这时会出现更多的字段。
5. 在**出站 LDAP**列表中选择您创建的 SSL 配置。请勿更改其他设置。
6. 单击**保存**。
7. 单击**更新主 BFClient.conf**。
8. 如果以前未启用 SSL，请执行以下操作：
 - a. 单击**启用 SSL** 将其设为“否”。
 - b. 单击**保存**。
 - c. 单击**更新主 BFClient.conf**。
9. 重新启动 Build Forge。
10. 在**管理 > LDAP**中，选择您的 LDAP 配置。
11. 单击**测试连接**。

关闭 LDAP/Active Directory 支持

要停止向 LDAP 或 Active Directory 进行认证：

1. 选择**管理 > LDAP**。
2. 删除所有域条目。单击域条目旁边的废纸箱图标可删除该条目。

不存在域时，Build Forge 只接受手动添加的用户的登录。

系统配置设置

可使用多种设置来配置管理控制台。这些设置可在**管理 > 系统**页面上找到。

单击**管理 > 系统**时，系统将显示设置列表。单击系统设置的名称以显示该设置的编辑面板。

注：对于接受数值的系统设置，管理控制台接受由一个或多个整数（0 到 9）组成的任意值。不支持数字分组字符，例如逗号 (,)、小数点 (.) 和其他非整数分隔符。




面板包含以下按钮：

- **保存：**保存对设置值所作的任何更改。
- **恢复为缺省值：**将设置复位为其缺省值。

下表描述了可用的设置。

设置	描述
帐户锁定递减	缺省值：60 分钟。经过该时间量之后，系统会丢弃最后失败的登录尝试。如果设置为负值，那么不会丢弃失败的登录。
帐户锁定最大尝试次数	缺省值：3。锁定帐户之前所需的连续失败登录尝试次数。如果为 0，将禁用帐户锁定。
帐户锁定重置计时器	缺省值：120 分钟。最小值：5 分钟。在锁定用户后允许其尝试再次登录之前所需的时间量。如果设置为负值，那么锁定永不到期：管理员必须重置用户以允许其尝试再次登录。
警报电子邮件限制	设置系统在指定的分钟数内发送警报电子邮件的最大数量。例如，10/60 将最大值设置为每小时 10 条消息。缺省值 0/0 由系统解释为消息数无限制。
应用内联的步骤容器环境	缺省值是“否”。若为“是”，将应用包含内联步骤的项目或库的环境。
最后应用服务器环境	缺省值是“否”。若为“是”，最后为步骤应用服务器环境。如果指定了服务器环境，这些环境将在步骤环境或项目环境之后应用。
审计最大字符串长度	缺省值：128（字符）。控制针对 DETAILED 更改可以记录的信息量。如果字符串值超过该字符数，那么将存储加密散列而不是原始值，以便节省存储空间。对于包含敏感信息的字段（如密码），不论字符串的长度如何，都将使用加密散列。

设置	描述
审计策略	<p>确定执行的审计日志记录量。</p> <p>NONE 不会记录任何审计信息。</p> <p>SECURITY 仅记录与安全相关的信息。示例：访问组创建和用户会话活动。</p> <p>BASIC 这是缺省设置。如果存在无效的审计策略值，也会使用缺省设置。仅会记录尽可能少的信息，如用户、对象类型、uuid 和操作。</p> <p>DETAILED 记录有关更改了哪些属性的详细信息，包括旧值和新值。并非所有对象都可以提供该信息。对于那些不能提供该信息的对象，BRIEF 和 DETAILED 策略之间没有区别。“审计最大字符串长度”系统配置设置可用于限制存储的数据量。</p>
自动注销分钟数	系统可自动注销空闲用户。此设置指定在系统注销用户之前，必须经过的空闲时间分钟数。当该设置为 0 时，系统不会自动注销用户。
自动清除审计日志天数 自动清除错误日志天数 自动清除信息日志天数 自动清除警告日志天数	这些值设置每种类别的日志条目保留在审计日志中的最大天数；较旧的条目将自动删除。如果值为 0，系统将永不删除该类别的日志条目。由于字符串值作为整数求值得到的结果都为 0，因此可使用“Never”之类的值代替 0。
构建取消检查频率	指定系统检查构建取消请求的频率，表示形式为两次检查之间的秒数。
控制台端口	Web 服务器用于侦听 Build Forge 请求的端口号。
控制台 URL	<p>Web 服务器用于侦听 Build Forge 请求的 URL。</p> <p>如果控制台是在非 80 端口上运行，就必须设置此项。如果设置了此项，其值将覆盖缺省的控制台 URL。它采用以下格式：<protocol>://<hostname>[:<port>]。示例：http://myHost:81。</p>
在失败时继续步骤部分	确定相同步骤中多条命令的运行行为。如果设置为缺省值“否”，步骤中失败的命令会导致该步骤失败。如果设置为“是”，失败的命令不会导致步骤失败。将运行步骤中的其他命令。
创建缺少的路径	缺省值是“否”。若为“是”，当项目路径尚不存在时，创建项目路径。
数据库大小阈值	<p>数据库大小的阈值，到达该值时控制台将发送通知。缺省值：2G。</p> <p>请注意：达到阈值后 UI 性能会降低。执行数据库清除或增加阈值大小可恢复性能。</p>

设置	描述
数据库大小阈值通知	达到“数据库大小阈值”时，将接收到电子邮件的用户名或通知组。如果使用了用户名，那么将使用该用户“电子邮件”字段中的地址。请参阅“管理 > 用户”。
设置一次日期	<p>确定 <code>.date</code> 命令值是每个作业设置一次还是每个步骤设置一次。行为取决于所分配的值，如下所示：</p> <ul style="list-style-type: none"> 否：这是缺省值。每个步骤中都会设置一次 <code>.date</code> 命令值。每个步骤都会使用代理程序机器的时区对 <code>.date</code> 值重新求值。代理程序会提供该值。重新启动之后会保持 <code>.date</code> 值。就像初始运行时一样，会对其进行重新求值。相同的变量对于使用它的每个步骤都具有不同的值，因为针对每个步骤，会重新解释 <code>.date</code> 值。 是：每个作业设置一次当前 <code>.date</code> 命令值。该日期值是通过引擎主机的时区（而不是代理程序的时区）生成的。重新启动之后会保持 <code>.date</code> 值。 <p>当您在电子邮件通知中使用包含 <code>.date</code> 的环境变量（例如，<code>.email</code> 命令或通知模板）时，请使用“是”。电子邮件中使用的环境变量随后会使用引擎的时区。</p>
缺省代理程序端口	设置用于连接到代理程序的缺省端口号。
缺省导入类	在导入的项目没有定义类，或者定义的类不存在时要使用的类。缺省值： <code>Production</code> 。
禁用对 XML 订阅源的认证	确定是否已认证 RSS 数据订阅源。
启用快速启动	<p>缺省值为“否”：所有项目都显示以下图标：。单击项目时，会检查该项目包括的所有变量中是否有类型为“必须更改”的变量。如果项目不包含“必须更改”变量，那么该项目已启动。如果项目包含“必须更改”变量，那么该项目不会启动，并会显示一个对话框以描述原因，而图标会更改为：。</p> <p>注：仅检查针对项目定义的变量。不检查针对项目中的步骤定义的变量。另请参阅第 258 页的『启动作业时更改项目变量』。</p> <p>如果设置为“是”，那么“项目”页面会检查页面中所有项目的所有环境以确定是否有类型为“必须更改”的变量。可以立即启动的项目用以下图标表示：。这是 V7.1.1.1 之前版本的缺省行为。</p>
硬运行限制	缺省值：否。如果为“是”，系统将启动项目的计划中的构建，前提是该启动操作不违反项目的 运行限制 设置。如果为“否”，系统会针对计划中的构建忽略该项目的 运行限制 设置。
导入缺省安全访问组	当 导入时附随安全访问 设置设定为“是”时，指定导入的项目的缺省访问组。
导入非安全缺省访问组	当 以安全访问导入 设置设定为“否”时，指定导入的项目的缺省访问组。该缺省组仅在导入文件缺少访问组时使用。

设置	描述
以安全访问导入	当设置为“是”时，系统会将先前的设置中列出的缺省访问组分配给导入的数据对象。这将覆盖导入的 XML 文件中指定的任何访问组，这样用户就无法通过导入数据来覆盖安全性。当设置为“否”时，系统将保留导入的文件中的任何访问组设置。
继承标记	当设置为“是”时，这将使通过链启动的作业使用与其调用者相同的作业标记。如果项目 MasterProject 的 BUILD_15 调用项目 ComponentProject，那么 ComponentProject 的该次运行的作业标记（和作业目录名）将变成 BUILD_15。 注： 被调项目总是继承调用者的原始标记；如果调用者的标记在运行期间更改（例如，由于 .retag 命令而更改），被调项目将仍然获得调用者起初的标记。
无效的相对目录字符	该项设置的字符如果用在项目名称中，系统会将这些字符更改为下划线。
LASTRUN 格式	输入表示 BF_LASTRUN 环境变量格式的值，使用为 .date 命令定义的日期格式字符（请参阅 第 336 页的『.date』）。
许可证服务器	许可证服务器主机名。该主机名在安装期间进行设置。示例：myhost.mycompany.com。值可能包含端口号。示例：myhost.mycompany.com:80。要更改许可证服务器，请参阅 第 23 页的『更改管理控制台的许可证服务器』。
链接调试方式	当设置为“是”时，如果作业有其定义的适配器链接，它们将运行对链接的测试，而不是运行相关联的项目。作业输出包含单个步骤，它包含适配器的输出。在对适配器接口进行故障诊断时，该数据十分有用。 注： 您可以通过将适配器链接的状态设置为“调试”来为个别适配器链接设置调试。该状态优先于“链接调试方式”设置。
链接手动作业	手动快速启动项目，而不从调度程序中运行时，确定系统是否通过适配器链接运行适配器。 该链接检查可能在该作业的 BOM 中产生其他输出。 如果设为“否”，那么运行作业时不会检查或运行该链接。
最大控制台进程数	设置控制台同时运行的最大进程数。用作控制台活动的常规限流口。系统将每个进程的标识存储在数据库中，并在启动新的外部进程之前检查进程总数，以此对进程进行管理。请确保此值比运行队列大小设置至少大 5；否则系统无法运行足够的进程来支持运行队列。
最大内联深度	控制系统允许项目内联的级数，以免项目无限制嵌套。缺省值是 32。如果此值设置为 0，系统使用 32。当系统达到内联限制时，不会运行超出该限制的内联项目，也不会包含项目中插入其步骤。将在系统消息列表中写入一则消息：“已放弃内联”。

设置	描述
最大并发服务器测试数	指定一次可运行多少服务器测试。根据系统资源，一次运行过多服务器测试可能严重减慢或锁死控制台。
最大同时清除数	控制可同时运行的清除操作数。您可以清除任意数量的构建，但可同时删除的构建数不会超过“最大同时清除数”。缺省值： 20
最大刷新数	页面自动刷新的最大次数。缺省值： 50
链接时覆盖类	确定系统是否将链式项目的类替换为其调用者的类。缺省值“是”会导致系统覆盖链式项目的类，并转用调用者的类。
密码到期天数	设置一个天数，密码设置为过期的用户在此天数之后必须更改其密码。超过此时间后，相关用户需要在下次登录时更改其密码。

设置	描述
密码格式	<p>使用由句点分隔的 6 个字段组成的格式字符串指定用户密码要求:</p> <p><i>length.char_types.upper.lower.numeric.special</i></p> <p>示例: 5.2.u1.l1.n1.s1。下面对此示例进行了解释。</p> <p>前两个字段指定如下内容:</p> <ul style="list-style-type: none"> • 最低密码长度 (字符数) • 四个剩余字段中将使用的最小字符类型数 (0 到 4 范围内的整数) <p>剩余字段指定字符类型和频率需求。每个字段包含一个类型和一个数字。</p> <ul style="list-style-type: none"> • 类型: u (大写)、l (小写)、n (数字) 或 s (特殊) 四者之一。大写 (U、L、N、S) 表示该字符是必需的。小写 (u、l、n、s) 表示字符是可选的。 • 数字: 对于必需的情况, 表示此类型字符必需的数量。对于可选的情况, 表示如果使用了此类型的字符, 那么需要提供多少个此类字符。 <p>类型如下:</p> <ul style="list-style-type: none"> • U 或 u 表示大写字符。这包括被认为是它们各自语言环境中的字母, 但不是小写。特别地, 这包括大写、首字母大写或不区分大小写的语言 (如中文) 中的任何字母。 • L 或 l 表示小写字符。这包括被认为是它们各自语言环境中的小写形式的所有字符。 • N 或 n 表示数字字符。这包括被认为是其各自语言环境中的数字的任何字符。 • S 或 s 表示特殊字符。与前三个类别均不同的任何字符。它包括所有既不是字母也不是数字的字符。 <p>示例: 字符串 5.2.u1.l1.n1.s1 表示以下密码要求:</p> <ul style="list-style-type: none"> • 至少 5 个字符长 • 必须包含四种类别 (大写、小写、数字和特殊) 中至少两个类别的字符。 • 每个类型一个字符即符合要求。 <p>abC1x 和 Abc2% 之类的密码为合格密码。</p>
暂停 Build Forge 引擎	<p>当设置为“是”时, 系统完成任何当前作业, 然后暂停引擎。设置为“否”则恢复正常操作。</p>
公共主机名	<p>如果设置了此项, 那么系统会将通知模板中 CONSOLEHOST 变量中的服务器主机名替换为此设置的值。</p>
清除检查时间	<p>设置系统检查要清除的作业的频率, 表示形式为两次检查之间的分钟数。</p>

设置	描述
快速报告公共目录	<p>公共报告设计的文件系统位置。</p> <p>在 7.1 中，使用该系统设置来指定公共报告的标准位置。报告设计必须位于该目录中才能进行自动迁移。</p> <p>在较早的发行版中，缺省文件位置 (<code>../reports/public</code>) 是相对于应用程序服务器安装目录的路径，例如: <code><bfinstall>/Apache/tomcat/webapps/quickReport</code>。</p>
快速报告临时目录	<p>在 7.1 中，使用此目录来指定服务层组件所在主机上的标准目录。服务层使用此工作目录来列出已成功迁移到数据库的报告设计。</p> <p>在较早的版本中，此目录用于在将“快速报告”报告设计保存到文件系统上的公共或专用目录之前临时存储这些报告设计。</p>
快速报告用户目录	<p>专用报告设计的文件系统位置。</p> <p>在 7.1 中，使用此系统设置来指定专用报告的标准位置。报告设计必须在此目录中才能自动迁移。</p> <p>在较早的发行版中，专用报告的缺省文件位置 (<code>../reports/users</code>) 是相对于应用程序服务器安装目录的路径，例如: <code><bfinstall>/Apache/tomcat/webapps/quickReport</code>。</p>
重新装入语言包	<p>缺省值为“否”。如果设置为“是”，控制台将在重新启动时重新装入语言包，并将此值复位为“否”。从 V7.0.1 开始不再需要。</p>
复位适配器模板	<p>使用此设置复位适配器模板（以将更改从更新复制到配置中）。要使用此项，请将其值设置为“是”，然后等待一分钟。系统将复位模板，然后将该值恢复设置为“否”。</p>
复位服务器作业计数	<p>使用此设置（是）同时将所有服务器的作业计数（BF_JOBS）复位为零。复位在运行清单检查时间间隔时发生。（缺省时间是每 10 秒钟。）</p> <p>复位所有服务器的 BF_JOBS 后，“复位服务器作业计数”的值还原为“否”（缺省值）。</p>
重新启动报告迁移	<p>缺省值：否。在 7.1 中，如果想要启动迁移而不重新启动服务层组件，请将此值设置为“是”。</p>
运行链链接	<p>控制已启动的链式项目是否还启动了任何连接的适配器链接。</p>
运行队列大小	<p>此值限制系统尝试同时运行的作业数。当队列中的运行数等于或超过此数时，系统将停止从“等待”队列将运行移入“运行”队列，直到作业数低于此值。如果更改了“运行队列大小”，请检查“最大控制台进程数”设置，该设置应比“运行队列大小”至少大 5。</p>
保存启动环境	<p>控制项目手册起始页上“保存环境”复选框的缺省值。当此设置为“是”时，该框缺省选中；否则为缺省不选中。当“保存环境”复选框选中时，您在起始页上对环境变量所作的任何更改将保存到数据库中的环境记录中，这样以后的运行将缺省为这些值。</p>

设置	描述
服务器环境在链之前	确定系统在步骤中设置链式项目的项目环境之前 (Y) 还是之后 (N) 设置步骤的服务器环境。第二次处理的环境中的变量会覆盖第一次处理的环境中的变量。缺省值为 Y，表示链式项目环境已经处理第二次，并覆盖了步骤的服务器环境。
服务器重试数	设置当找到匹配所有选择器条件的服务器时，系统尝试将步骤分配给服务器的次数，在此次数之后，系统将退出并使该步骤失败。
服务器测试频率	<p>与已启用服务器数一起使用，以确定测试和刷新服务器清单数据的频率。缺省值为 120 分钟（2 小时）。值 0 表示不检查服务器。</p> <p>在检查过程中，系统会与所有已启用服务器通信以验证以下事项：</p> <ul style="list-style-type: none"> • 是否仍可连接到服务器 • 服务器的登录信息是否正确 • 服务器的清单数据是否为最新 <p>每分钟至少对一台服务器执行服务器测试。如果服务器数量远远超出此设置，那么测试频率会增加。</p> <p>系统会在一段时间间隔内平均地分配测试。示例：</p> <ul style="list-style-type: none"> • 您具有 120 台服务器，时间间隔设置为 120 分钟。系统会尝试每分钟测试一台服务器（120 台服务器/120 分钟）。 • 您具有 12,000 台服务器，时间间隔设置为 1200 分钟。系统会尝试每分钟测试 10 台服务器。 • 您具有 10 台服务器，时间间隔设置为 120 分钟。系统会每分钟检查一台服务器。在这种情况下，在刷新时间间隔内，每台服务器最多会检查 12 次。 <p>从控制台启动的手工服务器测试优先于这些自动测试。</p> <p>服务器收集器的复杂性可能会影响吞吐量。用于执行许多手工命令以收集数据的收集器可能需要一分钟以上的时间完成。</p>
服务器使用连接超时	<p>设置在创建代理程序连接后，管理控制台在步骤失败前等待连接打开的秒数。</p> <p>设置与代理程序的现有连接的超时值。代理程序设计为每 15 秒与管理控制台通信一次。如果在超时时间段内无任何通信，那么表明代理程序可能已停止，或者可能出现网络通信问题。如果超出该值，那么步骤将失败。</p>
服务层认证 servlet URL	<p>如果已设置，它会覆盖指向服务层认证 servlet、通过编程构造的 URL。</p> <p>如果在使用别名或非缺省端口，将需要使用以下格式更新此设置：<code>http://server:port/rbf-services/AuthServlet</code></p>

设置	描述
SMTP 服务器	<p>设置在发送电子邮件通知时用作 SMTP 服务器的计算机。</p> <p>缺省设置为 localhost。</p>
SSO 远程用户	缺省值是“否”。若为“是”，将允许通过标准 web 服务器认证进行单点登录远程用户连接。
堆叠 BuildForge 环境变量	系统通常将传给链式项目的 BF_ 变量的名称更改为 BF_CALLER_；，此设置确定在链接超过一层深度时，系统是否堆叠构造变量名称。缺省值为“否”。当该设置更改为“是”时，从两层深度的主调项目派生的 BF_TAG 变量将名为 BF_CALLER_CALLER_TAG。
步骤最大重试次数	控制在首次连接尝试失败后，步骤尝试连接到代理程序的次数。如果步骤未在指定的重试次数内实现连接，那么它将失败。
本地存储用户认证	<p>确定系统是否将 LDAP/Active Directory 用户认证信息（以加密形式）保留在高速缓存中。缺省值为“是”。只有在使用 LDAP/Active Directory 认证时，系统才适合使用此选项。如果该设置为“是”，那么系统将以加密形式高速缓存用户认证信息，并且可将这些信息与 _USE_BFCREDS 和 _USE_BFCREDS_DOMAIN 特殊变量（这些变量将用户认证应用于服务器）一起使用。可选择将该值更改为“否”来关闭高速缓存；但是，如果关闭高速缓存，系统就无法将 _USE_BFCREDS 和 _USE_BFCREDS_DOMAIN 特殊变量用于在登录服务器时使用用户凭证。</p>
系统警报电子邮件	<p>系统将向此设置定义的地址发送警报电子邮件消息。</p> <p>缺省值为 root@localhost。支持以下电子邮件地址格式。<i>Username</i> 是电子邮件用户名。<i>Name</i> 是任意字符串，通常是某种形式的给定名称。</p> <pre>username@host.com name username@host.com name "<username@host.com>" name <username@host.com> name "username@host.com"</pre> <p>要发送至多个地址，请使用逗号 (,) 或分号 (;) 分隔这些地址。</p>
系统警报源	<p>当系统发送警报电子邮件消息时，会使用此设置中定义的地址作为发件人。</p> <p>缺省值为 root@localhost。支持以下电子邮件地址格式。<i>Username</i> 是电子邮件用户名。<i>Name</i> 是任意字符串，通常是某种形式的给定名称。</p> <pre>username@host.com name username@host.com name "<username@host.com>" name <username@host.com> name "username@host.com"</pre>
系统范围登录消息	允许您定义要显示在登录表单上面的消息。
系统范围用户消息	允许您定义要显示在每页顶部，紧靠在导航按钮下面的消息。

设置	描述
标记: 日期格式	定义用于在 BF_D 标记变量中显示日期的格式。使用字符 y、m 和 d 作为表示年、月、日的变量以显示所要的格式，并附带要作为分隔符的任何特殊字符。例如，对于日期 2005 年 9 月 21 日： 格式字符串....输出 ymd...050921 m/d/y...09/21/05
标记: 时间格式	定义用于在 BF_T 标记变量中显示日期的格式。作用类似于“标记: 日期”格式的设置，但是此格式使用字符 h、m 和 s 来表示时、分和秒。设置 h:m:s 将生成类似于 12:53:42 的输出。
邮件模板的尾部日志量	当通知模板中使用 TAILNORMALLOG 变量时，该项设置在通知中显示的日志末尾几行的行数。
终止线程	确定当线程化步骤失败后，是否停止同一项目中所有其他活动的线程块。

其他系统设置

有一些配置设置是在系统外进行设置的。

I/O 超时

缺省情况下，I/O 超时为 300 秒。该值可通过创建 BF_SOCKET_SELECT_TIMEOUT 作为操作系统变量并为其提供值（以秒为单位）来进行调整。不应将其设置为 60 秒以下的值。仅在启动时读取该值。您必须停止系统以对其进行更改，然后重新启动。如果值太低，那么系统在大量并行负载（例如，一次启动多个作业）之下可能会看似挂起。缺省值适用于大多数情况。

审计日志记录

审计日志记录提供了命令行界面以管理审计日志记录和查看审计日志。审计日志提供了对系统中对象所做更改的历史记录。这些更改存储在全文索引中。

如果使用审计日志记录，那么对于其跟踪的对象，其日志记录会替换其他情况下将在管理 > 消息中可视的消息传递。

要点: 审计日志记录仅适用于直接在正在运行的系统上执行的操作。它对导入系统的数据不起作用。bfimport 设施直接更新数据库，而不与审计日志记录设施交互。

管理审计策略

审计策略控制执行的审计量。它是由审计策略系统配置设置确定的。

NONE 不会记录任何审计信息。

SECURITY

仅记录与安全相关的信息。示例：访问组创建和用户会话活动。

BASIC

这是缺省设置。如果存在无效的审计策略值，也会使用缺省设置。仅会记录尽可能少的信息，如用户、对象类型、uuid 和操作。

DETAILED

记录有关更改了哪些属性的详细信息，包括旧值和新值。并非所有对象都可以提供该信息。对于那些不能提供该信息的对象，BRIEF 和 DETAILED 策略之间没有区别。“审计最大字符串长度”系统配置设置可用于限制存储的数据量。

“审计最大字符串长度”系统配置设置通过限制字符串数据字段长度上限来控制针对 DETAILED 更改可以记录的信息量。如果审计项超出该字符数，那么将存储 SHA 消息摘要来代替字符串值。对于包含敏感信息的字段（如密码），不论字符串的长度如何，都将使用消息摘要。

显示审计信息

命令行界面允许您指定显示哪些信息以及如何对其格式化。

需要“查看所有审计日志”许可权。缺省情况下，会将其分配给 Security and Build Engineer 访问组。

用法: `bfauditlog [connection-options][command-options][query]`

字面值可执行文件如下所示:

- Windows 系统: `bfauditlog.bat`
- UNIX 和 Linux 系统: `bfauditlog.sh`

连接选项

-H *hostname* | --hostname=*hostname*
连接的远程主机名。缺省值为 `localhost`。

-P *port* | --port=*port*
连接的通信端口。缺省值为 `3966`。

-d *domain* | --domain=*domain*
用于认证的域或区域。缺省值: 无。

-u *login* | --user=*login*
用于认证的用户名。缺省值为 `root`。

-p *password* | --password=*password*
认证用户的密码。缺省值为 `root`。

-E *encoding* | --encoding=*encoding*
要使用的输出字符集编码。如果指定的编码无效，那么会列出所有可用编码，并且退出该程序。请注意，以所需编码显示文本的能力取决于客户机上使用的终端软件的功能。缺省值取决于您的操作系统和 JVM 支持。

命令选项

-h 显示帮助。

-b | --bare

在输出 (bare) 中显示较少信息。确切的效果取决于输出格式 (请参阅 -t)。例如, 在 XML 输出中, 将省略开始标记、结束标记和行计数。在 CSV 中, 将省略行计数和列名称。

-c *auditClass[,auditClass ...]* | --audit.class=*class-list*

指定要在查询中使用的审计类的列表。用逗号分隔多个类名。如果指定多次, 那么会使用指定的所有审计类的并集。如果未指定该选项, 那么会使用用户可指定的所有审计类。使用 -L 选项来列出可用审计类。缺省值是授权用户查看的所有审计类。

-f *field[,field ...]* | --field=*field-list*

指定在输出中包含的字段。用逗号分隔多个字段名称。使用 -L 选项来列出可用字段。

-s *field[+|-][,field[+|-] ...]* | --field=*field-list*

列出要用于对输出进行排序的字段。缺省情况下, 输出将以 audit.stamp 的降序进行存储。+ 标记指定升序。- 标记指定降序。用逗号分隔多个字段名称。缺省值为 -audit.stamp。

-l [*offset*,]*count* | --limit=[*offset*,]*count*

用于对结果编页码。偏移量和计数均以条目数量表示。例如, “-l 100,50”表示在第 100 个条目后显示 50 个条目。不为 bare 的输出格式将在输出中包含匹配条目的总数。

-a *user[,user ...]* | --audit.user=*user-list*

将输出限制为与指定用户相关联的条目。用逗号分隔多个用户名。如果您使用多个 -a 选项, 那么该命令会使用选项中指定的所有用户的并集。用户可由登录标识或 UUID 指定。如果指定了不带有用户自变量的 --audit-user, 那么会使用用于认证的用户。

-t *format* | --format=*format*

用于输出的格式, CSV、XML、JSON 或 TEXT 之一。缺省值为 TEXT。

-L | --list

列出用户的可用审计类。以星号 (*) 标记的审计类可供用户访问。如果与 -c (审计类列表) 一起使用, 那么会显示每个类的可用字段列表。指定该选项时, 其他命令选项无效。具有 ViewAllAudits 许可权的用户可以从任何类查看审计日志。其他用户只能使用 -a 选项来查看其自己的审计日志。

查询

查询是使用通常由 Web 搜索引擎使用的格式的字符串: 规范的空格分隔列表。规范采用格式 '*field:value*' 以搜索指定字段中的值。如果未指定查询, 但是指定了至少一个选项, 那么会显示所有匹配的审计日志。在查询中无法指定审计用户和审计类。使用命令选项来指定审计用户和审计类。

注: 在 Windows 系统上使用双引号, 并在 UNIX 和 Linux 系统上使用单引号。

示例

1. 此示例列出了 Project 和 Step 审计类的所有已知查询字段。

- Windows
\$ bfauditlog.bat -c Project,Step -L
- UNIX 和 Linux

- \$ bfauditlog.sh -c Project,Step -L
- 此示例显示了用于创建项目的 10 篇最新审计日志，使用以逗号分隔的变量输出格式，并且无列标题或行数总计。针对每条记录仅提供该项目的唯一标识和名称。
 - Windows


```
$ bfauditlog.bat -c Project -t CSV -f audit.target,
  sxProjectName --bare -l 10 "audit.type:Create"
```
 - UNIX 和 Linux


```
$ bfauditlog.sh -c Project -t CSV -f audit.target,
  sxProjectName --bare -l 10 'audit.type:Create'
```
 - 此示例检索了使用提供的域、用户名和密码登录后，来自机器“buildforge.example.com”上的安装的 100 篇最新审计日志。
 - Windows


```
$ bfauditlog.bat -H buildforge.example.com
  -d ExampleLDAP -u apiuser -p "My$ecret" -l 100
```
 - UNIX 和 Linux


```
$ bfauditlog.sh -H buildforge.example.com
  -d ExampleLDAP -u apiuser -p 'My$ecret' -l 100
```
 - 此示例检索了根据对连接进行认证的用户的时间区设置，自 2011 年 5 月 23 日起至 2011 年 5 月 27 日记录的审计日志。
 - Windows


```
$ bfauditlog.bat "audit.stamp:[2011/05/23 TO 2011/05/27]"
```
 - UNIX 和 Linux


```
$ bfauditlog.sh 'audit.stamp:[2011/05/23 TO 2011/05/27]'
```
 - 该示例检索了访问组级别从旧值 4、5 或 6 更改为新值 3 的项目的审计日志。请注意，方括号用于表明包含范围，因此 [4 TO 6] 表示值 4 和 6 将匹配。另请注意，旧级别值和新级别值将只能用于搜索审计策略是否设置为“DETAILED”。
 - Windows


```
$ bfauditlog.bat -c Project "ioLevel:[4 TO 6] AND inLevel:3"
```
 - UNIX 和 Linux


```
$ bfauditlog.sh -c Project 'ioLevel:[4 TO 6] AND inLevel:3'
```
 - 此示例检索了访问组级别从旧值 4 或 5 更改为新值 2 或 3 的项目的审计日志。请注意，花括号用于表明排除性范围，因此 {3 TO 6} 表示值 3 和 6 将不匹配。另请注意，旧级别值和新级别值将只能用于搜索审计策略是否设置为“DETAILED”。
 - Windows


```
$ bfauditlog.bat -c Project "ioLevel:{3 TO 6} AND inLevel:(2 OR 3)"
```
 - UNIX 和 Linux


```
$ bfauditlog.sh -c Project 'ioLevel:{3 TO 6} AND inLevel:(2 OR 3)'
```
 - 该示例将查询限制为审计用户 root 和 sv 的结果，并在 CSV 文件中生成输出。它返回 Project 和 Step 类的数据，并包含类、类型、目标、用户和时间戳记的信息：
 - Windows


```
$ bfauditlog.bat -c Project,Step -f audit.class,
  audit.type,audit.target,audit.user,audit.stamp
  -a root,sv -l 10,5 -t CSV
```

- UNIX 和 Linux

```
$ bfauditlog.sh -c Project,Step -f audit.class,
audit.type,audit.target,audit.user,audit.stamp
-a root,sv -l 10,5 -t CSV
```

类

类是审计日志条目的标签。使用 -L 选项来列出可用类。

字段

字段表示分配给每个审计日志记录的信息。

- audit.class: 项目（字符串）
- audit.stamp: 时间戳记（整数）
- audit.stamp.localized: 表示 audit.stamp 整数值的本地化字符串
- audit.type: 更新（字符串）
- audit.target: 项目的 UUID（字符串）
- audit.user: 执行更新的用户的 UUID

审计日志记录中的某些字段将反映其状态（o 表示旧，n 表示新，x 表示扩展）及其数据类型（s 表示字符串，i 表示整数）。示例：

```
soDescription: Old project description (string)
snDescription: New project description (string)
soSelectorUuid: Old selector's UUID (string)
snSelectorUuid: New selector's UUID (string)
ioRunLimit: Old run limit (integer)
inRunLimit: New run limit (integer)
```

消息

系统消息页面显示您可以查看的系统和用户操作的日志。

选择**管理 > 消息**以显示消息列表。这些消息形成操作的审计跟踪。



仅会针对不是通过审计日志记录进行记录的项显示消息。请参阅第 217 页的『审计日志记录』。使用“审计”以外的某种严重性查看消息。

可以通过使用提供的一个或多个控件然后单击**过滤**来过滤该列表：

- 文本框：输入要搜索的字符串。
- 严重性：选择以下某个选项：
 - 全部（缺省值）
 - 错误

- 信息
- 警告
- 审计
- 持续时间：选择以下某个选项：
 - 全部（缺省值）
 - 12 小时
 - 24 小时
 - 2 天
 - 3 天
 - 4 天
 - 5 天

该列表更新时间取决于消息数和过滤器条件。过滤范围越小更新列表的速度越快。请注意：搜索所有严重性和所有时间段内的字符串需要耗用一定的处理时间。

对于每条消息，将显示以下三个字段：

- 时间戳记
- 用户
- 消息

在**管理 > 系统**中提供了以下设置以控制日志大小：

- 自动清除审计日志天数
- 自动清除错误日志天数
- 自动清除信息日志天数
- 自动清除警告日志天数

有关更多信息，请参阅第 208 页的『系统配置设置』。

已翻译的消息

将动态翻译非英语消息。

以非 en_US 语言使用 Build Forge 时，会对消息进行动态翻译。选择**管理 > 消息**以显示消息列表。以每 15 秒 250 批次的速度对消息进行翻译。

如果消息还未翻译，那么“消息”字段将显示“消息还未翻译”。

要点：如果添加了新用户并且其语言与当前设置不同，那么将使用该语言显示消息。不会使用先前语言或新用户的语言来显示现有消息。

预订 RSS 数据订阅源以获取作业状态

现在可以使用 RSS 数据订阅源跟踪和过滤各个作业的状态。Build Forge 的作业 RSS 数据订阅源显示的信息与 Build Forge 管理控制台上显示的服务器状态信息相同。

要预订 RSS 数据订阅源以获取作业状态，请执行以下操作：

1. 在 Build Forge 管理控制台中，选择**作业**。

Web 浏览器将检测 RSS 订阅源并在浏览器地址栏中显示 RSS 图标。

2. 在 RSS 聚合器工具中，装入 Build Forge RSS 数据订阅源。

例如，复制 URL 以将其添加到 RSS 数据订阅源列表中，或者拖放 RSS 图标以将 URL 添加到 RSS 数据订阅源列表。

3. 预订 RSS 数据订阅源以保存 URL 并在发生更新时获得通知。

注:

- 有关装入 URL 和预订 RSS 数据订阅源的信息，请参阅针对您 RSS 聚合器工具的文档。
- 若要使用非英语语言通过 RSS 数据订阅源查看 Build Forge 的作业状态、系统消息或服务器状态，RSS 聚合器工具必须支持 UTF-8 多字节字符编码。
- RSS 订阅源使用时间戳记报告活动。该时间戳记基于针对您的 Build Forge 用户帐户设置的时区。如果浏览器位于另一个时区，那么可将其设置为与您的 Build Forge 帐户相同的时区以获取准确的时间戳记。

过滤 RSS 数据订阅源以获取系统消息

可过滤在 RSS 数据订阅源中显示的消息。为此，可在管理控制台的管理 > 消息页面中使用过滤属性。

例如，将“严重性”属性更改为“审计”或将“最后一次”属性更改为“12 小时”。更改过滤属性时，“消息”页上的 RSS 图标的 URL 将自动更新。

要将更新后的 URL 装入 RSS 聚合器工具，请右键单击“消息”页上的 RSS 图标以复制链接，或者将更新后的 RSS 图标拖放到工具中的 RSS 数据订阅源列表。

注:

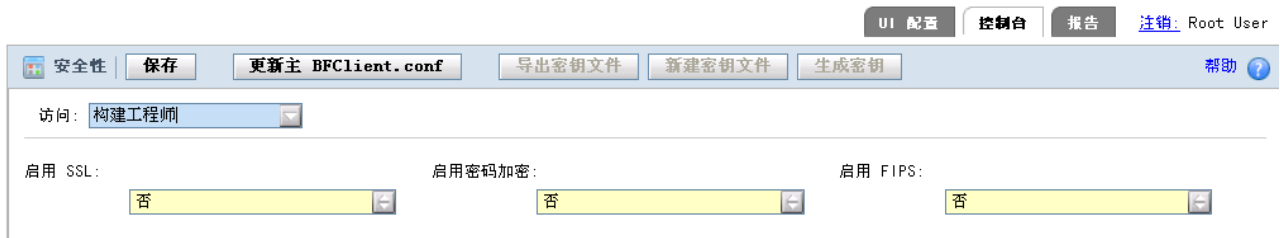
- 有关装入 URL 和预订 RSS 数据订阅源的详细信息，请参考 RSS 聚合器工具的文档。
- 若要使用非英语语言通过 RSS 数据订阅源查看 Build Forge 的系统消息或服务器状态，RSS 聚合器工具必须支持 UTF-8 多字节字符编码。
- RSS 订阅源使用时间戳记报告活动。该时间戳记基于针对您的 Build Forge 用户帐户设置的时区。如果浏览器位于另一个时区，那么可将其设置为与您的 Build Forge 帐户相同的时区以获取准确的时间戳记。

“安全”面板

您可在“安全”面板中启用安全服务:

- **SSL:** 在此面板中启用 SSL 只是在整个系统中启用 SSL 安全功能的一部分。在启用前后还需要完成其他工作。
- **密码加密:** 在此启用密码加密只是在整个系统中启用密码加密的一部分。

要查看“安全”面板，请选择管理 > 安全。



要点：在“安全”面板中只能完成部分设置。要了解更多信息，请参阅第 88 页的『安全性功能』。

启用 SSL

安装先决条件：安装过程中需要指定 SSL 配置使用的两项设置：

- **SSL 端口**，在“Web 和应用程序服务器”面板中指定。该端口必须与下面选择的配置中指定的端口匹配。安装期间和配置时的缺省端口是 8443。Apache Tomcat 上的认证 servlet 在登录时使用该端口对用户登录凭证进行编码或加密。
- **证书**，在“Web 和应用程序服务器”面板中指定。您或者提供了您自己的证书，或者已允许安装程序为您创建自签名证书。该证书存储在缺省密钥库中。密钥库的位置在指定的 SSL 配置中定义。
- 将**启用 SSL** 设置为“是”。其他属性如下所示：
 - **LDAP 出站：**指定通过 LDAP 进行的出站通信所用的配置。缺省值为“缺省 JSSE 出站 SSL”。
 - **引擎到代理程序缺省出站：**指定从引擎组件到代理程序的通信所用的配置。缺省值为“缺省 OpenSSL 出站 SSL”。
 - **服务层入站：**指定服务层组件接受来自 Web 接口组件和引擎组件的通信所用的配置。缺省值为“缺省 JSSE 入站 SSL”。
 - **服务层出站 (JSSE)：**指定引擎组件、Web 接口组件和服务层组件与数据库通信所用的 JSSE 配置。缺省值为“缺省 JSSE 出站 SSL”。
 - **服务层出站 (OpenSSL)：**指定引擎组件、Web 接口组件和服务层组件与数据库通信所用的 OpenSSL 配置。缺省值为“缺省 OpenSSL 出站 SSL”。
- 单击**保存**。
- 单击**更新主 BFClient.conf**。此步骤使用这些属性设置来编辑 BFClient.conf 文件。
- 重新启动 Build Forge。安全通信在系统使用这些设置启动后才生效。

您选择的配置在 **SSL** 面板中定义。

在此面板中启用 SSL 后的要求：

- **证书分发：**证书必须安装在代理程序主机、数据库主机和任何其他在运行的 Build Forge 安装（冗余配置）的密钥库中。
- **代理程序 SSL 启用：**如果要对引擎组件和代理程序之间的通信使用 SSL，每个代理程序都必须配置为使用 SSL。
- **API 客户机启用：**所有 API 客户机都必须配置 SSL 以与服务层组件通信。

启用密码加密

先决条件：

- **SSL 端口**，在“Web 和应用程序服务器”面板中指定。该端口必须与下面选择的配置中指定的端口匹配。安装期间和配置时的缺省端口是 8443。Apache Tomcat 上的认证 servlet 在登录时使用该端口对用户登录凭证进行编码或加密。
- **证书**，在“Web 和应用程序服务器”面板中指定。您或者提供了您自己的证书，或者已允许安装程序为您创建自签名证书。该证书存储在缺省密钥库中。密钥库的位置在指定的 SSL 配置中定义。
- 将**启用密码加密**设置为“是”。
- 单击**保存**。
- 单击**更新主 BFClient.conf**。此步骤使用这些属性设置来编辑 BFClient.conf 文件。
- 重新启动 Build Forge。密码加密在系统使用这些设置启动后才生效。
- 启用密码加密后，在控制台输入的任何新密码都将被加密，包括“服务器认证”密码和在控制台中为用户创建的用户密码。

其他要求:

启用加密后，需要执行以下操作:

- 在所有代理程序上启用加密。导出密钥，并用该密钥更新每个代理程序配置中的服务器认证密码。该密码必须在 BFAgent.conf 中手动更新。
- 对数据库访问启用加密密码。导出密钥，并用该密钥更新 Build Forge 用于登录到数据库的数据库密码。该密码必须在 buildforge.conf 中手动更新。

密钥库面板

密钥库面板包含各个密钥库的配置。在**管理 > 安全 > SSL** 面板上编辑 SSL 配置时，可以选择其中的各个配置使其成为 SSL 配置的一部分。

您可以创建配置，也可使用提供的配置:

- 缺省 JSSE 密钥库
- 缺省 JSSE 信任库
- 缺省 OpenSSL CA 库（认证中心）
- 缺省 OpenSSL 证书库（证书）
- 缺省 OpenSSL 密钥库

每个密钥库都具有以下属性:

名称

访问 定义哪些用户可以编辑或删除此密钥库的访问组。

位置 密钥库文件的位置。缺省密钥库都使用 Build Forge 密钥库的缺省位置: `<bfinstall>/keystore`。如果要 WebSphere Application Server 用作应用程序服务器，而不是提供的 Tomcat 应用程序服务器，请指定绝对路径。

密钥库类型

密钥库必须是下面的一个类型:

- JKS
- PKCS12
- JCEKS
- PEM

密码 指定访问密钥库时必须使用的密码。

验证密码

在此处再次指定密码以进行验证。

密钥库和 WebSphere Application Server

如果使用 WebSphere Application Server, 而不是提供的 Tomcat 应用程序服务器, 那么还有其他需求应用于密钥库配置操作:

- “位置”字段: 必须提供绝对路径而不是相对路径。
- 多个服务组件: 如果安装了多个 Build Forge 服务组件, 那么这些组件要安装在不同的主机上。为每个服务组件配置安全性。“位置”字段中指定的密钥库路径必须与每个服务组件相同。

SSL 面板

SSL 面板包含 SSL 的各个配置。在**管理 > 安全**面板上将**启用 SSL** 选择为“是”时, 可以选择其中的各个配置以使其成为启用 SSL 的一部分。

您可以创建自己的配置, 也可使用提供的配置:

- 缺省 JSSE 入站 SSL
- 缺省 JSSE 出站 SSL
- 缺省 OpenSSL 入站 SSL
- 缺省 OpenSSL 出站 SSL

每个配置都具有以下属性:

名称 此配置的名称。

访问 定义哪些用户可以编辑或删除此密钥库的访问组。

类型 选择 JSSE 或 OpenSSL。

客户机认证

选择下面的一个选项:

- 从不
- 受支持
- 必需

服务器证书别名

输入服务器证书的别名。

客户机证书别名

输入客户机证书的别名。

密钥库配置

选择某个密钥库配置。它们在“密钥库”面板中配置。

信任库配置

选择某个密钥库配置。它们在“密钥库”面板中配置。

握手协议

选择以下项之一:

- SSLv2

- SSLv3
- SSL
- TLSv1
- TLS
- SSL_TLS

密码套件组

选项有“高”、“中”、“低”或“全部”。高强度密码更加安全，但会降低性能。

密码覆盖列表

SSO 面板

SSO 面板包含单点登录（SSO）配置。您可以创建新的配置，也可使用提供的配置：

- 表单 SSO 拦截器
- SPNEGO SSO 拦截器

启用单点登录服务需要进行更多设置。在此面板中无法完成所有配置。请参阅第 89 页的『实施单点登录』。

每个配置都具有以下属性：

名称 SSO 配置的名称。

活动 选择“是”可激活此拦截器。

访问 定义哪些用户可以编辑或删除此配置的用户组。

Java 类

实施此 SSO 配置的 Java 类。提供了以下两个类：

- 表单拦截器：
`com.buildforge.services.server.sso.form.FORMSSOInterceptor`
- SPNEGO 拦截器：
`com.buildforge.services.server.sso.spnego.SPNEGOSOInterceptor`

密钥库类型

密钥库必须是下面的一个类型：

- JKS
- PKCS12
- JCEKS
- PEM

密码 指定访问密钥库时必须使用的密码。

验证密码

在此处再次指定密码以进行验证。

管理许可证

在创建用户时，系统将关于用户的数据存储在数据库中。在用户登录到系统中后，系统会为用户指定一个许可证。

授权用户

已将许可证分配给用户。该用户可以始终登录。登录耗用许可证。

浮动用户

浮动许可证用户仅在有用许可证时才能登录。如果所有许可证均在使用，他们可能无法登录。

令牌许可证

持有多个 Rational 产品许可证的客户可以选择使用令牌许可证。他们购买可用于跨这些产品使用的令牌许可证池。可在使用特定功能部件以及登录时使用令牌许可证。

令牌许可证支持可用于 Floating Server、Floating User、Adaptor 或 QuickReport 许可证。它不可用于固定许可证（授权服务器或授权用户）。如果 Server、Adaptor 和 QuickReport 许可证已检出，那么不会发布这些许可证，直至 Build Forge® 关闭。

令牌许可证支持可用于 Rational Build Forge V7.1.3.4 和更高版本。令牌许可证支持还需要 Rational Common Licensing V8.1.2 或更高版本。

有关令牌许可的更多信息，请联系您的 IBM 销售代表。

如果创建的用户数多于所具有的许可证数，那么新用户将不能登录，直到您购买额外的许可证或使用 root 用户帐户从某些用户除去许可证（或者从系统中删除某些用户）为止。

一个用户只能登录一次。如果您已在一台计算机上登录，然后使用相同的名称从另一台计算机登录，那么原先的会话会变为无效。

注：如果您购买的系统版本低于 V3.8，那么您的许可证模式可能不同于此处描述的模式。如果对于您的许可证有任何问题，请与客户支持联系。

输入新的许可证密钥

要更改您的许可证密钥，请选择**管理 > 系统**，然后在设置列表中找到“许可证服务器”设置。（在“过滤器”框中输入“许可证”可快速显示该设置，而无需对列表进行翻页）。

单击列表中的“许可证服务器”项，这样系统会在内容面板底部显示一个选项卡，其中包含可编辑的**许可证服务器**字段。编辑完该字段后，单击**保存**按钮。

令牌许可证日志

Rational Common Licensing 配置文件 RCL_Log_Config.txt 位于 Build Forge 安装目录中。缺省情况下，该安装目录位于以下位置：

- Windows 系统
C:\Program Files\IBM\Build Forge
- UNIX 和 Linux 系统
/opt/buildforge/Platform

缺省情况下，Rational Common Licensing 日志文件是在安装目录下的 temp/rc1 子目录中创建的。

要将配置文件和日志文件存储在相同位置，请执行以下步骤：

1. 停止 Build Forge
2. 将 RCL_Log_Config.txt 文件移至 `<bfinstall>/token_libs/rc1`。
3. 重新启动 Build Forge。现在，所有令牌日志都将在 `<bfinstall>/token_libs/rc1` 中生成。

管理引擎

本主题描述如何暂停、启动或停止引擎（`bfengine.exe/bfengine.pl`）。

注：如果运行引擎时出现问题，可以在前台运行它，以便查看它所生成的状态和错误消息。在 Windows 上，选择 **开始 > 所有程序 > IBM Rational Build Forge > 启动引擎（前台）** 在前台运行引擎，这样它会在控制台窗口中显示输出。当引擎在前台运行时，可以通过关闭其控制台窗口来停止它。

暂停引擎

如果希望暂时推迟新作业的处理，应暂停该引擎。

注：如果要直接对数据库执行操作（例如备份或复原数据库），首先请停止运行所有 Build Forge 服务和过程。在这种情况下，暂停引擎还不够，这样会导致出现不一致和/或损坏情况。

要暂停该引擎，请选择 **管理 > 系统**，然后找到 **暂停 Build Forge 引擎** 属性（使用过滤器框）。系统将显示用于编辑 **暂停 Build Forge 引擎** 设置的面板。请将该属性更改为 Y，然后单击 **保存配置** 按钮。

系统将暂停引擎，但是当前正在运行的所有项目会继续。您必须等待当前处于活动状态的所有作业完成，这样才能确保未向数据库写入任何数据。

要重新激活引擎，请将属性值重新改为 N，然后再次单击 **保存配置**。

启动和停止引擎

下面几节描述如何在 Windows[®]、Linux[®] 或 Solaris 上启动和停止引擎。

在 Windows 上启动和停止引擎

在 Windows 上：

- 在 **开始 > 程序 > IBM Rational Build Forge 管理控制台** 中，选择以下某项：
 - 启动引擎服务
 - 停止引擎服务

控制面板：也可使用 **管理工具 > 服务控制面板** 来启动或停止 **IBM Rational Build Forge 管理控制台** 服务。

在前台运行：如果在运行引擎时遇到任何问题，请在前台运行引擎以便查看它生成的状态和错误消息：**开始 > 程序 > IBM Rational Build Forge 管理控制台 > 启动引擎（前台）**。当管理控制台运行时，日志输出将显示在控制台窗口中。要在此模式下停止引擎，请关闭其控制台窗口。

在 Linux 或 Solaris 系统上启动和停止引擎

如果已安装 rc 文件，请使用以下命令启动和停止该产品：

```
$ /opt/buildforge/rc/buildforge start
$ /opt/buildforge/rc/buildforge stop
```

如果未安装 rc 文件，请使用以下命令启动该产品：

```
$ /<bfinstall>/Platform/buildforge &
```

通过确定其进程标识然后发出 kill 命令来停止它：

```
$ ps aux | grep buildforge
$ kill ${<PID>}
```

管理数据库

本主题描述在设置管理控制台数据库时（尤其是想要更改缺省配置时）的一些重要问题。

删除数据库日志文件

定期删除数据库日志文件。

系统将数据库调试信息存储在管理控制台安装目录的 db.log 文件中。您应当每月检查该文件的大小，如果需要释放控制台计算机上的空间，请将该文件删除。

错误消息

本主题描述在使用管理控制台时可能会看到的错误消息。

没有活动步骤

当调度条目的“下一个运行”列中显示该消息时，说明关联项目中的所有步骤都已被禁用。显示项目的步骤列表，并通过单击每个已禁用步骤旁的红圈来启用其中的一些步骤。

许可证密钥无效或 Build Forge 许可证密钥已损坏或丢失。

您的许可证密钥已过期，或者对于您所安装的产品版本是无效的。输入新的许可证密钥。请参阅第 228 页的『输入新的许可证密钥』。

需要数据库许可证

```
LicMgr: 5140: A database license is required.
```

如果控制台输出中出现上面的消息（在前台运行控制台或在 Linux® 或 UNIX® 系统上查看控制台日志时可以看到此消息），那么表明您在缺乏 Enterprise 许可证的情况下尝试了使用高级数据库功能。例如，您可能尝试了使用带非 DB2® 数据库的系统。如果要升级许可证，请联系支持。

第 17 章 服务器

本主题描述如何在管理控制台中设置和管理服务器资源。

关于服务器资源

控制台中的服务器资源代表您可运行项目或步骤的主机。

- 服务器包含 **清单**。清单是服务器属性列表。清单是收集器运行时填充的。如果没有为服务器分配收集器，一些属性会自动填入服务器清单中。
- 清单由收集器填充。收集器分配给服务器。收集器既设置属性值，又从服务器的代理程序上收集属性的值。
- **选择器**从清单中读取服务器属性。项目可使用选择器来确定哪个服务器将运行步骤。

作为设置系统的管理员：首先，您应创建服务器。然后创建可分配给服务器的收集器。随后运行收集器以填充服务器清单。完成这些任务后，构建工程师可创建项目，这些项目将使用选择器来确定在哪个服务器上运行项目步骤。

每个服务器资源指向一个已安装了代理程序的主机。当您添加服务器资源时，您要描述管理控制台如何访问以及使用特定主机。

在创建服务器之前，请确保其所依赖的数据对象已存在。为服务器指定以下项目：

项	必需或可选	描述
服务器认证	必需	指定用于该服务器的登录名和密码。
收集器	可选	定义系统从服务器收集的除缺省属性外的其他属性。
环境	可选	指定只要在服务器上运行项目就会指定的环境变量。

注：如愿意，可在管理控制台中为单台物理服务器创建多个服务器对象。这些服务器对象称为**逻辑服务器**。逻辑服务器会被经常使用，以便项目可以访问带有不同属性的相同硬件。示例，两台逻辑服务器可能使用不同的路径或不同的环境：

- 两台具有不同路径的逻辑服务器可在同一个服务器上创建不同的工作目录。您可以区分不同服务器执行的工作，因为所有输出都保存在不同目录中。
- 两台具有不同环境的逻辑服务器可使用不同的起始环境值运行其步骤。

关于“服务器”面板

使用“服务器”面板可指定运行项目的服务器。

要在管理控制台中管理服务器，请单击左侧菜单中的**服务器**。



“详细信息”选项卡

“详细信息”选项卡提供以下字段：

主机 运行 Rational Build Forge 代理程序的主机的名称

收集器 用于服务器的收集器

环境 用于服务器的环境

认证 要使用的服务器认证（登录和密码）

密码加密
确定是否对代理程序密码加密

文件 要和 .get 和 .put 命令一起使用的文件传输的类型

访问 可使用服务器的用户的访问组

启用 SSL
指示是否配置了 Rational Build Forge 系统使用 SSL

最大作业数
应该同时运行的最大作业数

“清单”选项卡

“清单”选项卡显示各种服务器属性。它提供一组缺省属性。可以使用收集器来包含其他属性。

“测试结果”选项卡

“测试结果”选项卡指示“管理控制台”和服务器之间连接的最后一次测试状态。

“注释”选项卡

“注释”选项卡提供一个方便的位置来存储和共享有关服务器的注释。

创建服务器

服务器资源表示您可用来运行项目和步骤的主机。

要创建服务器：

1. 选择左侧菜单中的**服务器**。系统将会在主内容面板底部显示“新建服务器”面板。若已选择了现有服务器，请单击**添加服务器**来擦除面板，以便添加一台新的服务器。
2. 提供服务器详细信息：



- **名称：**为服务器命名。此名称为服务器的 BF_NAME 属性。您在选择器中引用此名称来按名称指定服务器。
- **路径：**指定当服务器创建项目和作业目录时使用的目录，例如 c:/buildforgeprojects。系统创建构建目录时，使用此路径值作为起始点。

提示：系统不会创建服务器路径。在构建尝试访问服务器之前，该路径必须存在。如果该路径不存在，构建将失败。

- **主机：**提供运行代理程序的物理计算机的主机名。如果您要将管理控制台计算机定义为服务器，那么使用值 localhost。（代理程序还必须安装在 Management Console 上。）

注：您可以在主机名中包含端口号；例如 *host_name:port_number*。如果随主机名指定了端口号，它将覆盖“缺省代理程序端口”系统设置定义的端口号。（单击**管理 > 系统 > 缺省代理程序端口**。）

注：请勿在主机名前包含协议。例如，请不要使用 http://。

- **认证：**选择要用于此服务器的服务器认证。
- **访问：**选择可使用此服务器的用户的访问组。
- **收集器：**选择要用于此服务器的收集器。
- **环境：**选择一组环境变量，无论何时使用该服务器运行项目都将应用这组环境变量。这些变量将在所有其他变量之前应用。它们用于设置特定于服务器的参数。
- **文件：**定义可与 .get 和 .put 命令一起使用的该服务器的文件传输类型。可以选择不允许传输（None）、文件读取（.get）、文件写入（.put）或文件读写（.get 和 .put）。
- **最大作业数：**输入应同时运行的最大作业数。缺省值为 3。
- **密码加密配置：**如果想要加密代理程序密码，请选择“启用”。缺省值为“禁用”。
- **启用 SSL：**如果您已将 Build Forge 系统配置为使用 SSL 并且希望此服务器资源通过 SSL 与代理程序进行通信，请选择“是”。缺省值为“否”。

注：如果您选择了“是”但 Build Forge 没有正确配置 SSL，那么该服务器资源将完全不能与代理程序进行通信。

3. 单击**保存**。您的新服务器将在内容面板顶部的服务器列表中显示。要验证是否已正确配置了服务器，请在列表中选择服务器，然后单击**测试连接**。若无法与服务器通信，系统将报错。

测试服务器

您可在服务器上执行一组诊断测试。

要执行诊断测试：

1. 单击**服务器** -> *server_name*。
2. 单击**测试连接**。

将执行服务器连接测试并更新服务器清单。将在**测试结果**选项卡中显示一个进度条，直至刷新完成。

查看**测试结果**选项卡中的结果。单击**查看测试详细信息**以获取其他信息。

如果服务器未通过测试，请尝试下列某个操作：

- 验证用于该服务器的服务器认证中的用户名和密码是否正确。
- 验证使用的主机名是否正确。
- 在服务器上重新安装代理程序，或者验证是否安装了该代理程序。

更改服务器

您可以根据需要编辑服务器的属性。

要点： 如果满足以下条件，那么必须重新启动控制台：

- Host 参数使用代理程序主机的主机名
- 代理程序主机的 IP 地址更改

DNS 中的 IP 地址在托管 Build Forge 服务组件的 JVM 中进行高速缓存。必须重新启动以清除高速缓存，并允许服务组件在新 IP 地址处查找代理程序。

如果您在 Host 参数中使用 IP 地址，那么无需重新启动。代理程序主机 IP 地址更改后，需要将 Host 参数更改为新地址并保存。

启用和禁用服务器

可以临时禁用服务器。如果服务器资源已禁用，作业将不能在上面运行。

要禁用或启用服务器：

1. 单击**服务器**以显示服务器列表。
2. 单击要修改的服务器资源名称。系统显示该服务器的详细信息。
3. 在主面板底部的“详细信息”选项卡中选择**禁用**。
4. 单击**保存服务器**。

限制服务器上的并发作业

使用**最大作业数**服务器属性来指定特定服务器可以同时运行的最大作业数。

关于此任务

系统会限制尝试在任一服务器上运行的进程数量。**最大作业数**属性指定对进程数的限制。该属性的缺省值为 3。

注：服务器上可能还运行有其他程序。系统限制只应用于 Build Forge 作业。

注：内置变量 **BF_RESERVE** 和 **BF_EXCLUSIVE** 控制保留服务器的一个或所有作业插槽。请参阅第 250 页的『预设属性』以获取更多信息。

要为服务器指定非缺省值，请执行以下操作：

过程

1. 单击**服务器**。
2. 选择一个现有服务器，或者创建一个新的服务器。
3. 将**最大作业数**属性设置为需要的值。
4. 单击**保存**。

复位作业计数

将服务器上的作业计数复位为 0

在“服务器”页面中，使用**复位作业计数**可将选定服务器的作业计数（**BF_JOBS**）复位为 0。**BF_JOBS** 是服务器上当前正在运行的步骤或作业的数目；如果该数目设置过低，那么作业会由于作业争用而导致失败。

此选择允许您在作业完成、失败或被取消时未正确复位 **BF_JOBS** 的情况下对它进行复位。

例如，取消多个作业有时会无法复位 **BF_JOBS**。如果 **BF_JOBS** 未复位，它可能会达到“最大作业数”设置的限制，从而导致步骤或作业无法运行。

1. 选择**服务器**以显示服务器列表。
2. 选择服务器。
3. 单击**复位作业计数**。

将所有服务器上的作业计数复位为 0

使用**复位服务器作业计数**页面可同时将所有服务器的作业计数（**BF_JOBS**）复位为 0。**BF_JOBS** 属性是服务器上正在运行的步骤或作业的数目；如果该数目设置过低，那么作业会由于作业争用而导致失败。

复位所有服务器的 **BF_JOBS** 值后，“复位服务器作业计数”的值将还原为“否”（缺省设置）。

例如，当您取消几个作业时，构建系统偶尔会无法复位 **BF_JOBS** 值。如果 **BF_JOBS** 值未复位，它可能会达到缺省“最大作业数”系统设置的限制，从而导致步骤或作业无法运行。

1. 单击**管理 > 系统**来显示系统配置设置的列表。
2. 找到“复位服务器作业计数”。

3. 单击**复位服务器作业计数**。
4. 单击“详细信息”选项卡，然后单击**是**作为值。
5. 单击**保存**。

服务器状态的 RSS 数据订阅源

管理控制台运行服务器状态检查，以验证服务器是否可通过功能性测试，并验证代理程序是否可以登录。“测试结果”选项卡显示了状态检查的结果。每当创建或编辑服务器时，管理控制台会自动检查服务器状态。并且您可以随时（例如在运行项目之前）启动服务器状态检查。

Build Forge 的服务器状态 RSS 数据订阅源显示的信息与 Build Forge 管理控制台中“测试结果”选项卡上的信息相同。

要预订 RSS 数据订阅源以获取服务器状态，请执行以下步骤：

1. 在 Build Forge 管理控制台中，选择**服务器**。

Web 浏览器将检测 RSS 订阅源并在浏览器地址栏中显示 RSS 图标。

2. 在 RSS 聚合器工具中，装入 Build Forge RSS 数据订阅源。

例如，复制 URL 以将其添加到 RSS 数据订阅源列表中，或者拖放 RSS 图标以将 URL 添加到 RSS 数据订阅源列表。

3. 预订 RSS 数据订阅源以保存 URL 并在发生更新时获得通知。

注：

- 有关装入 URL 和预订 RSS 数据订阅源的信息，请参阅针对您 RSS 聚集器工具的文档。
- 若要使用非英语语言通过 RSS 数据订阅源查看 Build Forge 的系统消息或服务器状态，RSS 聚合器工具必须支持 UTF-8 多字节字符编码。
- RSS 订阅源使用时间戳记报告活动。该时间戳记基于针对您的 Build Forge 用户帐户设置的时区。如果浏览器位于另一个时区，那么可将其设置为与您的 Build Forge 帐户相同的时区以获取准确的时间戳记。

服务器认证

本主题描述如何授予要在服务器上运行的项目的访问权。

关于服务器认证

服务器认证用于将登录凭证关联到服务器。

服务器认证将登录名和密码存储为单一的已命名对象，您可以将该对象与一个或多个服务器相关联。“服务器认证”页面可用于创建和编辑服务器认证。

通过管理一组服务器认证，可以对多个服务器使用相同的凭证并全局更新这些凭证。

关于服务器认证面板

使用服务器认证面板可创建和管理服务器认证。

要查看该面板，请选择服务器 > 服务器认证。

UI 配置

控制台

报告

注销: Root User

服务器认证

添加服务器认证

帮助

过滤器

正在显示 1 - 1 (共 1 项)

全部显示

第 1 页 (共 1 页)

名称	用户名
Admin	admin

(新建服务器认证)

保存服务器认证

复制服务器认证

删除服务器认证

详细信息

名称:

访问: 构建工程师

登录:

密码:

已验证:

该面板提供以下字段:

- 名称 用于认证自身的名称
- 访问 可以使用此认证的用户的访问组
- 登录 登录时使用的帐户名称
- 密码 与登录相关联的密码
- 验证密码 再次输入密码

创建服务器认证

使用服务器认证可存储多组服务器的登录信息。

关于此任务

为每个服务器指定一个服务器认证，以便管理控制台能够以适当特权登录到服务器。服务器认证将登录信息与服务器记录分离，使您可以对多个服务器应用相同的登录信息。

要创建服务器认证:

过程

- 在 Build Forge 的左侧面板中，单击服务器 > 服务器认证。

构建系统会在主内容面板顶部显示现有服务器认证的列表，并在底部显示空的“服务器认证详细信息”面板。

提示: 当选择服务器认证时，系统会使用选定服务器的认证信息填充“服务器认证详细信息”面板。要清除面板以创建新认证，请单击添加服务器认证。

- 在名称字段中输入认证名称，这是一个用于在系统中标识服务器认证的逻辑名称。

3. 在**登录**字段中，指定服务器登录名。

注：如果登录名来自域用户，那么必须在该字段中包含域。例如，输入：
MYDOMAIN/joeuser。

4. 在**密码**字段中，输入密码。
5. 在**验证**字段中，重新输入密码。
6. 单击**保存服务器认证**。系统将使用您选择的名称存储新的服务器认证。

结果

构建系统将使用您选择的名称存储新的服务器认证。

覆盖服务器认证

通过使用特殊环境变量，您可以强制服务器使用管理控制台登录凭证，而不是使用为服务器指定的服务器认证。要覆盖指定的认证，请向项目或步骤使用的环境中添加此名为 `_USE_BFCREDS` 的变量，并将值设置为 1。如果将该变量添加到项目环境中，那么构建系统将对项目中的每个步骤使用覆盖设置。

当构建系统尝试运行环境中包含 `_USE_BFCREDS=1` 的步骤时，系统将使用启动了项目的用户的控制台登录凭证，以运行该步骤的命令。

注：如果正在使用 LDAP/Active Directory 认证，那么**本地存储用户认证**系统设置必须设置为“是”（其缺省值），以使 `_USE_BFCREDS` 变量生效。当该设置设为“是”时，系统会以加密形式高速缓存用户认证信息，然后可以访问用户认证信息来与 `_USE_BFCREDS` 一起使用。否则，系统不会存储 LDAP 信息，并且将无法使用该信息。

提示：在 Windows 上，也考虑设置变量 `_USE_BFCREDS_DOMAIN`。将此变量设置为值 1 将包含用户的域。

允许使用受限的服务器认证

使用“执行不可访问的服务器认证”许可权以允许用户通过其不具有访问权的服务器认证在服务器上执行步骤。

作为先决条件，该用户必须已经具有或被授予访问服务器的权限。（要授予访问服务器的权限，请单击**服务器 > 访问权限**。）

您可使用服务器认证来登录和访问服务器。服务器认证与访问组关联（**服务器 > 服务器认证 > 访问权限**）。

可以为服务器创建以下服务器认证：

- Dev/dev 服务器认证。将此服务器认证与服务器的“开发者”访问组关联。
- Qa/qa 服务器认证。将此服务器认证与 QA 访问组关联。
- Prod/prod 服务器认证。将此服务器认证与“构建”访问组关联。

要使仅可以访问 qa/qa 服务器认证的用户能够用 prod/prod 服务器认证来运行步骤，请向 QA 访问组添加“执行不可访问的服务器认证”许可权。

注：如果用户可以访问服务器但是无法通过“执行不可访问的服务器认证”许可权访问服务器认证，那么只有在将环境变量 `_USE_BFCREDS` 设置为覆盖服务器认证时才能运行步骤。

选择器

本主题描述选择器。

关于选择器

选择器选择要运行项目或步骤的服务器资源。

选择器包含一个变量列表。您可以为每个变量指定一个值和一个比较。例如，可以指定属性“`CompilerVersion = 1.1`”以仅选择具有该属性的服务器资源。也可以指定“`CompilerVersion >= 1.1`”以选择版本为 1.1、1.3、2 或 2.0 的服务器资源。

当项目或步骤运行时，为其分配的选择器将确定要运行它的服务器资源。

- 静态选择器使用 `BF_NAME` 变量按名称识别服务器资源。
- 动态选择器使用选择器变量指定的条件从系统的所有服务器资源中选择服务器资源。

关于“选择器”面板

使用“选择器”面板可创建和管理选择器，这些选择器用于选择运行项目或步骤的服务器资源。

要查看该面板，请选择服务器 > 选择器。



“详细信息”选项卡：“详细信息”选项卡提供以下字段：

名称 选择器的名称。

访问 可使用此选择器的用户的访问组。

“快照”选项卡：选择器快照是选择器实例。“快照”选项卡提供快照名称。使用此选项卡可查看或更改快照名称以及有关快照的注释。

选择器设置实践

选择器描述适合项目或步骤的服务器类型。它可按照名称或属性来指定收集器在清单中收集和存储的服务器。它可指定多个属性，包括必需属性和可选属性。

设置选择器有多种可行的方法：

- 按名称选择服务器资源。创建选择器并将其命名为服务器资源的名称。选择器按其 **BF_NAME** 值（系统中使用的唯一名称）来指定服务器资源。在希望指定要运行项目或步骤的服务器资源时使用这些选择器之一。
- 按服务器池选择服务器。您可以将服务器组织到已命名的池中，并为每个池创建选择器。将池名称定义为收集器中的属性（设置值属性）。然后为每个池名称创建选择器。将根据项目或步骤的当前负载来选择其服务器资源。
- 按服务器属性选择服务器。可以根据功能属性（例如可用硬盘空间、操作系统或 CPU 数量）来选择服务器。要实施动态选择，请执行以下操作：
 1. 创建用于收集和指定相应属性的收集器。
 2. 将收集器指定给相应的服务器。
 3. 为代表服务器选项的每个属性或每组属性创建一个选择器。

例如，您可以创建选择器来根据以下条件作出选择：

- 带有包含“Windows®”的操作系统的服务器资源。
- 带有多个 CPU 的服务器资源。
- 运行负载低于指定负载的服务器资源。
- 按嵌套收集器选择服务器资源。使用类型属性“包含”来创建指向其他收集器的收集器。收集器本身可由一组收集器指针组成。例如，您可能会希望为每个服务器创建单独的收集器，以使每个服务器可以具有您指定的某些唯一属性。可以使用“包含”类型来指向实用程序收集器。例如，您可以创建一个名为 **Version** 的收集器，该收集器指定环境中一些关键资源（如 Perl 和 Java）的版本号。

选择器变量类型

选择器用于定义如何在运行时为项目或步骤选择服务器资源。

可在选择器中定义下列变量类型：

- **标准属性：**针对此类型的变量，您可以指定以下内容：
 - **名称：**要使用的属性名称。您可以从自动生成的列表中选择，该列表由所有内置属性和收集器定义的所有设置值属性构成。

注：要通过设置值属性构建选择器，必须首先在一个或多个收集器中定义设置值属性。

- **运算符：**下列比较运算符之一：
 - **EQ** - 等于。该值必须精确匹配。该值可以是数字或字符串。
 - **NE** - 不相等。该值必须与指定的值不相等。该值可以是数字或字符串。
 - **GT** - 大于。
 - **GE** - 大于或等于。
 - **LT** - 小于。
 - **LE** - 小于或等于。

- Contains。
- **值**：运算符用于比较的值。
- **必需**：如果需要选择器才能与此变量匹配，请选择“是”；如果选择器是可选的，请选择“否”。
- **包含**：“包含”类型允许您定义通过嵌套由简单选择器构建的复杂选择器。您可以指定以下内容：
 - **选择器**：选择要包含的选择器。将包含选择器指定的所有属性。您可以通过包含多个简单选择器来构建复杂选择器。

选择器变量比较规则

当系统选择用于项目或步骤的服务器资源时，它会将选择器变量的值与相同名称的清单属性的值进行比较。

除非两个值都与针对数字的以下条件匹配，否则系统将执行字符串比较：

- 如果值以数字开始且仅包含数字和后面至少跟了一个数字的小数点，那么系统将执行数字比较。
 - 5、5.5、0.5、5.0 和 5.5.5 被视为数字。
 - 5.、.5、5..、5..5、5.4.6_05 和 5.6i5 全部被视为字符串
- 包含多个小数点的数字值将导致子版本数字比较，其中系统将比较每个以小数分隔的字段。尽管在正常的数字比较中 5.21 小于 5.3，但在子版本数字比较中 5.21.0 将大于 5.3。

注：对于“包含”运算符，系统将始终执行不区分大小写的字符串比较。

下表显示了如何应用比较规则的示例。

属性名称	清单属性值	运算符	选择器变量值	比较类型	是否匹配？
PerlVersion	v5.8.4	>=	5.2.1	字符串	是
PerlVersion	v5.8.4	>=	v.5.2.1	字符串	是
PerlVersion	v5.8.4	>=	v5.22.1	字符串	是
OS_VERSION	1.15	>=	1.1	数字	是
OS_VERSION	1.10	>=	1.1.0	子版本数字	是
BF_NAME	WinServer1	包含	win	字符串	是
BF_NAME	Server123	=	123	字符串	否

针对符合条件的服务器资源的选择器评估

要为项目或步骤选择服务器资源，系统会使用选择器来评估所有符合条件的服务器资源：

1. 系统将编译选择器中包含所有必需变量的服务器列表。如果没有与必需的选择器条件匹配的服务器资源，那么项目或步骤将失败，而系统将创建一个说明。
2. 如果有多个服务器资源满足必需条件，系统将对每个符合条件的服务器资源进行评级并指定分数，如下所示：
 - 为与之匹配的每个可选变量指定一个点数。如果选择器包含同一个变量的多个副本，那么系统会为每个副本指定一个点数。

- 系统将为带有最低 BF_LOADRATIO 值的服务器资源指定一个点数。
3. 系统将选择收到最多点数的服务器。如果有多个服务器资源都具有最高分数，那么系统将从它们之中进行选择。

您可以在选择器中重复可选变量。这样做会增加与之匹配的服务器的分数。例如，必需的内存为 1 GB，但是强烈希望内存为 2 GB 以上，可以按如下方式设置选择器变量：

- 指定必需的 MEM_TOTAL GE 1GB。
- 指定可选的 MEM_TOTAL GE 2GB 三次。

收集器

关于收集器

收集器定义为服务器资源收集和指定哪些属性。

“服务器”面板的“收集器”部分列出可用的收集器，并允许您创建新的收集器。

要打开“服务器”面板的“收集器”部分，请单击**服务器 > 收集器**。

收集器由一个或多个属性组成。每个属性指定清单中要包含的信息。

注：收集器支持使用动态选择器。要使用收集器，需要 Enterprise Edition。

关于收集器面板

使用“收集器”面板可创建、编辑和删除收集器。要查看该面板，请选择**服务器 > 收集器**。



该面板具有以下字段。

- 名称** 收集器的名称。
- 访问** 可以使用此收集器的用户的访问组。

创建收集器

创建收集器以针对您的项目启用动态服务器资源选择。

过程

1. 选择**服务器 > 收集器**。
2. 为收集器指定名称。
3. 为收集器指定访问权限。
4. 单击**保存**。 面板发生更改，因此您可以输入确定要收集的数据的变量。
5. 创建并保存一个或多个变量。 有关变量属性的信息，请参阅『收集器变量属性类型』。

收集器变量属性类型

收集器变量属性指定如何收集服务器清单的信息。

可以在收集器中定义以下类型的属性：

设置值 这些属性将已命名的静态值指定给服务器。您可以指定属性名和值。

值中可以使用特殊值来获得预定义响应。这些特殊值以下划线（_）字符开头。请参阅第 251 页的『收集器中的特殊属性』。

内置 这些属性返回有关为服务器资源指定的主机的信息。有关内置属性的列表，请参阅第 246 页的『内置属性』。

运行命令

该属性类型指定要运行的系统命令。该属性值设置为从运行命令输出。缺省情况下会使用输出的前 255 个字符。可以使用正则表达式来抽取输出的指定部分。

- **属性：** 属性名称
- **命令：** 要在为服务器资源指定的主机上运行的命令
- **正则表达式：**（可选）用于过滤输出的正则表达式。如果指定，构建系统将尝试使用来自命令的每行输出匹配正则表达式。第一次有一行匹配时，系统将检索值 \$1（PERL 约定）并将它用作该属性的值。正则表达式必须至少包含一组圆括号，这样才能返回值。有关构造 Perl 正则表达式的更多信息，请参阅 Perl 文档。

包含 此属性类型指定收集器列表。您可以嵌套收集器。在创建类型为“包含”的收集器时，请将另一个收集器的名称指定为其值。在构建系统创建或更新清单时，系统将插入来自引用的收集器的属性。

提示： 系统按照收集器变量属性在收集器中的排列顺序应用这些属性，同一个名称且排列在较后位置的属性将覆盖排在前面的属性。请在一个收集器中加入另一个收集器时使用此功能。如果要使用收集器的部分变量属性，而不是全部属性，请覆盖不希望使用的属性。

系统还将自动应用一些属性，如包含服务器的逻辑名的 BF_NAME 属性。这些属性被视为特殊清单属性的组成部分。请参阅第 250 页的『预设属性』。

清单和动态服务器选择

您可以同时使用收集器、清单和选择器在项目或步骤运行时选择服务器资源。

三个不同的数据对象允许系统动态选择服务器：

- 收集器是一个对象，用于定义系统从服务器资源收集或向服务器资源指定的一组属性。系统会在检查服务器资源的属性时运行收集器。收集的属性值存储在清单中。
- 清单是特定服务器的属性列表。它包含运行收集器的结果。
- 选择器是属性和比较的列表，例如 `MEM_TOTAL = 512`。系统可以将选择器的属性与清单比较，以查看服务器是否满足特定选择器的需求。项目和步骤将选择器指定为其属性之一。在项目或步骤运行时，会将选择器与所有定义的服务器资源清单进行比较，以选择要运行的服务器资源。

下面的示例显示如何创建和使用简单选择器：

1. 创建名为 `Mercury` 的服务器资源，并将它与一个代理程序关联。
2. 创建名为 `Mercury` 的选择器。设置该选择器以选择带有 `BF_NAME = Mercury` 的服务器。
3. 创建名为 `Lincoln` 的项目。将 `Mercury` 选择器指定给该项目。

当您运行 `Lincoln` 项目时，系统会选择名为 `Mercury` 的服务器资源。如果该服务器资源不可用，项目将失败。

下面的示例显示如何在—组服务器中设置动态服务器选择：

1. 创建名为 `RAMSIZE` 的收集器。设置该收集器以收集内置属性 `MEM_TOTAL`。
2. 创建与主机关联的服务器资源。将每个主机设置为使用收集器 `RAMSIZE`。
 - `Mercury`，具有 512 MB RAM 的主机
 - `Mars`，具有 1 GB RAM 的主机
 - `Jupiter`，具有 3 GB RAM 的主机
3. 创建名为 `BigRam` 的选择器。设置该选择器以选择标准属性（`property=MEM_TOTAL`、`Operator=GE`（大于或等于）且 `Value=2048`）。`MEM_TOTAL` 以 MB 来表示。该选择器只选择具有 2GB 或更大 RAM 的主机。
4. 创建名为 `SmallRam` 的选择器。设置该选择器以选择标准属性：`property=MEM_TOTAL`，`Operator=GE`（大于或等于）以及 `Value=2048`。该选择器只选择具有 256 MB 或更大 RAM 的主机。
5. 创建两个项目：
 - `HighMaint`：将该项目设置为使用选择器 `BigRam`。
 - `LowMaint`：将该项目设置为使用选择器 `SmallRam`。

当您运行 `HighMaint` 时，系统会选择服务器 `Jupiter`，因为它是满足至少 2 GB RAM 选择器需求的唯一服务器。

当您运行 `LowMaint` 时，系统会选择三个服务器资源中任意一个可用的服务器资源。

如果您稍后为具有 2 GB RAM 的主机添加名为 `Neptune` 的服务器资源，那么项目 `HighMaint` 将再一次运行，并且为该项目选择了 `Neptune` 或 `Jupiter` 中的任何一个。如果 `Jupiter` 因某些原因当机，那么将选择 `Neptune`。因为它是剩余的满足选择器的唯一服务器。

查看清单

要查看服务器清单，请执行以下操作：

1. 单击**服务器**。
2. 单击服务器资源的名称。
3. 单击**清单**选项卡。

您不能直接更改服务器的清单。清单内容由收集器来定义。

清单会按照服务器属性或系统设置确定的时间间隔自动刷新。要在时间间隔之间将其刷新，请单击**队列清单刷新**。

手动刷新清单

要手动刷新清单，请执行以下操作：

1. 单击**服务器**以显示服务器列表。
2. 选择服务器。
3. 单击**队列清单刷新**。

将执行服务器连接测试并更新服务器清单。将在**清单**选项卡中显示一个**刷新清单**进度条，直至刷新完成。

自动服务器清单刷新时间间隔通过**服务器测试频率**系统设置进行设置。在执行队列中的任何自动清单刷新之前，执行队列中的手动清单刷新。

设置服务器清单的更新频率

可以使用系统设置来控制系统向服务器查询其清单属性的频率。

关于此任务

缺省情况下，系统检查“内置”和“设置值”属性的频率会高于检查“运行命令”属性的频率。

您可以在收集器中定义一个名为 `_AGE` 的变量，以控制服务器在该收集器中查询其清单属性的频率。缺省值为 86400，提供每天一次的更新。除非您通过“运行命令”属性检索的值频繁更改，否则请勿设置低于此设置的值。要创建该变量：

过程

1. 单击**服务器 > 收集器**。
2. 选择一个现有收集器，或者创建一个新的收集器。
3. 向收集器添加一个变量。
 - a. 选择类型**设置值**。
 - b. 输入**变量名称** `_AGE`。
 - c. 指定以秒为单位的**值**。
 - d. 单击**保存按钮**。
4. 如有必要，更改多个服务器的收集器属性以与您刚刚编辑的收集器匹配。

静态和动态服务器选择的示例设置

下面的示例显示如何创建和使用简单的静态选择器:

1. 创建名为 `Mercury` 的服务器资源, 并将它与一个代理程序关联。
2. 创建名为 `Mercury` 的选择器。设置该选择器以选择标准属性: `property=BF_NAME`, `Operator=EQ` (等于), `Value=Mercury` 以及“必需”。
3. 创建名为 `Lincoln` 的项目。将 `Mercury` 选择器指定给该项目。

当您运行 `Lincoln` 项目时, 系统会选择名为 `Mercury` 的服务器资源。如果该服务器资源不可用, 项目将失败。

下面的示例显示如何在—组服务器中设置动态服务器选择:

1. 创建名为 `RAMSIZE` 的收集器。设置该收集器以收集内置属性 `MEM_TOTAL`。
2. 创建与主机关联的服务器资源。将每个主机设置为使用收集器 `RAMSIZE`。
 - `Mercury`, 具有 512 MB RAM 的主机
 - `Mars`, 具有 1 GB RAM 的主机
 - `Jupiter`, 具有 3 GB RAM 的主机
3. 创建名为 `BigRam` 的选择器。设置该选择器以选择标准属性: `property=MEM_TOTAL`, `Operator=GE` (大于或等于), `Value=2048` 以及“必需”。`MEM_TOTAL` 以 MB 来表示。该选择器只选择具有 2GB 或更大 RAM 的主机。
4. 创建名为 `SmallRam` 的选择器。设置该选择器以选择标准属性: `property=MEM_TOTAL`, `Operator=GE` (大于或等于) 以及 `Value=2048`。该选择器只选择具有 256 MB 或更大 RAM 的主机。
5. 创建两个项目:
 - `HighMaint`: 将该项目设置为使用选择器 `BigRam`。
 - `LowMaint`: 将该项目设置为使用选择器 `SmallRam`。

当您运行 `HighMaint` 时, 系统会选择服务器 `Jupiter`, 因为它是满足至少 2 GB RAM 选择器需求的唯一服务器。

当您运行 `LowMaint` 时, 系统会选择三个服务器资源中任意一个可用的服务器资源。

如果您稍后为具有 2 GB RAM 的主机添加名为 `Neptune` 的服务器资源, 那么项目 `HighMaint` 将再一次运行, 并且为该项目选择了 `Neptune` 或 `Jupiter` 中的任何一个。如果 `Jupiter` 因某些原因当机, 那么将选择 `Neptune`。因为它是剩余的满足选择器的唯一服务器。

属性引用

以下主题描述影响清单的属性:

- 『内置属性』
- 第 250 页的『预设属性』
- 第 251 页的『收集器中的特殊属性』

内置属性

管理控制台从服务器收集内置属性, 然后将值指定给服务器清单。

内置属性

构建系统中的以下不同数据对象使用内置属性：

- **选择器**可以将内置属性用作选择器变量，以将服务器与这些属性中的某些值匹配。
- **收集器**使用内置属性从服务器收集数据。
- **清单**在收集到内置属性时存储其值。

内置属性不会自动添加。您必须将内置属性添加到收集器，使属性可以显示在清单中。此表列出并描述了内置属性。

注：属性可用性根据平台而异。

表 8. 收集器和清单的内置属性

属性	描述
CPU_ARCH	返回值是体系结构名称的label，如下所示： <ul style="list-style-type: none">• HP-PA: HP 精度体系结构• IA-64: Intel Itanium• MVS: IBM S/390• PPC: PowerPC• PPC-64: PowerPC 64• SPARC: Sun SPARC• X86: 由 Intel、AMD、Cyrix 以及其他公司使用的符合 x86 的体系结构
CPU_LOAD（仅对于 Windows）	用容量百分比（0 到 100 之间）来表示 CPU 负荷（或 CPU 使用率）。
CPU_LOAD1	返回的值报告过去一分钟内正在运行或等待运行的进程的平均数量（平均负荷）。 CPU_LOAD1 是对 CPU 活动的一种测量。空闲的计算机的负荷数为 0。每个正在使用或等待使用 CPU 的进程代表 1 个负荷数。 注： 在 Windows 上，每个进程（无论是否处于活动状态）都会增加负荷数。此外，此信息由 bfdispatch 进程收集，并使用共享内存段发布到代理程序。如果用于连接到代理程序的用户凭证没有特权，那么这些统计信息将不可用。
CPU_LOAD5	返回的值报告过去 5 分钟内等待运行的进程的平均数量（平均负荷）。 CPU_LOAD5 是对 CPU 活动的一种测量。空闲的计算机的负荷数为 0。每个正在使用或等待使用 CPU 的进程代表 1 个负荷数。 注： 在 Windows 上，每个进程（无论是否处于活动状态）都会增加负荷数。此外，此信息由 bfdispatch 进程收集，并使用共享内存段发布到代理程序。如果用于连接到代理程序的用户凭证没有特权，那么这些统计信息将不可用。
CPU_LOAD15	过去 15 分钟内等待运行的进程的平均数量（平均负荷）。 CPU_LOAD15 是对 CPU 活动的一种测量。空闲的计算机的负荷数为 0。每个正在使用或等待使用 CPU 的进程代表 1 个负荷数。 注： 在 Windows 上，每个进程（无论是否处于活动状态）都会增加负荷数。此外，此信息由 bfdispatch 进程收集，并使用共享内存段发布到代理程序。如果用于连接到代理程序的用户凭证没有特权，那么这些统计信息将不可用。
CPU_MHZ	该属性报告了处理器速度（以兆赫为单位）。必须满足该属性的某些条件才能填写成功： <ul style="list-style-type: none">• 在 Linux 中：必须启用频率定标。• 在 Windows 中：必须存在 ~MHZ 注册表条目并且条目已填写。• 对于 x86 和 x86-64 处理器：内联组合件必须可工作。

表 8. 收集器和清单的内置属性 (续)

属性	描述
CPU_MANUFACTURER	<p>该属性返回处理器制造商的公司名称。如果没有直接提供信息，那么将根据体系结构来假定名称。如果可用的处理器信息不足，将不返回值。支持以下值：</p> <ul style="list-style-type: none"> • AMD: 表示 AMD x86 和 AMD64 处理器 • Cyrix: 表示 Cyrix x86 兼容处理器 • DEC: 对于 Alpha 和 VAX • HP: Hewlett-Packard 精度体系结构 • IBM: IBM S/390 和 PowerPC G5 • Intel: Intel x86 (包括 Intel64), IA-64 Itanium • Motorola: PowerPC G4 • NexGen: x86 兼容处理器 • National: National Semiconductor x86 兼容处理器 • Rise: Rise x86 兼容处理器 • Sis: Sis x86 兼容处理器 • Sun: Sun Microsystems SPARC • TransMeta: TransMeta x86 兼容处理器 • UMC: UMC x86 兼容处理器 • VIA: VIA Technologies x86 兼容处理器
CPU_MODEL	<p>该属性返回特定于制造商的 CPU 型号。值报告如下：</p> <ul style="list-style-type: none"> • x86 体系结构 <ul style="list-style-type: none"> – 386 – 486 – 586 – 686 – X86_64 • PowerPC 体系结构 <ul style="list-style-type: none"> – 6xx – POWER – RS64 – G3 – G4 – G5 – Cell
CPU_SERIAL	<p>该属性返回 CPU 或计算机的序列号。当前该功能仅限于以下体系结构：</p> <ul style="list-style-type: none"> • x86: 仅 Intel 或 Transmeta 序列号。注：大多数 x86 处理器将不报告序列号。这些情况下，不返回值。 • MacOS/X: 将从 I/O 注册表中检索分配的序列号。要报告该属性，它必须能够找到 CoreFoundation 和 IOKit 框架。

表 8. 收集器和清单的内置属性 (续)

属性	描述
DISK_FREE	<p>对于 UNIX 和 Linux, 该属性返回服务器“路径”属性指定的文件系统上的可用空间数量 (MB)。</p> <p>对于 Windows, 该属性指由服务器“路径”属性指定的驱动器上的可用磁盘空间 (MB)。</p> <p>例如, 将可用空间为 4 GB 的磁盘报告为 4096 MB。</p>
DISK_TOTAL	该属性返回总可用磁盘空间。将针对代理程序的基本路径报告该值, 这可能具有小于整个剩余磁盘或分区的不同分配值。磁盘空间管理在各个操作系统之间大不相同。
MEM_LOAD (仅对于 UNIX/Linux)	对于 UNIX 和 Linux, 当前正在使用的 RAM 或系统内存量用实际总内存的百分比 (0 到 100 之间) 来表示。
MEM_FREE	该属性返回 RAM 或系统内存的数量 (MB)。例如, 将 2 GB 可用 RAM 报告为 2048 MB。
MEM_PAGESIZE	该属性返回 RAM 或系统内存页面大小 (MB)。该图表示主机系统的标准页大小。例如, 将页面大小为 4 KB 的主机系统报告为 4096 MB。
MEM_TOTAL	<p>该属性报告 RAM 或系统内存总量 (MB)。</p> <p>例如, 将带有 2 GB RAM 的计算机报告为 2048 MB。</p>
NET_FQDN	该属性返回正在运行代理程序的计算机的标准域名 (FQDN)。该 FQDN 基于代理程序用于通信的地址进行报告。返回的地址可以基于 IPv4 或 IPv6, 这取决于实际使用的地址。另请参阅 NET_IPV、NET_IPV4 和 NET_IPV6。
NET_HWADDR	该属性返回 NET_IFACE 中所报告的接口的硬件地址。
NET_IFACE	<p>该属性返回代理程序用于通信的接口名称。</p> <ul style="list-style-type: none"> 在 Windows 中, 报告的名称为 ipconfig 命令返回的名称, 例如 Intel® PRO/100 VE Network Connection - Packet Scheduler Miniport 在其他操作系统中, 报告的名称为 ifconfig 返回的名称, 例如 en0 或 eth0 或 OSA1。
NET_IPV	该属性返回用于与代理程序通信的 IP 连接类型, 对于 IPv4 为 4, 对于 IPv6 为 6。
NET_IPV4	该属性返回代理程序用于通信的 IPv4 地址。对于 IPv6 连接, 如果代理程序能够识别同一接口的 IPv4 地址, 那么将报告该地址。
NET_IPV6	该属性返回代理程序用于通信的 IPv6 地址。
NET_SPEED (仅对于 Windows)	该属性返回接口速度 (以 Mb/秒为单位), 例如对于千兆以太网为 1000。
NUM_CPU	该属性返回计算机上的 CPU 数量。
OS_HOSTID	该属性返回 gethostid() 系统调用的结果。通常该结果不是非常有参考性, 除非系统管理员将 /etc/hostid 设置为参考值。
OS_SYSNAME	该属性返回服务器的操作系统名称。示例: Microsoft Windows XP、AIX 和 Macintosh OS。
OS_RELEASE	<p>该属性返回服务器的操作系统发行版级别。</p> <p>例如, 如果服务器操作系统是 Microsoft XP V5.1.2600, 那么该返回值是 5。</p>
OS_VERSION	<p>该属性返回服务器的操作系统版本。</p> <p>例如, 如果服务器操作系统为 Microsoft XP V5.1.2600, 那么该返回值是 1。</p>
WIN_SERVICEPACK (仅对于 Windows)	该属性返回服务器上安装的 Windows service pack 的版本号。例如, 对于 Service Pack 2, 该值为 2。

预设属性

一些清单属性会自动设置。

下列清单属性将自动设置。与内置属性不同，这些属性不必添加到收集器中以进行填充。

表描述中标记为**选择器**的属性可在选择器中使用。其他属性只负责提供清单中的信息。

表 9. 自动设置清单属性

属性	描述
BF_AGENT_VERSION	选择器 。服务器上安装的代理程序的版本号。
BF_EXCLUSIVE	选择器 。它是不具有任何运算符或值的标志。如果选择器包含此属性，那么在作业的持续时间内在选定服务器上保留所有槽。 如果作业中的某个步骤指定了要运行的不同服务器，那么在这个不同的服务器运行该步骤时，当前服务器上的所有当前槽都将继续保留。
BF_JOBS	选择器 。服务器资源同时运行的作业（步骤）数。该值在每次控制台向服务器指定一个步骤时更新，独立于其他清单属性更新。
BF_LAST_REFRESH	清单中内置属性的上次更新时间。该值按照 UNIX [®] 时间戳记来报告：自 1970 年 1 月 1 日开始的秒数。
BF_LASTJOBS	上次刷新清单时服务器正在运行的作业数。
BF_LAST_UPDATE	清单运行命令属性的上次更新时间。该值按照 UNIX [®] 时间戳记来报告，表示为自 1970 年 1 月 1 日开始的秒数。
BF_LOADRATIO	选择器 。作为比率报告的计算值：作业数（BF_JOBS）除以服务器所允许的最大作业数量（“最大作业数”设置）。如果服务器有 1 个作业在运行，而最大作业数量为 4，那么负载率为 0.25。
BF_NAME	选择器 。指定要用于运行的服务器资源。该值是服务器资源的名称。BF_NAME 属性不会在清单列表中显示。
BF_RESERVE	选择器 。它是不具有任何运算符或值的标志。如果选择器包含此属性，那么在作业的持续时间在选定服务器上保留一个槽。 <ul style="list-style-type: none">如果作业中的某个步骤指定了要运行的不同服务器，那么在这个不同的服务器运行该步骤时，选定服务器上的槽将继续保留。如果某个步骤明确指定了选定服务器，那么服务器会保留的槽用于该步骤。 该标志可帮助项目避免当一个或多个项目步骤在其他服务器上运行时由于丢失它们的槽而导致延时。

收集器中的特殊属性

当在收集器中命名某些特殊设置的值属性时，这些属性将导致系统中发生特定行为。这些属性以下划线字符开头。构建系统使用这些属性的值将行为应用于从收集器获取这些属性的服务器。

注：不可创建以字符串“BF_”开头的属性，因为这些是保留供系统使用的名称。

表 10. 收集器和清单的特殊设置值属性

属性	描述
_AGE	该属性定义应刷新清单的频率（单位为秒）。缺省值为 86400，表示每天刷新一次。值 3600 将使系统每小时更新清单一次。

使用快照新建选择器实例

对选择器创建快照来快速新建您要更改或修改的选择器实例。

选择器快照概述

查看这些主题可以了解选择器快照和理解如何使用选择器快照。

选择器快照用例

下面的示例描述了选择器快照的一些常见用例：


- 对选择器创建快照以对选择器配置作出更改，或者在继续使用现有选择器运行作业时执行新工具或脚本的测试。
- 将选择器快照作为临时备份或正式归档的一部分来存储。
- 对选择器创建快照以捕获与里程碑对应的时间点选择器配置，比如外部或内部发行版。

选择器快照概念和术语

快照引入了关于使用选择器的一些新概念和术语。


选择器快照：快照是现有选择器的新实例。需要紧记关于快照的以下几个要点：

- 快照是单独的选择器对象。更改快照集合中的一个快照并不会影响集合中的其他快照。
- 快照不是副本。如果您对与选择器关联的对象创建快照，快照将创建该对象的单独实例。副本用于映射对象之间的关系，它不会创建新的对象。
- 快照不是选择器的修订版：
 - 快照不支持比较两个选择器快照之间的更改。
 - 不会像在源控制系统中那样用版本号来跟踪或识别对选择器快照的更改。但是，可以使用包含版本号（例如，7.5.0、3.4.01）的快照命名方案将选择器快照与里程碑关联。


快照集：快照集是作为一个基本快照的后代的所有选择器快照的集合。该集合至少包括基本快照或父代快照及其子代快照。在 UI 中，选择器名称旁边的“快照”图标  表示该选择器存在快照集。

Base Snapshot: 最初，所有选择器都有一个快照名称 Base Snapshot。您可以将 Base Snapshot 更改为其他名称。基本快照是快照集中的父代。

缺省选择器快照：缺省选择器快照是当前工作选择器。一个集合中只能有一个缺省快照。如果未指定缺省快照，基本快照便是缺省快照。

- 在控制台中，缺省快照显示在选择器列表的顶级。选择**服务器 > 选择器**以显示选择器列表。
- 当您选择带有快照的选择器时，会使用缺省选择器快照，除非您选择了列表框中的其他选择器快照。
- 要访问和使用选择器快照集中的其他快照，必须单击“快照” 图标。

选择器快照视图

选择“快照”图标  以显示“快照”视图。“快照”视图显示集合中快照的层次结构：

- 基本快照位于顶级并且具有名称 Base Snapshot（如果没有为它指定唯一的名称）。
- 所有选择器快照都是基本快照的子代。同一个基本快照的子代缩进在“快照”列的同一级别。
- 通过子代快照创建的选择器快照将成为该子代快照的子代，并缩进在“快照”列中的下一级别。

选择器快照计划

复审选择缺省选择器快照和命名选择器快照的一些最佳实践。

• 在集合中选择缺省快照的策略

UI 仅识别快照集的一个缺省或当前选择器快照。请使用一致的策略来选择缺省快照：

- 使用基本快照作为缺省快照

将快照用作备份。仅对基本快照作出更改。不要对已备份的选择器快照作出更改。

- 使用最新快照作为缺省快照

您每次创建新选择器快照时，都将它创建为新的缺省选择器快照。不要对基本选择器快照或较早的选择器快照作出更改。

• 确定集合的快照命名方式

选择器快照名称在选择器快照集内必须是唯一的。

使用下面的条件可帮助您创建选择器快照名称：

- 名称应该是描述性的：它应指示快照的用法或用途。
- 命名方案应遵循已定义的标准。可以使用“快照”选项卡上的“注释”框描述命名方案。




• 使用集合的单个选择器名称

创建选择器快照后，您可以选择更改选择器的名称。如果您更改了选择器名称，将会为每个选择器快照更新该名称。

从现有选择器或选择器快照创建选择器快照

创建选择器快照会创建选择器的一个新实例。快照不是副本，而是新的选择器。

提示：要创建快照和更改缺省快照，用户必须具有所需的许可权。请参阅第 307 页的『验证和编辑访问组的快照许可权』。

- 1. 单击您要创建快照的选择器或选择器快照旁边的**编辑**图标：
 - 要对缺省选择器快照创建快照，在选择器列表（**服务器 > 选择器**）中，单击顶级快照旁边的“编辑”图标 。
 - 要对非缺省选择器快照创建快照，请单击“快照”图标 。快照视图会在集合中显示选择器快照。单击非缺省选择器快照旁边的“编辑”图标 。
- 2. 单击**新建快照**。
- 3. 在“快照”选项卡的**名称**字段中，输入快照名称。该名称将被指定给您使用该选择器创建快照的所有对象。

该名称在选择器快照集内必须是唯一的。

- 4. 选择要在创建选择器快照时创建快照的 **Build Forge** 对象。下表中描述了您可以选择的对象。

对象	描述
缺省	在 UI 中，缺省快照显示在选择器列表的顶级。 选择 服务器 > 选择器 以显示选择器列表。
包括下列选择器	如果选择器使用 包含 属性类型来包括其他选择器，将为那些选择器创建快照。 注： 包含变量类型将替换上一版本中提供的 <code>.include</code> 功能。



- 5. 单击**保存**以保存选择器快照。

更改缺省选择器快照

缺省选择器快照是选择器快照集中的顶级快照，它显示在选择器列表（**服务器 > 选择器**）中。

提示：要创建快照和更改缺省快照，用户必须具有所需的许可权。请参阅第 307 页的『验证和编辑访问组的快照许可权』。

要更改缺省选择器快照，请编辑您要作为新缺省快照的快照定义：

- 1. 选择**服务器 > 选择器**。
- 2. 在选择器列表中，单击缺省选择器快照的**快照**图标 。
- 3. 在快照列表中，单击要作为新的缺省快照的选择器快照的**编辑**图标 。

- 单击**设为缺省值**。
- 要点：**在弹出窗口中，选择“**确定**”或“**取消**”。



确定	更新引用： 如果任何对象引用先前的缺省选择器，请将这些对象更新为使用新的缺省选择器。
取消	不更新引用： 对于引用先前缺省值的任意对象，不将引用更新为新的缺省选择器快照。

更改选择器快照的快照名称

您可以更改选择器快照的快照名称，以及您在创建选择器快照时选择创建快照的对象快照名称。

对于基本快照（仅对于单个选择器快照，或对于所有当前和未来选择器快照），可以使用此选项将其缺省名称“**Base Snapshot**”更改为其他快照名称。

要更改快照名称，请执行以下操作：

- 选择**服务器 > 选择器**。
- 在选择器列表中，单击缺省选择器快照的**快照**图标 。
- 在快照列表中，单击选择器快照的**编辑**图标 。
- 选择**快照**选项卡。
- 在**名称**中，输入新的名称。
- 可选：**在**注释**中，输入注释。
- 要点：**在弹出窗口中，选择“**确定**”或“**取消**”。

确定	更改选择器快照名称和其他快照对象名称： 对于您在创建选择器快照时选择创建快照的对象，更改这些对象的名称以及选择器快照的称。 对于 Base Snapshot: 更改所有当前选择器快照和所有未来选择器快照的名称 Base Snapshot 。
取消	更改选择器快照名称，但不更改其他快照对象名称： 对于您在创建选择器快照时选择创建快照的对象，不更改这些对象的名称。仅更改选择器快照名称。 对于 Base Snapshot: 保留所有当前选择器快照和所有未来选择器快照的名称 Base Snapshot 。

访问和查看选择器快照集中的快照

对选择器创建快照来快速新建您要更改或修改的选择器实例。

创建选择器快照会创建一个快照集，它最少包含两个环境：基本选择器和新选择器快照。



要查看快照集中的所有选择器快照：

- 选择**服务器 > 选择器**。

选择器列表会显示选择器和选择器快照的列表。顶级快照是缺省的选择器快照。

2. 单击**快照**图标  以显示快照集中的选择器快照。

在“快照”视图中，您可以：



- 创建新的选择器快照。要开始操作，请单击**编辑**图标 。
- 更改选择器的缺省快照。单击**编辑**图标 ，然后单击**设置为缺省值**。
- 就象对标准选择器一样，编辑选择器快照定义。

删除选择器快照

可使用“删除”选项来删除选择器快照。

如果选择器正在被另一个对象使用，那么不能删除该选择器。例如，如果某个选择器包含于另一个选择器或者正被项目、步骤或调度使用，那么不能删除该选择器。

要删除选择器快照：

1. 选择**服务器 > 选择器**。
2. 在选择器列表中，单击基本快照的**快照**图标 。
- 快照视图会在集合中显示选择器快照。
3. 单击要删除的选择器快照旁边的**编辑**图标 。
4. 单击**删除**。

第 18 章 使用环境

本部分描述如何设置和管理环境。

关于环境

环境是一组命名的变量。

一旦定义，环境就可用于执行以下操作：

- 设置由作业中的步骤使用的变量。环境可以指定给服务器、项目和步骤。执行作业时，正在运行的步骤从以下三个环境中继承变量值：
 - 与运行作业的服务器关联的服务器环境
 - 与定义步骤的项目关联的项目环境
 - 与步骤显式关联的步骤环境
- 设置将由已调度作业使用的变量。针对已调度作业设置的环境将替换针对项目指定的环境。
- 设置要由适配器使用的变量。环境可以指定给适配器链接。它由项目的初始适配器步骤使用。

环境继承

在系统运行步骤之前，会创建步骤环境。步骤环境由适用于步骤的所有变量组成。这些变量依次继承自服务器环境、项目环境和步骤环境。下面提供了一些基本的例子。

1. 服务器环境：服务器环境变量将复制到步骤环境。
2. 项目环境：项目环境变量将应用于步骤环境。如果项目环境包含的变量与服务器环境中的变量相同，那么根据两个变量定义中的“变量操作”更新该值。
3. 步骤环境：步骤环境变量将应用于步骤环境。如果步骤环境变量的名称与继承自服务器和项目环境的变量名称相同，那么根据“变量操作”更新该值。

变量的变量操作直接影响值在继承时的应用方式。例如：

- 例 1：在变量操作为 Set 时通过继承覆盖值
 - Server environment: X = 1, action: Set
 - Project environment: X = 100, action: Set
 - Step environment: X = 3, action: Set
 - Final value during step execution: X = 3

变量 X 设置为 1，然后设置为 100，再设置为 3。每次应用新值时，变量操作 Set 都会替换该变量值。

- 例 2：由于变量操作 Set if Not Set 而继承值
 - Server environment: Y = 1, action: Set
 - Project environment: Y = 100, action: Set
 - Step environment: Y = 3, action: **Set if Not Set**
 - Final value during step execution: x = 100

变量 Y 设置为 1，然后，由于在服务器和项目环境中对 Y 执行 Set 操作，该值设置为 100。在步骤环境中，由于 Y 使用变量操作 **Set if Not Set**，所以它继承项目环境中设置的值。

继承的特例

下列情况会影响继承。

内联项目

通过在步骤的**内联**属性中指定一个项目，步骤将内联到项目中。步骤内联到项目中时，不会使用被调项目的服务器环境和项目环境。继承顺序如下：

1. 主调步骤的服务器环境。
2. 主调步骤的项目环境。
3. 主调步骤的步骤环境。
4. 被调项目中每个步骤的步骤环境（如果已指定）。

链式项目

项目或步骤可以将项目指定为“通过链”或“失败链”。如果以这种方式调用项目，那么项目在其自己的环境中运行。此外，它有权访问来自主调项目或主调步骤的所有变量。这些变量使用 **BF_CALLER_** 前缀被复制到新名称中。示例：
主调项目或步骤中的 **BF_NAME** 变量在被调项目和步骤中作为 **BF_CALLER_BF_NAME** 变量提供。

已调度作业

当根据**调度**中的某个调度执行项目时，可以选择对项目应用不同于缺省设置的环境。指定环境后，**环境**选项卡可用于设置该环境中变量的值。这些变量根据其在**项目**中属性设置显示在**环境**选项卡上。它们遵循的规则与非调度作业启动显示时相同。

覆盖继承顺序

使用系统设置**最后应用服务器环境**覆盖继承顺序。如果其值为“是”，那么继承顺序设置如下：

1. 项目环境
2. 步骤环境
3. 服务器环境

启动作业时更改项目变量

当用户启动作业时，可以更改项目变量，以覆盖在项目环境中设置的变量值。

用户启动作业时，来自项目环境的变量显示在作业启动页。用户可以根据每个变量的“在项目中”属性来更改显示的值。

在作业启动时所作的更改同样应遵循项目环境中所定义变量的继承规则。

例如：

1. 可以将 JavaEnv 环境定义为具有初始值的变量（**JavaVersion = 1.4**）。
2. 可以将 MyBuild 项目定义为使用 JavaEnv 环境。
3. 可以启动运行 MyBuild 项目的作业。在“作业启动”面板上，可以将 **JavaVersion** 的值更改为 1.5。

结果如下:

- 不覆盖项目环境 (JavaEnv) 的步骤继承修改后的 JavaVersion 值 1.5。
- 显式地将 JavaEnv 环境用作步骤环境的步骤使用在项目环境中定义的 JavaVersion 值: 1.4。

关于变量

变量在环境中定义。

除了值以外, 变量还有其他用于在解释变量时控制变量行为的属性。



要新建变量, 请执行以下操作:

1. 在左侧菜单中单击环境。
2. 单击添加环境变量。
3. 为变量指定名称。
4. 根据需要为变量指定其他属性。请参阅第 261 页的『环境属性』以获取更多信息。
5. 单击保存变量。

创建完成后, 可单击该变量进行编辑。

步骤中的变量解释

在步骤命令或环境变量定义中可以使用 UNIX[®] 样式或 Windows[®] 样式的变量语法。

系统会使用预处理器将 UNIX 样式 (\$VAR) 或 Windows 样式 (%VAR%) 的语法解释为适用于运行步骤的服务器的正确格式。预解析使步骤能够在基于 Windows 或 UNIX 的服务器上运行。

示例:

- 下面两个赋值语句在步骤中的作用是等同的:

```
echo %fooVar%      # Windows syntax
echo $fooVar       # UNIX or Linux syntax
```

- 不会对变量赋值进行预处理。因此，请避免在命令行中使用直接赋值，在无操作系统约束而选择服务器的情况下尤其如此。在环境中使用变量。

```
set fooVar=100      # Windows
fooVar=200          # UNIX or Linux syntax
```

变量的语法分析方式：

1. 预解析器对变量赋值等式进行求值。特殊字符（\$、%、{、}、"、'）将被吸收，除非使用反斜杠字符进行转义。如果预解析关闭，那么所有字符都被接受。
2. 变量赋值等式的两端都由目标环境进行求值。
3. 将执行已求值的变量赋值等式。

预解析器、Windows 环境以及各种 UNIX 和 Linux shell 对特殊字符的解释有所不同。请谨慎使用特殊字符和反斜杠转义字符。

未声明变量的解析

如果在步骤中调用变量，但变量未定义，返回的值取决于预解析功能是处于打开状态（缺省值）还是关闭状态。预先进行的语法分析行为可通过编辑 `bfagent.conf` 文件中的 `no_preparse` 命令或 `_NO_PREPARSE_COMMAND` 环境变量来设置。请参阅第 270 页的『触发器变量引用』。

变量格式	返回的值 - 预解析开启（缺省值）	返回的值 - 预解析关闭
echo %foo%	foo	Windows: %foo% UNIX 或 Linux: 空
echo \$foo	foo	Windows: \$foo UNIX 或 Linux: 空
echo \${foo}	foo	Windows: \${foo} UNIX 或 Linux: 空
echo \$[foo]	空字符串	Windows: \$[foo] UNIX 或 Linux: 系统错误

关于“环境”面板

“环境”面板用于为项目创建和管理环境。要查看该面板，请选择**环境**。



在“环境”面板中，还可以单击环境以显示和编辑其变量。

“详细信息”选项卡

环境属性

指定以下环境属性:

名称 环境的名称。

访问 可使用此环境的用户的访问组。

变量属性

在“环境”面板中, 选择变量以查看“变量”面板。使用“变量”面板指定以下属性:

名称 变量名称在名称中只能使用字母数字字符 (a-z, A-Z, 0-9) 和下划线字符 (_)。最大长度为 255 个字节:

- 单字节字符集: 255 个字符
- 双字节字符集: 127 个字符

值 变量值具有以下特性:

- **长度:** 值可以是任意长度 (如果操作系统有限制, 那么最大长度为操作系统的限制值)。
- **特殊字符:** 字符 %、\$、[,], {、}、@2122 " 和 ' 对预解析器 (操作系统求值前) 和操作系统求值而言具有特殊意义。避免使用这些特殊字符。特殊字符的转义使用反斜杠 (\) 进行传递。请参阅第 259 页的『步骤中的变量解释』。

如果值用单引号括起, 变量名称将作为字面值而不是解释值传递。例如, 变量指定 `MyEnv='$env'` 导致字符串 `$env` 赋值给 `MyEnv` 而不是变量 `$env` 的值。

- **值中的变量:** 如果变量位于变量值中, 那么在运行使用该变量的步骤时会解释该变量。
- **下拉值:** 要指定下拉变量的项, 请将变量类型设置为“下拉列表”, 保存该变量, 然后进行编辑。单击“下拉选项”选项卡可以向下拉列表添加项。请参阅第 264 页的『为变量创建下拉列表』。
- **用作值的点命令:** 某些点命令可用作环境变量的值; 在这些情况下, 系统使用其他值替换点命令。请参阅第 267 页的『在变量中使用点命令』。
- **回车符:** 变量不存储回车符。可将多行值赋予变量, 如下例所示。文件 `text.txt` 的内容赋值给变量 `test`:

```
.bset env "test = `type text.txt`"
```

文件中的行被并置。例如, 假设文件内容如下:

```
A first line  
And a second line
```

变量的值随后将成为如下所示:

```
A first lineAnd a second line
```

类型 向变量分配以下类型之一:

- **标准** - 缺省值。可以向变量分配值和操作。
- **包含** - 变量值指向要包含的另一个环境。该环境中的所有变量都会被包含在内。

注：“包含”变量类型取代了之前版本中提供的 `.include` 函数。

- **下拉列表** – 变量包含一组可供用户选择的值。下拉列表变量创建完毕后，就可以选中它，然后单击**下拉选项**选项卡来添加值。

操作

选择以下项之一：

- **设置**：缺省选项。向变量分配指定的值。如果变量不存在，会进行创建。
- **未设置时设置**：此操作仅在变量还不存在值时向变量赋值。请参阅第 259 页的『关于变量』。
- **追加**：将值追加到该变量的当前值末尾。在两个值之间添加操作系统特定的 PATH 定界符：
 - Windows：分号（;）
 - UNIX 或 Linux：冒号（:）
- **前置**：将值插入当前值之前。在两个值之间添加操作系统特定的 PATH 定界符：
 - Windows：分号（;）
 - UNIX 或 Linux：冒号（:）
- **清除**：将值设置为空字符串。如果“值”属性包含值，那么不使用此选项。
- **删除/取消设置**：从当前应用的环境中删除变量。如果“值”属性包含值，那么不使用此选项。
- **分配隐藏**：系统向变量赋值，但在日志中隐藏该值，使其显示为“*****”。使用该选项可向运行项目的用户隐藏变量值。

要点：如果将敏感信息存储到隐藏变量中，请谨慎使用该选项。可通过检查数据库或使用 API 客户机来读取隐藏变量的值。对于密码等敏感信息，该选项不安全。

缺省情况下，“指定隐藏”变量受到保护，无法编辑。不具有“编辑隐藏的标记”许可权的用户不能编辑这些值。但是，如果他们具有这些许可权，那么可以删除并重新创建这些值。具有“编辑隐藏的标记”许可权的用户可以直接编辑这些值，但必须重新输入要使用的值。此行为通过将“指定隐藏”变量的类型更改为“设置”，防止用户显示该变量的值。

注：如果将步骤中的变量设置为“指定隐藏”，那么该步骤中所有的变量都会视为已设置为“指定隐藏”。

注：“指定隐藏”变量仅适用于项目。如果将“指定隐藏”变量传递给适配器，那么该变量就不再保持隐藏状态。

系统通常会将命令行中的变量语法更改为适合您操作系统的相应格式（`%VAR%` 用于 Windows[®]，`$VAR` 用于 Linux[®] 和 UNIX[®] 系统）。但是对于隐藏的变量，不进行这种更改。该变量直接传递到服务器，且服务器的操作系统环境会解释此变量。

因此，请勿使用可由任意操作系统解析的变量名称。示例：

- \$name: 如果此变量设为“指定隐藏”，那么在 UNIX 或 Linux 系统上解释该变量时，操作系统会尝试在其环境中而不是在 Build Forge 环境中对它进行解析。
- %address%。如果此变量设为“指定隐藏”，那么在 Windows 系统上解释时，操作系统会尝试在其环境中而不是在 Build Forge 环境中对它进行解析。

在项目中

定义手动启动作业时变量的使用方式。此属性仅影响在分配给项目的环境中使用的变量。在作业运行期间，此属性不会影响变量。值可以是以下选择之一：

- **正常**：变量在分配到项目时行为正常。
- **必需**：变量必须存在值。具有此属性的变量会在“启动”面板中被突出显示。使用在变量定义中定义的值就可以了。如果值未定义，那么无法快速启动或启动作业。

如果包含“必需”变量的作业由调度程序而不是用户启动，那么当这些变量当前具有值时会保持不变，当没有值时将为空。

- **只读**：值无法更改。
- **禁止显示**：变量不显示在“启动作业”面板上。但是，变量存在且可用于步骤中。
- **必须更改**：变量值必须更改。具有此属性的顶级变量会在“启动”面板中突出显示；如果未输入新值，就无法快速启动或启动该作业。

注：如果包含“必须更改”变量的作业由调度程序而不是用户启动，那么变量值不会更改。此外，如果包含“必须更改”变量的作业由类属性启动，那么变量值不会更改。类属性包含“在清除时启动”、“在进入时启动”和“在退出时启动”。

“快照”选项卡

环境快照是环境的一个实例。“快照”选项卡提供快照名称。使用此选项卡可查看或更改快照名称以及有关快照的注释。

创建环境

过程

1. 在左侧菜单中，单击**环境**。
2. 单击**添加环境**。
3. 为环境指定名称。
4. 指定其成员应能够使用此环境的访问组。
5. 单击**保存**。将出现变量面板，可在其中将变量添加到环境。

使用变量

以下部分描述了使用变量完成常见任务的过程。

为变量创建下拉列表


关于此任务

您可以为变量定义多个可能的值。提供的值以下拉列表的形式显示为可选项。

要为环境变量创建下拉列表，请执行以下操作：

1. 选择**项目 > 环境**。
 2. 在列表中，选择环境。
 3. 单击**添加环境变量**。如下填入其属性。
 - 名称：输入环境变量名称。
 - 类型：选择**下拉列表**。
 - 操作：选择操作。
 - 在项目中：选择一个属性。
 4. 单击**保存**。
 5. 单击该变量。
 6. 单击**下拉选项**选项卡。
 7. 如下为下拉列表添加值：
 - a. 为下拉选项指定名称。用户将看到该名称显示为下拉列表中的选项名称。下拉名称只能使用字母数字字符（a-z、A-Z 和 0-9）以及下划线字符（_）。最大长度为 255 个字节：
 - 单字节字符集：255 个字符
 - 双字节字符集：127 个字符
 - b. 指定下拉选项的值。如果希望用户看到正在使用的值，那么可以设置与变量同名的值。如果在值中使用变量，那么会在运行步骤的操作系统上解释该变量。将不会在步骤运行前对该变量进行预处理或求值。
 - c. 单击**创建**。该选项将添加到列表中。
 - d. 为每个所需的值重复上述操作。
 8. 单击**保存变量**。
 9. 设置显示的缺省选项。请在填入**下拉选项**并保存此变量后执行此操作。单击该环境。在**详细信息**选项卡上，将**缺省值选项**属性从 **--NONE--** 更改为您希望在缺省情况下显示的值的名称。
- 请注意：在查看变量的**详细信息**选项卡时，如果没有设置**缺省选项**（其值为 **--NONE--**），那么缺省情况下，**值**列中会显示**下拉选项**列表中的第一项。
10. 单击**保存变量**。

您还可以如下对选项作进一步处理：

- 使用每个选项名称左侧的**编辑**图标 ，以在列表中定位选项或删除选项。
- 单击要编辑的选项。您可以编辑**名称**和**值**字段。完成后单击**保存**。
- 单击**清除**可清除**名称**和**值**字段。此操作一般在查看现有选项后执行，以创建新选项。

包括其他环境

使用“包含”变量类型可以包括其他现有环境中的所有变量。

1. 选择项目 > 环境。
2. 新建环境并单击保存。
3. 单击该环境名称。
4. 单击添加环境变量。如下填入其属性。
 - 名称: 输入环境变量名称。
 - 类型: 选择 包含。用户界面更改为显示包括环境下拉列表。将除去“操作”和“在项目中”属性。
 - 包括环境: 从列表中选择要包括的环境。
5. 单击保存变量。

在步骤执行期间更改变量值

变量可以在步骤、项目或永久作用域的执行过程中进行更改。

- 步骤作用域: 通过使用显式指定, 在步骤中使用命令可以覆盖变量值。这些值只在当前步骤中有效。
- 项目作用域: 在步骤中使用 .bset 命令可以更改正在运行的作业的作用域的变量值。可以使用 .bset 创建新变量。它们对作业的其余部分有效。使用 .bset 所作的更改将在使用 .bset 的步骤之后的步骤中生效。
- 永久作用域: 在步骤中使用 .set 命令会更改环境变量的定义。变量在服务器环境、项目环境和步骤环境中定义。使用 .set 更改服务器变量或项目变量不会更改当前作业的变量副本。更改过的变量在当前作业之后运行的作业中生效。但是, 如果 .set 命令更改步骤环境变量, 更改过的变量在使用该步骤环境的后续步骤中生效。.set 命令无法创建新变量。

例如, 如果使用名为 Java 的项目环境启动项目, 该环境中包含变量 JavaVersion = 1.4, 且您使用 .bset 将其值更改为 1.5, 那么所有继承该项目环境的步骤中该变量的值都为 1.5, 而所有特别引用 Java 环境的步骤中该变量的值则为 1.4。

请注意: 在系统启动作业时, 它会将项目环境变量复制到为作业留出的数据库记录中, 并在以后获取项目缺省值时引用该作业环境。如果用户在启动作业时修改了任意项目变量的起始值, 那么那些值会被记录到作业记录中。

映射 Windows 驱动器

Microsoft Windows[®] 操作系统以不同的方式管理映射的驱动器。代理程序尝试为用户帐户重新映射已记忆的连接, 但是在运行时可能无法成功完成映射。可使用特殊的环境变量 _MAP 变量来协助在 Windows 上的驱动器映射。如果设置了此变量, Windows 代理程序将在执行步骤之前映射驱动器。

使用 _MAP 变量的典型做法是将其指定给项目环境, 这样相同的驱动器映射就能通过环境变量继承传给所有步骤环境。请注意: 如果在步骤环境中也定义了 _MAP 变量, 那么步骤环境的值会覆盖项目环境, 因为对于一个特定步骤, 只能定义一个 _MAP 值。

虽然这是针对 Windows 环境, 但在 _MAP 变量中, 仍请使用正斜杠来分隔目录路径名。使用路径时, 代理程序将根据需要自动更正斜杠。

例如, 将 `_MAP` 设置为

```
X:>//server/share
```

, 这将定义将 X: 驱动器连接到 Windows UNC 路径名 `\\server\\share` 的运行时映射。

可映射多个驱动器, 方法是在 `_MAP` 变量中提供更多映射规范, 并用分号分隔它们:

```
X:>//server/share;Y:>//server/share2
```

缺省情况下, 在 Windows 上执行驱动器映射时使用为逻辑服务器定义的相同的用户名和密码。在映射后面的括号中添加用户名和密码可为不同的用户名映射驱动器, 如下例所示:

```
X:>//server/share(alternateusername,password)
```

注: 如果您的密码包含 `$` 字符, 请再使用一个 `$` 字符进行转义。示例: 对于密码 `pas$word`, 请输入 `pas$$word`。避免在密码中使用以下特殊字符: `%`、`[`、`]`、`{`、`}`、`"` 或 `'`。

通过 `_MAP` 变量映射的驱动器在命令完成后将解除映射。

即使驱动器映射成功, 但如果登录到系统控制台的用户正在使用所讨论的驱动器或共享, 那么 Windows 上的驱动器映射可能仍然不可访问。

使用下一个可用盘符映射 Windows 驱动器

可以让系统选择下一个可用的盘符。请使用以下语法:

```
<driveletter>?=//<directory path>
```

例如, 可如下设置 `_MAP`:

```
X?://server/share
```

在此例中, 系统不会将驱动器映射到 X, 而是映射到下一个可用盘符, 并将它选择的盘符存储在名为 `_MAP_X` 的变量中。如果选择的驱动器为 F, `_MAP_X` 变量的值将为“F:”。可使用该变量来访问映射。

您可以如下使用多个映射:

```
X?://server/share; Y?://server/public
```

可随意使用任何字母, 甚至可以使用多个映射, 如以下示例所示:

- F: 映射到 `//server/share`
- G: 映射到 `//server/public`

该示例还将创建以下变量:

- 值为“F:”的 `_MAP_X`
- `_MAP_Y`, 值为“G:”

注: 如果要在使用 Cygwin 的 Windows® 系统上使用下一个可用驱动器语法, 就必须用反斜杠转义问号, 如下所示:

```
Y\?://server/share
```

基于代理程序的驱动器映射

可以通过在代理程序中使用配置参数来映射驱动器。映射参数添加到 BFAgent.conf 文件后，将使用与 `_MAP` 变量相同的语法。可是使用此参数为特定服务器创建驱动器映射。如果还使用 `_MAP` 变量，那么其映射会覆盖代理程序映射。

在变量中使用点命令

一些点命令可用作环境变量的名称或值。

使用 `.source` 在运行命令之前运行脚本

系统提供了一种功能，通过定义名为 `.source` 的特殊环境变量可在系统运行命令之前运行服务器上的脚本。这样，您就可以从服务器上的源文件中装入一组环境变量，或者运行定制的准备命令。

要尝试此功能：

1. 在系统上创建一个名为 `mybatch.bat` 的批处理文件，其功能是回显某个语句。将批处理文件保存到 `C:\temp`。
2. 创建名为 `Step Variables` 的新环境。
3. 添加一个名为 `.source` 的变量，其值为 `C:\temp\mybatch.bat`。
4. 将步骤的环境设置为新创建的 `Step Variables` 环境。
5. 运行项目，然后检查该步骤的日志输出。

请注意附加的日志数据显示 `mybatch.bat` 文件已在步骤命令之前运行。有关 `.source` 的一些要点：

- 指定的路径不能包含参数。
- 在 Windows® 平台上，通过调用来运行脚本。
- 在 UNIX® 平台上，在正在运行的 shell 中运行脚本时，该脚本必须采用本机 shell 语法。

使用 `.date` 命令在变量中存储日期或时间

可使用 `.date` 环境点命令为变量提供当前日期或时间。将该命令用作变量值。运行使用该变量的项目时，系统使用 `.date` 命令的结果更新该变量。

例如，名为 `MONTH` 的变量值为 `.date %B` 当该变量包含在项目时，它将为五月期间的作业提供值“May”。

有关如何使用此命令的更多信息（包括日期格式代码的列表），请参阅 第 336 页的『`.date`』。

系统变量引用

系统定义的变量是可在变量中使用的变量。

系统在每个作业步骤中自动设置以下变量的值。这些变量是只读的。在步骤日志的 ENV 行中列出了它们在该作业中的值。前四项是项目级别的通知。所有其他 BF_ 变量用于步骤级别。

项目级别的变量	值
BF_D	日期。可在标记中使用。格式由“标记: 日期格式”系统设置决定。
BF_J	当年的第几天。可在标记中使用。
BF_T	时间。可在标记中使用。格式由“标记: 时间格式”系统设置决定。
BF_W	星期几，用 0（星期日）到 6（星期六）的数字表示。

步骤级别的变量	值
BF_D	日期。可在标记中使用。格式由“标记: 日期格式”系统设置决定。
BF_J	当年的第几天。可在标记中使用。
BF_T	时间。可在标记中使用。格式由“标记: 时间格式”系统设置决定。
BF_W	星期几，用 0（星期日）到 6（星期六）的数字表示。
BF_AGENT_PLATFORM	标识运行代理程序的操作系统平台的字符串。
BF_AGENT_VERSION	当前服务器的代理程序的版本号。
B	缺省标记变量，从 1 开始，按每个作业进行递增。可用在以 BF_TAG 表示的标记中。
BF_BID	作业标识号，对于同一个项目的作业而言是唯一的。
BF_CALLER_	应用于从主调项目传递到链式项目的变量的前缀。
BF_CLASS	项目的 Build Forge 类
BF_ENGINE	唯一标识引擎的字符串。此值也存储在安装目录中的文件中: engine.id。示例: D8531015-6C07-1014-8CA0-BD58317220B3。
BF_HOST	逻辑服务器的主机名（TCP/IP 主机名）。（该变量是服务器环境的一部分。）
BF_ITERATION	成功启动 While 循环中某个步骤的次数。该数目在步骤条件的求值结果为 true 时递增。作业重新启动使用此变量的值作为重新启动的迭代。
BF_ITERATION_MAX	While 循环可以运行的最大次数。它在步骤属性中设置。如果达到此迭代次数，那么 BF_ITERATION_MAX_REACHED 设为“是”。
BF_ITERATION_MAX_REACHED	缺省情况下不创建或不设置此变量。当 while 循环迭代达到 BF_ITERATION_MAX 值时，while 循环步骤的步骤唯一标识 (BF_SSID) 会追加到此变量的末尾。如果一个项目中的多个 While 循环步骤达到其 BF_ITERATION_MAX 值，那么此变量包含多个值，每个达到最大迭代次数的步骤都有一个值。

步骤级别的变量	值
BF_LASTGOODRUN	同一个项目的上一个通过的作业的日期，如果没有通过的作业，就为上一个作业的日期。
BF_LASTGOODTAG	上一个通过的作业（如果没有存储同一个项目通过的作业，就为上一个作业）的标记。
BF_LASTGOODUNIX	与 BF_LASTGOODRUN 相同，但是以 UNIX [®] 格式的日期表示。
BF_LASTRUN	当前作业上一次运行的日期。
BF_LASTTAG	同一个项目的前一个作业的标记字符串。
BF_LASTUNIX	与 BF_LASTRUN 相同，但是以 UNIX 格式表示日期。
BF_ONFAIL	步骤的“暂停/继续”标志。
BF_PID	项目标识号。
BF_PROJECTNAME	此作业的项目名称。
BF_PROJECTNAME_PHYS	用于创建项目目录的项目名。系统将无效的 相对目录字符 系统设置中指定的字符更改为下划线字符，以创建项目目录。例如，如果此设置包含空格，那么名为 My Project 的项目会得到名为 My_Project 的项目目录。
BF_ROOT	作业的基本工作目录，来源于作业属性。另请参阅 BF_STEP_ROOT。
BF_SERVER	运行当前作业的服务器的名称（该变量是服务器环境的一部分）。
BF_SERVER_ROOT	在服务器属性中分配给逻辑服务器的路径（该变量是服务器环境的一部分）。
BF_SID	构建内步骤结果的序号。
BF_SPID	如果当前作业被另一个作业调用，此变量包含主调项目的标识。否则，此值与 BF_PID 相同。
BF_STEPNAME	步骤名称。在步骤属性中设置。
BF_STEP_ROOT	步骤的基本工作目录，来源于步骤属性。另请参阅 BF_ROOT。
BF_SSID	步骤标识，是项目中当前步骤的唯一标识。
BF_TAG	作业标记。标记定义可以包含变量。此变量包含的值是通过启动作业时对这些变量进行解释而得到的。
BF_TAG_PHYS	作业标记，用下划线替换 BF_TAG 值中存在的任何空格。如果选择了步骤的“绝对”选项，那么 BF_TAG_PHYS 与 BF_TAG 的作用相同。
BF_USER	作业所有者的用户名。

触发器变量引用

系统监视以下变量名称。当步骤的环境包含这些变量的其中之一（特意包含或者从项目或服务器继承）时，系统将执行操作。

变量	目录
_CI_BUILD_DELETE	将此变量设置为任意值以在作业运行后删除构建以及关联的构建数据。（如果未运行其他项目构建，标记变量将复位为已删除构建之前的初始值。）
_CI_BUILD_KEEP	将此变量设置为任意值将在作业运行后保留构建以及关联的构建数据。例如，如果作业包含适配器链接并且适配器步骤失败，那么其他项目步骤将不会运行。您可能想保留该作业的构建记录副本以作它用，例如用于调试。
CLEARCASE_VIEW	启动指定的 ClearCase 视图。此变量中指定的视图必须存在，而且必须将使用此变量的步骤设置为“absolute”。在运行 Microsoft Windows 的系统上，该变量必须与 bfagent.conf 中代理程序的 cc_suppress_server_root 配置选项一起使用。
_CLEARCASE_VIEWS	指定要在命令执行之前启动的 ClearCase 视图的列表。请将其值设置为视图的逗号分隔列表；例如，“View1,View2,View3”。
_CLEARCASE_VOBS	指定要在命令执行之前挂装的 ClearCase VOB 的列表。请将其值设置为 VOB 的逗号分隔列表；例如，“\Vob1,\Vob2,\Vob3”。
_CONTEXT_LOG_RANGE	使用此变量将日志输出限制为过滤器匹配项前后的那几行。它接受一个正整数值，并使系统只输出每一个过滤字符串命中项前后的一系列行（行数范围等于该变量的值），而忽略其他日志输出。例如，如果将变量设置为 5，日志中仅显示含有过滤器匹配项的行，以及这些匹配行的前后各 5 行。

变量	目录
<code>_ERROR_THRESHOLD</code>	<p>确定允许的最大错误（由已定义的“设置失败”过滤器捕获的错误）数量。使用此变量，您可以为各个步骤或一个项目建立失败或消息阈值。</p> <p>使用以下某种格式：</p> <ul style="list-style-type: none">• 值 5 或 F5 表示如果出现的错误多于 5 个，该作业应失败。• 值 N7 表示出现的错误多于 7 个时，系统应向作业说明添加一条消息。该消息表示已满足该阈值。 <p>当在一个步骤中使用该变量时，系统将统计该单个步骤中的错误数。有其他形式可用：</p> <ul style="list-style-type: none">• 类似 W9 的值表示在 9 次错误之后，不管过滤器以后是否捕获到错误，都会将该步骤置于警告状态。• 类似于 C8 的值表示在 8 次错误之后，会将该步骤设为故障状态，但是任何“清除故障”过滤器都可清除该故障。 <p>注： 此变量统计的错误数定义为将过滤器与“设置失败”操作进行匹配的字符串，并且该字符串将分配给项目中的步骤。由过滤器标识为故障的每一个字符串将作为一个错误计入步骤错误总计以及项目错误总计。</p>
<code>_EXITCODE_MAP</code>	<p>指定一个数字列表（用逗号、空格、分号或冒号分隔），系统应接受这些数字作为步骤成功的指示符。缺省情况下，退出代码 0 表示成功；当指定了此变量时，该变量中列出的任何值也都将表示成功。</p>

变量	目录
_InterfaceLoggingLevel	<p>控制 Build Forge 在运行适配器步骤时应记录多少日志数据。（在适配器环境中）创建名为 _InterfaceLoggingLevel 的环境变量。赋予其 0 到 8 之间的某个整数值。日志记录级别包含下级级别，例如，级别 2 包含级别 1 和 0 的信息。</p> <ul style="list-style-type: none"> • 0: Exec 行以及服务器连接错误或取消通知；没有其他内容 • 1: 解析过的命令（将发送到服务器的命令） • 2: 未解析的命令（设置其局部变量之前的命令） • 3: 构建和环境变量 SET 行 • 4: 临时和内部变量 SET 行 • 5: 环境求值，电子邮件组添加，BOM 文本记录行 • 6: 锁和子锁开始/结束行 • 7: （缺省记录级别）针对匹配模式进行检查的代理程序输出，以及与模式匹配的行。 • 8: 所有代理程序输出
_LOG	<p>指定一个路径名，以创建包含 Build Forge 代理程序原始输出的日志文件。</p> <p>注：该日志不包含时间戳记，除非还指定了 _LOG_TIMESTAMP。该文件中的日志数据的格式通常如下：代理程序代码、日志存储区和消息。</p> <p>使用此变量在服务器上保存作业日志的副本。如果文件已存在，系统会将日志追加到其中。</p>
_LOG_TIMESTAMP	<p>_LOG 的每一行输出都以时间戳记为前缀。此变量值应当为 .date 和 .gmdate 环境命令使用的同一 strftime 语法中的格式字符串。</p> <p>注：需要 _LOG。</p>
_MAP	<p>有关如何使用此变量的讨论，请参阅 第 265 页的『映射 Windows 驱动器』。</p>
_NO_PREPARSE_COMMAND	<p>系统在向代理程序发送命令之前通常会尝试解析环境变量的值。当定义了 _NO_PREPARSE_COMMAND 变量时（定义为任何值），系统将向代理程序直接发送这些环境变量，而不进行解析。使用此变量可确保操作系统 shell 处理那些环境变量。</p>

变量	目录
_PRISM_DIR_POSTCMD	用于 IDE 插件。指定在运行项目步骤后要对目录运行的命令。请参阅第 469 页的『测试项目的特殊变量』。
_PRISM_DIR_PRECMD	用于 IDE 插件。指定在为项目步骤将目录复制到服务器之前对目录运行的命令。请参阅第 469 页的『测试项目的特殊变量』。
_PRISM_FILE_POSTCMD	用于 IDE 插件。指定在运行项目步骤后要对文件运行的命令。请参阅第 469 页的『测试项目的特殊变量』。
_PRISM_FILE_PRECMD	用于 IDE 插件。指定在为项目步骤将文件复制到服务器之前对文件运行的命令。请参阅第 469 页的『测试项目的特殊变量』。
_SUPPRESS_ENV_OUTPUT	<p>指定系统忽略日志中的环境消息。缺省情况下，不会设置该变量，环境中的所有变量值都会在运行步骤命令前打印出来。值显示为步骤日志中的 ENV 条目。可将变量设置为以下值：</p> <ul style="list-style-type: none"> • ALWAYS: 始终忽略 ENV 消息 • 任何其他值: 忽略 ENV 消息。然而，如果命令失败，将在命令消息后打印 ENV 消息。在调试命令执行失败时，此信息可能非常有用。
_SUPPRESS_AGENT_LOG_OUTPUT	<p>如果设置为 1，将阻止代理程序向引擎发送日志数据。这与 _SUPPRESS_LOG_OUTPUT 不同，后者允许从代理程序发送日志数据，但由引擎丢弃。</p> <p>注： 使用此变量可防止过滤器匹配项。</p>
_SUPPRESS_LOG_OUTPUT	<p>如果设置为 1，会导致引擎丢弃几乎所有从代理程序接收的日志输出。某些控制台日志消息会保留。会显示过滤器匹配项。</p>
_TIMEOUT	<p>该值覆盖项目中的一个或全部步骤的“超时”属性。</p>
_TRAP	<p>当前步骤失败时要运行的字符串；该字符串可设置为可执行文件或命令的名称。注： 命令的输出不会返回到控制台，因为当步骤失败时，控制台和代理程序之间的连接将关闭；如果要保留通过 _TRAP 发出的命令的输出，请使命令将其输出写入文件，以备以后检索。</p>

变量	目录
<code>_USE_BFCREDS</code>	<p>当设置为 1 时，系统使用用户的登录凭证来登录到服务器，而不是附加于服务器的服务器授权中存储的凭证。系统使用启动项目的用户的管理控制台登录凭证来运行项目中的命令。可以为单个步骤设置此变量，也可整个项目设置。</p> <p>注：如果使用 LDAP/Active Directory 认证，本地存储用户认证系统设置必须设置为“是”（其缺省值），这样 <code>_USE_BFCREDS</code> 函数才能起作用。当该设置设为“是”时，系统会以加密形式高速缓存用户认证信息，然后可以访问用户认证信息来与 <code>_USE_BFCREDS</code> 一起使用。</p> <p>提示：在 Windows 上，也考虑设置变量 <code>_USE_BFCREDS_DOMAIN</code>。</p>
<code>_USE_BFCREDS_DOMAIN</code> （仅限于 Windows）	<p>设置为 1 时，系统使用用户的域，以及 <code>_USE_BFCREDS</code> 用于登录服务器的登录凭证。</p>
<code>_XSTREAM_PROTOCOL</code> <i>type</i>	<p>启用代理程序之间的直接文件传输。</p> <p>要点：某些操作系统上的代理程序对于直接文件传输存在限制或者不支持。请参阅第 158 页的『配置代理程序之间的直接文件传输』。</p> <p>引擎、发送方代理程序和接收方代理程序都必须支持直接文件传输。如果任何一方不支持，那么会忽略 <code>_XSTREAM_PROTOCOL</code>，而不发出警告，并且会使用普通的文件传输方法。</p> <p>接收方代理程序必须能够在发送方代理程序主机上创建 TCP 连接。如果它们都存在，那么必须配置防火墙以允许连接。</p> <p>协议的 <i>type</i> 确定编码数据的方法，采用以下类型之一：</p> <p>AES-CBC 使用强加密算法进行数据编码。必须通过 OpenSSL 编译两个代理程序，并且使用 SSL 与引擎进行通信。加密密钥从引擎处获取。</p> <p>PRNG 使用伪随机数生成器来掩盖文件内容。</p> <p>PLAIN 不进行编码，按原样传输文件。</p>

环境快照

对环境创建快照，以便为要更改或修改的环境快速创建新实例。

环境快照概述

回顾这些主题以了解环境快照及其使用方法。

环境快照用例

以下示例描述了环境快照的一些常见用例：

- 对环境创建快照后可以更改环境配置或测试新工具或脚本。
- 将环境快照另存为临时备份或正式归档的一部分。
- 对环境创建快照可以捕获与里程碑（如外部发布或内部发布）相符的时间点环境配置。

环境快照概念和术语


在 UI 中，快照引入了一些用于环境的新概念和新术语。

环境快照：快照是现有环境的新实例。需要紧记关于快照的以下几个要点：

- 快照是独立的环境对象。更改快照集合中的一个快照并不会影响集合中的其他快照。
- 快照不是副本。


如果对与环境关联的对象创建快照，将创建该对象的新实例。副本用于映射对象之间的关系；它不会创建新的对象。

- 快照不是环境的修订版：
 - 快照不支持对两个环境快照的更改进行比较。
 - 对环境快照所作的更改不像在源控制系统中那样可以通过版本号进行跟踪或识别。但是，可以通过使用包括版本号（例如，7.5.0 和 3.4.01）的快照命名方案，将环境快照与里程碑相关联。

快照集合：快照集合是一个基本快照的所有后代环境快照的集合。该集合至少包括基本快照或父代快照及其子代快照。在 UI 中，环境名称旁边的快照图标  表示该环境存在快照集合。

Base Snapshot：最初，所有环境的快照名称都是 Base Snapshot。您可以将 Base Snapshot 更改为其他名称。基本快照是快照集合中的父代。

缺省环境快照：缺省环境快照是当前的工作环境。一个集合中只能有一个缺省快照。如果未指定缺省快照，基本快照便是缺省快照。

- 在 UI 中，缺省快照显示在环境列表的顶级。选择**环境**可显示环境列表。
- 选择具有快照的环境时，除非列表框中选择其他环境快照，否则将使用缺省环境快照。
- 要访问和处理环境快照集中的其他快照，必须单击“快照”图标 .

环境快照视图

选择“快照”图标  以显示“快照”视图。在 UI 中，“快照”视图显示集合中的快照层次结构。

- 基本快照位于顶级并且具有名称 Base Snapshot（如果没有为它指定唯一的名称）。
- 所有环境快照都是基本快照的子代。同一个基本快照的子代缩进在“快照”列的同一级别。
- 从子代快照创建的环境快照成为该子代快照的子代，缩进为“快照”列的下一个级别。

环境快照计划

回顾一些选择缺省环境快照和命名环境快照的最佳实践。

• 在集合中选择缺省快照的策略

此 UI 在快照集合中识别唯一的一个缺省或当前环境快照。请使用一致的策略来选择缺省快照：

- 使用基本快照作为缺省快照

使用此策略，可将快照创建为时间点备份，并且不更改已备份的环境快照。可更改基本快照。

- 使用最新快照作为缺省快照

此策略可以在创建新环境时使用，以便使其成为新的缺省环境快照。不更改基本快照或早期的环境快照。

• 确定集合的快照命名方式

环境快照名称在环境快照集合中必须唯一。

使用以下条件有助于您创建环境快照名称：

- 名称应该是描述性的：它应指示快照的用法或用途。
- 命名方案应遵循已定义的标准。可以使用“快照”选项卡上的“注释”框描述命名方案。

• 对集合使用单个环境名称

创建环境快照后，可以选择更改环境名称。如果更改了环境名称，会对每个环境快照更新该名称。

创建环境快照

创建环境快照将创建环境的新实例。快照不是副本，而是新环境。

您可以从环境或环境快照中创建环境快照。

提示：要创建快照和更改缺省快照，用户必须具有所需的许可权。请参阅第 307 页的『验证和编辑访问组的快照许可权』。

1. 单击要创建快照的环境或环境快照旁边的**编辑**图标：

- 要建立缺省环境快照，请在环境列表（**环境**）中，单击顶级快照旁的“编辑”图标



- 要建立非缺省环境快照，请单击“快照”图标 。“快照”视图显示集合中的环境快照。单击非缺省环境快照旁的“编辑”图标 .

- 2. 单击**新建快照**。
- 3. 在“快照”选项卡的**名称**中，输入名称。名称便分配给通过环境创建快照的所有对象。

名称在环境快照集合内必须唯一。

- 4. 选择创建环境快照时要拍摄快照的 Build Forge 对象。下表中描述了您可以选择的对象。

对象	描述
缺省	在 UI 中，缺省快照显示在环境列表的顶级。 选择 环境 可显示环境列表。
下列环境包含	通过使用“包含”环境变量类型对环境所包括的环境拍摄快照。

- 5. 单击**保存**保存环境快照。

更改缺省环境快照

缺省环境快照是快照集中的顶级快照，显示在环境列表（**环境**）中。

提示：要创建快照和更改缺省快照，用户必须具有所需的许可权。请参阅第 307 页的『验证和编辑访问组的快照许可权』。

要更改缺省环境快照，对您希望用作新缺省快照的快照编辑其定义：

- 1. 选择**环境**。
- 2. 在环境列表中，单击缺省环境快照的**快照**图标。
- 3. 在快照列表中，对要用作新缺省快照的环境快照单击**编辑**图标。
- 4. 单击**设为缺省值**。
- 5. **要点：**在弹出窗口中，选择“确定”或“取消”。



确定	更新引用： 对于引用先前缺省环境快照的任何对象，会将先前缺省环境快照中的引用更新为引用新的缺省环境快照。
取消	不更新引用： 对于引用先前缺省环境快照的任何对象，不更新为引用新的缺省环境快照。

更改环境快照名称

您可以更改环境快照的名称，也可以更改在创建环境快照时选择来创建快照的对象的快照名称。

对于基本快照，可以使用此选项仅为单个环境快照或为所有当前和未来环境将缺省名称 Base Snapshot 更改为其他快照名称。

要更改快照名称，请执行以下操作：

- 1. 选择**环境**。
- 2. 在环境列表中，单击缺省环境快照的**快照**图标 。
- 3. 在快照列表中，单击环境快照的**编辑**图标 。

4. 选择**快照**选项卡。
5. 在**名称**中，输入新的名称。
6. **可选**：在**注释**中，输入注释。
7. **要点**：在弹出窗口中，选择“确定”或“取消”。

确定	<p>更改环境快照名称和其他快照对象名称：对于在创建环境快照时选择来创建快照的对象，更改这些对象和环境快照的名称。</p> <p>对于 Base Snapshot：更改所有当前环境快照及所有未来环境快照的名称 Base Snapshot。</p>
取消	<p>更改环境快照名称，但不更改其他快照对象名称：对于在创建环境快照时选择来创建快照的对象，不更改这些对象的名称。仅更改环境快照名称。</p> <p>对于 Base Snapshot：保留所有当前环境快照及所有未来环境快照的名称 Base Snapshot。</p>


访问和查看快照

创建环境快照会创建至少包含两个环境的快照集合：基本快照环境和新环境快照。



要查看快照集合中的所有环境快照，请执行以下操作：

1. 选择**环境**。

环境列表列出环境和环境快照。顶级快照是缺省环境快照。

2. 单击**快照**图标  以显示快照集中的环境快照。

在“快照”视图中，您可以：

- 新建环境快照。要开始操作，请单击**编辑**图标  。
- 更改环境的缺省快照。单击新缺省快照的**编辑**图标  ，然后单击“设为缺省值”。
- 就像编辑标准环境那样编辑环境快照定义。

删除环境快照

可使用“删除环境”选项删除环境快照。


如果环境正在由其他对象使用，那么无法删除。例如，包括在其他环境中或正在由项目、步骤、调度或服务器使用。

要删除环境快照，请执行以下操作：

1. 选择**环境**。

2. 在环境列表中，单击基本快照的**快照**图标  。

“快照”视图显示集合中的环境快照。

3. 单击要删除的环境快照旁的 **编辑**图标  。
4. 单击**删除环境**。

第 19 章 使用项目

本主题描述如何在管理控制台中创建和管理项目。

关于项目

项目是可执行的多组步骤，拥有其自己的环境组和服务器属性。

除了项目，还可以创建库。库类似于项目，但不含用于确定步骤运行位置的选择器。

关于“项目”面板

使用“项目”面板可创建新项目并查看或编辑现有项目。要访问“项目”面板，请单击左侧菜单中的项目。

“项目”面板提供以下选项卡：

- 『“详细信息”选项卡』
- 第 281 页的『“标记”选项卡』
- 第 281 页的『“寄存器”选项卡』
- 第 281 页的『“注释”选项卡』
- 第 281 页的『“快照”选项卡』

“详细信息”选项卡

在“详细信息”选项卡上，您可以指定有关项目的各种属性。



名称	<p>项目的名称。系统使用此名称来引用列表和数据库中的项目。</p> <p>在运行项目时，项目名称用于构造项目目录。因为项目可能包含运行于不同操作系统的步骤，所以请避免在项目名称中使用特殊字符和符号。如果项目名称必须使用导致无效目录名的字符，请将这些字符添加到“无效的相对目录字符”系统配置设置中。在创建项目目录时，此设置中列出的字符将转换为下划线(_)。使用 & 符号可能导致无意生成 HTML 实体作为产品名称的一部分。</p>
访问	<p>有权查看和使用项目的访问组。访问属性随许可权一起用于确定用户可以执行的操作。例如，要启动某个作业，您必须是为项目指定的访问组的成员，并且还必须是具有“执行作业”许可权的组的成员。有关访问组的更多信息，请参阅第 195 页的『访问概述』。</p>
禁用	<p>选中此复选框可禁用项目。当用户尝试运行禁用的项目时，Rational Build Forge 会显示一条消息，表明该项目处于不活动状态并且不会运行项目。</p>
最大线程数	<p>允许项目启动的并行处理的最大数量。使用该字段可防止项目使用过多的系统资源。每个支持线程的步骤和任何内联项目（它们自己可能启动支持线程的步骤）都会产生并行处理，但所有这些处理都计入父项目的最大数量。当达到“最大线程数”值时，系统会停止启动新的并行处理，并等待项目的并行处理数量降低到低于“最大线程数”值，然后再继续。有关线程化的更多信息，请参阅第 320 页的『线程化：并行运行步骤』。</p>
运行限制	<p>运行限制属性设置项目在同一时刻允许的最大作业数。</p> <ul style="list-style-type: none"> 如果启动项目且活动作业数等于“运行限制”，那么新作业在“等待”队列中等待，直到至少完成一个作业为止。 如果调度尝试启动项目而活动作业数等于“运行限制”且“硬运行限制”系统配置设置具有值“是”，那么系统不会启动新作业。如果“硬运行限制”设置为“否”，那么系统忽略对已调度构建的“运行限制”设置。 通过内联启动的项目不被视为原始项目的实例，不计入其“运行限制”内。
类	<p>必须将每个项目指定给一个类，类用于为作业组指定全局属性。有关更多信息，请参阅第 295 页的『类』。</p>
选择器	<p>在选择项目服务器时使用的选择器的名称。对于项目内未指定它们自己的选择器的所有步骤，系统使用该选择器作为缺省的选择器。请参阅第 239 页的『选择器』。如果未指定选择器，那么项目将被添加到“库”面板，而不是“项目”面板。库使用调用步骤的选择器；如果该步骤没有选择器，那么库将使用调用项目的选择器。</p>
通过链，失败链	<p>选择在项目构建通过或失败时运行的项目。在项目级别设置通过/失败链使您可以根据项目的通过/失败状态来调用单独的通过/失败操作。该功能类似于在项目内的步骤级别设置通过/失败操作。在项目级别，通过/失败操作是由项目运行状态而不是步骤状态触发。</p>
环境	<p>在“服务器”环境之后、“步骤”环境之前应用的环境。有关环境如何协同工作的更多信息，请参阅第 259 页的『关于变量』。</p>
固定	<p>启用“固定”复选框可以强制项目中使用缺省项目选择器的所有步骤都停留在同一服务器上，如果该服务器正忙，那么等待它变为可用。有关此选项的更多信息，请参阅第 284 页的『使步骤固定在服务器中』。</p>

启动通知、通过通知、失败通知

通过对这些字段中的一个或全部选择访问组，使用这些字段可以指示系统在项目启动、通过和/或失败时发送通知电子邮件。

“标记”选项卡

使用“标记”选项卡可管理项目的构建标记。

标记格式

使用纯文本和标记变量引用定义项目标记的字符串。有关标记格式的更多信息，请参阅第 327 页的『作业期间更改构建标记』。

标记同步

同步两个项目的标记变量。选择要将其标记变量与当前项目进行同步的项目。当两个项目进行同步时，它们的变量从同一个池中提取，所以当它们依次运行时，一个项目获得值 1，下一个获得值 2，如此类推。有关更多信息，请参阅第 287 页的『同步标记』。

标记名称

变量的名称。当您以标记格式使用标记变量时，请使用格式 `$<Tag Name>` 引用其名称。例如，要创建使用 `MainVer` 和 `B` 变量的标记，请使用标记格式 `“Build_$(MainVer).$(B)”` 获取类似 `Build_005.1` 的标记。

初始值 设置标记变量的值。若不使用**自动增加**选项，变量将保留该值直至您更改它。

填充 如果选择了除 `None` 外的**填充值**，那么如需在标记中使用变量来确保数字数量等于**填充值**时，系统将向变量值前添加零。例如，如果变量当前为 2，而“填充”为 3，那么系统会将值表示成 002。“填充”范围为 1 到 8。

自动增加

若设置为 `Yes`，系统对于项目的每个作业按 1 递增变量值。

“寄存器”选项卡

使用项目寄存器可存储构建之间持久存在的信息。

寄存器 项目寄存器的名称。

目录 寄存器的值。

有关更多信息，请参阅第 330 页的『项目寄存器』。

“注释”选项卡

使用“注释”选项卡可存储项目中感兴趣的项。

对于每条注释，选项卡显示：

- 用户创建注释的日期和时间
- 输入注释的用户
- 注释

“快照”选项卡

项目快照是项目实例。“快照”选项卡提供快照名称。

使用此选项卡可查看或更改快照名称以及有关快照的注释。

更改项目属性

要更改项目级别属性，请选择项目，然后单击所需项目名称旁边的**编辑**图标 。

项目名称

项目的名称。系统使用此名称来引用列表和数据库中的项目。

在运行项目时，项目名称用于构造项目目录。因为项目可能包含运行于不同操作系统的步骤，所以请避免在项目名称中使用特殊字符和符号。如果项目名称必须使用导致无效目录名的字符，请将这些字符添加到“无效的相对目录字符”系统配置设置中。在创建项目目录时，此设置中列出的字符将转换为下划线(_)。使用 & 符号可能导致无意生成 HTML 实体作为产品名称的一部分。

访问 有权查看和使用项目的访问组。**访问**属性随许可权一起用于确定用户可以执行的操作。例如，要启动某个作业，您必须是为项目指定的访问组的成员，并且还必须是具有“执行作业”许可权的组的成员。有关访问组的更多信息，请参阅第 195 页的『访问概述』。

标记格式

使用明文和标记变量引用定义项目标记的字符串。有关标记格式的更多信息，请参阅第 327 页的『作业期间更改构建标记』。

标记同步

同步两个项目的标记变量。选择要将其标记变量与当前项目进行同步的项目。当两个项目进行同步时，它们的变量从同一个池中提取，所以当它们依次运行时，一个项目获得值 1，下一个获得值 2，如此类推。有关更多信息，请参阅第 287 页的『同步标记』。

最大线程数

允许项目启动的并行处理的最大数量。使用该字段可防止项目使用过多的系统资源。每个支持线程的步骤和任何内联项目（它们自己可能启动支持线程的步骤）都会产生并行处理，但所有这些处理都计入父项目的最大数量。当达到“最大线程数”值时，系统会停止启动新的并行处理，并等待项目的并行处理数量降低到低于“最大线程数”值，然后再继续。有关线程化的更多信息，请参阅第 320 页的『线程化：并行运行步骤』。

运行限制

运行限制属性设置项目在同一时刻允许的最大作业数。

- 如果启动项目且活动作业数等于“运行限制”，那么新作业在“等待”队列中等待，直到至少完成一个作业为止。
- 如果调度尝试启动项目而活动作业数等于“运行限制”且“硬运行限制”系统配置设置具有值“是”，那么系统不会启动新作业。如果“硬运行限制”设置为“否”，那么系统忽略对已调度构建的“运行限制”设置。
- 通过内联启动的项目不被视为原始项目的实例，不计入其“运行限制”内。

类 每个项目都必须被指定到一个类；类用于为成组的项目指定全局属性。有关更多信息，请参阅第 295 页的『类』。

选择器 在选择项目服务器时使用的选择器的名称。对于项目内未指定它们自己的选择器的所有步骤，系统使用该选择器作为缺省的选择器。请参阅第 239 页的『选择器』。如果未指定选择器，那么项目将被添加到“库”面板，而不是“项目”面板。库使用调用步骤的选择器；如果该步骤没有选择器，那么库将使用调用项目的选择器。

通过/失败链

选择在项目构建通过或失败时运行的项目。在项目级别设置通过/失败链使您可以根据项目的通过/失败状态来调用单独的通过/失败操作。该功能类似于在项目内的步骤级别设置通过/失败操作。在项目级别，通过/失败操作是由项目运行状态而不是步骤状态触发。

环境 在“服务器”环境之后、“步骤”环境之前应用的环境。有关环境如何协同工作的更多信息，请参阅第 257 页的『环境继承』。


固定 启用“固定”复选框可以强制项目中使用缺省项目选择器的所有步骤都停留在同一服务器上，如果该服务器正忙，那么等待它变为可用。有关此选项的更多信息，请参阅第 284 页的『使步骤固定在服务器中』。

启动通知、通过通知、失败通知

通过对这些字段中的一个或全部选择访问组，使用这些字段可以指示系统在项目启动、通过和/或失败时发送通知电子邮件。

复制项目

要制作现有项目的副本，请完成以下步骤：

1. 选择项目。
2. 选择想要复制的项目旁边的**编辑**图标 。
3. 单击**复制项目**。

在复制项目时，系统将以下引用从现有项目复制到新项目：

- 步骤及其在步骤的**详细信息**选项卡上列出的所有属性
- 在项目的**详细信息**选项卡上列出的所有项目属性，例如，项目的类、选择器和其他属性
- 项目的标记格式（位于项目属性面板上的**标记**选项卡中）

系统不会复制以下属性：

- 标记变量（位于项目属性面板上的**标记**选项卡中）；然而，复制标记格式中使用的标记变量
- 项目寄存器（位于项目属性面板上的**寄存器**选项卡中）
- 步骤注释（位于步骤属性面板上的**注释**选项卡中）
- 项目注释（位于项目属性面板上的**注释**选项卡中）

删除项目


有两种删除项目的方法，这取决于项目是否有任何与之关联的作业。

选择以下一个选项来删除项目：

删除项目按钮

注：删除无法撤销的项目。


删除项目按钮将删除没有作业的项目。要使用此按钮删除项目，首先必须删除项目的所有作业。该按钮位于项目属性编辑页面和项目步骤列表中。要查看项目属性编辑页面，请完成以下步骤：

1. 在左侧菜单中，选择项目。
2. 单击所需项目名称旁边的编辑图标 。

破坏按钮

注：破坏操作将删除项目，即使项目已锁定，也是如此。而且，您无法撤销项目破坏。

破坏按钮可从 Build Forge 数据库删除项目及其所有相关作业。在破坏项目前，系统将要求确认。该按钮位于项目属性编辑页面上。要查看项目属性编辑页面，请完成以下步骤：

1. 在左侧菜单中，选择项目。
2. 单击所需项目名称旁边的编辑图标 。

使步骤固定在服务器中

如果项目步骤的选择器允许，步骤可以在不同的服务器上运行。但是，您可能会希望项目的所有或大部分步骤都在同一服务器上运行，无论您是否预先指定了该服务器。项目级别**固定**属性为您提供了该选项。

要查看项目级别属性，请选择项目，然后单击所需项目名称旁边的编辑图标 。

“固定”属性仅应用于项目中未指定自己的选择器的步骤。如果步骤的选择器选项不是“缺省值”，那么系统使用该选择器来为步骤选择服务器 - 即使选择器与项目的选择器相同，也是如此。

在设置了“固定”属性时，项目对选择器字段设置为“缺省值”的每个步骤使用同一服务器。该属性在项目的重新启动之间持续存在。

在系统启动内联项目时，系统使用内联项目的选择器作为内联步骤的缺省选择器。调用的项目的“固定”属性不会影响内联项目，内联项目遵从它自己的“固定”属性（如果已设置）。

在系统启动内联库时，它遵从以下规则：

- 未选中“固定”属性的内联库：使用调用的步骤的选择器作为内联步骤的缺省选择器。
- 选中“固定”属性的内联库：使用主调步骤的服务器作为内联步骤的缺省服务器。

注：在作业期间，可以使用 `.bset` 服务器命令来更改项目的缺省服务器。在 `.bset` 命令之后出现的步骤使用由该命令设置的新缺省值，并固定在这个新服务器中。

链：其他项目或库的条件执行

可以在项目级别调用两种类型的链：

- 通过链：指定项目或库在项目通过时运行。
- 失败链：指定项目或库在项目失败时运行。

该功能具有多种用途：

- 在项目级别使用条件执行。在步骤级别提供其他流控制的功能。此外，步骤可具有自己的“通过链”和“失败链”。请参阅第 313 页的第 20 章，『使用步骤』中的第 318 页的『控制执行流程』。
- 维护步骤的常用组，同时与依赖于这些组的项目隔绝。库也可用于此用途。
- 在项目通过或失败后清除文件。
- 在软件构件项目通过时调用自动测试和部署项目。

来自于调用项目或库的链继承

链式项目或库从调用项目中继承某些特征：

- 缺省情况下，链式项目继承调用项目的类。可以在**管理 > 系统**中更改该行为，方法是将“链接时覆盖类”设置为“否”。
- 链式库继承调用步骤的选择器，因为库没有自己的选择器。如果该步骤没有选择器，那么库继承构建选择器。链式库的步骤使用继承的选择器，除非这些步骤具有显式选择器。

否则，链式项目或库将利用自己的特征运行：

- 由其自己的选择器指定服务器。
- 使用自己的属性，包括自己的通知设置和链设置。
- 在调用项目环境后应用其环境。来自于调用项目的变量将重新命名，并可用于被调用的项目。另请参阅『链式项目中的环境变量继承』。

链嵌套

在连接项目时，被调用的项目将嵌入调用项目中。最大嵌套级别为 32 个级别。嵌套级别可能还受运行“管理控制台”的主机上可用内存的限制。

运行和中断链

如果在链式项目中使用 `.break` 命令，那么系统会停止该链式项目，但是将控制返回给调用项目，然后该项目将继续。请参阅第 335 页的『`.break`』。

链式项目中的环境变量继承

如果项目是从通过/失败链启动的，那么系统将应用来自于调用项目中的环境变量。被调用方项目按以下顺序设置调用方项目环境及其自己环境中的变量：

1. 被调项目服务器环境。
2. 调用项目的变量集，“BF_”变量名更改为“BF_CALLER_”。
3. 被调用项目的服务器环境（第二次应用，以防它被调用方变量修改）。
4. 被调项目环境。
5. 运行时的步骤环境（如果已指定）。

启用等待时取消链式项目

通常，系统不会取消链式项目。将“通过等待”或“失败等待”属性设置为“是”可以让系统自动取消“通过链”或“失败链”的被调项目。当取消了主调项目或主调步骤时，系统会取消被调项目。

定义标记

系统使用标记来标识项目的特定作业，并用于构造缺省情况下在其中发生进程活动的作业目录的名称。系统根据项目的标记格式属性构造作业的标记，该属性可包含静态文本以及数字标记变量。

项目的缺省标记格式为 **BUILD_\$B**，它使用缺省标记变量 **B**，这是系统为每一个项目定义的自动递增值。此缺省标记格式将产生如下的构建标记串：

BUILD_1

BUILD_2

BUILD_3

但是，您并非局限于只能使用这些标记。您可以定义自己的标记变量，并设置自己的标记格式以形成各种标记类型。您还可以在作业期间使用 **.retag** 命令将标记更改为任意字符串。（有关更多信息，请参阅第 347 页的『**.retag**』。）

当前作业的标记在作业期间可用作系统定义的环境变量（**BF_TAG**），这样您就可以访问该标记，并用它来标注源存储库，或者用于其他跟踪或标注目的。（有关这些变量的更多信息，请参阅第 267 页的『系统变量引用』。）

您可以同步两个项目的标记变量；这样就创建了一个链接，当其中任一个项目运行时，就可以使用相同的标记变量值。（有关更多信息，请参阅第 287 页的『同步标记』。）

本部分的主题描述如何设置标记格式和标记变量，以生成反映您所需要的值的动态标记。

编辑项目的标记格式

标记格式定义了系统如何构造标记。标记格式由纯文本和 **\$** 符号表示的变量引用组成。您在标记格式中使用的所有变量都必须来自上一节中的系统定义的标记变量列表，否则您必须在项目运行前为项目定义这些变量。未定义的变量视作静态文本。

标记格式是一种项目属性。要编辑它，请单击“项目”按钮以显示项目列表，然后单击要编辑的项目的项目名称。系统显示项目中步骤的列表；单击页面顶部的项目名称显示项目属性。

在您的标记格式中，使用 **\$** 符号表示标记变量的开始。您可以包含多个标记变量（如愿意）。例如，可以为项目的主修订版（**\$MAJ**）定义非递增变量，为次修订版（**\$MIN**）定义递增变量，然后使用一个反映项目版本号的标记格式，例如 **Version\$MAJ.\$MIN**。这样允许您手动控制主版本号，但将对每个发行版自动增加次版本号，并产生类似以下标记：

Version1.1

Version1.2

同步标记

可利用项目级“标记同步”属性同步不同项目的标记，这样两个或两个以上项目就能使用相同的变量值。当您将 Project B 的“标记同步”属性设置为等于 Project A 时，即在 Project A（父）和 Project B（子）之间建立了父子关系。

运行带有“标记同步”属性的项目时，系统将查看子项目中的任何标记变量是否与父项目中的标记变量匹配。如果找到，那么子项目的变量将设置为父项目的上次使用的值。

如果子项目的标记格式中没有变量与父项目的标记格式中的变量匹配，“标记同步”属性无效。

同步只对标记变量的值有效。将不同步“自动递增”和“填充”属性。

只有该标记中的变量会同步，因此您仍然可以区分不同的项目。

例如，假设定义了下表所示的两个项目：

项目	标记格式	自动增加	标记同步
Project A	Project_A_\$B	是	-- 无 --
Project B	Project_B_\$B	是	Project A

然后您交替运行这两个项目（从 Project A 开始），完成后的作业列表将显示如下标记。最后一次运行的先显示，相同方式的作业将显示在完成后的作业列表中。

项目	标记
Project B	Project_B_4
Project A	Project_A_3
Project B	Project_B_2
Project A	Project_A_1

如果您只在父项目上将“自动递增”属性设置为“是”，那么结果是不同的。项目设置如下：

项目	标记格式	自动增加	标记同步
Project A	Project_A_\$B	是	-- 无 --
Project B	Project_B_\$B	否	Project A

然后您交替运行这两个项目（从 Project A 开始），完成后的作业列表将显示如下标记。最后一次运行的先显示，相同方式的作业将显示在完成后的作业列表中。

项目	标记
Project B	Project_B_2
Project A	Project_A_2
Project B	Project_B_1
Project A	Project_A_1

标记的系统定义变量

可在作业标记中使用下列预定义变量:

变量	值
B	作业号: 从 1 开始的整数值, 每出现一个新作业, 作业号将增 1。
BF_D	日期, 格式由“标记: 日期格式”系统设置设定。该变量在执行时求值, 因此不能在项目起始页上的标记示例中求值。如果使用, 请以 { } 将其括起。
BF_J	当年的第几天。该变量在执行时求值, 因此不能在项目起始页上的标记示例中求值。如果使用, 请以 { } 将其括起。
BF_T	时间, 格式由“标记: 时间格式”系统设置设定。该变量在执行时求值, 因此不能在项目起始页上的标记示例中求值。如果使用, 请以 { } 将其括起。
BF_W	星期几 (数字值, 范围从 0 到 6) 该变量在执行时求值, 因此不能在项目起始页上的标记示例中求值。如果使用, 请以 { } 将其括起。

起始页上系统定义的变量和标记示例


某些变量不能在起始页上的标记示例中求值, 原因是这些变量是在作业启动时求值的。如果您使用这些变量, 请以 { } 将其括起。

示例:

```
BUILD_$B.{BF_T}
BUILD_$B.{BF_D}.{BF_T}
```


创建或编辑标记变量

您可以定义自己的标记变量来包含在标记格式中。标记变量采用数字值, 且可以针对每个作业由系统自动递增 (如需要)。

要为项目添加新变量或编辑现有变量，请选择项目，然后单击所需项目名称旁边的编辑图标 。项目属性显示在面板底部；单击**标记**选项卡以显示项目的标记变量。

系统显示项目的标记变量列表。



- 要编辑某个标记变量，请单击其名称旁边的编辑图标 。系统使用标记变量值填写面板，并将**创建**更改为**保存**按钮。更改值，并单击**保存**来存储您的更改。
- 要删除标记变量，单击其名称旁的废纸箱图标。
- 要添加新的变量，输入变量的属性，并单击**创建**按钮。

有关属性“标记名称”、“初始值”、“填充”和“自动增加”的信息，请参阅第 281 页的『“标记”选项卡』。

库

库是“选择器”属性设置为“无”的任意项目。库用于在其他项目中运行。它们在调用它们的步骤的服务器资源上运行。

当您保存选择器为“无”的项目时，系统警告您该项目将另存为库。库列在**库**面板中。

要从某个步骤调用库，请在该步骤的“内联”、“通过链”或“失败链”属性中选择该库。

关于库

“库”面板包含一些库，这些库是未指定选择器的项目。

库使用调用它们的任何步骤的选择器。如果调用步骤无选择器，那么库使用步骤的项目的选择器。库通常由其他项目作为某个步骤的“内联”或某个步骤的“通过链”或“失败链”来调用。

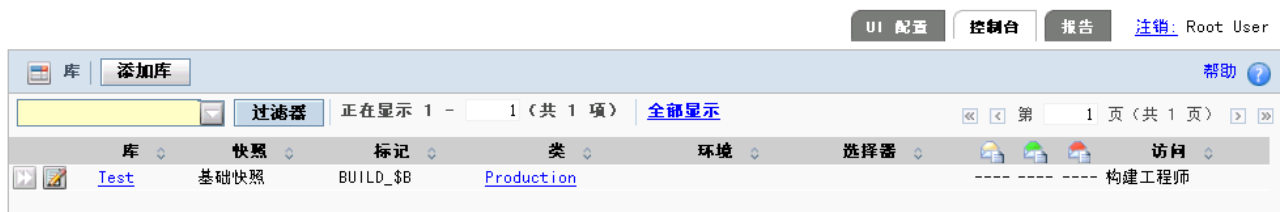
从“库”面板中，您可以查看、编辑、创建或启动库。可以让库独立运行，但在执行此操作时必须指定一个选择器。

注：启动库时，缺省情况下将指定列表中的第一个选择器。如果想指定不同的缺省选择器，请使它成为一个项目。

您可以通过编辑项目并为其选择一个选择器，将库更改为项目。在保存带有选择器的库时，它将变为项目，并从“库”列表中消失。


除了缺少选择器之外，对待库与对待任何其他项目的方式都相同。

要访问“库”面板，请单击左侧菜单中的库。



复制库

要制作现有库的副本，请完成以下步骤：

1. 选择库。
2. 选择想要复制的库旁边的编辑图标 .
3. 单击复制库。

在复制库时，系统将现有库的以下方面复制到新库中：

- 步骤及其在步骤的详细信息选项卡上列出的所有属性
- 在库的详细信息选项卡上列出的所有库属性，例如，类和其他属性
- 库的标记格式（位于库属性面板上的标记选项卡中）

系统不会复制以下属性：

- 标记变量（位于库属性面板上的标记选项卡中）；然而，复制标记格式中使用的标记变量
- 库寄存器（位于库属性面板上的寄存器选项卡中）
- 步骤注释（位于步骤属性面板上的注释选项卡中）
- 库注释（位于库属性面板上的注释选项卡中）

日志过滤器

本主题描述如何创建和使用日志过滤器。

关于日志过滤器

使用日志过滤器可指定步骤的成功条件。过滤器存储一个或多个正则表达式。

若未设置过滤，那么 Build Forge 将按步骤命令的退出状态确定该命令是成功还是失败，其中 0 表示成功，1 表示失败。如果在“命令”属性中使用了多个命令，那么只有最后发出的命令的退出状态会影响步骤结果状态。请注意：某些命令始终返回 0 退出状态。类似 net use 的报告命令打印出映射的网络驱动器列表。该命令始终是成功的，即便列表不包含所需的驱动器。

日志过滤器用于评估步骤命令的输出（而不是退出状态）。

例如，对于 net use 命令，可以使用日志过滤器来查找特定驱动器，并在找到该驱动器时将步骤标记为成功。

关于“日志过滤器”面板

使用“日志过滤器”面板可创建、编辑和删除日志过滤器。要查看该面板，请选择项目 > 日志过滤器。



该面板具有以下字段。

名称 日志过滤器的名称。

访问 可使用此过滤器的用户的访问组。

创建日志过滤器

日志过滤器可以包含一个或多个过滤器模式。每个过滤器模式都与一个操作相关联，并可选择与一个访问组进行关联，以发送通知。您先定义一个日志过滤器，然后将日志过滤器与项目中的步骤相关联。

要创建日志过滤器，请执行以下操作：

1. 选择项目 > 日志过滤器。“管理控制台”会显示“日志过滤器”列表和“新建日志过滤器”面板。
2. 在**名称**中，输入日志过滤器的名称，然后单击**保存**。“管理控制台”将会保存日志过滤器，并显示“新建模式”面板。
3. 对于为日志过滤器定义的每个过滤器模式，执行以下操作：
 - a. 在**模式**字段中输入一个正则表达式。（正则表达式必须与 Perl 兼容。）在项目运行时，Build Forge 会在步骤输出中搜索该模式。有关详细信息，请参阅第 292 页的『过滤器模式』。
 - b. 在**操作**字段中，选择在找到过滤器模式时执行的过滤器操作。缺省属性“设置失败”会将步骤状态设置为“失败”。有关详细信息，请参阅第 293 页的『过滤器操作』。
 - c. （可选）在“通知”中选择一个访问组，以便在过滤器被激活时向其成员发送通知电子邮件。
 - d. 单击**保存**。

要使用日志过滤器，请选择一个项目步骤，并将该步骤的**结果**属性设置为新的日志过滤器。请参阅第 292 页的『向步骤指定日志过滤器』。

向步骤指定日志过滤器

要使用日志过滤器，必须使用步骤**结果**属性将日志过滤器指定给项目步骤。当您向步骤指定日志过滤器时，每当项目运行时，都将在步骤输出上运行日志过滤器中的过滤器模式。但是，日志过滤器不能应用于适配器输出。

请注意：当您向步骤指定日志过滤器时，日志过滤器设置的步骤结果将覆盖所有其他用于确定该步骤是成功还是失败的条件。这其中包含步骤命令的退出状态或所有步骤属性。例如，如果步骤运行时间超出步骤“超时”属性指定的时间，那么步骤停止。但是，其状态不会视作失败，除非其关联的日志过滤器操作使其设置为失败。

要向步骤指定日志过滤器，请执行以下操作：

1. 选择**项目**或**库**来访问步骤。
2. 选择包含该步骤的项目或库。
3. 选择步骤来打开步骤“详细信息”面板。
4. 在**结果**中，选择要在每次步骤运行时运行的日志过滤器。

过滤器模式

过滤器模式定义希望在步骤输出中进行匹配的字符串或表达式。所创建的每个过滤器模式都与单个过滤器操作相关联。过滤器模式和操作都定义在过滤器日志集中。在日志过滤器中包含多个过滤器模式并将其应用于单个步骤的输出，这一能力使您无需构造复杂的表达式即可以使用多个搜索条件。

要创建日志过滤器，请选择**项目 > 日志过滤器**。有关详细信息，请参阅第 290 页的『日志过滤器』。

过滤器模式语法

查看以下过滤器模式创建准则：

- 过滤器模式定义为正则表达式，必须使用与 Perl 兼容的语法。关于构造 Perl 兼容表达式的详细信息，请参阅 Perl 文档。
- 系统会添加定界正斜杠字符 (`/<expression>/`)，所以指定表达式时，**不要**在其前后添加正斜杠定界符（表达式）。
- 如果表达式中含有元字符（例如，`a/b`），那么元字符前必须带有反斜杠转义字符（`a\b`）。

下表显示了一些标准正则表达式的语法。

表达式	匹配项
<code>Production</code>	匹配字符串中任意位置的 <i>Production</i> 。
<code>^Production</code>	匹配字符串开始位置的 <i>Production</i> 。
<code>Error.*[0-9]\$</code>	匹配符合以下条件的行：含有 <i>Error</i> ，其后跟随任意字符串，并且字符串尾部以一个数字终止。
<code>[Ww]arning</code>	匹配 <i>Warning</i> 或 <i>warning</i> 。

表达式	匹配项
.*	匹配任意字符 0 次或多次。点号 (.) 匹配任意字符，星号 (*) 匹配任意字符 0 次或多次。

同一行上的多个模式匹配

要构造模式过滤器，了解系统如何搜索模式匹配项非常重要。

对于每一个输出行，系统将按顺序根据所有过滤器模式来检查匹配项；在找到匹配项时，它将停止，并继续处理下一个模式。所以，如果模式在一行中出现两次，系统可能不会发现它。例如，考虑以下输出行：

```
exception retrying exception
```

使用下表中的过滤器模式，系统将匹配第一个 *exception*，并将步骤结果设置为“失败”，匹配 *retrying* 并将步骤结果设置为“通过”，然后继续处理下一行，而不匹配第二个 *exception*。

过滤器模式	过滤器操作	示例描述
[Ee]xception [Rr]etrying	设置失败 - 失败 清除失败 - 通过	这对于 Java 项目非常有用；它在产生异常时将步骤设为失败，但在重试时清除失败。如果重试失败，那么将产生新的异常，所以命令的最终状态是有效的。

解决该问题的一种方式是将表中的过滤器模式替换为以下过滤器模式：

```
retrying.*exception
```

过滤器操作

过滤器操作定义了当在步骤输出中找到过滤器模式时执行何种操作。所创建的每个过滤器模式都与单个过滤器操作相关联。过滤器操作和模式都在日志过滤器中定义。

要创建日志过滤器，请选择项目 > 日志过滤器。有关详细信息，请参阅第 290 页的『日志过滤器』。

过滤器操作	定义	步骤结果
设置失败（缺省值）	当系统找到过滤器模式时，它会将步骤结果状态设置为失败，并继续在当前行中搜索集合中的过滤器模式。	失败
设置失败/中止	当系统找到过滤器模式时，它会将步骤结果状态设置为失败，停止在设置中的当前行搜索过滤器模式，跳至下一行，并再次开始模式搜索。	失败
清除失败	当系统找到过滤器模式时，它会将步骤结果状态设置为通过，并继续在设计中的当前行搜索过滤器模式。	通过
清除失败/中止	当系统找到过滤器模式时，它会将步骤结果状态设置为通过，停止在设计中的当前行搜索过滤器模式，跳至下一行，并再次开始模式搜索。	通过
中止	当系统找到过滤器模式时，它将停止在设计中的当前行搜索过滤器模式，跳至下一行，并再次开始模式搜索。它不会更改步骤结果状态。	不适用

过滤器操作	定义	步骤结果
包含	包含允许您引用其他日志过滤器中的一个或多个过滤器。您指定要包含在“模式”字段中的日志过滤器，并在“操作”字段中选择“包含”。	不适用
警告	当系统找到过滤器模式时，它会将步骤结果状态设置为“警告”，并继续在设置中的当前行搜索过滤器模式。（注：“警告”状态为通过状态；指定给此步骤的任何通过链都将运行。）	警告
清除警告	当系统找到过滤器模式时，它会将步骤结果状态设置为通过，并继续在设置中的当前行搜索过滤器模式。	通过
清除警告/中止	当系统找到过滤器模式时，它会将步骤结果状态设置为通过，停止在设置中的当前行搜索过滤器模式，跳至下一行，并再次开始模式搜索。	通过
通知更改人	<p>要使用“通知更改人”，必须在项目中包含一个创建关系列表的适配器，并且该适配器步骤的运行必须先于包含“通知更改人”日志过滤器的步骤。</p> <p>适配器关系列表将用户和对象（例如已更改的文件）组成对。有关详细信息，请参阅“适配器 XML 引用”。</p> <p>在适配器运行并创建关系后，如果“通知更改人”操作的日志过滤器在步骤输出行中匹配其过滤模式，将再次扫描该行以尝试匹配关系列表中的对象。如果找到匹配的对象，将向与该对象配对的用户发送电子邮件通知。</p> <p>例如，在以下步骤输出行中，与过滤模式 Error 匹配的对象是 MyFile.c。因此将向关系列表中与 MyFile.c 对象配对的用户发送有关错误的电子邮件通知。</p> <p>Error: Invalid token on line 55 of MyFile.c</p>	不适用
停止构建，结果为失败	当系统找到过滤器模式时，它会将作业结果设置为失败，并退出该作业。该步骤结果根据其结果而设置。不运行更多步骤。	结果
停止构建，结果为通过	当系统找到过滤器模式时，它会将作业结果设置为通过，并退出该作业。该步骤结果根据其结果而设置。不运行更多步骤。	结果
停止构建，结果为警告	当系统找到过滤器模式时，它会将作业结果设置为警告，并退出该作业。该步骤结果根据其结果而设置。不运行更多步骤。	结果

过滤器通知

对于日志过滤器中的每个过滤器模式，都可以有选择地设置通知来向访问组发送电子邮件，以通知各成员步骤的模式过滤器已激活。

错误阈值

可以使用特殊的环境变量 `_ERROR_THRESHOLD` 来为个别步骤和/或项目建立阈值。随即系统计算过滤器匹配项的数量，当达到阈值时令步骤或项目失败，或在作业说明中注明已达到阈值的事实。

有关更多信息，请参阅第 270 页的『触发器变量引用』。

错误和警告计数

如果将过滤器与步骤相关联以确定步骤是成功还是失败，系统将显示由过滤器捕获的错误数和警告数。在**作业 > 已完成**选项卡中，这些数字在**结果**列中作业结果后的括号内显示。格式为（<失败匹配数> / <警告匹配数>）。

示例：作业结果**失败（1 / 0）**表明作业已失败，有一个匹配的“失败”过滤器，没有匹配的“警告”过滤器。

类

本主题描述如何创建和使用类。

关于类

类是一组作业。每个作业必须且只能是一个类的成员。使用类可以对类中的每个作业应用不同的全局管理行为。作业从其项目的属性中获取缺省类。当从**作业 > 启动**页启动作业时，还可以为其手动选择不同的类。

类有用于管理下列活动的属性：

- 自动删除作业
- 在系统清除此类的作业时，或者在现有作业与此类的作业相互转化时启动作业。

注：可在作业完成后更改该作业的类。要更改作业的类，请选择**作业 > 已完成**来查看作业，然后单击作业标记（例如，BUILD_5）。在**类**字段中选择其他类。

关于“类”面板

使用“类”面板可添加、编辑和删除类。要查看该面板，请选择**项目 > 类**。系统将显示类列表。选择一个类以编辑其属性。

UI 配置

控制台

报告

注销: Root User

类

添加类

帮助

过滤器

正在显示 1 - 2 (共 2 项)

全部显示

构建类	访问	删除	天	计数	哪一个
Production	访问	控制台数据	始终	无限制	所有构建
Scratch	访问	所有内容	5	无限制	所有构建

(新建类)

保存

复制

删除

详细信息

名称:

访问:

构建工程师

删除文件:

在清除时启动:

所有内容

-- 无 --

天:

在进入时启动:

始终

-- 无 --

计数:

在退出时启动:

无限制

-- 无 --

哪一个:

所有构建

类的**访问**属性根据您指定的访问组来控制哪些用户可查看或更改类。

用于自动删除作业的属性:

类的大多数属性用于控制删除何种项目数据，以及在什么情况下删除它们。

系统按“清除检查时间”系统设置定义的时间间隔（缺省为 15 分钟）检查要删除的作业。

注：也可使用调度来表示应执行清除的时间，这样当系统被占用时就不必尝试运行清除。可以使用此功能执行清除，例如只在晚上或每周发生一次。请参阅第 295 页的『类』。

当运行清除作业时，系统对作业进行归档，然后根据类中的设置删除数据。

删除文件

确定要删除何种数据。它具有以下选项:

所有内容

从数据库删除有关作业的所有信息，并从运行该作业的服务器删除作业目录。

控制台数据

从数据库删除作业的所有相关信息，但原封不动地将作业目录保留在服务器上。

日志与文件

删除作业目录和日志，但保留作业 > 已归档页面上的步骤通过/失败信息。

仅日志 仅删除作业日志。

仅文件 删除运行作业的服务器上的作业目录。日志和某些其他信息（例如，步骤通过/失败状态）仍然保留在数据库中；作业记录则移至作业 > 已归档页面。

天数* 作业被删除前必须经过的天数。

计数* 允许的最大作业数。当作业数超出“计数”值时，系统将调度清除作业来删除多余的构建。缺省值 Unlimited 防止系统因存在的作业数删除作业。

条件 “条件”属性设置删除作业前必须满足的附加条件。它具有以下选项:

所有构建

选择此选项时，“哪个”属性对作业删除无效。

仅已失败

系统只删除失败的作业。

仅已通过

系统只删除通过的作业。

保持 1 次通过

系统始终保留最近通过的作业，即便它满足其他删除条件。

* 当超出“天数”或“计数”值时，系统将删除作业。例如，如果将“计数”设为 10，“天数”设为 2，而且有 8 个作业，但 3 个超出 2 天的时长，那么将删除这 3 个作业。类似地，若有 12 个作业，全部都小于 2 天的时长，那么将删除两个最旧的作业。

当运行清除作业时，系统对作业进行归档，然后根据类中的设置删除数据。

在事件发生时启动项目的属性:

您可以在发生某些与类相关的事件时启动（链）项目。使用以下属性，您可以在进程中建立状态进展模型。

以下类属性使您可以在发生某些事件时启动作业:

在清除时启动

该属性在类中的任何作业被清除时（也即，每当系统对具有该类的作业启动清除作业时），启动指定的项目。可以使用该属性来确保删除一些不会随清除自动删除的特定文件。

在进入时启动

该属性在作业的属性更改为该类型时启动指定的项目。可以使用该属性将某个进程关联到作业重新分类；例如，您可以创建一个 `Test` 类，并在作业提升为 `Test` 类时启动一些标准测试。

在退出时启动

该属性在作业的属性从该类更改为其他类时启动指定的项目。

这些属性将项目作为链来启动。

注：如果这些属性启动一个包含“必须更改”变量的作业，那么变量值不会更改。

设置通知

系统可在项目或步骤通过或失败时，或者发生其他特定事件时发出电子邮件通知。本部分描述如何配置电子邮件通知，以及如何修改控制电子邮件通知外观的通知模板。

通知是发送给访问组，因此在设计访问组时不仅要考虑安全性，还要考虑到通知。通知总是发送到组，而不是直接发给单个用户，但是如果需要，可设置只包含一个用户的组。

如果已将访问组设置为分层（访问组包含子组），那么通知行为如下:

- 如果通知的访问组为父组且包含一个或多个用户，那么通知将只发送给父组中的用户。
- 如果通知的访问组为父组但不包含用户，只包含子组，那么通知将发送给所有子组用户。

要创建通知事件，请为通知属性选择一个访问组。

- 对于项目，可选择“启动通知”、“通过通知”和“失败通知”组。
- 对于步骤，可选择“通过通知”和“失败通知”组。

只要为这些属性选择了访问组，系统将在发生相应事件时向组成员发送电子邮件。对这些电子邮件消息中时间的任何引用都基于发送通知的“管理控制台”的时区。

当项目包含作为内联项目的其他项目时，系统将忽略内联项目的启动、通过和失败通知设置，但是其步骤的所有通知设置仍然有效。请参阅第 304 页的『内联项目的通知』以了解更多详细信息。

在可以使用通知之前，您必须:

- 配置 SMTP 服务器系统设置，使系统知道应使用哪个 SMTP 服务器来发送电子邮件。缺省值为 localhost。可能还需要设置系统警报源的设置。此地址可用作源地址，并且大部分 SMTP 服务器要求有效源地址。缺省值为 root@localhost。
- 创建一个或多个通知组并为它们分配用户。
- 为各个项目和/或步骤选择要通知的组。

此外，通过编辑通知模板，还可以配置系统发出的通知电子邮件。请参阅第 301 页的『定制通知模板』。

关于通知模板

通知模板提供了一种向用户发送关于系统中的事件的定制消息的方法。

提供的模板类型

系统包括一组通知模板类型，如下所示：

- 更改工件匹配消息
- 过滤器匹配消息
- 作业 .break 消息
- 作业失败消息
- 作业通过消息
- 作业启动消息
- 作业警告消息
- 清除失败消息
- 步骤 .email 消息
- 步骤失败消息
- 步骤通过消息
- 步骤警告消息
- 系统警报消息

每种类型包含一组已转换的消息。消息使用变量提供上下文。

全局用法

可以全局使用提供的模板，如下所示：

- 项目：每个项目中的属性允许您针对项目启动、通过和失败指定通知类型。针对作业完成时带有警告的情况，提供了“作业警告”通知。
- 步骤：每个步骤中的属性允许您针对步骤启动、通过和失败指定通知类型。针对步骤失败后作业继续的情况，提供了“步骤警告”通知。
- 其他通知类型如下所示：
 - 更改工件匹配消息 - 工件已更改，并且与某个用户相关
 - 作业 .break 消息 - 作业遇到包含 .break 命令的步骤
 - 步骤 .email 消息 - 步骤执行 .email 命令
 - 系统警报消息

转换中存储的消息

使用的通知消息存储在与模板类型相关联的转换中。每个转换指定与其通知消息相关的信息：

- 语言
- 描述
- 主题（用作通知电子邮件中的主题）
- 正文（用作通知电子邮件中的消息体）

可以编辑提供的通知模板和创建的通知模板中的转换。

消息格式

可以使用以下两种消息类型之一发送消息：

- 纯文本消息（缺省值）
- MIME - “正文”字段中包含某些 HTML 标记的通知将自动作为 MIME 消息发送。
请参阅第 302 页的『在“正文”字段中使用 HTML』。

模板中的变量

系统会解析模板以获取一些变量（请参阅第 302 页的『使用环境变量和寄存器变量』）。

指定项目和步骤的定制模板

您可以根据提供的通知类型创建定制的通知模板。定制的模板必须与项目或项目内的个别步骤相关联。请参阅第 301 页的『为特定项目和步骤新建模板』。

关于“模板”面板

使用“模板”面板可设置关于事件的通知。要查看该面板，请选择项目 > 模板。



该面板列出 Rational Build Forge 随附的所有模板。创建新模板时，指定以下字段：

类型 消息的类型。选择现有的模板类型。

项目 必需：要为其发送消息的项目。

步骤 可选：要为其发送消息的指定项目中的步骤。

关于“转换”面板

使用“转换”面板以设置模板使用的消息。要查看该面板，请选择**项目 > 模板**，然后从列表中选择模板。此时将显示模板的转换列表。列表中的每项显示以下信息：

语言 必需。消息的语言。

描述 用作列表中“转换”的标题。

发件人 可选：要在消息的“发件人”字段中使用的电子邮件地址。支持以下电子邮件地址格式。*Username* 是电子邮件用户名。*Name* 是任意字符串，通常是某种形式的给定名称。

```
username@host.com
name username@host.com
name "<username@host.com>"
name <username@host.com>
name "username@host.com"
```

主题 必需：要在消息的“主题”字段中使用的字符串。它可以包含变量。

正文 必需：消息的文本。它可以包含变量。

配置 SMTP 服务器

要配置 SMTP 服务器，选择**管理 > 系统 > SMTP 服务器**。系统将显示一个有关 SMTP 服务器值的编辑面板。输入您站点的 SMTP 服务器的名称。缺省值为 localhost。

您可能还需要设置**系统警报源**参数，因为该参数将用作源地址，并且大多数 SMTP 服务器需要有效的源地址。缺省值为 root@localhost。

有关更多信息，请参阅第 208 页的『系统配置设置』。

设置项目和步骤的通知属性

当用于通知的 SMTP 和组配置就位时，可将项目和步骤配置为在某些特定事件发生时发送通知。

- 对于项目，可设置“启动通知”、“通过通知”和“失败通知”属性。这些是项目属性。
- 对于步骤，可设置“通过通知”或“失败通知”属性。这些是步骤属性。

如果您在单个步骤内设置了多个通知，那么仅第一个匹配项生成电子邮件通知。

通知练习

以下过程描述如何设置和尝试电子邮件通知。练习需要 SMTP 服务器和电子邮件帐户。

1. 按照『配置 SMTP 服务器』中的描述设置 SMTP 服务器

请确保具有一个用户帐户，可通过该帐户向您可访问的电子邮件帐户发送电子邮件。

2. 选择**管理 > 访问组**
3. 创建名为 Email Test 的新访问组，并将您选择的用户设置为“初始成员”。
4. 选择一个项目（例如，Hello World 项目），并编辑其项目属性。在“启动通知”、“通过通知”和“失败通知”字段中选择 Email Test 组。
5. 运行项目。

6. 验证是否接收到两封电子邮件：一封指示项目已启动，另一封指示它已成功或失败。如果未收到电子邮件，请验证是否使用了正确的 **SMTP** 服务器值。

定制通知模板

您可以按如下所示定制通知模板：

- 为项目和步骤创建定制模板：您可以为个别项目以及项目内的个别步骤创建定制模板。使用所提供的模板类型创建定制模板。可以在创建的定制模板中创建转换。
- 针对所提供的模板编辑转换：您可以在每个提供的通知模板中添加或编辑转换。不能编辑所提供模板的基本特性。

为特定项目和步骤新建模板

系统随附系统中可能发生的许多事件的模板。可以创建特定于特定项目或项目内步骤的新模板。在创建定制通知之前，必须在系统中创建项目或步骤。

要创建新模板，请执行以下操作：

1. 选择**项目 > 模板**。系统显示当前模板列表。
2. 单击**添加模板**。
3. 指定模板的信息：
 - **类型**：选择现有通知类型以用作新模板的基础。
 - **项目**：选择新模板适用于的项目。模板仅用在通知消息上，这些消息在运行选定项目时生成。

注：如果为项目定义了多个快照，那么将显示快照列表。选择要使用的快照。通知仅将应用于使用该快照运行的项目。如果您指定“缺省快照”，那么使用缺省值时会指定快照。

- **步骤**：可以选择一个特定步骤（因此模板仅适用于该步骤的通知）或选择“项目事件”以使模板适用于选定项目的所有通知。
4. 单击**保存**。
 5. 为模板创建转换。
 - a. 从**项目 > 模板**中选择模板
 - b. 单击**添加转换**。
 - c. 指定模板的信息：
 - **语言**：必需 - 消息的语言。
 - **描述**：必需 - 该字段用作转换的标题。
 - **发件人**：可选 - 要在消息的“发件人”字段中使用的电子邮件地址。
 - **主题**：必需 - 要在消息的“主题”字段中使用的字符串。它可以包含变量。
 - **正文**：必需：消息的文本。它可以包含变量。如果它包含某些 **HTML** 标记，那么消息会作为 **MIME** 消息（而不是纯文本）发送。
 - d. 单击**保存**。
 - e. 根据需要，对其他转换重复以下过程。

编辑通知模板中的转换

每个通知模板都使用转换列表以存储用于通知的消息。

可以针对提供的通知模板以及为个别项目和步骤创建的通知模板编辑转换。

要编辑转换，请执行以下操作：

1. 从项目 > 模板中选择要编辑的模板。此时会显示转换列表。
2. 单击要编辑的转换。
3. 指定模板的信息：
 - **语言：**必需 - 消息的语言。
 - **描述：**必需 - 该字段用作转换的标题。
 - **发件人：**可选 - 要在消息的“发件人”字段中使用的电子邮件地址。
 - **主题：**必需 - 要在消息的“主题”字段中使用的字符串。它可以包含变量。
 - **正文：**必需：消息的文本。它可以包含变量。如果它包含某些 HTML 标记，那么消息会作为 MIME 消息（而不是纯文本）发送。
4. 单击保存。

在“正文”字段中使用 HTML

缺省情况下，通知会作为纯文本进行发送。如果您使用以下任何 HTML 标记或部分，那么通知会作为 MIME 消息进行发送并且会解释所有 HTML 标记：

```
<html
<body
<a
<b>
<p>
<table
```

创建转换时，系统会解析“正文”字段。它会确定当时要使用的格式。系统仅扫描列出的标记和标记部分。

如果消息未包含要扫描的任何元素，那么消息会作为纯文本进行发送。不会解释消息中的 HTML 标记。标记将包含在消息中。

使用环境变量和寄存器变量

只要使用了 \${VAR} 语法，就可在通知模板中引用环境变量（您定义的环境变量以及标准系统变量）。

还可在通知模板中包含项目的寄存器变量。如果引用空寄存器，系统将返回空字符串。

特殊通知模板变量

下表列出了可用于通知模板的特殊变量。某些变量与上下文相关，仅在相关时可用（例如，不会对项目级别的通知而仅对步骤级别的通知设置 STEPNAME 变量）。

变量	包括
ACTION	对于构建清除，描述执行的删除类型。

变量	包括
BID	指定作业标识号。用于构造返回到管理控制台以用于访问报告的链接。
CONSOLEHOST	“管理控制台”计算机的主机名。
CONSOLEPORT	管理控制台使用的端口号。便于您在通知模板中构造有效的 URL。
CONTEXTLOGLINKS	列出日志中以“FILT:”开头的行，每个条目三行上下文。系统在消息中提供了管理控制台日志条目的链接。
DURATION	对于步骤，指定步骤以及其内联的任何步骤的运行时间，单位为秒。
EID	指定环境标识号。用于构造返回到管理控制台以用于访问报告的链接。
FULLNORMALLOG	显示作业中每个步骤的日志信息，但不包括详细日志中出现的环境设置操作。
LINK	对于构建，指定链接名称。
MESSAGE	包含故障或警报消息的错误或消息文本。
ONFAIL	对于步骤，保存步骤的连续属性。
PATH	对于步骤，为服务器或步骤之类的数据项，指定相应的路径。
PID	指定项目标识号。用于构造返回到管理控制台以用于访问报告的链接。
PROJECTNAME	包含项目的名称。
RUNACTION	指定电子邮件模板使用的变量。
SELECTOR	包含步骤或项目的选择器名称。
SERVER	包含步骤或项目的选择器名称。
SID	对于步骤，指定步骤标识号。用于构造返回到管理控制台以用于访问报告的链接。
SRVRHOST	对于步骤，包含步骤的服务器的 TCP/IP 主机名。
START	包含作业启动的日期/时间。
STEPNAME	对于步骤，包含步骤的名称。

变量	包括
STEPNORMALLOG	对于步骤，显示作业中当前步骤的日志信息，但不包括详细日志中出现的环境设置操作。
TAG	包含作业的标记字符串。值与 \$BF_TAG 相同。
TAILNORMALLOG	对于步骤，作用类似于 STEPNORMALLOG，但是只显示日志的结尾部分。 对于构建，作用类似于 FULLNORMALLOG，但只显示每个步骤的日志的结尾部分。 显示的行数由邮件模版的尾部日志量系统设置控制。
UID	指定用户标识号。用于构造返回到管理控制台以用于访问报告的链接。
USEREMAIL	包含作业/事件的所有者的电子邮件地址。
USERNAME	包含作业/事件的所有者的全名。

内联项目的通知

系统按照将内联项目的步骤视为是嵌入调用的项目的方式来处理内联项目的通知。

- 当某个项目含有内联项目时，内联项目的项目级别通知设置会被忽略。当项目 A 将项目 B 内联时，将不会发送任何关于项目 B 的启动、通过或失败的消息。
- 步骤级别通知设置不受影响。如果步骤设置了“通过通知”或“失败通知”访问组，那么无论步骤是处于顶级项目还是内联项目中，都将会发送相应的消息。
- 内联步骤会对确定调用的项目是成功还是失败产生影响。例如，如果在内联步骤中发生失败，那么或者会导致调用项目失败，或者，如果步骤设置为“失败时继续”，那么将调用项目的状态更改为“失败但继续”。

使用快照新建项目实例

对选择器创建快照来快速新建您要更改或修改的项目实例。项目快照是单独的可执行项目。您也可以使用快照来创建库的新实例。

项目快照概述

查看这些主题可以了解项目快照和理解如何使用项目快照。

项目快照用例

下面的示例描述了项目快照的一些常见用例：

- 对项目创建快照以对项目配置作出更改，或者在继续使用现有项目运行作业时执行新工具或脚本的测试。
- 将项目快照作为临时备份或正式归档的组成部分来存储。
- 对项目创建快照以捕获与里程碑对应的时间点项目配置，比如外部或内部发行版。

项目快照概念和术语


在 UI 中，快照引入了关于使用项目的一些新概念和术语。

项目快照：快照是现有项目的新实例。需要紧记关于快照的以下几个要点：

- 快照是单独的项目。更改快照集合中的一个快照并不会影响集合中的其他快照。
- 快照是可执行的项目。它与您在创建项目快照时也选择创建快照的对象一起运行，或者与和源项目关联的对象一起运行，也称作基本快照。
- 快照不是副本。


如果您对与项目关联的对象创建快照，快照将创建该对象的单独实例。复制项目将复制对象之间的关系，但它不会创建选择器、环境或内联或链式项目的新实例。

- 快照不是项目的修订版：
 - 快照不支持比较两个项目快照之间的更改。
 - 不会像在源控制系统中那样用版本号来跟踪或识别对项目快照的更改。但是，可以使用包含版本号（例如，7.5.0、3.4.01）的快照命名方案将项目快照与里程碑关联。

快照集：快照集是作为一个基本快照后代的所有项目快照的集合。该集合至少包括基本快照或父代快照及其子代快照。在 UI 中，项目名称旁边的快照图标  表示已为该项目创建快照集。

Base Snapshot：最初，所有项目都有一个快照名称 Base Snapshot。您可以将 Base Snapshot 更改为其他名称。基本快照是快照集中的父代。

缺省项目快照：缺省项目快照是当前工作项目。一个集合中只能有一个缺省快照。如果未指定缺省快照，基本快照便是缺省快照。

- 在 UI 中，缺省快照在项目列表的顶级显示。选择项目或作业 > 启动以显示项目列表。
- 当您选择带有快照的项目作为内联项目或链式项目时，将使用缺省项目快照，除非您选择了列表框中的其他项目快照。
- 要访问和使用项目快照集中的其他快照，必须单击快照图标 .

项目快照视图

选择快照图标  以显示“快照”视图。在 UI 中，“快照”视图显示集合中快照的层次结构：

- 基本快照位于顶级并且具有名称 Base Snapshot（如果没有为它指定唯一的名称）。
- 所有项目快照都是基本快照的子代。同一个基本快照的子代缩进在“快照”列的同一级别。
- 从子代快照中创建的项目快照将成为该子代快照的子代，并缩进在“快照”列中的下一级别。

项目快照计划

复审选择缺省项目快照和命名项目快照的一些最佳实践。

- 在集合中选择缺省快照的策略

UI 对于一个快照集仅识别一个缺省或当前项目快照。请使用一致的策略来选择缺省快照:

- 使用基本快照作为缺省快照

使用此策略，您可以将快照作为时间点备份来创建，并且不对备份的项目快照作出更改。您只对基本快照作出更改并继续使用基本快照项目来运行作业。

- 使用最新快照作为缺省快照

当您创建新项目时，可使用此策略将它作为新的缺省项目快照。您不对基本快照或早期项目快照作出更改。最新的快照是用于运行作业的快照。

- 确定集合的快照命名方式

项目快照名称在项目快照集内必须是唯一的。

使用下面的条件可帮助您创建项目快照名称:

- 名称应该是描述性的：它应指示快照的用法或用途。
- 命名方案应遵循已定义的标准。可以使用“快照”选项卡上的“注释”框描述命名方案。

- 使用集合的单个项目名称

创建项目快照后，您可以选择更改项目的名称。如果您更改了项目名称，将会为每个项目快照更新该名称。

项目快照选项

在您创建快照时，必须选择要在快照中包含哪些对象。下表描述了可用于 Build Forge 对象的选项。该表包含这些对象对应的列:

- 自动包含在项目快照内的对象。
- 如果您在创建项目快照时选择了这些对象，将选择创建和包含这些对象。

对于这些对象，将在 UI 中使用与项目快照相同的快照名称来创建新对象。例如，如果您的项目快照名称为 release_7.1，那么环境、选择器、内联项目或库以及链接项目或库的快照名称也为 release_7.1

- 对象不包含在项目快照中；必须手动创建这些对象并将它们添加到项目。

自动包含在项目快照中	可选择包含在项目快照中	不包含在项目快照中
	* 仅复制；不创建单独的实例	
项目步骤及其设置（日志过滤器和通知组等）	项目及其步骤的环境	项目注释
项目标记	快照环境的类型为“包含”的环境变量添加的环境	
	内联项目或库及其步骤	
	链接项目或库及其步骤	

自动包含在项目快照中	可选择包含在项目快照中 * 仅复制；不创建单独的实例	不包含在项目快照中
	项目及其步骤的选择器	
	快照选择器的类型为“包含”的选择器属性添加的选择器	
	*项目寄存器（已复制）	
	*项目标记变量值（已复制）	
	*通知模板（已复制）	
	*适配器链接（已链接）	

验证和编辑访问组的快照许可权

验证用户是否具有创建快照和设置缺省快照所需的许可权。若没有，使用访问组为用户指定许可权。

可通过访问组向用户指定许可权，访问组可由 Build Forge 提供或由 Build Forge 管理员创建。

要验证和编辑为快照许可权指定的访问组：




1. 选择**管理 > 许可权**。
2. 在许可权列表中，选择**显示全部**以列出所有许可权。
3. 验证相应的访问组和用户是否有权使用以下快照许可权：

创建快照	创建项目、环境和选择器快照所需的用户许可权。
设置缺省快照	设置或更改项目、环境和选择器的缺省快照所需的用户许可权。

从现有项目或项目快照创建项目快照

创建项目快照会创建您选择创建快照的项目和对象的新实例。快照不是副本；它是项目的可执行实例。

注： 管理控制台不会显示前 2999 个项目后的项目的快照。

1. 单击您要创建快照的项目或项目快照旁边的**编辑**图标：
 - 要对缺省项目快照创建快照，请在项目列表（**项目**）中，单击顶级快照旁边的**编辑**图标 。
 - 要对非缺省项目快照创建快照，请单击**快照**图标 。快照视图会在集合中显示项目快照。单击非缺省项目快照旁边的**编辑**图标 。
2. 单击**新建快照**。
3. 在“快照”选项卡的**名称**字段中，输入快照名称。

该名称在项目快照集内必须是唯一的。该名称将被指定给您使用该项目创建快照的所有对象。

4. 在创建项目快照时，选择要创建快照的 Build Forge 对象。下表中描述了您可以选择的对象。



对象	描述
缺省	在 UI 中，缺省项目快照在项目列表的顶级显示。 选择 项目或作业 > 启动 以显示项目列表。
包括项目环境	对项目和项目中的步骤环境创建快照。
下列环境包含	如果选择了“包括项目环境”，那么还要对类型为“包含”的环境变量包含的所有其他环境创建快照。
包括项目选择器	对项目和项目中包含的步骤选择器创建快照。
包括下列选择器	如果选择了“包括项目选择器”，那么还要对类型为“包含”的选择器属性包含的所有其他选择器创建快照。
克隆项目适配器链接	将适配器链接作为快照的一部分来复制。 适配器链接将适配器添加到项目。适配器作为项目中的第一个步骤（步骤 0）运行。
克隆项目寄存器	将项目寄存器作为快照的一部分来复制。
克隆项目标记变量值	复制项目标记变量的标记值。标记变量可自动复制，但是它们的值不行。如果您不复制标记值，它们将重置为 1。
克隆项目模板	复制在项目级别和步骤级别设置的通过和失败通知事件的通知模板。
包括已链接项目	对在项目级别或步骤级别引用的已链接项目或库及其步骤创建快照。 链会被项目的通过/失败或步骤的通过/失败条件触发。
包括项目内联	对在步骤级别引用的内联项目及其步骤创建快照。 内联会被步骤触发，并在步骤完成后运行。

5. 单击**保存**以保存项目快照。

更改缺省项目快照

缺省项目快照是项目快照集中的顶级快照，它显示在项目列表（**项目**）中。

要更改缺省项目快照，请编辑您要作为新缺省快照的快照定义：

1. 选择**项目**。
2. 在项目列表中，单击缺省项目快照的 **快照**图标 .
3. 在快照列表中，单击要作为新的缺省快照的项目快照的**编辑**图标 .
4. 单击**设为缺省值**。
5. **要点：**在弹出窗口中，选择“确定”或“取消”。



确定	更新引用： 对于引用先前缺省值的任意对象，将先前缺省项目快照的引用更新为新的缺省值。
取消	不更新引用： 对于引用先前缺省值的任意对象，不将引用更新为新的缺省项目快照。

更改项目快照的快照名称

您可以更改项目快照以及您在创建项目快照时选定要创建快照的对象的快照名称。

对于基本快照（仅单个项目快照或所有当前和未来项目快照），可以使用此选项将其缺省名称“Base Snapshot”更改为其他快照名称。

要更改快照名称，请执行以下操作：

1. 选择项目。
2. 在项目列表中，单击缺省项目快照的 **快照**图标 。
3. 在快照列表中，单击项目快照的**编辑**图标 。
4. 选择**快照**选项卡。
5. 在**名称**中，输入新的名称。
6. 可选：在**注释**中，输入注释。
7. **要点**：在弹出窗口中，选择“确定”或“取消”。


确定	<p>更改项目快照名称和其他快照对象名称：对于您在创建项目快照时选定要创建快照的对象，更改这些对象的名称以及项目快照的名称。</p> <p>对于 Base Snapshot：更改所有当前项目快照和所有未来项目快照的名称 Base Snapshot。</p>
取消	<p>更改项目快照名称，但不更改其他快照对象名称：对于您在创建项目快照时选定要创建快照的对象，不更改这些对象的名称。仅更改项目快照名称。</p> <p>对于 Base Snapshot：保留所有当前项目快照和所有未来项目快照的名称 Base Snapshot。</p>

访问和查看项目快照集中的快照



创建项目快照会创建一个快照集，它最少包含两个项目：基本项目和新项目快照。

要查看快照集中的所有项目快照：

1. 选择项目。

项目列表会显示项目和项目快照的列表。顶级快照是缺省的项目快照。
2. 单击**快照**图标 ，以显示快照集中的项目快照。


在“快照”视图中，您可以：

- 创建新的项目快照。要开始操作，请单击**编辑**图标 。
- 更改项目的缺省快照。单击**编辑**图标 ，然后单击**设置为缺省值**。
- 就象对标准项目一样，编辑项目快照定义。

启动缺省项目快照的作业

使用“快速启动”图标或“启动项目”页面来启动缺省项目快照。

要使用“快速启动”图标来启动缺省项目快照:

1. 选择项目。
2. 在项目列表中, 单击**快速启动**图标 。

作业会使用选择器、类、标记格式和环境的缺省值来运行。

要使用“启动项目”页面来启动缺省项目快照:

1. 选择**作业** → **启动**。
2. 在项目列表中, 单击缺省快照的项目名称以显示其“启动项目”页面。

在“启动项目”页面上, 您可以更改环境变量、选择器、类和标记格式。

如果选择器包含选择器快照, “快照”字段下面的字段将列出可以选择的选择器快照。


注: 要快速选择缺省选择器快照, 请选择“缺省快照”。“缺省快照”将映射到缺省选择器快照的名称。

3. 单击**执行**。

启动非缺省项目快照的作业

使用“快速启动”图标或“启动项目”页面来启动非缺省项目快照。

要使用“快速启动”图标来启动非缺省项目快照:

1. 选择项目。
2. 在项目列表中, 单击基本快照的**快照**图标 。

快照视图会在集合中显示项目快照。

3. 单击非缺省项目快照旁边的**快速启动**图标 。

作业会使用选择器、类、标记格式和环境的缺省值来运行。

要使用“启动项目”页面来启动非缺省项目快照:

1. 选择**作业** → **启动**。
2. 在项目列表中, 单击基本快照的**快照**图标 。

快照视图会在集合中显示项目快照。

3. 单击非缺省项目快照的项目名称以显示其“启动项目”页面。

在“启动项目”页面上, 您可以更改环境变量、选择器、类和标记格式。

如果选择器包含选择器快照, “快照”字段下面的字段将列出可以选择的选择器快照。

注：要快速选择缺省选择器快照，请选择“缺省快照”。“缺省快照”将映射到缺省选择器快照的名称。

4. 单击**执行**。

删除项目快照

可使用“删除项目”或“破坏项目”选项来删除项目快照。

- 当项目快照没有作业且项目快照没有被其他对象（类或其他项目）作为内联或链项目引用时，可使用“删除项目”选项。
- “破坏”选项会从 Build Forge 数据库删除项目及其作业，并除去其他对象对该项目的引用。

要删除项目快照：

1. 选择**项目**。

2. 在项目列表中，单击基本快照的**快照**图标 。

快照视图会在集合中显示项目快照。

3. 单击要删除的项目快照旁边的**编辑**图标 。

4. 单击**破坏或删除项目**。

第 20 章 使用步骤

本主题描述如何在管理控制台中创建和管理步骤。

关于步骤

步骤是项目的组成部分。项目作为作业运行时，每个步骤将按顺序发出。步骤包含一个或多个命令，并具有影响其行为的步骤属性。

关于步骤面板

“详细信息”选项卡

步骤属性指定了步骤运行方式、其输出的处理方式以及步骤完成时要执行的操作。步骤还可以运行其他项目或库。

要查看步骤属性，请选择项目内的一个步骤。缺省情况下会显示“详细信息”选项卡。它显示步骤属性。

如果未明确设置步骤属性，那么其值将从项目中继承。步骤的步骤属性集将覆盖继承的值。

步骤: [添加新步骤]

保存步骤

步骤信息

注释 (0)

名称:

活动:

已启用

访问:

-- 缺省 --

目录:

/

路径:

相对

步骤类型:

正则

内联:

-- 无 --

命令:

环境:

-- 无 --

选择器:

-- 缺省 --

广播:

否

超时 (分钟):

5

结果:

-- 退出代码 --

在失败时:

中止

线程:

否

通过链:

-- 无 --

通过等待:

否

通过通知:

-- 无 --

链失败:

-- 无 --

等待失败:

否

通知失败:

-- 无 --

步骤属性包括:

- 名称

步骤的名称。它在系统和日志中用作步骤的标签。
- 活动

指定是否运行步骤。缺省情况下启用步骤。选择“禁用”可阻止步骤运行。禁用的步骤不可在作业中运行。
- 目录

设置用于运行步骤命令的位置。系统自动为每个作业创建一个唯一目录。目录字段提供了在作业运行期间在项目所构造的目录中运行命令的一种便捷方法。
(Build Forge 不会构建在目录字段中提到的目录。)

路径 指定目录为绝对路径还是相对路径。

- **相对**: 步骤命令将在由服务器、项目、作业和步骤目录叠加在一起的路径下运行。
- **绝对**: 步骤命令将在由服务器和步骤目录叠加在一起的路径下运行。该选项允许您访问不在项目目录结构中的目录。示例: 它可用于启动永久安装在服务器上的应用程序。

步骤类型

确定如何运行步骤。该属性会对“命令”的内容和“内联”中指定的项目（如果存在）产生影响。

- **常规**: 运行步骤一次。
- **条件**: 如果“条件”属性中的表达式的求值结果为 **true**, 将运行步骤一次。选择“条件”会显示“条件”、“Else 内联”和“Else 命令”属性。如果“条件”属性的求值结果为 **false**, 那么不会运行“命令”和“内联”。但是, 如果已指定“Else 命令”和“Else 内联”, 那么会运行这两个属性。
- **While 循环**: 可多次运行步骤。它将一直执行, 直到“条件”属性中表达式的求值结果为 **false** 或者达到最大迭代数。选择“While 循环”将导致显示“条件”和“最大迭代数”属性。

针对每个“While 循环”迭代对选择器求值, 以确定用于迭代的服务器。

内联 指定要利用当前项目运行内联的项目或库。使用当前项目的环境和大部分属性来运行项目或库的步骤。但是, 系统使用内联项目的选择器作为内联步骤的缺省选择器。该行为就像是在当前步骤后复制指定项目中的步骤。

访问 选择一个访问组以定义允许哪些用户使用该步骤。可使用此属性限制对项目特定步骤的访问。如果用户不是该步骤的访问组成员, 那么当他启动包含此步骤的项目时, 该步骤将跳过。

选择项目缺省值会使步骤继承项目的访问属性。

最大迭代数

仅当步骤类型为“While 循环”时才显示。指定在一个循环中可以运行步骤的最大迭代数。系统强加的缺省值为 100。该步骤在步骤日志中显示为成功完成（通过）。使用**如果达到最大值步骤失败**使步骤在达到最大迭代数时失败。

执行作业时, 只读变量 **BF_ITERATION** 包含成功进入的迭代数。如果停止作业又重新启动, 将在 **BF_ITERATION** 指示的迭代处重新启动。

如果达到最大值步骤失败

如果选择“是”, 当达到最大迭代数时, “While 循环”步骤将失败。如果选择“否”, 步骤将通过。

Else 内联

仅当步骤类型为“条件”时才显示。指定当所指定的条件为 **false** 时要运行内联的项目。缺省值为“否”。

命令

一条或多条命令。命令可以是操作系统命令、点命令或两者的组合。请参阅第 318 页的『步骤如何运行』。

条件 仅当您已选择步骤类型“条件”或“while 循环”时才显示。

- **条件**: 如果条件的求值结果为 **true**, 将运行该命令。

- **While 循环**: 可多次运行命令, 只要条件的求值结果为 `true`。可以使用“最大迭代数”设置该限制。

条件可以是在选定服务器资源上运行的函数或命令。

- 如果使用函数, 那么必须在“条件”字段的开头使用。它通过 **Build Forge** 引擎进行求值。不会将它发送到服务器资源。要获取函数的列表以及有关如何使用这些函数的指示信息, 请参阅第 323 页的『条件函数』。
- 命令在选定服务器上运行。此处使用的任何命令在代理程序的 `shell` 环境中都必须有效。执行的返回码决定条件通过还是失败。

项目的 **Build Forge** 变量可在条件表达式中使用。请参阅第 259 页的『步骤中的变量解释』, 以获取有关如何表示变量以及如何对变量求值的更多信息。

Else 命令

仅当您已选择步骤类型“条件”时才显示。指定当条件的求值结果为 `false` 时要运行的命令。

环境

指定要在执行命令前应用的环境。此环境中的值将覆盖从服务器环境、项目环境和步骤变量中继承的所有值。

选择器 指定要用于选择该步骤的服务器的选择器。如果保留为缺省值, 将在项目选择器确定的服务器上运行该步骤。

广播 如果选中此框, 将在与当前选择器 (若已指定, 为步骤选择器; 否则为项目选择器) 匹配的**所有**服务器上运行步骤。在运行时, 系统将广播步骤替换为一系列步骤, 一个步骤对应一个服务器, 然后根据广播步骤的**线程**属性来顺序或并行运行这些命令。

重新启动时的广播步骤行为: 当重新启动广播步骤时, 它不会广播。该设置仅适用于该步骤的新启动。重新启动时, 引擎会为此步骤随机选择一个服务器。

超时 (分钟)

指定系统等待当前命令产生输出的分钟数 (缺省值为 5 分钟)。值 0 表示如果该步骤正确连接到代理程序, 那么该步骤不超时。如果达到了超时值, 系统将使该步骤失败。项目也将失败, 除非该步骤设置为“失败时继续”。

结果 “结果”属性确定系统如何判断步骤是成功还是失败。使用退出码的缺省值可根据命令 `shell` 返回的退出码来决定是否成功。您也可以选择日志过滤器来检验命令输出。要选择日志过滤器, 必须首先创建。

在失败时

指定在步骤失败时是停止还是继续作业。缺省情况下, 系统会停止作业。

线程 如果选择“是”, 将与其他步骤并行运行此步骤。将此属性设置为“是”将允许线程化运行此步骤 (此步骤与其他步骤并行运行)。将该属性设置为“否”以避免线程化。将该属性设置为“连接”会将线程化的步骤块分开。第一组步骤必须首先完成, 然后“连接”步骤后面的下一组线程化步骤才能启动。

通过通知

指定在步骤通过时要通知的访问组。

通过链 指定在当前步骤通过时要启动的项目。 (具有“警告”状态的步骤被视为通过, 并且将启动通过链)。

通过等待

如果选中此框，系统将暂挂当前项目直到通过链式项目完成。如果取消此步骤（或其项目），那么链式项目也将被取消。如果未选中，那么链式项目将异步启动，并且当前项目继续执行下一步。

通知失败

指定在步骤失败时要通知的访问组。

链失败 指定在当前步骤失败时要启动的项目。（设置为故障时继续的步骤被视为失败，并且将启动指定给此步骤的任何失败链。）

等待失败

如果选中此框，系统将暂挂当前项目直到失败链式项目完成。如果取消此步骤（或其项目），那么链式项目也将被取消。


“注释”选项卡

注释选项卡包含对步骤作出的带时间戳记的注释的列表。您可以手动创建注释。它本身不会自动记录对步骤的编辑。该选项卡显示当前注释编号，例如**注释（2）**。


要添加注释：

1. 单击**注释**选项卡。
2. 在文本字段中写入新注释。
3. 单击**提交**。

要编辑注释：

1. 单击**注释**选项卡。
2. 单击要编辑注释旁边的**编辑**图标 。进行编辑。
3. 单击**提交**。

要删除注释：

1. 单击**注释**选项卡。
2. 单击**废纸篓**图标 。系统会提示您是否确定要删除该注释。
3. 单击**确定**。

添加步骤

关于此任务

本过程将步骤添加到项目末尾。有关如何将步骤作为第一个步骤插入或插入在现有步骤之间的信息，请参阅第 317 页的『其他步骤操作』。

过程

1. 选择**项目**，然后单击项目名称。系统将显示所选项目的步骤列表。
2. 单击主面板顶部的**添加步骤**。系统显示一个空的步骤详细信息面板。
3. 输入属性的值。**名称**和**命令**为必填。
4. 单击**保存步骤**。


编辑步骤

过程

1. 选择项目，然后单击项目名称。系统将显示所选项目的步骤列表。
2. 单击步骤名称。系统将在面板下部显示该步骤的属性。
3. 编辑属性。
4. 单击保存步骤。

禁用步骤

过程

1. 选择项目，然后单击项目名称。系统将显示所选项目的步骤列表。
2. 单击步骤名称前面的复选框。步骤被选中  时，会对项目产生以下影响：
 - 项目作为作业运行时，不运行该步骤。
 - 正常启动作业时（选择作业 > 启动，然后单击作业名称，最后单击作业步骤），该步骤将在步骤列表中灰显。步骤可见，但无法为正在启动的作业激活该步骤。

注:

还可以通过在详细信息选项卡中将步骤的活动属性设置为“禁用”来禁用该步骤。保存步骤时，该复选框是选中的。

其他步骤操作

要处理步骤，请选择项目，然后单击项目名称。系统将显示所选项目的步骤列表。

单击步骤名称前面的操作图标  以显示其他选项:

- **插入新步骤:** 在选中的步骤上方插入一个步骤。
- **克隆步骤:** 复制步骤及其所有属性。通过在步骤结尾处添加数字来更改名称。复制名为 Build 的步骤会产生名为 Build COPY 0 的新步骤。数字将自动设置。可以复制到以下位置：
 - 顶部: 列表的开头
 - 上方: 就在当前步骤之前
 - 下方: 就在当前步骤之后
 - 底部: 列表的底部
- **移动步骤:** 将步骤移动至列表中的其他位置。可以移动至以下位置：
 - 顶部: 列表的开头
 - 向上: 上移一个位置
 - 移动至...: 将显示对话框来请求步骤号。步骤将移动至该位置，并按照需要调整其他步骤号。
 - 向下: 下移一个位置
 - 底部: 列表的底部
- **删除步骤**

控制执行流程

步骤内有几个功能可用于控制项目中的执行流程:

- **内联:** 使用步骤的“内联”属性可指定项目或库。运行此步骤的命令后, 将立即以内联方式运行项目或库的步骤。内联项目或库的步骤将在步骤日志中缩进显示。
- **通过和失败链:** 步骤可具有其自己的通过链和失败链, 不同于为项目指定的链。
- **线程化:** 您可以并行运行标记为线程化的步骤。使用步骤的“线程”属性可以将该步骤标记为线程化。
- **广播:** 您可以在多个服务器上运行标记为广播的步骤。使用步骤的“广播”属性。
- **条件:** 仅当条件为 `true` 时才可设置步骤运行。如果条件为 `false`, 您可以设置一组备用命令以及一个内联项目或库运行。将“步骤类型”属性设置为有条件的, 并将相关的“条件”和“Else”属性用于该功能。
- **While 循环:** 您可在每次条件求值结果为 `true` 时运行循环中的步骤。将“步骤类型”属性设置为“While 循环”并将相关的属性用于此功能。
- **点命令:** `.run` 和 `.runwait` 命令可从步骤命令启动库或项目。

复杂执行流程的常见用法是作业优化, 即仅在需要时执行步骤。

在软件构建工程环境中, 作业优化意味着根据需要只构建应用程序的某些部分而不是整个应用程序。作业可对照上次编译的二进制文件检查源代码状态, 并仅当存在源代码更改时运行编译。对于复杂应用程序, 执行流程可对模块依赖性以及源代码状态作出响应。

步骤如何运行

在步骤定义中, 命令属性包含操作系统命令、点命令或两者的组合。

您可以在单个步骤中执行多个命令。将各条命令分开, 一行一条命令。

注: 如果将缺省的“退出码”设置用于步骤的**结果**属性, 那么整个步骤的成功或失败由步骤的最后一条命令返回的退出码确定。为了检测任意命令中的故障, 请创建“日志过滤器”并在**结果**属性中指定其用法。

在系统运行步骤之前, 会构造步骤环境。使用服务器环境、项目环境和步骤环境中指定的值来设置变量。请参阅第 257 页的『环境继承』。缺省情况下, 变量首先进行语法分析, 然后才能用于命令。请参阅第 259 页的『步骤中的变量解释』。

指定 shell

可以使用 `#!` 伪指令来指定要用于运行命令的 shell。这在 Linux[®] 和 UNIX[®] 系统上有效, 在 Windows[®] 上同样有效 (Windows 代理程序处理将命令传给指定解释器的操作。要从步骤中将命令发送到 Windows 上 `C:\perl\bin` 中的 Perl 副本, 请使用 `#!C:\perl\bin\perl.exe`。如果使用带有 Cygwin 的 Windows 代理程序, 但是需要将命令定向到 Windows shell `cmd.exe`, 可以使用下面的命令行, 该命令行利用了 Windows 隐式路径:

```
#!cmd.exe /C
```

请注意 `/C` 选项对于 `cmd.exe` 是必需的, 因为如果不指定的话, `cmd.exe` 将在您的步骤命令传给它之后等待其他命令。在 UNIX 或 Linux 计算机上可使用 `#!/bin/perl` 命令。

注：在 Linux 或 UNIX 系统上使用 #! 命令时，系统不会切换到标准缺省目录（由服务器路径、项目、名称和步骤路径字段的组合构成的路径），因为它无法预测必需的语法；您必须包含自己的目录切换命令。使用系统创建的特殊环境变量，如 BF_SERVER_ROOT 和 BF_PROJECTNAME_PHYS 来引入自己的切换目录命令。

系统如何拆分步骤

Rational® Build Forge® 可以拆分步骤，其中每个部分由一组操作系统命令或单个点命令组成。例如，以下步骤分为六个部分。

```
cmd1          # Part 1
cmd2
.dot_cmd1     # Part 2
cmd3         # Part 3
cmd4
.dot_cmd2     # Part 4
.sleep 30    # Part 5
.dot_cmd3     # Part 6
```

步骤的各个部分按顺序执行。每个部分生成的环境将传递给下一个部分。在任何一个部分发生错误时，处理将立即停止。

先前的发行版要求上述步骤的每个部分都成为一个单独的步骤。

注：请勿在部分之间创建引用。请勿在部分之间创建跳跃，例如，在一个部分中使用 GOTO 而其目标标签位于另一部分中。

内联：包含项目或库的步骤

使用步骤的“内联”属性来包含指定项目或库中的所有步骤。

要在步骤中使用内联，请将“内联”属性设置为项目或库的名称。在步骤运行时，执行了以下操作：

1. 步骤运行“命令”属性中的一条或多条命令。
2. 步骤运行在“内联”属性中指定的项目或库的步骤。

如果想要运行内联步骤而不使用“命令”属性，请在“命令”属性中使用 `.sleep 0`。

来自于调用步骤的内联继承

来自于被调用项目或库的所有步骤都将在调用步骤的上下文中运行。内联的步骤将继承主调步骤的环境。

但是，系统使用内联项目的选择器作为内联步骤的缺省选择器。

内联步骤状态对调用步骤状态的影响

如果主调步骤的命令通过，然后才会考虑内联步骤的执行状态，如下所示：

- 如果步骤命令失败，那么作业正常停止。然而，如果步骤的“失败时”属性设置为“继续”，并且指定“内联”，那么内联将运行。
- 如果所有步骤返回的执行状态都是“通过”，那么主调步骤将标记为“通过”。如果为主调步骤指定了“通过链”，那么其将运行。
- 如果任何内联步骤返回的执行状态是“失败”，那么主调步骤将标记为“失败”。如果为主调步骤指定了“失败链”，那么其将运行。

此行为使您通过深度嵌套的内联和链轻松地跟踪执行状态。

内联嵌套

在内联项目或库的步骤时，被调用的步骤将嵌套在调用步骤中。

最大嵌套级别由系统设置**最大内联深度**设置。缺省值为 32。作业启动时不会测试嵌套。如果运行中的作业超出该限制，那么将在超出限制的点失败。

嵌套级别可能还受运行“管理控制台”的主机上可用内存的限制。

步骤的通过链和失败链

单个步骤可以具有一个“通过链”和一个“失败链”。

步骤的“通过链”和“失败链”的运行独立于项目的“通过链”和“失败链”。

“通过链”和“失败链”属性将设置为项目或库的名称。其工作方式与项目的链集相同。请参阅第 285 页的『启用等待时取消链式项目』。

线程化：并行运行步骤

线程化使步骤可以在同一个服务器或不同服务器上并行运行。线程化由步骤的“线程”属性设置控制。缺省情况下，“线程”属性设置为“否”。当存在很多可以彼此独立运行的项目部分时，线程化可帮助减少项目执行时间。

当多个相邻步骤的“线程”属性设置为“是”时，系统将尝试并行运行该步骤。这样的步骤视为**启用线程**，并且在作业的其余步骤继续运行的同时，每个步骤可以单独运行。线程化遵循下列规则。

- 为了实现线程化，必须至少依次有两个步骤的“线程”属性设置为 **Yes**。依次的一组线程步骤称为**线程块**。线程块可以继续进入作为“内联”一部分的步骤。例如，如果项目中的某步骤包含“内联”且该“内联”的第一个步骤也是线程化的，那么这两个步骤是同一线程块的一部分。它们并发运行。线程块执行线程化步骤，包括嵌套的“内联”步骤，直到遇到“加入”步骤或非线程化步骤。使用嵌套的“内联”步骤时注意避免竞争状态。根据结果不同，竞争状态可能是由内联的线程化步骤引起的，也可能是线程化父步骤中的数据引起的。
- 线程块会被“线程”属性设置为**加入**的步骤终止，或在遇到非线程式步骤时终止。在该点上步骤将再次依次执行。
- 当系统遇到启用线程的步骤时，它将尝试启动该步骤。如果后面的步骤也启用线程，系统将尝试启动该步骤并继续运行下一个步骤，将一直重复进行，直至再没有启用线程的步骤或者达到了作业限制为止。如果项目的选择器指定了服务器池，那么概念上作业限制就是池中服务器的作业限制的总和。

注： 启用线程步骤的开始时间取决于打算运行该步骤的服务器的可用性。如果步骤无法启动，系统将等待，然后重试。您不能明确控制首先启动哪些步骤。

- 步骤最终可在一台服务器上同时运行（根据该服务器的容量），也可在多台服务器上同时运行（取决于有多少服务器与该选择器匹配）。
- 要强制所有步骤在一台服务器上运行，可使用项目的**固定**属性。
- 如果存在多个线程块，必须先完成第一个线程块，才能启动下一个线程块。

在下面的示例中，步骤 2、3 和 4 必须先完成，在此之后，步骤 5 和 6 才能启动。

项目	步骤的线程属性
步骤 1	否
步骤 2	是
步骤 3	是
步骤 4	加入
步骤 5	是
步骤 6	是
步骤 7	否

- 使用项目的**最大线程数**属性可限制同时运行的线程数。每个启用线程的步骤及其任何内联项目（若有）都会导致并行处理。将对所有进程计数，直至达到父项目的最大值。当达到**最大线程数**值时，系统会停止启动新的并行处理。它会等到项目的并行处理数量降低到低于**最大线程数**值，然后再继续。

将步骤广播到多个服务器

具有在多个服务器上可以有用地执行的活动时，可使用广播功能在多个服务器上重复同一个步骤。

通常一个步骤仅在一个服务器上运行。但是，每个步骤都有一个**广播**复选框。选中步骤的**广播**框时，运行时系统会将此步骤替换为一组非广播步骤，每个与步骤的选择器匹配的服务器对应一个非广播步骤。

注：如果步骤的选择器仅与一个服务器匹配，那么该步骤仅运行一次。

对广播的潜在用途包括：

- 重新引导一组服务器。
- 在一组服务器上运行测试。
- 将同一组文件复制到整个服务器组。
- 将同一组源代码检出到多个服务器，以便为稍后使用一个易于维护的步骤执行的单独任务准备好这些服务器。

重新启动广播步骤

请参阅第 362 页的『重新启动广播步骤』。

广播步骤中的线程化

在运行时为广播步骤创建替换步骤时，系统将如下所示执行步骤：

- 如果广播步骤的**线程**属性设置为**否**，那么替换步骤将获取相同的**线程**值，这样将连续执行所有步骤。必须完成每个步骤，才能开始下一个步骤。
- 如果广播步骤的**线程**属性设置为**是**，那么替换步骤也将获取相同的**线程**值。这将导致一组步骤彼此之间以并行方式运行，并且与广播步骤前后的任何线程步骤之间也以并行方式运行。
- 如果广播步骤的**线程**属性设置为**加入**，那么系统将创建把**线程**设置为**是**的替换步骤，但最后一个步骤例外，因为它标记为**加入**。这将导致一组步骤彼此之间以并行方式执行，并且与位于其前面的任何线程步骤之间也以并行方式执行，但是整组步骤必须完成，然后才能启动广播步骤后的步骤。

从广播步骤启动其他项目

可以广播包含内联项目的步骤，或者在步骤通过或失败时链接项目的步骤（**通过链/失败链**）。广播用于启动（链接）另一个项目的步骤时，请注意：此广播步骤不覆盖所启动项目的选择器。如果您的目的是在与广播步骤的选择器匹配的每个服务器上启动此项目，那么通常在从广播步骤启动项目时使用库（没有自己的选择器的项目）。

如果不使用库，那么广播步骤的每个副本在不同的服务器上运行，但是内联或已连接项目将服从自己的选择器，从而可能不会选择与广播步骤的副本相同的服务器。可以以每个广播步骤在不同的服务器上运行结束，而来自内联项目的所有步骤多次在同一个服务器上运行。

注：如果目的是使用**广播**在与一个选择器匹配的每个服务器上启动一次库，请务必也在该库上设置**固定**选项，这样它的所有步骤（使用缺省项目服务器的步骤）将在同一个服务器上运行。

条件步骤执行

条件执行实施步骤的 if-then-else 分支。

简单 IF-THEN 执行：

1. 将步骤类型设置为“条件”。
2. 将条件设置为可以求值的表达式。
3. 填写要运行的命令。
4. 如果需要，指定要运行的内联。（如果已设置内联，可将命令保留为空。）

如果当作业运行时“条件”求值结果为 **true**，将运行该步骤。如果已指定，也将运行内联项目或库。如果表达式求值结果为 **FALSE**，将跳过该步骤并且作业执行将继续下一个步骤。

If-Then-Else 执行：

如果您希望在“条件”返回 **FALSE** 时运行不同的命令和/或内联，请填写其他属性：

- 填写要运行的 **Else** 命令。
- 如有需要，指定要运行的 **Else** 内联项目或库。（如果已设置 **Else** 内联，**Else** 命令可为空。）

作业执行期间，如果条件已成功求值并且“命令”或“**Else** 命令”中的命令已成功运行，那么步骤结果将标记为“通过”。要确定使用哪个路径，必须查看日志。

另请参阅第 323 页的『条件函数』。

While 循环执行

While 循环执行可用于根据条件重复某个步骤。

要按照 while 循环实施步骤：

1. 将步骤类型设置为“While 循环”。
2. 将条件设置为命令或可以求值的表达式。
3. 填写要运行的命令。

4. 如果需要, 指定要运行的内联。(如果已设置内联, 可将命令保留为空。)
5. 如果需要, 将“最大迭代数”设置为您希望运行步骤的最大次数。在开发过程中使用此限制, 直到您指定的条件正确工作。缺省值为 100。
6. 如果您希望该步骤在达到“最大迭代数”时失败, 请将**如果达到最大值步骤失败**设置为“是”。否则在达到“最大迭代数”时, 该步骤将通过。

步骤的每个迭代都会记录在日志中。每个迭代结果都会根据结果条件设置为“通过”或“失败”。

另请参阅『条件函数』。

条件函数

条件函数用于 Condition 步骤属性和适配器 XML 元素的 condition 属性中。

- 对于使用 Condition 属性的步骤: 如果在 Condition 字段的开头使用了以下函数, 那么引擎将对这些函数求值, 并且, 除非该条件的求值结果为 true, 否则不会向选定服务器发送信息。如果条件的求值结果为 true, 将在选定服务器上运行该步骤。

要点:

- 不要尝试对服务器资源的 shell 环境中设置的变量使用函数。求值对 Build Forge 引擎进行, 因此对于该步骤, 它们仅对在 Build Forge 环境中定义的变量起作用。
- 请勿在求值字符串中使用运算符字符。运算符字符有 =, <, >, !
- 对于适配器模板: 以下函数可用于具有 condition 属性的 适配器 XML 元素中。它们用于指定适配器的运行方式。重要信息: 条件函数在条件属性中必须用双引号引起: condition="condition_function"。

以下函数可用:

true(expression)

如果 expression 为真, 则返回 true。

false(expression)

如果表达式为 false, 那么返回 true。

contains(a,b)

如果字符串 a 包含字符串 b, 那么返回 true。a 和 b 参数可以是文字串或变量。不应将文字串加引号。如果将文字串加上引号, 引号将成为求值的字符串的一部分。

注: 在 Rational Build Forge V7.1.2 之前, 如果字符串 a 在字符串 b 中, 此函数会返回 true。

hastext(var)

如果该变量不为空, 那么返回 true。Var 是 Build Forge 中的变量集。

isempty(var)

如果该变量为空, 那么返回 true。Var 是 Build Forge 中的变量集。

a eq b

如果 a 等于 b, 那么返回 true。a 和 b 参数可以是 Build Forge 中的变量集, 也可以是文字值。可使用字符和数值类型。在参数和运算符之间使用一个空格。

a ne b

如果 *a* 不等于 *b*，那么返回 **true**。*a* 和 *b* 参数可以是 Build Forge 中的变量集，也可以是文字值。可使用字符和数值类型。在参数和运算符之间使用一个空格。

a contains b

如果字符串 *b* 可在字符串 *a* 中找到，那么返回 **true**。不应将文字串加上引号。如果将文字串加上引号，引号将成为求值的字符串的一部分。可使用字符和数值类型。在参数和运算符之间使用一个空格。

函数中的表达式

true() 和 **false()** 函数中的 *expression* 参数可使用以下运算符：

a==b 测试两者是否相等。参数可以是字符串或数字。参数可以是 Build Forge 中定义的字面值或变量。

a eq b 测试两者是否相等。参数可以是字符串或数字。参数可以是 Build Forge 中定义的字面值或变量。在参数和运算符之间使用一个空格。

a!=b 测试两者是否不等。参数可以是字符串或数字。参数可以是 Build Forge 中定义的字面值或变量。

a ne b 测试两者是否不等。参数可以是字符串或数字。参数可以是 Build Forge 中定义的字面值或变量。在参数和运算符之间使用一个空格。

a>b 测试 *a* 是否大于 *b*。参数必须为数值。参数可以是 Build Forge 中定义的字面值或变量。字面值可使用算术运算符，例如 2+2。

a<b 测试 *a* 是否不大于 *b*。参数必须为数值。参数可以是 Build Forge 中定义的字面值或变量。字面值可使用算术运算符，例如 2+2。

a>=b 测试 *a* 是否大于等于 *b*。参数必须为数值。参数可以是 Build Forge 中定义的字面值或变量。字面值可使用算术运算符，例如 2+2。

a<=b 测试 *a* 是否不大于等于 *b*。参数必须为数值。参数可以是 Build Forge 中定义的字面值或变量。字面值可使用算术运算符，例如 2+2。

a contains b 测试字符串 *b* 是否可在字符串 *a* 中找到。参数可以是 Build Forge 中定义的字面值或变量。不应将文字串加引号。

条件函数的示例:

在下表中的示例中，变量设置如下：

- \$AVAL 包含值 String。
- \$BVAL 包含值 3。

条件	求值为	注释
A String contains \$AVAL	TRUE	字符串比较
A String contains "String"	FALSE	String 两边的引号成为比较的一部分
true(A String contains \$AVAL)	TRUE	字符串比较

条件	求值为	注释
<code>\$AVAL contains String</code>	TRUE	字符串比较
<code>\$AVAL contains "String"</code>	FALSE	<code>String</code> 两边的引号成为比较的一部分
<code>contains(A String,\$AVAL)</code>	TRUE	字符串比较
<code>true(A String contains "\$AVAL")</code>	FALSE	<code>\$AVAL</code> 两边的引号成为比较的一部分; “A String”中的“String”部分两边没有引号。
<code>A String != \$AVAL</code>	TRUE	字符串比较
<code>A String ne \$AVAL</code>	TRUE	字符串比较
<code>false("Not Here" contains \$AVAL)</code>	TRUE	测试字符串比较
<code>true(2+1 == \$BVAL)</code>	TRUE	用于测试相等性的数字表达式
<code>false(2+2 < \$BVAL)</code>	TRUE	用于测试不等性的数字表达式
<code>\$AVAL eq \$AVAL</code>	TRUE	测试字符串比较
<code>true(\$AVAL ne Linus)</code>	TRUE	测试字符串比较
<code>true(\$BVAL > 2+2)</code>	FALSE	<code>(3 > 2+2)</code> 不为 <code>true</code>
<code>contains(Not Here, \$AVAL)</code>	FALSE	测试字符串比较

从步骤启动项目

在某步骤中使用 `.run` 或 `.runwait` 命令来启动项目。

以该方式启动的项目的行为与链相同。项目使用其自身的选择器和环境运行。请参阅第 349 页的『`.run` 和 `.runwait`』。

定制日志输出

这些主题显示使用步骤中的命令功能定制日志输出的一些方法。

标记步骤的日志输出

创建标签以将步骤输出列在步骤日志中其自己的类别下。

开始之前

该任务假定您已经创建了选择器、服务器和项目。它假定您使用的服务器具有接受 `echo` 命令的操作系统（Windows、Linux 或 UNIX）。

关于此任务

您可以在任意输出行的开始位置包含一个大写标签。在该步骤结束或遇到新的标签之前，将一直使用该标签。

标签采用以下语法:

- 仅包含大写字母，后跟冒号（标签不能包含空格、标点符号、数字或小写字母。
"SPACESHIPS:" 有效。"Space Ships:" 无效。）
- 至少含三个字符

您可以在步骤中使用 **echo** 命令来创建标签。如果 **echo** 命令遵循标签语法，那么系统将其第一个自变量识别为标签。

注：如果 **echo** 命令遵循该语法，那么即使任何构建输出中行开始处的文本不是该命令的自变量，系统仍将其识别为标签。

正如示例中所示，您可以将标签设置为现有输出标签名称，也可以设置为新名称。

过程

1. 在项目中创建新步骤。此示例中的项目名为 Say_hi。
2. 将步骤命名为 LabeledLogOutput。
3. 在“命令”字段中输入以下文本：

```
echo SPACESHIPS: Voyager I
echo Voyager II
echo EXEC: You can add text to existing categories as well
```

4. 运行项目。
5. 当作业完成时，查看日志。

结果

注意类别标题中的 SPACESHIPS 复选框和输出行 354 和 355（标记了 SPACE-SHIPS）。

```
79      7/9/10 10:24 AM EXEC      Locale set to 'English_United States.1252'
250     7/9/10 10:24 AM EXEC      Locale set to 'English_United States.1252'
349     7/9/10 10:24 AM EXEC      Performing variable expansion on command line
353     7/9/10 10:24 AM EXEC      start [C:\data\BuildForge\buildforge_projects\Say_hi\BUILD_1@RBF-14]
354     7/9/10 10:24 AM SPACESHIPS      Voyager I
355     7/9/10 10:24 AM SPACESHIPS      Voyager II
356     7/9/10 10:24 AM EXEC      You can add text to existing categories as well
357     7/9/10 10:24 AM EXEC      end [C:\data\BuildForge\buildforge_projects\Say_hi\BUILD_1@RBF-14]
```

注：您可以选择或清除 SPACESHIPS 复选框以显示或隐藏类别。

注：在将文本添加到 EXEC 类别时，请一定要避免使用特殊字符（例如，圆括号）。在某些系统上，加上引号后可行，但总体来说最好只使用纯文本。

将步骤输出突出显示为彩色或活动链路

日志查看器可识别输出文本中的 [STATUS] 和 [URL] 命令。这些命令不区分大小写。您可以在 echo 命令或产生输出的任何命令中使用这些命令。开始和结束标记必须出现在输出的同一行上。

- [STATUS=*condition*] 和 [/STATUS] 标记将文本标记为突出显示。 *condition* 以下列方式设置突出显示颜色：
 - 通过 – 绿色
 - 警告 – 黄色
 - 失败 – 红色
 - 运行 – 蓝色
- [URL] 和 [/URL] 标记将文本标记为活动的超链接。

使用 [status] 的示例：

```
echo [STATUS=WARN]Access to source control timed out[/STATUS]
```

使用 [url] 的示例:

```
echo See the support forums at [url]http://www.ibm.com[/url]
```

日志将文本显示为指向指定 URL 的活动超链接。单击它可打开新的浏览器窗口或选项卡并显示该页面。但是, 请注意, 匹配 [URL] 地址部分的日志过滤器将导致链接失效。

使用作业数据

这些主题显示使用命令功能来修改项目和作业的一些方法。

在项目文件中嵌入构建号

可使用 .strsub 命令将文件中的一个字符串替换成另一个字符串; 常见的用法是将标准标记替换为系统变量, 比如提供当前作业号的 \$B 变量。

可使用 .strsub 点命令将构建号或版本号嵌入代码文件。通过在项目前面的部分放置 .strsub 命令, 后面的步骤就能编译包含更新信息的文件。

例如, 下面的步骤设置一个项目以嵌入构建号:

1. 将 _BUILD_ 之类的唯一字符串添加到项目中的某个文件。例如, 修改文件 README.TXT, 然后如下更改版本声明:

```
Application version 5.0.123
Application version 5.0._BUILD_
```

2. 项目中前面的某个步骤应检出要处理的文件。在检出 README.TXT 之后添加一个步骤, 该步骤将 _BUILD_ 替换为 \$B 系统变量。使用的命令如下:

```
.strsub _BUILD_ $B README.TXT
```

3. 运行项目, 然后验证 README.TXT 文件是否包含当前作业号。对于项目的第三次运行, README.TXT 文件应包含下面一行:

```
Application version 5.0.3
```

增强功能

可按照下面方式改进此方法:

- 使用附加的环境变量。例如, 创建名为 \$MAJORVERSION 和 \$MINORVERSION 的变量, 然后如下使用它们:

```
.strsub _MAJORVERSION_ $MAJORVERSION README.TXT
.strsub _MINORVERSION_ $MINORVERSION README.TXT
```

- 启动项目时, 更新您的环境变量。选择**作业 > 启动**以启动项目, 您可看到当前环境变量, 并可在启动项目之前编辑变量的值。例如, 您可以将某些注释作为变量包含在作业中。对注释变量使用项目操作**必须更改**以强制用户在运行项目时输入新值。

作业期间更改构建标记

可以在作业期间通过使用 .retag 命令将标记的值更改为全新的值, 该命令的语法如下:

```
.retag <new tag value>
```

下面是简单用法的示例:

```
.retag MyProject
```

也可以使用更复杂的用法:

```
.retag Job_${B}_${BF_D}
```

此示例将标记设置为使用运行增量和当前日期系统变量。可以对服务器的命令解释器使用命令来设置结果。要在点命令内使用命令，请将该命令以反引号（` `）字符括起来：

```
.retag `hostname`
```

本示例将标记设置为运行该步骤的服务器上运行主机名命令的结果。

注：切勿将反引号格式与命令的标准指定格式混用。

作业期间更改环境变量的值

可以使用 `.set`、`.bset` 和 `.tset` 命令从步骤内更改环境变量。这些命令如下所示更改现有环境变量的值：

- 使用 `.set` 命令可更改环境的主记录。系统运行项目时，将从主记录创建项目环境的副本，然后将该副本用作项目的缺省副本。作用如下：
 - 如果 `.set` 命令修改项目环境，使用缺省环境的后续步骤将不会发现这些更改，因为系统不会向后引用主记录。
 - 如果使用 `.set` 命令修改环境，并且后续步骤明确使用相同的环境，该步骤将发现您所作的更改。此步骤选择特定环境时，系统将向后引用环境的主记录。即使命名的组与项目缺省组相同，只要步骤的环境设置不是“缺省值”，这仍然适用。
 - `.set` 命令所作更改将在作业完成之后永久存在。以后的作业将使用先前运行的 `.set` 命令创建的值。

请使用下面的基本语法：

```
.set env <EnvGroupName>[(<SnapshotName>)] "<VariableName>=<DesiredValue>"
```

- 使用 `.bset` 命令在作业执行期间添加或更改变量值。更改将在 `.bset` 出现于其中的步骤后的步骤中生效。它们对作业的其余部分有效。

```
.bset env "<VariableName>=<DesiredValue>"
```

注：与 `.set` 命令不同的是，如果对 `.bset` 命令进行了设置，对该命令指定的变量无需存在，这样就可以使用 `.bset` 命令在作业期间新建变量。该变量的值在当前作业过去之后不会永久存在。

- 使用 `.tset` 命令在作业执行期间添加或更改变量值。更改在当前步骤中生效。它们对于步骤中的任何其他命令以及步骤中指定的任何内联有效。该变量的值在当前步骤过去之后不再存在。

```
.tset env "<VariableName>=<DesiredValue>"
```

注：与 `.set` 命令不同，如果对 `.tset` 命令进行了设置，为该命令指定的变量就无需存在，这样就可以使用 `.tset` 命令在作业期间新建变量。

设置多个变量

通过包含用空格分隔的多个“变量/值”对，您可以同时为这些命令设置多个变量，如下示例所示：

```
.set env MyGroup "X=5" "X2=45"
```

```
.bset env "Y=7" "CompilerVersion=4.511"
```

```
.tset env "Z=9" "Z2=54"
```

使用命令输出来设置值

通过将命令发送给服务器的命令解释器，可为 .set 或 .bset 命令生成变量值。要在点命令中使用命令，将该命令括在反引号字符中。例如，命令：

```
.set env SetupGroup "PerlVer=`perl --version`"
```

将变量 PerlVer 设置为 perl --version 命令的输出。

变量只能存储 256 个字符；如果分配给变量更多字符，其值将截短。

缺省情况下，系统将反引号中的命令的完整输出赋给变量，但是可以在方括号中使用范围命令来选择要将命令输出中的哪些行赋予变量。范围编号使用基于 0 的索引（第一行编号为 0，第二行为 1，依此类推）来指定输出中的行。在下面的示例中，

```
.set env SetupGroup "WindowsIPinfo[0,5-8]=`ipconfig`"
```

变量 WindowsIPinfo 接收 ipconfig 命令输出的第 1 行和第 6 至 9 行。

以下都是有效的范围修饰符，它们选择单个行、多组行或者两者的组合：

- [5]
- [4-6]
- [1,2,5,8-11]

系统不加分离地将这些行组合在一起；不添加任何空格或回车符。

注：切勿将反引号格式与命令的标准指定格式混用。

使用寄存器

寄存器是步骤可用于存储持久数据的常规用途缓冲区。普通的寄存器可以有单字母名称或以字母开始的多字符名称。

底层数据库的区分大小写确定在创建寄存器名称时可使用的大小写。

可以将寄存器变量加入通知模板；引用通知模板中的寄存器时，请使用 \${X} 符号括起来的格式。引用空寄存器将返回空字符串。

使用 .push 和 .pop 点命令可将信息存储在寄存器中并从中检索信息。另请参阅 .poptag 命令（第 346 页的『.poptag』），该命令使当前作业标记等同寄存器的内容。

注：不能在命令中将寄存器用作变量。必须首先将寄存器的值出栈到文件，才能使用寄存器。

表 11. 专用寄存器

寄存器	包括
!	包含与“失败”过滤器模式匹配的命令输出行。 注：该寄存器仅在应用过滤器的步骤的作用域中可见。在 Rational Build Forge 完成过滤器步骤的处理后，此步骤过滤器可能设置的寄存器的内容将不再可见。

表 11. 专用寄存器 (续)

寄存器	包括
@	包含与“通过”过滤器模式匹配的命令输出行。 注： 该寄存器仅在教育过滤器的步骤的作用域中可见。在 Rational Build Forge 完成过滤器步骤的处理后，此步骤过滤器可能设置的寄存器的内容将不再可见。
=	指定作业的说明数据库。允许步骤将来自文件的数据作为说明添加到作业。此寄存器不同于其他寄存器： <ul style="list-style-type: none">• 只能写入（推送到）此寄存器，不能从中读取。• 推送到此寄存器的数据始终附加到该寄存器，而不是覆盖前面的数据。• 系统为时间戳记和用户标识提供附加的数据。这样可以保留作业说明的审计跟踪。

项目寄存器

项目寄存器不同于普通寄存器。它们在构建之间持久存在，可以通过管理控制台界面创建和查看，这使之成为存储一些配置信息的理想方式。

例如，您可以将 IBM® Rational® ClearCase® 配置规范存储为项目寄存器，然后通过一个步骤使用 `.pop -p` 命令来提取规范，并通过 `cleartool setcs` 命令来配置构建。这使您可以随同项目一起管理配置。

如果具有名为 ALPHA 的项目寄存器，您还可以具有名为 ALPHA 的普通寄存器，两者的内容完全不同。项目寄存器是一组独立的值。



您可以通过两种方式创建和访问项目寄存器：

- 通过点命令（`.push` 和 `.pop`），添加 `-p` 选项。在使用 `-p` 选项时，命令将引用项目寄存器，而不是普通寄存器。

例如，命令

```
.push -p ALPHA register.txt
```

将文件 `register.txt` 的内容放入名为 ALPHA 的项目寄存器中。

- 通过管理控制台界面。选择项目，然后单击所需项目名称旁边的编辑图标 。项目属性出现在面板的下半部；单击寄存器选项卡以显示项目的寄存器。该选项卡提供了用于管理寄存器的面板：
 - 要新建寄存器，请输入名称和内容，然后单击创建。
 - 要删除寄存器，请单击面板右侧列表中的寄存器名称旁的废纸箱图标。
 - 要编辑寄存器，请单击列表中该寄存器名称旁边的编辑图标 。系统会使用寄存器的内容填充寄存器面板。进行更改，然后单击保存已编辑的寄存器按钮。

所有有权访问项目的人员都可以查看和编辑项目的项目寄存器。

注：底层数据库的区分大小写确定在创建寄存器名称时可使用的大小写。

向/从步骤的服务器资源复制文件

可以使用点命令将一个服务器资源的文件复制到另一个服务器资源。本主题描述如何使用 `.get` 和 `.put` 命令（对于单个文件）以及 `.rget` 和 `.rput` 命令（复制整个目录树）。

要点： 服务器资源必须已启用了文件复制功能。缺省情况下，不启用该功能。请参阅『在服务器资源上启用文件复制』。

在服务器资源上启用文件复制

服务器的缺省设置不允许通过使用点命令来复制文件。要允许项目向/从服务器资源复制文件，请更改服务器的文件属性。

要更改设置，请执行以下操作：

1. 选择服务器 > **<ServerName>**。
2. 对于文件属性选择除“无”以外的值。可以启用从服务器复制文件和/或将文件复制到服务器。

从服务器获取文件

要从服务器获取文件的副本并将它放置在相对于当前步骤工作目录的目标位置中，请使用 `.get` 命令。例如，如果有一个名为 `winbuildserver1` 的服务器，并且在其 `config` 目录中含有 `config.txt` 文件，那么可以向项目中添加以下步骤，用以将文件复制到当前服务器的 `config` 目录中：

```
.get winbuildserver1:./config/config.txt ./config/config.txt
```

有关更多信息，请参阅以下内容：

- `.get` 的“参考”条目
- 第 348 页的『`.rget`』的“参考”条目
- 第 371 页的『作业的工作目录』中提供了有关如何解析步骤中的路径的描述

将文件置于服务器

要将当前服务器中的文件复制到其他服务器，请使用 `.put` 命令。下面的示例步骤假设在当前服务器上的 `config` 目录中有 `config.txt` 文件，并且从当前路径下可访问到文件：

```
.put ./config/config.txt winbuildserver1:./config/config.txt
```

有关更多信息，请参阅以下内容：

- `.put` 的“参考”条目
- `.rput` 的“参考”条目
- 第 371 页的『作业的工作目录』中提供了有关如何解析步骤中的路径的描述

配置直接文件传输

当 `_XSTREAM_PROTOCOL` 触发器变量出现在作业的环境中时，它支持在代理程序之间进行直接文件传输。发送方代理程序、接收方代理程序和引擎都必须支持直接文件传输。一些参数控制编码、网络和文件传输特性。

请参阅第 270 页的『触发器变量引用』和第 159 页的『`bfagent.conf` 引用』参考资料中的 `xstream_*` 参数。

对步骤处理进行故障诊断

如果您在步骤处理中遇到问题，请复审本主题中的信息，以查看是否存在可接受的变通方法或解决方案。

执行 **ANT** 构建命令后作业不会处理任何步骤命令

问题描述:

将不处理步骤中 **ANT** 构建命令后的命令。

在以下步骤示例中，**echo** 命令将不运行。

```
<path to ant bin directory> ant -f <path to Java project>\build.xml build  
echo "Ant build complete"
```

说明: 无论 **ANT** 构建失败还是成功，它都会返回错误代码 1。

在步骤的“命令”属性中，如果使用了多个命令，那么只有最后运行的命令的退出状态会影响步骤结果状态。当服务器运行包含 **ANT** 构建命令的步骤的命令脚本时，错误状态 1 会导致 **ANT** 构建后的所有命令失败。

解决方案:

创建一个步骤日志过滤器来处理 **ANT** 构建产生的步骤输出。该步骤日志过滤器用于设置步骤结果并确保处理作业中的下一个步骤。

1. **ANT** 构建应为步骤中的唯一命令或步骤中的最后一条命令。

没有日志过滤器，**ANT** 构建将返回错误代码 1 并且步骤结果设置为失败。

2. 创建日志过滤器可搜索步骤输出以查找相应的故障文本字符串（**BUILD FAILED**）并有效地控制步骤处理。

如果发现了该文本字符串，请使用“设置失败”操作将步骤结果设置为失败。使用步骤日志过滤器时，如果未发现该文本字符串，步骤结果将始终设置为通过。

有关设置步骤的日志过滤器的详细信息，请参阅第 290 页的『日志过滤器』。

执行 **Windows** 批处理命令后步骤中的命令不运行

问题描述:

执行 **Windows** 批处理命令后不再处理步骤命令。

在以下步骤示例中，第一个批处理文件后的两个命令不再运行。

```
C:\script1.bat  
C:\script2.bat  
echo "Performed both batch commands"
```

说明: 一个步骤中的所有命令都放在将由服务器运行的 Windows 批处理文件中。如果该步骤命令包含对批处理文件的引用, 那么在该批处理文件运行后该步骤将退出。在批处理文件引用后的步骤命令不会运行。

解决方案:

在步骤中使用 `call` 命令来运行批处理文件。`call` 命令在步骤批处理文件中运行。

```
call C:\script1.bat
call C:\script2.bat
echo "Performed both batch commands"
```

点命令参考

您可以在步骤的“命令”字段中使用点命令。它们提供对系统内的特殊功能和功能的访问。

可在步骤中将点命令和常规命令混合使用, 但是在单个步骤中可有多个点命令。但是, 不要在一个步骤中使用多个 `.scan` 命令, 因为如果这样做的话, 系统将无法准确报告命令的结果。

存在一个单独的点命令列表, 用于和环境变量一起使用。请参阅第 267 页的『在变量中使用点命令』。

点命令语法

语法规则: 每个点命令的描述都包含一个使用以下注释的语法规则:

- 用户提供的值显示在尖括号中: `<value>`
- 可选文本显示在方括号中: `[optional text]`

除非特别说明, 否则可以对命令参数使用环境变量。

如果点命令接受参数的环境变量, 您还可以使用:

- 反引号 (```) 表示命令执行
- 方括号 (`[` 和 `]`) 表明范围 (也称为范围表示法。)

请考虑以下 `.set` 示例:

```
.set envgroupname "F00[1,3-5]=`cat foofile`"
```

在此例中, 代理程序运行 `cat foofile`, 该命令列出 `foofile` 的内容。使用范围表示法, 抽取第 1 和第 3~5 行。然后, `.set` 命令使用这些行来更新名为 `envgroupname` 的环境组中的现有变量 `'FOO'`。

.bom

```
.bom addcategory "category"
.bom setcolumn "category" "section" "column" [...]
.bom data category "section" "column=value" [...]
```

`.bom` 命令用于将数据添加到构建的材料清单 (BOM) 中。用它可以添加类别、部分和数据。

类别 类别是显示在 BOM 上的标题。使用 `addcategory` 选项指定类别。

```
.bom addcategory "category"
```

部分 部分用于定义类别中的数据列。使用 `setcolumn` 选项指定部分和部分内的列。部分名称不会打印出来。部分的一组列标题会打印在部分的开头处。可以使用 `-p` 选项并指定父部分来嵌套部分。

必须先定义部分和列，然后 `data` 选项才能尝试将数据添加到部分和列中。

```
.bom setcolumn "category" "section" "columnheader" [...]  
.bom setcolumn "category" "section" -p parentsection"columnheader" [...]
```

数据 数据用于填充部分中定义的列。部分和列必须已使用 `setcolumn` 定义。

```
.bom data category" "section" "column=value" [...]
```

以下示例显示了指定类别、部分和列以及数据时必须采用的顺序。

```
.bom addcategory "Spaceships"  
.bom setcolumn "Spaceships" "Section1" "ShipName" "WarpSpeed" "Tonnage"  
.bom setcolumn "Spaceships" "Subsection1" -p "Section1" "ShippingDate" "ShippingManifest"  
.bom data "Spaceships" "Section1" "ShipName=SpaceShipOne" "WarpSpeed=9" "Tonnage=10000"  
.bom data "Spaceships" "Subsection1" "ShippingDate=123" "ShippingManifest=456"  
.bom data "Spaceships" "Section1" "ShipName=Freighter" "WarpSpeed=6" "Tonnage=20000"
```

以下示例显示了 BOM 中的内容:

ShipName	WarpSpeed	Tonnage		
SpaceShipOne	9	10000		
			ShippingDate	ShippingManifest
			123	456
Freighter	6	20000		

与其他点命令一起使用时，可以使用此命令中的环境变量。类似

```
.bom data "Spaceships" "${SECTION}" "ShipName=${NAME}" "WarpSpeed=${SPEED}" "Tonnage=${TONNAGE}"
```

的命令将使用早期的命令装入环境变量中的数据填充 BOM。

可以创建任意数量的列，但是填充最后一列之前，系统不会向 BOM 写入行。

如果省略了数据行中的列，系统将使用上一行的值，如下例所示。

```
.bom addcategory "Spaceships"  
.bom setcolumn "Spaceships" "Section1" "ShipName" "WarpSpeed" "Tonnage"  
.bom data "Spaceships" "Section1" "ShipName=SpaceShipTwo" "WarpSpeed=3" "Tonnage=30000"  
.bom data "Spaceships" "Section1" "ShipName=Tanker" "Tonnage=50000"
```

结果是重复 SpaceShipOne 中的 WarpSpeed 值:

ShipName	WarpSpeed	Tonnage
SpaceShipTwo	3	30000
Tanker	3	50000

.bomexport

描述

`.bomexport` 点命令用于将作业的 BOM 导出为 XML 文件。收集 BOM 信息之后，`.bomexport` 会将这些信息保存到您指定的文件和位置中。

路径和文件名为可选。缺省情况下，Build Forge 将 BOM 报告保存到服务器上该步骤的工作目录中，并使用标记名称作为文件名 (`<build_tagname>.xml`)。

请将 `.bomexport` 命令指定为项目中的最后一步。

语法

`.bomexport [path_name][file_name]`

选项

选项	描述
path_name	可选路径名。如果提供，那么路径必须相对 Build Forge 服务器上步骤的工作目录。如果省略，文件将保存到步骤的工作目录。
file_name	可选文件名。作业的 BOM 以 XML 格式保存到文件中。如果未提供文件名，那么将根据构建标记名和字符串 <code>_BOM</code> 来构造文件名： <code><build_tagname>_BOM.xml</code> 。

示例

```
.bomexport
.bomexport myproj.xml
.bomexport path/to/myproj.xml
.bomexport /path/to/myproj.xml
```

.break

`.break [<notification_group_name>]`

`.break` 命令用于在重新启动作业之前执行作业中止。系统遇到使用 `.break` 命令的步骤时，运行结果将为“已停止”。请使用**重新启动**图标来重新启动作业，以便继续执行 `.break` 步骤之后的步骤。

如果在链式作业中执行 `.break` 命令，系统将停止该链式作业，但将控制权返还给主调作业，该作业继续处理步骤。

您可以在该命令中包含作为可选自变量的访问组；如果执行此操作，系统将在停止作业时向指定访问组发送一封电子邮件消息。

.bset

```
.bset env "<VarName>=<Value>" [...]
.bset selector <SelectorName> [<SelectorSnapshotName>]
.bset server <ServerName>
.bset buildserver <ServerName>
```

`.bset` 命令用于在作业期间临时更改项目设置。

注：`.bset` 命令会对后续步骤使用的基本构建环境产生影响。请勿指定依赖于同一步骤中先前 `.bset` 命令的 `.bset` 命令。例如，请勿指定与同一步骤中的以下组合相似的步骤。

```
.bset env "VAR1=VALUE1"
.bset env "VAR2=$VAR1"
```

此命令的选项如下：

- **env** 更改在运行的作业的一个或多个项目环境变量的值。更改在当前步骤中立即生效。有关以这种方式使用此命令的更多信息，请参阅第 327 页的『作业期间更改构建标记』。可以设置尚不存在的变量。使用 `.bset` 命令设置的值将写入作业记录。如

果为同时在项目或步骤环境中定义的变量设置新值，该新值只有在作业执行期间才会生效。项目或步骤的环境不会发生更改。

- **selector** 更改作业期间的项目选择器。新选择器在下一个步骤中生效（包括当前步骤的任何“内联”步骤）。新选择器仅用于无显式选择器设置（步骤选择器设置为项目缺省值）的步骤。如果使用“内联”步骤中的 **selector** 选项，该选项会影响内联项目中的所有后续步骤，包括“内联”步骤中内联的所有步骤。它不会影响调用步骤或不低于调用步骤所在级别的任何其他步骤。

有显式选择器设置的步骤不受影响。

使用可选的 `<SelectorSnapshotName>` 来定义所指定选择器的快照。

- **server** 更改作业期间的缺省项目服务器。新服务器设置在下一个步骤中生效（在当前步骤的任何“内联”步骤后）。步骤仅使用指定的服务器；它们没有显式选择器设置。

反引号语法：可以使用倒单引号将服务器名称设置为命令的输出。例如，以下命令运行 `SelectAServer.sh` 脚本并提供其输出作为 `.bset server` 服务器命令的服务器名称：

```
.bset server `SelectAServer.sh`
```

- **buildserver** 更改作业期间的缺省项目服务器。新服务器设置在下一个步骤中生效（在当前步骤的任何“内联”步骤后）。它不同于 **server** 选项，因为它应用于内联步骤以及项目级别的步骤。

注：避免在线程步骤中使用多个 `.bset` 命令。

.buildstatus

```
.buildstatus <result>
```

该命令强制构建在完成后处于指定的 `<result>`，而不管构建步骤的结果如何。

`<result>` 的有效值包括：

- P** 将结果设置为“已通过”
- F** 将结果设置为“已失败”
- W** 将结果设置为“警告”
- B** 将结果设置为“已停止”（如同已指定 `.break` 命令）

.date

```
.date <conversion_specifier>
```

将 `.date` 命令与作为参数的一个或多个转换说明符一起使用，可在项目运行时生成当前日期时间信息。

`.date` 命令及其参数必须定义为某个环境中的环境变量。然后，可对项目或步骤指定该环境。

例如，在某个环境中，定义环境变量 `DayofWeek` 并为其指定值 `.date %A`。将该环境指定给某个项目或步骤。如果该项目在星期三运行，作业会将文本 `Wednesday` 指定给环境变量 `DayofWeek`。

要点: 不能直接在步骤“命令”字段中引用 `.date` 命令。

注: 每个步骤中都会对 `.date` 命令重新求值。要保留某个特定时间，请将 `.date` 与 `.bset env` 配合使用。例如，使用 `$ORIGTIMESTAMP=.date %d-%b-%Y.%H:%M:%S`，然后使用 `.bset env "TIMESTAMP=`echo $ORIGTIMESTAMP`"`。

`.date` 命令基于 POSIX `strftime` 函数构建，它接受 ANSI C89 标准可识别的转换说明符。下表提供了转换说明符的日期时间值。

注: 并非所有的转换说明符都可以在各个语言环境和操作系统之间移植。请在计划使用 `.date` 命令的服务器操作系统上和语言环境中测试该命令的结果。

日期转换说明符	描述
%a	根据当前语言环境的星期几缩写。
%A	根据当前语言环境的星期几全称。
%b	根据当前语言环境的月数缩写。
%B	根据当前语言环境的月数全称。
%c	当前语言环境的首选日期和时间表示方式。
%d	十进制数字形式的月度日期（范围为 01 到 31）。
%H	十进制数字形式的 24 小时制小时数（范围为 00 到 23）。
%I	十进制数字形式的 12 小时制小时数（范围为 01 到 12）。
%j	十进制数字形式的年度日期（范围为 001 到 366）。
%m	十进制数字形式的月数（范围为 01 到 12）。
%M	十进制数字形式的分钟数（范围为 00 到 59）。
%p	根据提供的时间值为“AM”或“PM”，或当前语言环境的相应字符串。下午视为“pm”，午夜视为“am”。
%S	十进制数字形式的秒数（范围为 00 到 61）。
%U	十进制数字形式的当年周数，范围为 0 到 53，第一个星期天（Sunday）作为第 01 周的第一天。

日期转换说明符	描述
%w	十进制形式的周度天数，范围为 0 到 6，星期天（Sunday）是 0。
%W	十进制数字形式的当年周数，范围为 0 到 53，第一个星期一（Monday）作为第 01 周的第一天。
%y	十进制数字形式，不带世纪数的年数（范围为 00 到 99）。
%Y	十进制数字形式，包含世纪数的年数。
%Z	时区或名称或缩写。
%%	字面值“%”字符。

.defect

描述 .defect 命令用于将针对缺陷跟踪应用程序的适配器添加到项目步骤中。缺陷适配器是一个 Build Forge 对象；它基于针对缺陷跟踪应用程序的适配器模板。项目运行时，将运行此步骤的适配器代码。

语法

```
.defect <adaptor_name> [entry_name]
```

<adaptor_name> 是必需的，它是在管理控制台中分配给适配器的名称。<adaptor_name> 的大小写应与控制台中使用的大小写匹配。

如果适配器模板具有多个接口函数，请使用 entry_name 选项指定一个要运行的函数。entry_name 必须与为适配器模板中 <interface> 元素指定的 name 属性匹配。如果 entry_name 中指定的 <interface> 元素不存在或无法找到，那么将改为运行缺省的 <interface> 元素。在以下示例中，条目名称为 DefectFunction。

如果在使用适配器链接，将自动调用适配器，并运行适配器模板中的第一个接口函数。要在适配器模板中运行其他接口，请将要运行的接口的 default 属性设置为 true (default="true")。

示例

```
.defect MyClearCaseQuestAdaptor
.defect MyClearCaseQuestAdaptor DefectFunction
```

注释 要创建适配器或查看适配器列表，请选择项目 > 适配器。

Build Forge 产品随附的适配器模板位于：

```
<bfinstall>/interface
```

.drill

```
.drill [through]
    <"var1,var2,var3"|${EnvVar|-r[p] Register>
    [grouped by] "{}"
    [separated by] ","
    [exec] "Command $1 $2"
```

`drill` 命令用于通过一个命令进行循环，从而对一系列值的每个成员执行一次此命令。可以在命令行上指定这些值，或者从环境变量或寄存器进行绘制。系统运行 `.drill` 命令时，系统将使用 `.drill` 语法来构造一系列命令行，并将这些命令行发送到代理程序来执行。

例如，命令 `.drill "A,B,C,D" "echo value $1"` 将创建下列命令：

```
echo value A
echo value B
echo value C
echo value D
```

分组

可以对这些值进行分组，并使用 `$n` 语法引用各组中的多个值。`$1` 引用组中的第一个值，`$2` 引用组中的第二个值，依此类推。例如，`.drill through "(A,B,C,D,E),(B,C,D,E,F),(C,D,E,F,G)" grouped by "(" separated by "," exec "echo 1[$1] 2[$2] 3[$3] 4[$4] 5[$5]"` 将创建下列命令：

```
echo 1[A] 2[B] 3[C] 4[D] 5[E]
echo 1[B] 2[C] 3[D] 4[E] 5[F]
echo 1[C] 2[D] 3[E] 4[F] 5[G]
```

注：没有缺省的分组字符。但是存在缺省的分隔符字符，即逗号。如果不指定 `grouped by`，系统将查找提供的以分隔符字符分隔的值，并将每个此类字符串视为一个值。例如，命令 `.drill "(A,B),(C,D)" "echo $1 $2"` 将解析为下列命令：

```
echo (A 2
echo B) 2
echo (C 2
echo D) 2
```

数据源

对于 `.drill` 命令从何处获取它循环的数据，您有几个选择。此命令的第一个参数是数据源。可以加入可选的命令字“`through`”来指示数据源。

- 可以在命令行中明确列出数据，在下面的命令中，将通过值 `one`、`two` 和 `three` 循环：
`.drill through "one,two,three" exec "echo $1"`
- 可以从环境变量绘制数据。下面的命令假定环境变量 `FILENAMES` 是以逗号分隔的文件列表，并且使用 `DOS` 命令删除该列表中的所有文件：
`.drill through $FILENAMES exec "del $1"`
- 可以从寄存器或项目寄存器绘制数据。如果 `RegisterA` 包含以逗号分隔的文件名列表，那么下面这个发出到 Linux 系统的命令将写出各文件的内容：
`.drill -r RegisterA exec "cat $1"`

而下面的示例执行的操作相同，但使用的却是项目寄存器：

```
.drill -rp ProjectRegisterA exec "cat $1"
```

.edit

```
.edit /<search_expression>/<replace_expression>/ [<relative_path>/]file [file ...]
```

.edit 命令用于搜索或替换一个或多个文件中的文本字符串。.edit 命令替换指定的各文件中每行上的字符串（search_expression）的第一个实例。除非指定了相对路径，否则假定文件位于此步骤的工作目录中。

.edit 命令实施用于匹配和替换的 POSIX 标准正则表达式，包括在替换模式中使用 () 子串选择和 \N 替换。

.edit 命令缺省情况下使用 POSIX 扩展正则表达式语法。如果通过 Perl 兼容的正则表达式支持编译了代理程序，那么替换表达式会后跟“p”字符来指定 PCRE 语法。

将反斜杠用作转义字符需要进行不同的处理，具体取决于代理程序主机上的操作系统。

- 在 UNIX 和 Linux 上，代理程序处理会将表达式解释两次，因此需要使用四个反斜杠来代表一个反斜杠。例如，使用四个反斜杠和一个句点与字面的句点字符匹配。

```
\\\\.
```

- 在 Windows 上，只需两个反斜杠以与相同的字面句点匹配。

```
\\..
```

不能将定界符转义。在看似需要的情况下，正确的方法是使用备用定界字符。示例：您有一个仅包含字符 abc 的文件 abc.txt。您希望将其替换为字符串 a/c：

- 错误 - 在这些情况下，字符串保持不变，因为定界字符优先于转义字符。

```
.edit /abc/a\c/ abc.txt
.edit /abc/a\\c/ abc.txt
.edit /abc/a\\\c/ abc.txt
```

- 正确 - 更改定界字符允许使用 a/c 进行替换。

```
.edit "abc"a/c" abc.txt
```

注：必须明确列出一个或多个文件名，不得使用通配符。

例如：下面的命令替换 drivemakefile 文件中 winXPdriver 和 win2000driver 之类的字符串。

```
.edit /win.*driver/linuxdriver/ drivemakefile
```

.edit 命令类似 .strsub 命令，不同之处包括：

- 在大文本文件或多个文件中执行替换时，.strsub 命令的速度比 .edit 快。
- .edit 命令可以执行正则表达式搜索和替换。
- .edit 命令只能替换指定的各文件中每行上的字符串（search_expression）的第一个实例。
- .strsub 命令则可替换各文件中每行上的字符串（source）的每个实例。

.email

```
.email <recipient>
```

该命令使用 step_email 邮件模板向 recipient 发送电子邮件。理想情况下，所指定的收件人是 Build Forge 用户，但 .email 还会将邮件发送到任何有效的电子邮件地址。

.export

```
.export [path_name][file_name]
```

`.export` 命令将主调项目的项目定义保存到步骤工作目录中的 XML 文件。该 XML 文件描述了项目及其步骤。它不描述其他相关联的项目，例如服务器。

导出的 XML 文件可用于将项目定义导入到管理控制台。

`.export` 命令可以采用可选的路径和/或文件名。路径必须是相对路径。其从步骤的工作目录进行应用。

如果未提供文件名，将从当前项目标记 \$BF_TAG.xml 构造文件名。

.get

```
.get server:[[<relative_path>/]file/]file [[[<relative_path>/]file/]file]
```

`.get` 命令用于将文件从一个逻辑服务器传送到另一个服务器。`.get` 操作在当前服务器/路径中运行，并从指定的服务器/路径检索文件。目标路径名相对于该步骤的当前工作目录。源路径名是指定服务器基本路径的相对路径。服务器必须指定允许对文件执行 `.get` 操作的逻辑服务器（请参阅第 331 页的『在服务器资源上启用文件复制』）。只能传送单个文件。

指定路径时可以包含环境变量。这样您就可以相对于特定作业使用的路径指定文件。请参阅第 371 页的『作业的工作目录』中作业所用路径的描述。

如果使用的服务器名称中包含空格，请用引号括起服务器名称和文件路径。例如：`.get [<relative_path>/]file server:[[<relative_path>/]file]"`

传输的速度并不快，因此，您可能希望选择其他方法来传输大型文件。预期速度不会超过 40 KB 每秒，传输 70 MB 的文件可能需要 45 分钟到 1 个小时。

注：如果目标文件已经存在，那么目标文件会被覆盖而不会显示任何警告。

.load

```
.load [-o] [-e] [-v] [-j] [<relative_path>/]<filename>  
.load -r|-p <registername>  
.load -s `<命令名>`
```

`.load` 命令从 XML 文件装入项目，并将所装入项目的步骤添加到当前项目中，运行了 `.load` 命令的步骤执行完毕后，将允许项目在运行时动态创建和装入步骤。您可以通过使用选项使 `.load` 命令从寄存器或从命令的输出提取其数据。

要为 `.load` 命令编写 XML 文件，请首先使用来自现有项目的导出文件来提供适当的基本结构。还可以在系统内创建项目，然后将其导出以便在 `.load` 命令中使用。本主题包含示例 XML 代码。

`.load` 命令装入的步骤可以包含对内联或已连接项目的引用。缺省情况下，系统在 XML 文件内查找内联项目的定义并装入这些项目的步骤；请参阅本主题后面部分的 `-e` 选项以了解让系统从数据库获取内联项目定义的方法。无论对通过链式项目还是失败链式项目，系统始终在数据库中查找项目定义。

注：对于通过 `.load` 运行的 JPO 步骤，已标识为“else 内联”的项目或库必须是系统中已经存在的项目或库，否则不会运行内联。

XML 文件中的多个项目

由于系统将内联项目随其调用方项目导出，因此 XML 文件可能包含若干项目。`.load` 命令运行在文件中标注为主项目的项目。该项目在其 `<project>` 元素中具有属性 `primary="1"`。

命令选项和参数

最简单的命令格式为 `.load <filename>`。可以在文件名的前面加上可选的路径名（相对于作业目录）。例如，命令

```
.load ../../project.xml
```

将从服务器目录（包含项目和作业目录的目录）装入文件 `project.xml`，假定步骤的路径属性为“/”（缺省值）。

注：正常步骤启动内联项目时，系统将访问数据库以获取该项目的当前定义；当 `.load` 命令导入的步骤启动内联项目时，系统将在 XML 文件内查找该内联项目的定义。有关避免这种情况的方法，请参阅下面对 `-e` 选项的描述。

注：由 7.1 之前版本的 Build Forge 生成的步骤 XML 文件不受支持，可能会失败，因为这些元素通过名称而非标识引用。

注：`.load` 使用导入实用程序，因此会根据导入实用程序的设置来应用访问组所有权。请参阅第 406 页的『访问组如何指定给导入的对象』以了解更多信息。

此命令的选项如下：

-r 或 -p

这些选项可使系统从寄存器装入步骤。请通过命令行使用这些选项。

```
.load -r|-p <registername>
```

`-r` 选项从普通寄存器装入步骤，而 `-p` 选项则从项目寄存器装入步骤。可以在项目前面的步骤中，在寄存器内构建数据，然后使用此命令从寄存器装入步骤。

-s 此选项可使系统运行命令并将该命令的输出用作要装入的数据。请使用命令行

```
.load -s <命令名>
```

-e 设置了 `-e` 选项时，系统将从数据库而不是从装入的 XML 文件获取内联项目。它将 `chainID` 的值视为对数据库内某个项目标识的引用。这将使您的 XML 文件能够引用内联项目的最新版本（而不是引用 XML 文件中的项目），或引用 XML 文件中不包含的项目。

-o `-o` 选项用于禁用 XML 文件内的内联项目。使用此选项时，系统将忽略主项目内的任何内联项目。包含对内联项目的引用的步骤将运行其命令，但是忽略其内联。

-j 如果 XML 文件中的最后一组步骤和 `.load` 命令后的步骤都已线程化，请使用 `-j` 选项。`-j` 选项将最后线程化的步骤转化为连接步骤。否则，已线程化的步骤将成为 `.load` 命令后线程化步骤块的一部分。

-v 将装入的 XML 文件的内容发送到显示终端（`stdout`）以便查看。

示例 XML

下面的示例显示要与 .load 命令一起使用的 XML 文件。该 XML 文件是通过导出名称为 HelloWorldPlusInline 的项目创建的。

请注意此示例 XML 文件的下列详细信息：

- 此 XML 文件包含两个 <project> 元素。
- 此 XML 文件中的第一个项目是主项目；它具有属性 name="HelloWorldPlusInline" 和 primary="1"。
- 此 XML 文件中的第二个项目称为 Sleepytime，具有指示它不是主项目的属性 primary="0"。
- HelloWorldPlusInline 的第一个步骤称为 EchoHelloWorld，其中包含一个 echo 命令和一个 chainID 属性。chainID 属性有一个值 2，指示系统应内联标识为 2 的项目，即 Sleepytime 项目。

注：请忽略步骤属性 inline；不推荐此属性，因为已不再使用该属性。所有步骤都具有这个属性（值为 N）。要确定步骤是否具有内联项目，请查找属性 chainID。chainID 的值代表项目的标识。缺省情况下，系统在 XML 文件内查找内联项目，但是如果在 .load 命令中使用 -e 选项，那么系统会将该值视为数据库内的项目标识。这样，您就可以创建自己的 .load 文件夹，而无需在其中加入内联项目。

注：不支持针对由 V7.0.x 和更早版本的 Build Forge 生成的 XML 文件使用 .load，在上述情况下该命令无法正常运行，因为较早的语法通过名称而非 UUID 来引用对象。

- 每个项目都有一个 id 属性。此标识值与数据库中该项目的标识相同。可以通过从安装目录执行下面的命令来获取项目标识列表：

```
bfexport -l
```

- commandStore 属性包含 <command> 元素的内容副本，以及通过 XML 转义的特定字符。例如，在 <command> 中，可能包含 "、'、&、<、> 或换行符。这些字符在 commandStore 中分别由 "、'、&、<、> 和
 表示。

```
<?xml version="1.0" encoding="UTF-8"?>

<buildforge schema="7.115014" comment="">
  <project access="6" active="Y" name="HelloWorldPlusInline" primary="1"
    selectorId="Choose_local" maxthread="0" increment="Y" tagsync="0"
    buildclass="Production" sticky="N" envId="0" tag="BUILD_$B" id="19" exclusive="0">
    <tagvar autoincrement="Y" name="B" id="1">2</tagvar>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="1" passwait="N" inline="N" threadable="N" chainId="2"
      access="6" active="Y" passnotify="0" description="EchoHelloWorld" onfail=" "
      failnotify="0" envId="0" commandStore="echo Hello World">
      <command>echo Hello World</command>
    </step>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="2" passwait="N" inline="N" threadable="N" access="6"
      active="Y" passnotify="0" description="export proj to build and server folders"
      onfail=" " failnotify="0" envId="0"
      commandStore=".export $BF_PROJECTNAME_PHYS.xml&#10;copy /Y $BF_PROJECTNAME_PHYS.xml ..\">
      <command>.export $BF_PROJECTNAME_PHYS.xml
copy /Y $BF_PROJECTNAME_PHYS.xml ..\"</command>
    </step>
  </project>
  <project access="6" active="Y" name="Sleepytime" primary="0"
```

```

selectorId="Choose_local" maxthread="0" increment="Y" tagsync="0"
buildclass="Production" sticky="N" envId="0" tag="SLEEP_$B" id="2" exclusive="0">
  <tagvar autoincrement="Y" name="B" id="1">21</tagvar>
  <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
    timeout="300" id="1" passwait="N" inline="N" threadable="N" access="6"
    active="Y" passnotify="0" description="Sleep, perchance to dream" onfail=" "
    failnotify="0" envId="0" commandStore=".sleep 0">
    <command>.sleep 0</command>
  </step>
</project>
<class maxdays="0" access="1" entranceprojectId="1" name="Production" keepfiles="B"
  deletechangedata="N" purgeprojectId="2" exitProjectId="5" candidates="AnyBuild "
  maxbuilds="0"></class>
  <selector operator="" required="" access="6" value="" name="Choose_local"
    selectorId="" property=""></selector>
</buildforge>

```

.lock

```
.lock
```

.lock 命令可使系统在作业完成之后将其锁定。这样可以防止作业由于其类属性而被自动删除；并且，锁定的运行不会在**作业 > 已完成**选项卡上列出，而是在**已锁定**选项卡上显示。此命令不采用参数；它锁定已在使用的作业。

.mkdir

```
.mkdir <relative_path>
```

.mkdir 命令用于创建目录。<relative_path> 参数解释为从当前步骤目录开始的相对路径。如果指定的路径名称中的目录不存在，会创建这些目录。不允许使用绝对路径和包含盘符的路径（如 C:\ ）。

.monitor

```
.monitor [-c] [-w] <interval> [<relative_path>/]<filename>
```

.monitor 命令可使系统在查看文件以了解文件大小何时停止变化时中止项目。当步骤发出此命令时，系统将检查指示的文件；然后每隔 <interval> 秒重新检查此文件。当两个时间间隔之间文件大小不再变化时，系统将继续到下一步。

如果使用 -c 选项，系统确定文件的大小不再变化之后，会将受监视的文件的内容写到步骤日志中，然后继续到下一步。

如果此文件不存在，系统将不等待，而是在第一个时间间隔之后立即继续。使用 -w 选项可强制系统在开始监视进程之前等待创建此文件。

.pack

描述 .pack 命令用于为打包应用程序向项目步骤添加适配器。打包适配器是一种 Build Forge 对象，它基于打包应用程序的适配器模板。项目运行时，将运行此步骤的适配器代码。

语法

```
.pack <adaptor_name> [entry_name]
```

`<adaptor_name>` 是必需的，它是在管理控制台中分配给适配器的名称。
`<adaptor_name>` 的大小写应与控制台中使用的大小写匹配。

如果适配器模板具有多个接口函数，请使用 `entry_name` 选项指定一个要运行的函数。`entry_name` 必须与为适配器模板中 `<interface>` 元素指定的 `name` 属性匹配。如果 `entry_name` 中指定的 `<interface>` 元素不存在或无法找到，那么将改为运行缺省的 `<interface>` 元素。在以下示例中，条目名称为 `PackageFunction`。

如果在使用适配器链接，将自动调用适配器，并运行适配器模板中的第一个接口函数。要在适配器模板中运行其他接口，请将要运行的接口的 `default` 属性设置为 `true` (`default="true"`)。

示例

```
.pack MyPackagingAdaptor
.pack MyPackagingAdaptor PackageFunction
```

注： 要创建适配器或查看适配器列表，请选择项目 > 适配器。

Build Forge 产品随附的适配器模板位于：

```
<bfinstall>/interface
```

.pop

```
.pop [-p] <register_name> [+][<relative_pathname>|-]
.pop [-p] <register_name> [>|>>]<register_name>
```

将寄存器的内容写入文件、步骤日志或另一个寄存器。

可选的 `-p` 参数用于使命令引用项目寄存器。项目寄存器与普通的寄存器隔开，而项目寄存器将在作业结束之后永久存在。

下列示例显示 `.pop` 命令的各种用法：

- `.pop A data.txt`

将寄存器 `A` 写入文件 `data.txt`，该文件位于此步骤的工作目录中。

- `.pop ver +data.txt`

将寄存器 `ver` 的内容追加至文件 `data.txt`。

- `.pop Alpha`

将寄存器 `Alpha` 的内容写入此步骤的日志中。

- `.pop ALPHA > BETA`

使寄存器 `BETA` 的内容与寄存器 `ALPHA` 的内容相同。

- `.pop A >> B`

将寄存器 `A` 的内容附加到寄存器 `B` 中。

注： 使寄存器出栈不会将其清空。要更改寄存器的内容，请使用 `.push` 命令将一个新值推送到其中。

.poptag

```
.poptag [-p]<registername>
```

`.poptag` 命令用于更改当前标记，将其替换为指定寄存器的内容。

可选的 `-p` 参数用于使命令引用项目寄存器。项目寄存器与普通的寄存器隔开，而项目寄存器将在作业结束之后永久存在。

.purge

```
.purge
```

使用 `.purge` 命令可设置一个使作业在完成之后立即被清除的标志。但是，在 `.purge` 命令之后运行的 `.lock` 命令将导致作业被保存。可以使用此命令创建仅当成功完成了自己的所有步骤时才保存的作业。要创建这样一个作业，可将 `.purge` 命令作为项目中的第一个步骤，而将 `.lock` 命令作为最后一个步骤。

.push

```
.push [-p] [+]<register_name> [<relative_pathname> | -]
```

将 `<relative_pathname>` 的内容放入寄存器 `<register_name>` 中。将替换 `<register_name>` 的当前内容。

如果要追加而不是替换，请在 `<register_name>` 前面放置一个加号 (+)。

如果要清除寄存器，请使用连字符代替 `<relative_pathname>`。

可选的 `-p` 参数用于使命令引用项目寄存器。项目寄存器与普通的寄存器隔开，而项目寄存器将在作业结束之后永久存在。

`<relative_pathname>` 在没有启用步骤的“绝对”属性的情况下是项目或标记路径下的相对路径。

下列示例假定没有为步骤启用“绝对”属性：

- `.push Alpha data.txt`

该步骤的工作目录中的 `data.txt` 文件的内容将放入寄存器 `Alpha`。

- `.push +B ../newdata.txt`

该步骤的工作目录的父目录中 `newdata.txt` 文件的内容将追加到寄存器 `B`。

- `.push ALPHA -`

将清除寄存器 `ALPHA`。

.put

```
.put [<relative_path>/]file server:[[<relative_path>/]file]
```

`.put` 命令用于将文件从一个逻辑服务器传输到另一个逻辑服务器。`.put` 操作在当前服务器/路径中运行，并将指定的文件发送到远程服务器。目标路径名与目标服务器的基本

路径相对。源路径名与步骤的当前工作目录相对。远程服务器必须指定允许对文件执行 `.put` 操作的逻辑服务器（请参阅第 331 页的『在服务器资源上启用文件复制』）。只能传送单个文件。

指定路径时可以包含环境变量。这样您就可以相对于特定作业使用的路径指定文件。请参阅第 371 页的『作业的工作目录』中作业所用路径的描述。

如果使用的服务器名称中包含空格，请在名称两边加上引号。

传输的速度并不快，因此，您可能希望选择其他方法来传输大型文件。预期速度不会超过 40 KB 每秒，传输 70 MB 的文件可能需要 45 分钟到 1 个小时。

注：如果目标文件已经存在，那么目标文件会被覆盖而不会显示任何警告。

.rem

描述

`.rem` 点命令用于注释。记录步骤中命令的有关信息时，它很有用。

示例

```
.rem This is some comment text here
.rem all text after .rem is ignored.
```

.report

将在“快速报告”中创建的报告的报告输出添加到作业 BOM。快速报告是 Rational Build Forge 的独立许可功能。

描述 `.report` 命令用于将报告添加到项目。

`.report` 命令运行报告并将报告输出添加到作业 BOM。运行作业时，BOM 都会根据当前数据显示报告结果。

语法

```
.report <report_name>
```

`<report name>` 具有以下限制：

- `<report name>` 是必需的。
 - 该名称区分大小写。
 - 如果名称中包含空格，请在名称前后使用引号。例如，`"my_report"`。
- 指定的 `<report_name>` 必须另存为公共报告。
- BOM 报告类型不受支持。

.retag

```
.retag <new_tag>
```

步骤中的 `.retag` 命令用于在作业期间更改作业的标记。可将变量或命令用作新标记的值。

.retry

```
.retry <count> <command>
```

使用 `.retry` 命令可允许命令在失败时重试。`.retry` 命令采用指定次数的一个计数参数来重试命令。要运行的命令被视为这些参数的剩余部分，因此 `.retry` 命令必须是步骤的最后一个点命令。例如，请考虑以下命令：

```
.retry 3 myscript.sh arg1 arg2 arg3
```

将在舍弃此步骤之前最多运行“myscript.sh arg1 arg2 arg3”三次。第一个返回成功状态的命令调用将停止重试进程。

.rget

```
.rget server:[<path>] [<path>]
```

`.rget` 命令的作用类似于 `.get` 命令，但它以递归方式复制整个目录树。必须提供目录名称来充当参数。例如，命令

```
.rget winbuildserver1:config myconfig
```

会将服务器 `winbuildserver` 上的目录 `config` 的内容复制到当前服务器上的 `myconfig` 目录中。

注：

- 请勿在此命令中使用环境变量。
- 请勿在此命令中使用符号链接。
- 在尝试使用命令之前，指定的目录必须存在。如果指定的目录不存在，该命令将失败。

.rmdir

```
.rmdir <relative_path>
```

`.rmdir` 命令用于除去 `<relative_path>` 指定的目录。系统将除去路径名指定的基本目录，包括所有内容和后代。

.rput

```
.rput [<relative_path>] server:[<relative_path>]
```

`.rput` 命令的作用类似于 `.put` 命令，但它以递归方式复制整个目录树。提供的相对路径必须是目录，而不得为文件。例如，命令

```
.rput myconfig linuxserver5:feb2005
```

会将目录 `myconfig` 的内容从当前服务器复制到服务器 `linuxserver5` 的目录 `feb2005` 中。

源路径相对步骤的工作目录，因此包含（或不包含）项目和标记目录，具体取决于步骤的“绝对”属性的值。目标路径仅相对于目标服务器的“路径”属性。有关系统如何构造路径的更多信息，请参阅第 371 页的『作业的工作目录』。

注：

- 请勿在此命令中使用环境变量。
- 请勿在此命令中使用符号链接。
- 在尝试使用命令之前，指定的目录必须存在。如果指定的目录不存在，该命令将失败。

.run 和 .runwait

```
.run [-c "<condition>"] "<ProjectName>" ["<ProjectSnapshotName>"]  
.runwait [-c "<condition>"] "<ProjectName>" ["<ProjectSnapshotName>"]
```

.run 和 .runwait 命令可用于从步骤命令启动链式项目。要指定项目的快照，请使用可选的 <ProjectSnapshotName> 参数。

这些命令的行为与它们启动了项目之后的行为不同：

- .run 命令将指定项目作为链启动，遵循用于已连接项目的环境变量继承的规则。
- .runwait 命令启动指定的项目。启动步骤将等待已启动项目完成。已启动的项目完成时，系统将根据已启动项目的完成状态将启动步骤的结果值设为 pass 或 fail。

要点： 包含具有 .runwait 的步骤的项目在运行时使用两个作业插槽。如果可用的作业插槽不足，该步骤将由于出错而失败。

要点： 任何通过 .run 或 .runwait 命令启动的项目都不会在 Build Results 页面上生成链链接图标。

根据条件启动

可以使用可选的 -c 参数设定根据条件启动。可以在条件中使用环境变量。条件可以有几种形式：

字符串比较

可使用等于 (=) 或不等于 (!=) 运算符对字符串求值。如果比较求值结果为 true，那么启动链。

数字比较

可以使用 <、>、<>、>< 或 = 运算符比较两个数字值。

命令成功

可以将用反引号括起来的命令用作 -c 参数的值。系统运行此命令；如果成功，那么启动链。

示例

```
.run "BuildWindowsDriver"
```

系统启动 BuildWindowsDriver 项目。正在启动的项目将立即继续执行下一步。

```
.runwait "BuildWindowsDriver"
```

系统启动 BuildWindowsDriver 项目。系统在 .runwait 步骤暂停正在启动的项目。当 BuildWindowsDriver 项目完成并通过时，.runwait 步骤的状态将设置为 pass。

```
.run -c "$HOMEDRIVE=C:" "Simple Echo"
```

当且仅当 HOMEDRIVE 变量的值为 C 时，系统才运行 Simple Echo 项目。

此命令会生成如下日志输出（在步骤日志的 EXEC 部分）：

- 当 HOMEDRIVE 为 C: 时

```
.run Condition: 'C:' = 'C:' satisfied.
```

```
Queueing Project "Simple Echo" on server [WinBox].  
Queued Build 'BUILD_202' of project 'Simple Echo'.
```

- 当 HOMEDRIVE 不为 C: 时

```
.run -c "$HOMEDRIVE=C:" "Simple Echo"
```

```
.run Condition: 'D:' = 'C:' unsatisfied, no project queued.
```

如果字符串包含数字，系统将比较这些字符串的数字部分。例如，将以如下方式进行处理。

```
.runwait -c "a12b<c42d" "Simple Echo"
```

```
.run Condition: '12' < '42' satisfied.
Queueing Project "Simple Echo" on server [WinBox].
Waiting for .run build (4411) to complete.
.run build is now running.
.run build has finished.
Build 'BUILD_203' of project 'Simple Echo' completed.
```

```
.runwait -c "f43g<h43i" "Simple Echo"
```

```
.run Condition: '43' <> '43' unsatisfied, no project queued.
```

下列示例显示如何将命令用作条件：请注意：必须将命令用引号和反引号括起来。

```
.run -c "`exit 1`" "Simple Echo"
```

```
Env .run encountered an error during variable expansion,
parameter [exit1] expanded to [].
Expansion returned non-zero exit, project will not be queued.
```

```
.run -c "`exit 0`" "Simple Echo"
```

```
Expansion returned zero exit, project will be queued.
Queueing Project "Simple Echo" on server [WinBox].
Queued Build 'BUILD_204' of project 'Simple Echo'.
```

使用 `.runwait` 但构建失败时，日志将类似如下内容。

```
.runwait "Fail Build"
```

```
Queueing Project "Fail Build" on server [WinBox].
Waiting for .run build (4413) to complete.
.run build is now running.
.run build has finished.
Build 'BUILD_3' of project 'Fail Build' Failed, setting step status to fail.
```

.scan

```
.scan [-v][-i <ignorepattern>] baseline | checkpoint
```

`.scan` 命令用于扩充 BOM 中存储的作业数据。它跟踪步骤的工作目录中的文件，以及每个文件的 MD5 值。

.scan baseline

在步骤的工作目录中存储包含所有文件的列表。系统将在作业 BOM 中把此列表显示为类别。您可以在一个作业中具有多条基线命令，但在每条命令运行时，它们都会将列表重置为此步骤的工作目录的状态。最终的 BOM 将仅显示一个基线类别。

.scan checkpoint

存储自作业中上次执行 `.scan baseline` 或 `.scan checkpoint` 命令以来所有

新建、已更改和已删除文件的列表。系统在 BOM 中显示该列表。每个 checkpoint 命令都会在 BOM 中新建一个类别。

您必须在作业的第一个 .scan checkpoint 命令之前使用 .scan baseline 命令。在 .scan baseline 命令之前的 .scan checkpoint 命令会被忽略。

命令选项:

- v** 在作业日志中记录更改信息的副本。
- i** 忽略与提供的模式匹配的目录。模式可以与路径的开始、结尾或任何目录部分匹配。可使用此选项从更改列表中除去源控制目录。

CVS 示例:

```
.scan -i CVS checkpoint
```

示例命令从报告中除去 CVS 目录。

子版本示例:

```
.scan -i .svn baseline
```

如果将 -v 与 -i 一起使用, 系统将记录对源控制目录的更改, 但这些更改不会包括在 BOM 中。

注: 请勿在单个步骤中使用多个 .scan 命令。如果在一个步骤中使用多个 .scan 命令, 系统将无法提供这些命令的准确输出。

有关使用这些命令的更多信息, 请参阅第 364 页的『使用 .scan 命令添加基线和检查点』。

.semget

```
.semget <semaphore_name>
```

当步骤发出此命令时, 系统将检查是否存在带有这个指定名称的信号。

- 如果不存在这样的信号, 系统将创建一个, 并将其分配给步骤的作业。然后将继续执行到下一步。
- 如果别的某个作业已经声明了这个信号名称, 作业将在 .semget 步骤挂起, 直到其他项目释放了此信号。

有关使用此命令的更多信息, 请参阅第 514 页的『信号量 (semaphore)』。

.semput

```
.semput <semaphore_name>
```

.semput 命令用于发出名为 <semaphore_name> 的信号。有关使用此命令的更多信息, 请参阅第 514 页的『信号量 (semaphore)』。

.set

```
.set env <EnvGroupName>[(SnapshotName)] "<VariableName>=<DesiredValue>" [...]
```

`.set` 命令对环境变量指定值。可以指定其他变量和值。用引号将每个变量及其值括起来。除非您指定了快照，否则该命令为环境组的缺省快照设置该变量。如果指定快照，请将其用括号括起来，在环境组名与快照名称之间不留空格。

注：必须已经存在此命令设置的变量。

使用 `.set` 命令可更改环境的主记录。系统运行项目时，将从主记录创建项目环境的副本，将这个副本存储在系统记录中，然后将该副本用作项目的缺省副本。

步骤运行时，它使用环境的作业副本而非主记录。因此，使用 `.set` 有下列影响：

- 在某个步骤中运行 `.set` 命令时，使用缺省步骤环境的后续步骤不会看到更改。系统对该步骤使用缺省环境作业副本。
- 当对指定环境运行 `.set` 命令时，指定该环境的后续步骤可以看到您所作的更改。如果步骤指定环境，系统将读取该环境的主记录。即使指定的步骤环境与项目缺省环境相同，也是如此。
- `.set` 命令所作的更改在作业完成之后继续有效。以后运行 `.set` 命令时将使用之前运行此命令时所创建的值。

有关使用此命令的更多信息，请参阅第 327 页的『使用作业数据』。另请参阅类似的命令 第 335 页的『`.bset`』。

.sleep

`.sleep <seconds>`

`.sleep` 命令用于指定步骤暂停的秒数。因为由管理控制台处理此命令，所以不会创建到远程服务器的连接。也可以将“`.sleep 0`”命令用作独立于平台的空命令。

.snapshot

`.snapshot` 命令用于新建主调项目的实例并将实例以项目快照的形式存储在数据库中。项目快照是可执行项目。

描述 `.snapshot` 命令对您选择来创建快照或进行复制的项目及其关联对象创建快照。使用 `.snapshot` 选项可指定要创建快照或进行复制的对象，如下表所述。如果未指定任何选项，那么只会包括项目定义、步骤和标记。
快照名称是必需的，且对项目快照集合必须是唯一的。快照名称同时还指定给您要创建快照的其他对象。

运行 `.snapshot` 的项目完成后，项目快照将作为主调项目的子项目显示在 UI 中。其他快照对象则作为其基本快照或父级对象的子代显示在 UI 中。

语法

`.snapshot -v <"snapshot_name"> [-c <"comment">] [-e[f]] [-s[f]] [-pI] [-pC] [-a] [-t] [-r] [-g]`

选项	描述
-v <"snapshot_name">	项目快照名称是必需的。快照名称对项目必须是唯一的。而且名称必须用引号括起来。
-c <"comment">	将可选注释另存为快照的一部分。注释必须用引号括起来。

选项	描述
-e	创建项目快照时对项目和步骤环境创建快照。
-ef	如果添加 f 选项，那么还会对快照环境通过“包含”环境变量类型包含的任何环境创建快照。
-s	创建项目快照时对项目和步骤选择器创建快照。
-sf	如果添加 f 选项，那么还会对快照选择器通过“包含”选择器属性类型包含的任何选择器创建快照。
-pl	创建项目快照时对内联项目或库及其步骤创建快照。 内联项目或库由步骤触发，在步骤完成后运行。
-pC	创建项目快照时对链式项目或库及其步骤创建快照。 项目或库可以由项目通过/失败条件或步骤通过/失败条件触发。将为两种类型的条件链式项目或库创建快照。
-a	创建项目快照时复制适配器链接。 适配器链接将适配器添加到项目并运行适配器代码。
-t	针对在项目级别及步骤级别设定的通过/失败通知事件，复制通知模板。
-r	创建项目快照时复制项目寄存器。
-g	复制项目标记变量的标记值。 标记变量将自动复制，但变量值不会自动复制。如果您不复制标记值，它们将重置为 1。

.source

描述 .source 命令用于为源代码应用程序向项目步骤添加适配器。源代码适配器是一种 Build Forge 对象，它基于源代码应用程序的适配器模板。项目运行时，将运行此步骤的适配器代码。

语法

```
.source <adaptor_name> [entry_name]
```

<adaptor_name> 是必需的，它是在管理控制台中分配给适配器的名称。
<adaptor_name> 的大小写应与控制台中使用的大小写匹配。

如果适配器模板具有多个接口函数，请使用 *entry_name* 选项指定一个要运行的函数。*entry_name* 选项必须与为适配器模板中 <interface> 元素指定的 name 属性匹配。如果 *entry_name* 中指定的 <interface> 元素不存在或无法找到，那么将改为运行缺省的 <interface> 元素。在以下示例中，条目名称是 By Date。

如果在使用适配器链接，将自动调用适配器，并运行适配器模板中的第一个接口函数。要在适配器模板中运行其他接口，请将要运行的接口的 default 属性设置为 true (default="true")。

示例

```
.source MyClearCaseAdaptor  
.source MyClearCaseAdaptor "By Date"
```

注释 要创建适配器或查看适配器列表，请选择项目 > 适配器。

Build Forge 产品随附的适配器模板位于:

`<bfinstall>/interface`

.stop

`.stop <构建的状态>`

`.stop` 强制构建立即停止处理。使用此命令可以终止构建。构建的可能状态包括: 通过 (P)、失败 (F)、警告 (W) 和中断 (B)。

.strsub

`.strsub <source> <replacement> file [file ...]`

`.strsub` 命令用于在一个或多个文本文件中执行基本的字符串替换。系统扫描目标文件以查找 `<source>` 字符串; 如果发现匹配项, 系统会将 `<source>` 字符串替换为 `<replacement>`。`.strsub` 命令则可替换各文件中每行上的字符串 (`source`) 的每个实例。

`.strsub` 命令在所有操作系统上都可以运行, 并不依赖于服务器上可以使用的任何特定命令。

要替换文件 `about.c` 中的字符串 `_VERSION_`, 请使用类似以下内容的命令:

`.strsub _VERSION_ 2.34 about.c`

必须严格指定一个或多个文件名, 不得使用通配符。例如, 如下命令将失败:

`.strsub _VERSION_ 2.34 *.txt`

但是, 可在此命令中使用变量, 因此, 如果环境中定义了 `VERSION` 和 `FILENAME` 变量, 那么可运行如下命令。

`.strsub _VERSION_ ${VERSION} ${FILENAME}`

注: 请在命令中使用空格来分隔参数。

`.strsub` 命令类似于 `.edit` 命令, 两者的区别是:

- 在大文本文件或多个文件中执行替换时, `.strsub` 命令的速度比 `.edit` 快。
- `.edit` 命令可以执行正则表达式搜索和替换。
- `.edit` 命令仅替换各文件中每行上的字符串 (`search_expression`) 的第一个实例。
- `.strsub` 命令则可替换各文件中每行上的字符串 (`source`) 的每个实例。

`.edit` 命令缺省情况下使用 POSIX 扩展正则表达式语法。如果通过 Perl 兼容的正则表达式支持编译了代理程序, 那么替换表达式会后跟“p”字符 (表示应改用 PCRE 语法)。

在这两种情况下, 表达式通过代理程序处理解释两次。因此, 在通常使用一个反斜杠的地方都应使用四个反斜杠。例如:

Four slashes escape a literal period:
\\ \\.

或者, 也可以使用 `/x` 标志而不使用反斜杠:

`.strsub/x`

.test

描述 .test 命令用于为测试应用程序向项目步骤添加适配器。测试适配器是一种 Build Forge 对象，它基于测试应用程序的适配器模板。项目运行时，将运行此步骤的适配器代码。

语法

```
.test <adaptor_name> [entry_name]
```

<adaptor_name> 是必需的，它是在管理控制台中分配给适配器的名称。
<adaptor_name> 的大小写应与控制台中使用的大小写匹配。

如果适配器模板具有多个接口函数，请使用 *entry_name* 选项指定一个想要运行的函数。*entry_name* 必须与为适配器模板中 <interface> 元素指定的 name 属性匹配。如果 *entry_name* 中指定的 <interface> 元素不存在或无法找到，那么将改为运行缺省的 interface 元素。在以下示例中，条目名称为 TestFunction。

如果在使用适配器链接，将自动调用适配器，并运行适配器模板中的第一个接口函数。要在适配器模板中运行其他接口，请将要运行的接口的 default 属性设置为 true (default="true")。

示例

```
.test MyTestAdaptor  
.test MyTestAdaptor TestFunction
```

注释 要创建适配器或查看适配器列表，请选择项目 > 适配器。

Build Forge 产品随附的适配器模板位于：

```
<bfinstall>/interface
```

.tset

```
.tset env "<VariableName>=<DesiredValue>" [...]
```

.tset 命令用于在步骤中临时更改项目设置。可以使用 .tset 设置尚未存在的变量。

命令在当前步骤中生效。它对于步骤中的所有命令以及步骤中指定的任何内联有效。但是，线程技术可以影响此行为。示例：

1. 步骤是线程化的且指定为内联。
2. 内联的首个步骤也是线程化的。

在本例中，.tset 命令对第一步（1）生效，而不是对内联（2）的第一步生效。两个步骤均是同一线程块的一部分。它们独立运行。请参阅第 320 页的『线程化：并行运行步骤』。

.tset 命令设置的值将写入作业记录中。这些值不会更新环境的数据库记录。后面的作业不会受到这些更改的影响。

.unlock

```
.unlock
```

.unlock 使系统释放使用 .lock 命令锁定的作业。解锁的运行当前在通过作业 > 已完成打开的页面上列出，并且已从“已锁定”页面上的列表中除去。

第 21 章 使用作业

本主题描述如何在管理控制台中运行、查看和管理作业。

关于作业

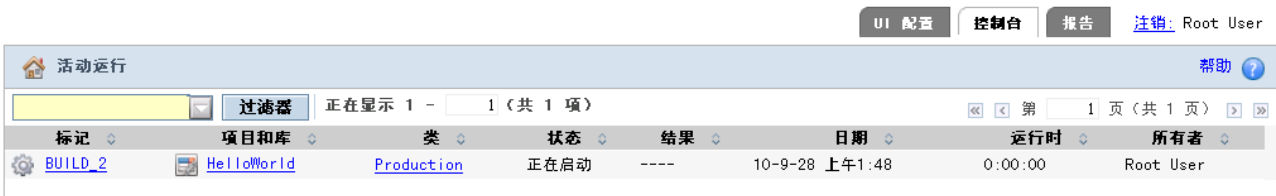
作业是正在执行或已经执行完毕的项目。

可以使用主页面板和作业面板来监视作业。此外，您还可以使用作业面板来启动作业、查看作业结果以及管理信号量。

关于主面板

主页面板提供有关最近作业的信息和系统消息。

要访问主页面板，请单击左侧菜单中的主。



使用主页面板可查看最近作业或当前作业。还可使用它来查看通知和系统消息。选择菜单项可查看以下内容：

- 活动运行 - 当前正在执行的作业
- 已完成的运行 - 已完成的作业
- 系统消息 - 系统消息日志（然而要查看消息，请使用管理 > 消息。）

关于“作业”面板

使用作业面板可处理正在运行的作业和查看作业结果。

要访问作业面板，请单击左侧菜单中的作业。



作业选项卡

可以使用“作业”选项卡以按状态查看作业列表：全部、已完成、正在运行、已归档或已锁定。

单击某个选项卡可按类型查看作业。每个选项卡用于在列表中管理作业选项都不同。

全部 先显示所有正在运行的作业（按开始时间排序）。接着显示所有已完成的作业（按完成时间排序）。复选框变灰。您可以单击任一链接。不提供任何特别的操作。

已完成 显示所有已完成的作业（按开始时间排序）。不显示正在运行、已归档和已锁定的作业。您可以清除或锁定已完成的作业。选中所需作业的复选框，然后单击**清除或锁定**。清除已完成的作业后，如果不是完全删除，作业将移至**已归档**中。

正在运行

显示所有正在运行的作业（按开始时间排序）。您可以取消一个或多个正在运行的作业。选中所需作业的复选框，然后单击**取消**。

已归档 显示所有已归档的作业（按开始时间排序）。已归档的作业是删除了部分信息的作业，该删除通常是通过作业类的类定义中指定的自动清除完成的。您可以清除已归档的作业。选中所需作业的复选框，然后单击**清除**。

已锁定 显示所有已锁定的作业（按开始时间排序）。您可以对一个或多个锁定的作业解除锁定。选中所需作业的复选框，然后单击**解锁**。

注：您不能直接删除已锁定的作业。要删除已锁定的作业，请先对其解锁，再转至**已完成**选项卡，然后使用**清除**。

作业列表字段

字段针对每个作业显示信息。

按完成时间显示任何选项卡中的作业列表，最近的作业列在最前。在列中为每个作业显示以下信息：

标记 构建标记，显示为链接。单击可查看作业步骤的日志。

项目和库

显示为链接。单击可查看作业所用的项目。

类 显示为链接。单击可查看作业所用的类的类定义。

状态 正在等待、正在运行、已完成、已归档或已锁定之一。

- “正在等待”状态表示作业尚未启动，原因是“正在运行”队列已满或作业正在等待信号。“正在运行”队列大小是通过系统设置**运行队列大小**定义的。
- “正在运行”队列中的作业通常处于正在执行状态。但是，如果“正在运行”队列包含同一项目的多个作业，那么其中当前正在执行一个作业还是多个作业取决于项目中的**运行限制**属性。请参阅第 282 页的『更改项目属性』。

结果 通过、失败、失败但继续、警告、已停止或已取消之一。

日期 启动作业前：请求作业的日期和时间。

启动作业后：启动作业的日期和时间。

运行时 对于正在运行的作业，为当前耗用时间。对于已完成的作业，为总耗用时间。

所有者 启用作业的用户。

对作业列表进行过滤和排序

您可以过滤和排序任何选项卡中的作业列表。

过滤使作业列表仅显示与您指定的字符串匹配的作业。排序按任意列中的值对列表进行排序。

- **过滤器**：您可以过滤作业列表以查看包含指定字符串的作业。如果列中存在字符串，那么与过滤器匹配。
 - **新过滤器**：在框中输入字符串，然后单击**过滤器**。
 - **现有过滤器**：单击框旁边的箭头，然后选择要使用的过滤字符串。
- **排序**：您可以按任何一个列的值对作业列表进行排序。单击列名旁边的箭头。如果单击多次，排序按照以下排序顺序循环。
 1. 升序（列标题中突出显示向上箭头）
 2. 降序（列标题中突出显示向下箭头）
 3. 无过滤器

运行作业和查看结果

您可以运行、调度、取消和重新启动作业并查看作业结果。

启动作业

有多种方式可以启动作业。

开始之前

如果项目要作为作业运行，那么它必须定义环境和选择器。

过程

- 从“作业”面板：选择**作业 > 启动**并单击项目名称。与单击**启动项目**时一样，此方法也会显示“启动项目”页面。

作业详细信息

作业步骤

项目参数

运行链接:

快照:

选择器:

类:

标记格式:

标记示例:

基础快照

Selector1

Production

BUILD_\$B

BUILD_1

项目环境

保存环境


作业将使用缺省环境启动。

项目标记

可编辑的标记


B

1

- 从“项目”面板: 单击快速启动图标 。


将检查以下条件。如果通过检查，那么项目立即启动。

- 该项目包含一个或多个步骤。
- 该项目的环境中没有变量在“在项目中”变量属性中设置为“必须更改”。

如果检查失败，那么将禁用快速启动图标: 。

通过该方式运行项目将使用它的选择器、类、标记和环境变量的缺省值。

如果“启用快速启动”系统设置设置为“是”，那么“项目”页面将检查所有项目以确定它们是否应该快速启动。如果定义了大量项目，那么“项目”页面可能需要很长时间来

显示项目列表。指示项目可以启动的另一个图标: 。这是 V7.1.1.1 之前版本的缺省行为。

- 从“步骤”面板: 查看项目的步骤时，单击启动项目。该方法将显示项目的“启动项目”页面，在其中您可以更改项目参数、环境变量值和选择要从运行中排除的步骤:
 - 选择项目参数的新值。
 - 编辑项目标记变量值。
 - 编辑项目环境变量值。如果希望将更改保存为这些变量的新缺省值，请单击保存环境复选框。
 - 选择“作业步骤”选项卡以显示项目步骤的列表。您可以选择个别的步骤，仅将它们从该运行中排除。

进行选择后，单击执行以启动项目。

结果

项目运行时，查看作业 > 正在运行页面以检查项目状态。

要查看作业结果，请选择作业 > 已完成以显示已完成的作业。单击“标记名称”可访问用于查看作业结果的选项。

查看作业结果

关于此任务

您可以在各个作业选项卡中查看作业结果。

过程

1. 单击**作业**。 **已完成**选项卡随即选中。
2. 单击所需的选项卡。
3. 单击您要查看的作业的标记。 作业结果最初显示如下：
 - 主面板中显示步骤列表。每个步骤有若干个信息列。
 - 步骤：步骤编号
 - 步骤名称：该步骤的名称。步骤名称前面的图指示步骤类型：步骤启动、已线程化、广播和连接。单击步骤名称可查看步骤日志。
 - 结果：步骤结果（通过、失败和已取消）
 - 服务器（选择器）：运行步骤的服务器名称，加上选择服务器的选择器。
 - 运行时：步骤的耗用时间（时:分:秒）
 - 链：如果步骤启动了另一个作业，那么显示对该作业的链接。
 - 在左侧菜单底部显示一个菜单，其中包含以下项：
 - 结果：在主面板中显示步骤列表。
 - 材料清单：在主窗格中显示材料清单（BOM）。BOM 包含显示作业步骤、步骤清单和检查点（如果使用检查点）的链接。
 - 说明：显示为此作业输入的说明。
 - 步骤日志（此项打开，并显示步骤名称）：显示步骤列表。单击某个步骤可查看步骤日志。
 - 链：显示作为链启动的所有作业的作业号（使用 `.run` 和/或 `.runwait` 启动的任何作业都不会生成此图标）。
4. 要查看步骤日志：从任意步骤列表单击某个步骤名称。最初，所有类别均处于选中状态。

如果是在查看“正在运行”作业中的步骤，并且该步骤尚未完成，您将看到部分日志，并且可能看不到所有类别。要更新正在运行的步骤的日志视图，请单击构建标记（在顶部，显示为**作业 >>** 标记），然后单击步骤。查看步骤日志时，单击**全部显示**以显示所有步骤。要查看更少步骤，请在**显示第 1 - m 步（共 n 步）** 控件中输入 m 的值并按 **Enter**。

5. 要过滤步骤日志：选中或取消选中类别，然后单击**刷新**。

```
1 7/8/10 4:43 PM STEP      Step using selector 'My selector'.
2 7/8/10 4:43 PM MANIFEST   BF_LAST_UPDATE=1278618783
3 7/8/10 4:43 PM MANIFEST   BF_NAME=rbf-14
4 7/8/10 4:43 PM MANIFEST   BF_LOADRATIO=0.3333333333333333
5 7/8/10 4:43 PM MANIFEST   BF_JOBS=1
6 7/8/10 4:43 PM MANIFEST   BF_AGENT_VERSION=version
7 7/8/10 4:43 PM MANIFEST   BF_LAST_REFRESH=1278618783
```

作业状态的 RSS 数据订阅源

可以使用 RSS 数据订阅源跟踪和过滤各个作业的状态。Build Forge 的作业 RSS 数据订阅源显示的信息与 Build Forge 管理控制台上显示的服务器状态信息相同。

要预订 RSS 数据订阅源以获取作业状态，请执行以下操作：

1. 在 Build Forge 管理控制台中，选择**作业**。

Web 浏览器将检测 RSS 订阅源并在浏览器地址栏中显示 RSS 图标。

2. 在 RSS 聚合器工具中，装入 Build Forge RSS 数据订阅源。

例如，复制 URL 以将其添加到 RSS 数据订阅源列表中，或者拖放 RSS 图标以将 URL 添加到 RSS 数据订阅源列表。

3. 预订 RSS 数据订阅源以保存 URL 并在发生更新时获得通知。

注：

- 有关装入 URL 和预订 RSS 数据订阅源的信息，请参阅针对您 RSS 聚集器工具的文档。
- 若要使用非英语语言通过 RSS 数据订阅源查看 Build Forge 的作业状态、系统消息或服务器状态，RSS 聚合器工具必须支持 UTF-8 多字节字符编码。
- RSS 订阅源使用时间戳记报告作业活动。该时间戳记基于针对您的 Build Forge 用户帐户设置的时区。如果浏览器位于另一个时区，那么可将其设置为与您的 Build Forge 帐户相同的时区以获取准确的时间戳记。

重新启动失败的作业

如果作业失败，可以重新启动作业。重新启动将在同一个标记下启动新运行。它从失败点继续。

要重新启动作业，请单击构建列表中的作业标记（在**作业 > 已完成**选项卡上）。系统显示有关构建的信息，并在面板顶部附近包含一个**重新启动作业**按钮。

1. 单击**重新启动作业**；系统显示“重新启动”页面。
2. 选择相应选项。如果希望系统在其步骤中从项目记录获取对命令的任何更新，请设置**同步命令**属性；如果不设置，那么命令将按照它们在作业最初启动时的原样运行。
3. 单击“重新启动”页的**重新启动**按钮。

重新启动的作业在以下几个方面不同于新作业：

- 它使用的标记的数量与失败的运行相同，并在“已完成”列表中替代失败的运行。
- 缺省情况下，它从失败的步骤开始，并且不重复在前次运行中已通过的任何步骤。但是，您在重新启动作业时，可以选择实际运行哪些步骤。
- 缺省情况下，系统将提供您在前次运行时所提供的相同的环境变量值；但是，在重新启动作业之前，您可以更改这些变量值。
- 系统仅根据在重新启动的作业期间所运行的步骤来评估作业是否成功。前次运行中的失败不会影响重新启动的作业的状态。

有关重新启动具有适配器链接步骤的作业的信息，请参阅第 429 页的『关于适配器链接』。

重新启动广播步骤

在运行时，标记为广播步骤的步骤显示为要运行该步骤的每个服务器的步骤。

如果广播步骤失败，那么最多所有步骤都会显示为失败。重新启动时请执行以下步骤：

1. 仅选择一个失败的步骤。
2. 设置**同步命令**属性。
3. 设置任何其他期望的选项。
4. 单击“页面”页面上的**重新启动**。

重新启动 While 循环

如果步骤的类型为 While 循环，系统将根据 BF_ITERATION 系统变量的值，在步骤失败时的迭代位置重新启动步骤。作业流程示例：

1. 作业进入 While 循环类型步骤
2. 条件求值结果为 true
3. BF_ITERATION 设置为 1
4. 命令和内联运行成功
5. 步骤循环
6. 条件求值结果为 true
7. BF_ITERATION 设置为 2
8. 执行命令或内联过程中作业停止

如果上述作业重新启动，那么从迭代 2 处重新开始。它试图为该步骤运行命令和内联。

使用材料清单

系统在每个作业之后会生成一个材料清单（BOM）。BOM 包含有关作业中的步骤以及从中产生的文件更改的信息。BOM 可提供给作业的使用者（如质量保证部门），以帮助理解新作业的内容。它可作为构建和发布过程的审计解决方案。通过 BOM，您可获得作业内容的完整文档。它可包含结果、说明、环境、文件列表以及代码更改。您可使用该信息来比较和汇总整个企业中构建的状态。

系统自动为每一个作业生成 BOM，但是您可以在步骤中使用点命令，使系统存储有关文件在构建前后的状态的其他信息。

显示材料清单（BOM）

当您查看已完成的构建时（作业 > 已完成），系统在缺省情况下将显示**步骤**选项卡。单击 **BOM** 选项卡来显示材料清单。

单击任一类别旁的 + 来扩展它。您所看见的实际类别取决于项目以及系统的配置方式：

- “项目步骤”类别出现在每个作业中，并显示有关此作业所运行步骤的信息。
- “源更改”类别仅在系统包含源代码适配器且项目与适配器之间存在链接时显示。请参阅第 429 页的『适配器与作业结果』以了解详细信息。配置适配器时，可以更改“源更改”类别的格式，甚至名称。
- 基线和检查点部分仅在项目包括了 .scan 命令时显示。

使用 .scan 命令添加基线和检查点

可以使用 .scan 命令向 BOM 添加更多信息。在运行 .scan 命令时，系统将有关文件状态的信息存储在步骤的工作目录中。本部分显示其使用示例。另请参阅第 350 页的『.scan』的参考信息。

此命令有两种格式。

.scan baseline

将所有文件的列表存储在步骤的工作目录树中，每个文件都带有 MD5 值。系统在作业的 BOM 中显示列表。您可能希望在执行某些设置步骤和检出相应的文件集之后发出此命令。可以在一个项目中使用多个 baseline 命令，但是 .baseline 命令执行时，每个命令都会将列表重置为步骤的工作目录状态。

.scan checkpoint

存储自上次在项目中执行 .scan baseline 或 .scan checkpoint 以来的所有新建、已更改和已删除文件的列表，每个文件带有 MD5 值。使用 .scan baseline 命令时，系统将在 BOM 中显示此列表。必须在项目的第一个 .scan checkpoint 之前发出一个 .scan baseline 命令。在 .scan baseline 命令之前的 .scan checkpoint 命令会被忽略。

以下示例显示 .scan baseline 与 checkpoint 命令如何共同运行：

序号	步骤	步骤之后的文件	BOM 数据
1	检出初始文件	config.c execute.c	
2	.scan baseline	config.c execute.c	基线: config.c execute.c
3	添加数据文件	config.c execute.c data.txt	
4	.scan checkpoint	config.c execute.c data.txt	检查点 1: 添加的 data.txt
5	添加更多数据文件	config.c execute.c data.txt data2.txt data3.txt	
6	删除 data.txt	config.c execute.c data2.txt data3.txt	
7	.scan checkpoint	config.c execute.c data2.txt data3.txt	检查点 2: 添加的 data2.txt、 data3.txt 删除的 data.txt

将 BOM 导出为 XML 文件

本主题描述了 bfbomexport 命令的相关语法、用途和选项描述。

Build Forge 命令位于 Build Forge 安装目录（在 Windows 上）和 *<bfinstall>/Platform* 目录（在 UNIX 和 Linux 上）。

描述

使用 `bfbomexport` 命令将作业的材料清单（BOM）导出至 XML 文件。收集 BOM 信息后，`bfbomexport` 将其保存至您指定的位置和文件名。

要指定要将哪个 BOM 保存至 XML 文件，必须指定项目和作业的构建号。

必须从管理控制台的 Build Forge 安装目录以及从 UNIX 或 Linux 上的 */Platform* 目录运行 `bfbomexport` 命令。

语法

```
bfbomexport [-f filename] [-p projectID | -P projectName]
            [-b buildID | -t buildTag] [-L] [-H]
```

选项

选项	描述
-f filename	可选路径和/或文件名。作业的 BOM 以 XML 格式保存。若未提供文件名，那么 BOM 将写入标准输出（stdout）。若未提供路径名，那么使用当前工作目录。
-p projectID	作业的项目标识。（带有 -l 选项的 <code>bfexport</code> 命令可列出项目标识。）
-P projectName	项目的名称。
-b buildID	构建标识。
-t buildTag	构建标记名称。
-L	包括步骤日志。
-H	帮助消息。

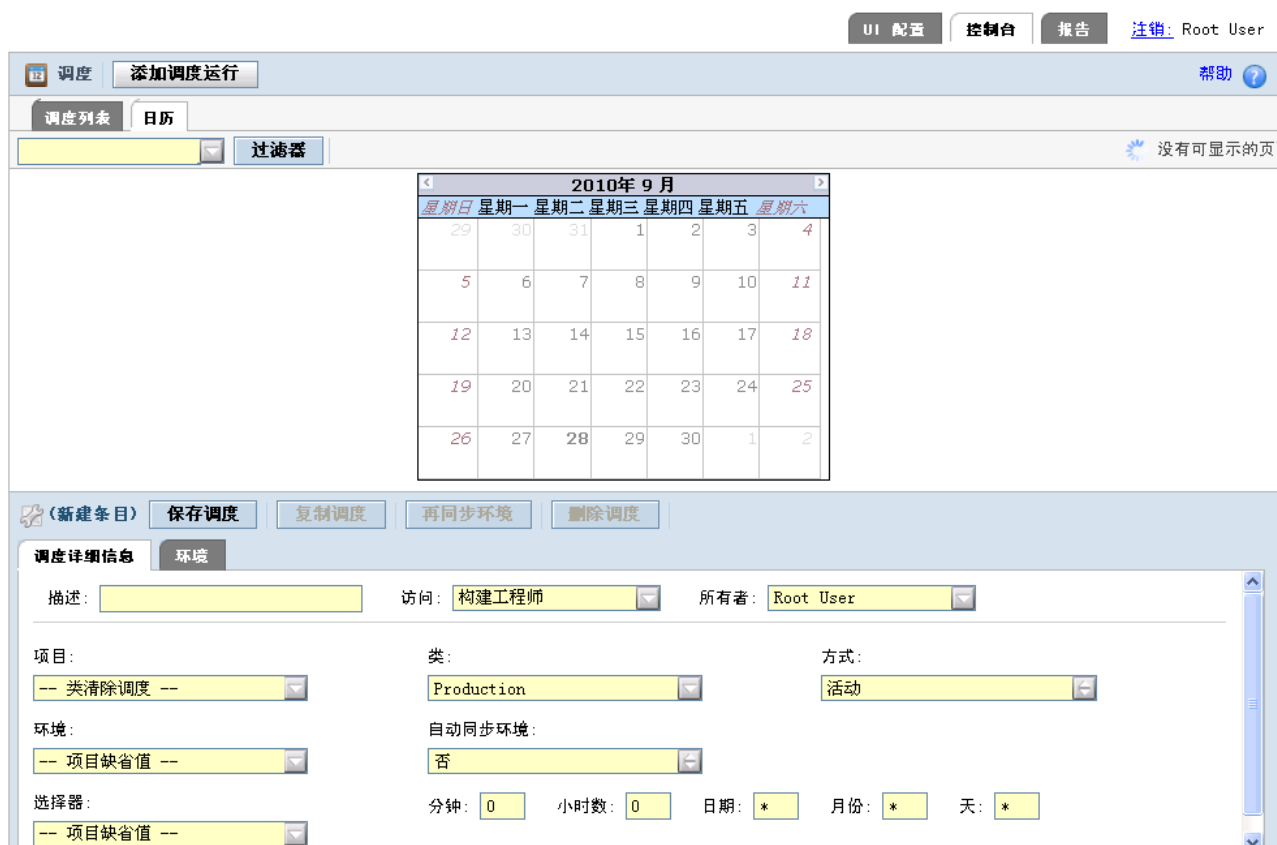
调度作业

使用**调度**面板可调度项目在以后的某个时间运行，或者以重复的时间间隔定期运行。例如，可将项目设置为每小时或每天运行。

要查看作业调度，请单击**调度**。此时将显示当前月份的日历，日历下方有一个面板，用于创建和修改调度条目。

日历中的每个日期都显示为当天安排的项目数。将鼠标光标放在日期上方可以看到为当天调度的作业的名称和调度参数。

如果调度了多个项目，系统将显示一个下拉框和**过滤**按钮，以便于您按项目过滤日历。



要将项目放入调度程序中，请执行以下操作：

1. 单击左侧菜单中的**调度**。
2. 单击**添加调度运行**。
3. 在**描述**中为调度条目输入相应描述。
4. 从项目列表中选择项目。
5. 在**项目**下的字段中显示项目的当前快照。如果希望为已调度作业运行其他快照，请从列表选择快照。
6. 对于**方式**，选择“活动”（缺省值）、“非活动”或“一次”。如果设置为“一次”，那么作业只在下次时间设置匹配时运行一次。如果设置为“活动”，那么作业在每次时间设置匹配时都将运行。
7. 可选：您可以覆盖以下项目属性。您在此处选择的设置只影响已调度作业的该实例。
 - **访问**：如果已指定，已调度作业将使用指定的“访问”属性。
 - **所有者**：如果已指定，已调度作业的运行相当于是由指定的所有者手动启动的。
 - **类**：如果已指定，已调度作业使用指定的类。
 - **环境**：在该字段中，已为此调度的作业生成了环境的副本（无论其显式设置还是保留在“项目缺省值”）。如果指定了环境，那么还可以设置变量的开始值。

要点：生成副本后，未在为已调度作业生成的环境副本中自动更新对原始环境及其变量的更改。可以手动或自动更新副本以反映对原始环境所作的更改。对于手动更新，请单击**再同步环境**。对于自动更新，请将**自动同步环境**设

置为“是”。使用自动更新，将在每次运行已调度的作业时更新环境。但请注意，“环境”选项卡仅可用于将**自动同步环境**设置为“否”的调度。

- **选择器**：如果已指定，作业将使用指定的选择器。

8. 指定时间设置。填写“分钟”、“小时”、“日期”、“月份”和“天数”的值。请参阅『调度参数』了解要使用的值。
9. 单击**保存调度**。

“下次运行”列将显示“正在计算”，然后显示安排该作业下次运行的时间。

创建日程表后，日程表名称将显示在日历中。其旁边的图标将显示其“方式”。您可以通过单击该图标来更改方式。每次单击后，方式都将更改。

- 绿色圆圈：活动
- 蓝色圆圈：一次
- 红色圆圈：非活动

已调度作业尝试运行时，系统检查作业队列、**硬运行限制**系统设置，还可能检查项目的**运行限制**属性。如果调度激活时项目的一个或多个作业已经在运行，那么系统行为取决于**硬运行限制**系统设置（如下表中所解释）。

硬运行限制值	系统检查运行限制值？	系统启动作业？
是	是	如果运行作业数小于 运行限制 值，那么为“是” 如果运行作业数等于 运行限制 值，那么为“否”
否	否	是

注：如果先前的运行尚未完成，那么将**运行限制**值设置为 1 并将**硬运行限制**设置为“是”以跳过运行。

调度参数

本部分描述可用于指定何时运行作业的参数。

使用一系列字段指定运行作业的时间。

字段	描述	范围
分钟	分钟数。	0-59
小时	一天中的某小时。	0-23
日期（一月中的某一天）	某月中的某一天。	1-31
月份	一年中的某月。	1-12
天（星期几）	星期几（星期天 = 0）。	0-6

在字段中输入的数字值可以表示如下：

- 使用星号（*）表示范围内的所有有效值。星号可以后跟正斜杠（/）和步长值。例如，“小时”字段中的 *2 值表示每 2 小时运行一次作业。
- 注：**星号在“日期”和“星期几”字段中有特殊的含义。它符合在 UNIX/Linux cron 设施中的含义。如果一个字段有字面值，而另一个字段有星号，那么仅使用字面值的频率。例如，如果“月份”为 *，“日期”为 *，“星期几”为 1，那么每个星期一执行作业。另请参阅下面的“星期几/日期”示例。
- 使用由连字符分隔的数字范围。例如，“小时”字段中的 8-11 即指定在 8、9、10 和 11 点执行。范围可以后跟正斜杠（/）和步长值。例如，“小时”字段中的 0-23/2 表示每隔一小时运行一次作业。
- 使用由逗号分隔一组数字（或范围）的逗号分隔列表。例如，1, 2, 3-5, 8。

使用字段中指定的值构造调度。示例:

值					
所要的日程	分钟	小时	日期	月份	天
每天下午 5 点运行作业。	0	17	*	*	*
每周星期一的下午 4:30 运行作业。	30	16	*	*	1
每个工作日（周末除外）每半小时运行一次作业	*/30	*	*	*	1-5
每隔一天的中午 12:30 运行作业	30	0	*/2	*	*

“星期几”和“日期”字段使用星号的方式很特殊。

- 如果一个字段有星号值，而另一个字段有字面值，那么根据含有字面值设置的字段运行作业。
- 如果两个字段都含非星号值，那么在任一条件发生时运行作业。
- 如果两个字段都含有星号值，那么每天运行作业。

星期几/日期示例:

值					
所要的日程	分钟	小时	日期	月份	天
每月第一天凌晨 1:01 运行作业。“日期”使用字面值；“星期几”使用星号。	1	1	1	*	*

值					
所要的日程	分钟	小时	日期	月份	天
该月的每个星期一凌晨 1:01 运行作业。“日期”使用星号；“星期几”使用字面值。	1	1	*	*	1
该月的每个星期一和该月的第一天（无论该日是否为星期一）凌晨 1:01 运行作业。“日期”和“星期几”都使用字面值。	1	1	1	*	1
每天凌晨 1:01 运行作业。“日期”和“星期几”都使用星号值。	1	1	*	*	*

调度作业类清除

通过创建作业类的日程表，可控制系统何时进行旧作业的清除。创建这些日程表就像创建启动项目的日程表一样，但是您要选择“类清除日程表”选项，而不是选择项目。执行此操作时，系统将进行检查，以查找属于您在日程表中所选择的类的要清除的作业。对于符合条件的每一个作业，系统将创建一个清除作业，并将该作业放入等待队列。

缺省情况下，系统将按照由“清除检查时间”系统设置设定的时间间隔（缺省为每 15 分钟）来检查应清除的作业（基于定义自动删除规则的类型属性）。这一行为可能导致与常规作业争用系统资源。

如果创建了针对某一类作业的日程表，那么系统在该日程表激活时将仅检查要清除的作业。如果对于某个特定类没有时间表存在，那么系统将使用该类的作业的缺省行为。如果要将所有清除限制在某些特定时间，那么必须为每个类创建至少一个日程表。

要定义清除日程表：

1. 照常创建一个日程表，但在“项目”字段中选择“类清除日程表”。
2. 在“类”字段中选择一个类。

管理作业

您可以锁定、归档和删除作业。

锁定作业

您可以锁定项目，以防止它被自动删除。

要锁定作业，请单击**作业**面板中的**已完成**选项卡，并选择一个构建，然后单击**锁定**。

要解锁作业，请单击**作业**面板中的**已锁定**选项卡，并选择一个锁定的构建，然后单击**解锁**。

系统不会自动清除已锁定的作业（如运行的类清除属性要求删除作业时）。但您可以手动删除已锁定的作业。

单击**已锁定**选项卡可查看处于已锁定状态的作业。此类项目不会显示在**已完成**页面中。

删除作业

以下各个主题描述删除作业的若干种方法。

从已完成选项卡删除作业

您可以从**作业**面板的“**已完成**”选项卡手动删除作业。

可以从该列表手动删除一个或多个作业。当您执行此操作时，系统将根据每次运行的类启动清除作业。过程与类属性触发的自动删除相同。系统对作业进行归档（如果需要完全删除作业）并删除该类作业的指定数据。请参阅第 295 页的『类』。

要删除作业：

1. 选择**作业** > **已完成**页面来显示已完成的运行列表。
2. 单击表右端的一个或多个复选框以选择要删除的构建。
3. 从列表底部的列表框，选择“清除”选项。
4. 单击“执行”按钮。

如果类设置为删除输出文件但保留控制台数据，那么从**已完成**列表中删除运行也将删除输出文件，并将作业的条目从**已完成**列表移至**已归档**列表。

从归档列表完全删除作业

选择**作业** > **已归档**来显示归档列表。该列表显示有关其文件已被删除的作业的数据。可以在此处删除作业，就像从**作业** > **已完成**列表删除一样。从归档列表删除作业将从数据库除去该作业的所有跟踪，并从应用程序报告的统计数据中除去它。

自动删除作业

当作业的类型属性确定应删除某个作业时，系统将自动删除该作业。可以使用此功能来防止数据累积和删除作业组。

如果创建了类的调度，系统将在调度激活时仅检查要清除的作业。请参阅第 369 页的『调度作业类清除』。

要确保删除希望删除的作业，请在启动或调度作业时检查下列设置：

1. 设置一个或多个类的删除属性可允许系统在累计到达特定天数和/或特定作业数之后删除这些作业。
2. 将正在处理的项目的类属性设置为相应的类。

如果生成了一些作业，然后希望将其删除，那么可以临时更改相关类的删除属性。也可以在**已完成**选项卡上选择多个构建，然后将其删除（单击各作业旁边的复选框以将其选中，然后单击**清除**按钮）。

例如，如果前一天生成大量的 Production 作业，那么可使用以下过程将其删除：

1. 记下 Production 类的当前设置。
2. 将 Production 类的“天数”属性更改为 1，然后单击**保存类**按钮。延迟 15 分钟之后，系统将开始删除时间超过一天的作业。
3. 作业删除之后，将 Production 类的属性改回原始设置。

作业的工作目录

系统会为每个作业构造工作目录，以便每个运行都具有一个经过标注且隔离的区域，在其中进行工作。它使用为服务器路径、项目名称和标记提供的值来生成工作目录名称。

在运行命令时，系统会在为步骤指定的目录中启动该命令。缺省情况下，该目录是作业的工作目录，但您也可以指定相对于服务器的**路径**属性的某个其他目录。本部分中的主题描述如何创建路径和目录。

注：在运行项目时，系统会构造项目目录（如果尚不存在）和作业目录。它不会构造服务器目录（在服务器的**路径**属性中指定）或步骤目录（步骤的 **Dir** 属性）。

作业的工作目录名称

以下示例显示了系统如何使用多个值来构造作业目录（在作业中发生在单个服务器上）：

系统值	创建的目录
服务器的路径字段: C:\BuildForge 项目名称: My Project 标记: Job 5	C:\BuildForge\My_Project\Job_5\ 系统仅创建路径的 粗体 部分。必须在运行项目前创建服务器目录，否则项目将运行失败。

注：当创建项目目录时，系统会将**无效相对 Dir** 字符系统设置中指定的字符更改为下划线字符。缺省情况下，设置包含空格和反引号字符，因此名为 My Project 的项目会收到名为 My_Project 的项目目录。

如果在多个服务器上运行作业，那么系统将在每个服务器上创建一个作业目录。由于项目中的每个步骤都能指定一个不同的服务器，因此系统可能会创建多个目录。以下示例描述了使用两台服务器的项目：

系统值	创建的目录
项目服务器：ServerA，其路径值为 C:\BuildForge 项目中的第三步指定了 ServerB，其路径值为 C:\deployments。（所有其他步骤使用缺省（项目）服务器，ServerA）。 项目名称：My Project 标记：Job_6	在 ServerA 上： C:\BuildForge\My_Project\Job_6\ 在 ServerB 上： C:\deployments\My_Project\Job_6\ 系统仅创建路径的 粗体 部分。

在上述示例中，可以认为缺省情况下要创建的步骤的所有输出文件都位于 C:\BuildForge\My_Project\Job_6 目录中，但第三步除外，它使用 C:\deployments\My_Project\Job_6 目录。

构造步骤的目录路径

当系统运行步骤时，它可以为作业构造的目录启动，或者如果步骤的**路径**选项设置为“绝对”，那么它可忽略项目和标记目录。

- 当步骤的**路径**设置为“绝对”时，系统通过同时添加**服务器路径**和步骤的**目录**字段来为步骤构造路径。**目录**字段中的值是服务器工作目录的相对路径。

步骤的值	命令的结果路径
服务器的 路径 字段：C:/BuildForge 步骤的 目录 字段：/bin 步骤的 路径 设置：绝对	C:\BuildForge\bin 使用此表单来访问位于服务器目录中的目录。
服务器的 路径 字段：C:/BuildForge 步骤的 目录 字段：/（缺省值） 步骤的 路径 设置：绝对	C:\BuildForge
服务器的 路径 字段：C:/BuildForge 步骤的 目录 字段：C:/temp 步骤的 路径 设置：绝对	C:\BuildForge\C:\temp （本示例将引发错误；步骤将失败。）

注：您可以输入带有反斜杠或正斜杠的路径值。系统将使用反斜杠存储它们，并在 Windows[®] 计算机上根据需要将其更改为正斜杠。

要点：将命令的路径设置为根目录可能导致无意中删除或修改系统文件。如果可能，请在不同目录中运行命令。但是，如果您正在自动执行系统管理任务并且要求路径为根目录，请将服务器的**路径**设置为根目录，将**目录**设置为 **../**，并将步骤的**路径**设置为“绝对”。

- 当步骤的**路径**选项设置为“相对”时，系统将通过同时添加**服务器路径**、**项目名称**、**标记**和步骤的“**目录**”字段来为步骤构造路径。**目录**字段中的值将成为作业工作目录的相对路径。

步骤的值	命令的结果路径
服务器的 路径 字段: C:/BuildForge 项目名称: My Project 标记: Job_5 步骤的 目录 字段: /bin 步骤的 路径 设置: 相对	C:\BuildForge\My_Project\Job_5\bin 路径的 粗体 部分由系统构造（若它们还不存在）。 注: 当创建项目目录时，系统会将 无效相对 Dir 字符系统设置中指定的字符更改为下划线字符。缺省情况下，设置包含空格和反引号字符，因此名为“My Project”的项目会收到名为“My_Project”的项目目录。

如果在服务器的**路径**字段中或步骤的**目录**字段中指定的目录不存在，那么步骤将失败；系统不会创建这些目录。步骤的“**目录**”字段指定的路径部分必须通过上一步骤在项目期间显式创建。

通常，项目中的早期步骤会从源代码控制检出一个目录树，以下步骤将对这些目录执行操作。

当您添加新的步骤时，系统会记住您最近一次就**路径**选择的设置（“相对”或“绝对”），并将该设置用作新步骤的缺省设置。

路径中的斜杠

当您在**路径**或**目录**字段中输入路径时，系统会将所有反斜杠转换为正斜杠。当系统创建实际路径时，如果使用的服务器是 Windows[®] 计算机，系统会将所有正斜杠转换为反斜杠。因此，您可以在路径中使用任何类型的斜杠。

注: 系统不会修改步骤的命令字段中的斜杠。请根据需要针对目标服务器使用反斜杠或正斜杠。

信号量

信号量是系统中设置互斥（mutex）资源的全局信号标志。使用信号量可使一部分进程等待其他进程完成。

使用**作业 > 信号量**来查看正在使用的作业信号量。也可以清除信号量，当挂起或取消作业未能释放其信号量时可能需要此操作。



通过下面的一对点命令可实现信号量，即：.semget 和 .semput 命令。使用 .semget 命令可“抓取”一个标签：在某个步骤获得一个标签后，尝试获取同一个标签的任何其他步骤（任何项目中）都必须等待，直到原始请求者释放此标签。

注：包含 .semget 命令的步骤将等待，直到信号量释放。如果作业失败并使其信号量处于活动状态，那么必须先手动清除信号量才能再次运行使用信号量的任何作业。

例如，假设有一个创建打印机驱动程序的程序，您希望该程序在任何时刻都只能由一个进程使用。在调用该程序的每一个项目内，设置带有以下命令行的三个步骤：

步骤	命令行
获取信号量	.semget \$BF_PROJECTNAME_PHYS
运行驱动程序创建器	printdrivermaker.exe windows
释放信号量	.semput \$BF_PROJECTNAME_PHYS

可为组织中的关键资源建立信号量，例如负载较重的服务器或者只有单用户许可证的软件程序。使用您要保护的资源的每一个步骤应包含在 .semget 和 .semput 命令之间。

信号量遵循以下规则：

- 使用时，信号量点命令必须是该步骤唯一的点命令。
- 系统原样使用标签；因此应避免在标签中使用特殊字符，或者结尾空格，因为这可能扰乱标签解析器。
- 信号量是全局的，除了同步线程锁之外，还可以用来同步不同的项目。
- 信号量是在首次访问它们时创建的。
- 项目获得的信号量在项目终止后自动释放。
- 如果两个步骤同时请求信号量，那么无法肯定是哪个步骤获得了锁。

手动清除信号量

系统会在项目终止时自动释放项目创建的所有信号量。但是，如果异常的项目终止导致某个信号量未被释放，您可以手动将它清除。

1. 选择**作业** > **信号量**来显示信号量列表。
2. 选择要清除的信号量。
3. 单击**清除**。

第 22 章 使用报告

此主题描述了随 Rational Build Forge 提供的内置报告。它还描述了如何使用“快速报告”（单独许可的报告工具）创建和运行报告。

关于报告


有两种报告类别：通过“性能”和“查询”面板访问的内置报告和使用“快速报告”工具创建的报告。

要开始操作，请单击右上角的**报告**选项卡以显示报告选项：

- 选择**性能**面板以查看标准报告。
- 选择**查询**面板以查看并运行预定义的查询报告。
- 如果想要创建自己的报告，请选择**快速报告**面板。快速报告是 Rational Build Forge 的许可选项。

“性能”和“查询”面板

“性能”面板使用 Build Forge 数据来创建多个标准报告。当您打开一个性能报告时，结果会自动更新，这样您总是能看到当前数据。

“性能”面板还有一个“查询”面板，其中包含预定义的查询报告。要运行这些报告，请提供必需数据，然后单击“快速启动” 图标。“查询”报告针对当前数据运行。只要重新运行报告即可看到基于最新数据的结果。

不能修改“性能”和“查询”面板中的报告。根据设计，您不能修改字段或更改报告显示。

“快速报告”工具

“快速报告”是基于 BIRT（基于 Eclipse 的开放式源代码报告系统）的报告工具。使用“快速报告”，您可以创建自己的报告设计。可以使用提供的报告类型作为起点，然后选择所需的报告字段。所有报告类型都使用 Rational Build 数据库中的数据作为其数据源。

对于您在“快速报告”中设计的任何报告，您可以从几种报告格式（比如表或图表）中进行选择。另外，可以使用分组和过滤功能来控制报告显示。

在报告输出中显示数据的先决条件

所有报告（包括样本报告）都会显示来自 Build Forge 数据库的数据。要填充该数据库，您必须首先创建项目并运行作业。

注：如果您在数据库包含数据之前创建并运行了报告，报告结果将为空。此外，您可能遇到报告错误。

显示报告中数据的访问许可权

正在运行报告的用户的访问许可权决定报告结果中包含和可见的项目数据。

用户的项目访问许可权决定报告结果中将包含的项目数据，如下所示：

- 对于“性能”面板中的标准报告，使用的许可权是打开报告页面的用户的许可权。
- 对于“查询”面板中的预定义报告，使用的许可权是运行报告的用户的许可权。
- 对于“快速报告”，使用的许可权是运行报告的用户的那些许可权，而不是报告设计者的许可权。

用户可使用其他访问许可权来修改报告设计和运行报告。请参阅第 383 页的『快速报告的报告组许可权』。

将内置报告结果导出到 CSV 文件

一些内置报告可将报告结果导出到 CSV 文件。如果您想要保存结果或将结果导入其他报告应用程序，请使用此功能。

对于包含“保存到 CSV”链接的所有内置报告，您都可以导出报告结果。

注：此功能不是为快速报告提供的。

要将报告结果导出到 CSV 文件格式：

1. 运行报告以生成结果：
 - 选择**分析报告**链接以打开步骤和服务器性能报告（**报告 > 性能**）。
 - 运行“查询”面板中的任何报告（**报告 > 查询**）。
2. 从报告结果页面，单击**保存到 CSV**。
3. 在“文件下载”对话框中，单击**打开或保存**：
 - 使用支持 CSV 文件格式的任意应用程序打开 CSV 文件。例如，Microsoft Excel 或您希望使用的基于文本的编辑器。
 - 将该文件保存到您的本地工作站或您网络中的其他位置。

“性能”的标准报告

当您打开标准报告页面时，标准报告显示基于当前数据的结果。

“性能”面板使用当前报告和作业数据来创建多个标准报告：

- 所有项目的作业性能报告
- 项目的作业持续时间报告
- 项目的步骤和服务器分析报告

查看项目的作业性能统计信息

“性能”面板类似一个作业性能报告，其中总结了所有项目的高级别作业统计信息。

- 对于上一个作业，该报告会显示完成日期时间和作业持续时间。
- 对于全部作业，统计信息包含通过、失败或通过但带有警告的作业数量。

数据源：Rational Build Forge 数据库

报告格式: 表

显示选项: 创建过滤器以控制列表中显示哪些项目。

字段:

- 置信区间: 在作业平均持续时间前后 (+/-) 的值范围 (秒)。此值范围有 95% 的可能性包含下一个作业持续时间。例如, 如果置信区间为 5.88 秒, 那么有 95% 可能性包含下一个作业持续时间的值范围为在作业平均持续时间前后的 5.88 秒。

要查看作业性能统计信息报告:

1. 打开报告选项卡。
2. 单击性能面板。

所有项目的作业性能报告会以结果表的形式显示。

查看项目的作业持续时间

“性能”面板包含一个作业持续时间报告, 其中图示了项目中所有作业的持续时间。

数据源: Rational Build Forge 数据库

报告格式: 图形

要查看项目作业持续时间报告:

1. 打开报告选项卡。
2. 单击性能面板。
3. 单击列表中的项目名称。

项目作业持续时间报告将显示为作业时间的图形。

查看项目的步骤和服务器性能

“性能”面板包含一个项目的步骤和服务器分析报告。它还包含项目中步骤的关键路径分析报告。

对于每个项目步骤, 步骤和服务器分析报告会列出运行该步骤的服务器和以不同运行速度执行该步骤所需的时间, 例如, 最快或最慢运行。步骤时间分类为通过、失败或警告。这些类别会防止您错误地比较时间, 例如将通过时间与失败时间比较。

步骤的关键路径分析报告将列出已通过作业的一些关键度量值。

数据源: Rational Build Forge 数据库

报告格式: 表

关键字: -/- 关键字提供了两条步骤信息: 步骤结果和步骤持续时间 (秒), 如以下示例中所示:

14/-/-	步骤已通过, 总运行时间为 14 秒。
通过/警告/失败	

-/1 通过/警告/失败	步骤失败，总运行时间为 1 秒。
-1/- 通过/警告/失败	步骤已通过但带有警告，总运行时间为 1 秒。

字段:

- 已通过步骤: 所有作业的步骤通过的时间数。
- 平均运行时间: 已通过作业的平均作业持续时间。
- 置信区间: 在作业平均持续时间前后 (+/-) 的值范围 (秒)。此值范围有 95% 的可能性包含下一个作业持续时间。例如，如果置信区间为 5.88 秒，那么有 95% 可能性包含下一个作业持续时间的值范围为在作业平均持续时间前后的 5.88 秒。
- 最长持续时间概览: 下个作业与已通过作业的最长持续时间相等的可能性。
- 最短持续时间概览: 下个作业与已通过作业的最短持续时间相等的可能性。

要查看分析报告，请执行以下操作:

1. 打开**报告**选项卡。
2. 单击**性能**面板。
3. 在列表中选择个项目，然后单击**分析**。

步骤和服务器的分析报告和关键路径分析报告会以结果表的形式显示。

“查询”的预定义报告

提供日期范围或其他所需数据，以便使用当前项目和作业数据来运行预定义的报告。

“查询”面板使用当前数据来创建多个预定义报告:

- 项目和步骤的选择器使用情况历史记录
- 所有服务器的当前服务器清单
- 作业结果通过/失败/警告历史记录
- 某个日期范围内作业的服务器使用情况历史记录
- 基于作业文件的 MD5 值搜索作业文件

查看选择器使用情况历史记录

一份显示所有项目的选择器使用情况的报告。

“查询”面板包含一份报告，其中显示所有项目的选择器使用情况。使用此报告可以查看为项目和步骤指定了哪个选择器。当您使用选择器来动态选择服务器以运行项目或步骤时，此报告会特别有用。

数据源: Rational Build Forge 数据库

报告格式: 表

显示选项: 选择**平铺报告输出**以展开树形层次结构并在列表中显示所有报告结果。

要查看步骤和作业的服务器使用情况报告：

1. 打开**报告**选项卡。
2. 单击**查询**面板。
3. 单击项目选择器和步骤服务器报告的“快速启动”图标 。

选择器使用情况报告会以结果表的形式显示。

查看按服务器列出的当前服务器清单

一份列出所有服务器的服务器清单属性和值的报告。

“查询”面板包含一份报告，其中列出了所有服务器的服务器清单属性和值。使用此报告可以比较服务器属性，标识可被同一个选择器选择的服务器或标识需要的服务器属性。

此报告将显示由分配给服务器的收集器为该服务器指定的所有清单属性。要包含自动分配给所有服务器的特殊清单属性，请选择**显示 BF_properties**。

数据源： Rational Build Forge 数据库

报告格式： 表

显示选项： 选择**平铺报告输出**以展开树形层次结构并在列表中显示所有报告结果。

要查看服务器的当前服务器清单报告：

1. 打开**报告**选项卡。
2. 单击**查询**面板。
3. （可选）选择**显示 BF_properties** 以在报告结果中包含特殊清单属性。
4. 单击当前服务器清单报告的“快速启动”图标 。

当前服务器清单报告会以结果表的形式显示。

查看作业通过/失败/警告结果

一份列出指定日期时间范围内完成作业的通过/失败/警告结果的报告。

“查询”面板包含一份报告，其中列出了指定日期时间范围内完成作业的通过/失败/警告结果。此报告适用于所有项目的所有作业。使用此报告可快速检查已完成作业的运行状况。

数据源： Rational Build Forge 数据库

报告格式： 表

显示选项： 选择**平铺报告输出**以展开树形层次结构并在列表中显示所有报告结果。


要查看项目的作业结果报告：

1. 打开**报告**选项卡。
2. 单击**查询**面板。

3. 使用缺省时间范围，或为报告结果指定时间范围：使用 UNIX 时间格式指定时间。UNIX 时间格式是从 1970 年 1 月 1 日开始的秒数。可以使用如位于 http://www.onlineconversion.com/unix_time.htm 上的工具按照传统的日期时间格式来计算 UNIX 时间。

一天从午夜（00:00）开始，到第二天的午夜结束。例如，要返回 2009 年 7 月 21 日完成的所有作业，请输入以下 UNIX 时间：

开始时间： 07/21/09 00:00 AM	1248134400
结束时间： 07/22/09 00:00 AM	1248220800

4. 单击作业结果报告的“快速启动”图标 。

作业结果报告会以结果表的形式显示。

查看某个日期范围内作业的服务器使用情况

列出项目所有作业的服务器的服务器使用情况报告。

“查询”面板包含一个服务器使用情况报告，其中列出了用于项目的所有作业的服务器。对于每个步骤，该报告会标识哪个服务器运行该步骤并列出步骤持续时间。使用此报告可以比较不同服务器的步骤持续时间。

数据源：Rational Build Forge 数据库


报告格式：表

显示选项：选择平铺报告输出以展开树形层次结构并在列表中显示所有报告结果。

要查看步骤的服务器使用情况报告：

1. 打开报告选项卡。
2. 单击查询面板。
3. 选择开始和结束日期。

一天从午夜（00:00）开始，到第二天的午夜结束。

4. 单击服务器使用情况报告的“快速启动”图标 。

服务器使用情况报告会以结果表的形式显示。

使用作业文件的 MD5 值搜索作业文件

一份搜索作业 BOM 以查找与 MD5 值匹配的文件。MD5 散列值是文件的数字指纹。

“查询”面板包含一份搜索作业 BOM 以查找与 MD5 值匹配的文件报告，如果找到匹配，该报告会列出作业标记和文件位置。当您需要搜索特定版本的文件并需要了解其 MD5 值时，可使用此报告。

作为先决条件，您的项目必须包含 `.scan baseline` 或 `.scan checkpoint` 命令以存储 MD5 值。有关详细信息，请参阅第 350 页的『`.scan`』的参考信息。该 `.scan` 命令将完成以下任务：


- 在步骤的工作目录中创建文件列表
- 在工作目录中生成文件的 MD5 值并将 MD5 值存储在作业 BOM 中

数据源：Rational Build Forge 数据库

报告格式：表

显示选项：选择**平铺报告输出**以展开树形层次结构并在列表中显示所有报告结果。

要搜索作业 BOM 查找 MD5 值：

1. 打开**报告选项卡**。
2. 单击**查询**面板。
3. 在 **MD5 搜索值**中，输入您要查找的 MD5 值。
4. 单击 MD5 文件报告的“快速启动”图标 。

MD5 文件报告会以结果表的形式显示。

使用快速报告创建报告

可使用快速报告提供的报告类型创建您自己的报告设计。快速报告是 Rational Build Forge 的许可选项。

在快速报告中，您可以创建、保存、编辑和运行报告设计。将报告保存到公用目录以与其他用户共享报告。

快速报告中的报告类型使用 Build Forge 数据库中的数据来创建报告。不同的报告类型包含不同的报告字段和描述其功能的名称，例如，“容量”和“构建”。

当您创建报告时，可以从几个报告格式（比如表或图表）中进行选择，并可使用分组和排序功能来控制报告显示。

您可以选择在作业 BOM 中运行和查看报告结果。请参阅第 347 页的『`.report`』的参考信息。

要开始操作，请单击**报告选项卡**以显示报告选项，然后选择**快速报告**。

快速报告的报告组许可权

您可将报告存储在公用或专用存储库中以控制哪些人可运行、修改、保存和删除“快速报告”中创建的报告。

专用报告仅允许创建和保存该报告的 Rational Build Forge 用户来访问。

公用报告可由属于具备相应许可权的访问组的任意用户来访问。缺省情况下，“构建工程师”组中的用户具有对您另存为公用报告的报告的读取/ 写入/编辑权限。

要授予权限，可为用户的访问组指定下列报告许可权中的一个或多个许可权。

读取公共报告	运行保存到“快速报告”中的公用报告存储库的报告的许可权。
保存、编辑或删除报告	保存、编辑或删除保存到“快速报告”中的公用存储库的报告的许可权。

下表列出了“报告”许可权组的许可权并标识了为访问组指定的缺省许可权。

要管理许可权，请在“控制台 UI”中，选择**管理 > 许可权**。

	构建工程师	开发人员	访客	运算符	安全性	系统管理员
读取公共报告	是	是			是	是
保存公共报告	是					
编辑公共报告	是					
删除公共报告	是					

快速报告的报告类型引用

报告类型具有可描述其内容的功能性名称，例如，“容量”和“构建”。对于每个报告类型，本主题描述了报告用途、报告字段和报告示例。

分析 **描述：** 分析报告类型可用于通过使用步骤持续时间和步骤执行顺序来报告步骤级别的作业性能。

报告示例：

- 创建一个表报告以显示作业中包含的每个步骤的步骤持续时间和步骤顺序。按项目名称为作业分组。
- 创建一个折线图报告以显示每个构建标记的步骤持续时间。按步骤名称为构建标记分组。

字段描述：

字段名称	描述
构建标记	作业标记是基于项目标记格式的唯一标识。
项目名称	用户指定的项目名称。
步骤持续时间	步骤以及内联的任何步骤的总运行时间，单位为秒。
步骤名称	用户指定的步骤名称。
步骤顺序	标识步骤运行顺序的序号。

构建 **描述：** 构建报告类型可用于报告项目级别的作业性能。

报告示例：

- 创建一个表报告以显示每个构建标记的构建结果、开始时间和持续时间。按项目为构建标记分组。
- 创建一个条形图报告以显示每个项目的构建计数。按项目分组。

字段描述:

字段名称	描述
构建计数（聚集字段）	作业总计数，包括已完成的作业和失败的作业。
构建持续时间	作业总运行时间（秒）。
构建结果	作业结果：通过、通过但带有警告或失败。
构建开始时间	作业开始时间。
构建状态	构建状态：正在运行、完成、归档或锁定。
构建标记	作业标记是基于项目标记格式的唯一标识。
项目名称	用户指定的项目名称。
选择器名称	用户指定的选择器名称。
用户登录名	启动该作业的用户 Build Forge 登录名或用户名。
用户名	启动该作业的用户名。

容量 **描述:** 容量报告类型可用于按项目报告作业性能。

报告示例:

- 创建一个表报告以显示每个构建标记的构建开始时间、构建持续时间、构建平均持续时间和构建结果。按项目为构建标记分组。
- 创建一个图表报告以显示每个项目的平均构建时间。按选择器为项目分组。

字段描述:

字段名称	描述
构建平均持续时间（聚集字段）	基于作业总数（包括已完成作业和失败作业）的平均作业运行时间。
构建持续时间	作业总运行时间（秒）。
构建结果	作业结果：通过、通过但带有警告或失败。
构建开始时间	作业开始时间。
构建标记	作业标记是基于项目标记格式的唯一标识。
上次构建持续时间（聚集字段）	上个作业的运行时间。完成上个作业所用的总时间（秒）。
项目名称	用户指定的项目名称。
选择器名称	用户指定的选择器名称。

项目 **描述:** 项目报告可用于按项目报告服务器、环境和步骤的使用情况以及按项目报告步骤性能。

报告示例:

- 创建一个表报告以显示项目、类名和项目环境。按服务器为项目分组。
- 创建一个表报告以显示步骤、步骤结果、步骤顺序、步骤环境和服务器。按项目为步骤分组并按步骤顺序排序。

- 创建一个条形图报告以显示每个步骤的步骤计数。按步骤结果为步骤分组。

字段描述:

字段名称	描述
类名	用户为项目指定的类，例如： 生产或测试。
失败步骤计数（聚集字段）	所选分组字段（例如： 项目或服务器或其他字段名称）的失败步骤数。
通过步骤计数（聚集字段）	所选分组字段（例如： 项目或服务器或其他字段名称）的通过步骤数。
项目环境名称	用于定义项目环境变量的项目环境名称。
项目级别	用户指定的项目的访问组。
项目名称	用户指定的项目名称。
服务器名称	用户指定的服务器名称。
步骤计数（聚集字段）	所选分组字段（例如， 项目或服务器）的步骤总数。
步骤环境名称	用于定义步骤环境变量的步骤环境名称。
步骤级别	用户指定的步骤的访问组。
步骤名称	用户指定的步骤名称。
步骤结果	步骤结果： 通过、通过但带有警告或失败。
步骤顺序	标识步骤运行顺序的序号。

步骤度量值

描述: 步骤度量值报告可用于按项目报告步骤成功和失败统计信息。

报告示例:

- 创建一个表报告以显示步骤名称、步骤计数以及通过和失败步骤的百分比。按项目为步骤分组。
- 创建一个折线图报告以显示每个构建标记的步骤持续时间。按步骤名称为构建标记分组。

字段描述:

字段名称	描述
构建标记	作业标记是基于项目标记格式的唯一标识。
失败步骤百分比（聚集字段）	失败的步骤占步骤总计数的百分比。步骤总计数针对您选择的分组字段（例如， 项目）。
通过步骤百分比（聚集字段）	通过的步骤占步骤总计数的百分比。步骤总计数针对您选择的分组字段（例如， 项目）。
项目名称	用户指定的项目名称。
服务器名称	用户指定的服务器名称。
步骤平均持续时间（聚集字段）	基于步骤总数（包括已完成步骤和失败步骤）的平均步骤运行时间。
步骤计数（聚集字段）	步骤总计数，包括已完成的步骤和失败的步骤。步骤总计数针对您选择的分组字段（例如， 项目）。
步骤持续时间	步骤以及内联的任何步骤的总运行时间，单位为秒。
步骤名称	用户指定的步骤名称。

字段名称	描述
步骤结果	步骤结果：通过、通过但带有警告或失败。
步骤顺序	标识步骤运行顺序的序号。
步骤开始时间	步骤开始时间。

质量 **描述：** 质量报告可用于按项目报告作业成功和失败统计信息。

报告示例：

- 创建一个表报告以按构建标记显示构建结果。按项目为构建标记分组。
- 创建一个表报告以按项目显示已通过和失败的所有构建的百分比。按项目分组。
- 创建一个条形图报告以按项目显示构建计数。按构建结果为项目分组。

字段描述：

字段名称	描述
构建计数（聚集字段）	作业总计数，包括已完成的作业和失败的作业。
构建结果	作业结果或状态：通过、通过但带有警告或失败。
构建开始时间	作业开始时间。
构建标记	作业标记是基于项目标记格式的唯一标识。
失败构建百分比（聚集字段）	失败的构建占构建总计数的百分比。构建总计数针对您选择的分组字段（例如，项目）。
通过构建百分比（聚集字段）	通过的构建占构建总计数的百分比。构建总计数针对您选择的分组字段（例如，项目）。
项目名称	用户指定的项目名称。

资源 **描述：** 资源报告可用于按项目和服务器报告步骤和作业性能。

报告示例：

- 创建一个表报告以按服务器显示步骤运行时间。选择步骤顺序、步骤名称、服务器名称、步骤开始时间和步骤持续时间。按步骤顺序和开始时间排序。
- 创建一个表报告以按服务器显示作业运行时间。选择构建标记、服务器名称、构建开始时间、构建持续时间和构建结果。按项目为构建标记分组并按构建开始时间排序。

字段描述：

字段名称	描述
构建持续时间	作业总运行时间（秒）。
构建结果	作业结果：通过、通过但带有警告或失败。
构建开始时间	作业开始时间。
构建标记	作业标记是基于项目标记格式的唯一标识。
项目名称	用户指定的项目名称。

字段名称	描述
选择器名称	用户指定的选择器名称。
服务器名称	用户指定的服务器名称。
步骤持续时间	步骤以及内联的任何步骤的总运行时间，单位为秒。
步骤顺序	标识步骤运行顺序的序号。
步骤开始时间	步骤开始时间。

BOM 描述:

BOM 报告可用于使用作业 BOM 中的信息来创建报告。BOM 报告可针对以下任意数据集运行:

- 数据库中的所有项目（缺省值）
- 单个项目
- 单个项目中的一个或多个构建

您选择的报告字段因数据集而异。报告字段可能包含步骤清单属性、.scan 命令输出、适配器记录的输出以及使用 .bom 命令指定的用户定义列。

报告示例:

- 创建一个表报告以显示多个项目的 .scan 命令输出。选择构建标记、BOM 数据、BOM 路径和 BOM 类型字段。按项目为构建标记分组。
- 创建一个表报告以显示多个项目中调用过滤操作的次数。选择步骤名称、过滤器事件计数类型和过滤器计数。按项目为步骤分组。

字段描述:

字段名称	描述
BOM 数据	如果您在项目包含 .scan checkpoint 命令，那么对于扫描的文件，“BOM 数据”字段将显示 MD5 值。
BOM 路径	如果您在项目包含 .scan checkpoint 命令，那么对于扫描的文件，“BOM 路径”字段将显示文件路径。
BOM 类型	如果您在项目包含 .scan checkpoint 命令，那么对于扫描的文件，“BOM 类型”字段指示路径为 D（目录）、F（文件）还是 S（符号链接）。
构建标记	作业标记是基于项目标记格式的唯一标识。
过滤器事件计数	作为在步骤输出中查找模式匹配的结果，过滤器调用过滤操作的次数。
过滤器事件计数类型	在步骤输出中找到模式匹配时，过滤器调用的过滤操作。
项目名称	用户指定的项目名称。
结果描述	结果记录中的描述字段，通常包含步骤名称的副本。
步骤名称	用户指定的步骤名称。
选择器名称	用户指定的选择器名称。
服务器名称	用户指定的服务器名称。
步骤持续时间	步骤以及内联的任何步骤的总运行时间，单位为秒。
步骤结果	步骤结果：通过、通过但带有警告或失败。

快速报告的报告格式和显示引用

快速报告提供了几个常见的报告格式：表、条形图、折线图和饼图。

注：随快速报告提供的样本报告不包含任何数据。要查看报告数据，必须首先在“管理控制台”中创建项目并运行作业。

表报告格式

样本

要查看样本表报告，请在快速报告中选择 **SampleAnalytic-StepDuration** 报告。

要查看显示选项，请选择 **SampleAnalytic-StepDuration** 报告的编辑图标 。

要求

对于表，请遵循以下需求：

- 选择至少一个报告字段。
- 如果您选择了聚集报告字段，必须同时选择分组字段。请参阅第 392 页的『报告格式和聚集报告字段的选择需求』。

报告字段

在“报告”字段中，选择一个或多个要作为报告中表列的字段。在选择列表中，顺序是有关系的。第一个字段将作为第一个表列，最后一个字段将作为最后一个表列。第一个表列还可用于分组报告结果。



要试验报告字段，请选择 **Sample Analytic-StepDuration** 报告的编辑图标 。在选择列表中，更改字段顺序，保存选择并运行报告以查看表列顺序的更改。

分组字段

在“分组”字段中，可以选择用于对表行进行分组的字段。您选择的组会作为树控件添加到用于分组报告结果的第一个表列。您可以选择多个分组字段，并可选择没有作为表列显示的分组字段。

要试验分组，请选择 **SampleAnalytic-StepDuration** 报告的编辑图标 。选择“项目名称”作为分组字段，保存您的选择，然后运行该报告。“项目名称”的树节点将添加到“构建标记”列。

排序字段

在“排序”字段中，可以选择带有待排序数据的表列。可使用箭头来指定排序方向。向上箭头  按升序对数据进行排序（最低到最高），向下箭头  按降序对数据进行排序（最高到最低）。

要试验排序，请选择 **SampleAnalytic-StepDuration** 报告的编辑图标 。选择“项目名称”作为排序字段，可使用箭头来指定排序方向，保存您的选择并运行该报告。

条形图报告格式

样本

要查看样本条形图，请在快速报告中选择 **SampleCapacity-RunTimeByProject** 报告。

要查看显示选项，请选择 **SampleCapacity-RunTimeByProject** 报告的编辑图标



垂直图表元素（y 轴）

选择一个字段作为您要为一组数据元素（x 轴）进行比较的数据值（y 轴）。字段名称和数据值单位会在图表的垂直轴（或称为 y 轴）上显示。只能选择一个字段作为垂直图表元素。

水平图表元素（x 轴）

选择一个字段作为您要使用数据值（y 轴）进行比较的数据元素集合（x 轴）。字段名称显示为 x 轴标签。

数据值在柱条上显示，值与柱条高度相等。只能选择一个字段作为水平图表元素。

分组字段

您可以通过选择分组字段对图表的水平（或称为 x 轴）上的数据元素进行分组。

要求

对于图表，请遵循以下需求：

- 选择 x 轴和 y 轴报告字段。
- 为 x 轴和 y 轴选择不同的报告字段。
- 为 x 轴数据选择分组字段。
- 如果您选择了聚集报告字段，必须同时选择分组字段。请参阅第 392 页的『报告格式和聚集报告字段的选择需求』。

示例：

要试验条形图，请尝试创建以下报告：

- 对于“步骤”报告类型的每个“步骤名称”（x 轴），比较“失败步骤百分比”（y 轴）并按“构建标记”对结果进行分组。
- 对于“容量”报告类型的每个“项目”（x 轴），比较“上次构建持续时间”（y 轴）并按“项目名称”对结果进行分组。
- 对于“资源”报告类型，比较每个“构建标记”（x 轴）的“构建持续时间”（y 轴），并按“项目名称”对“构建标记”进行分组。

折线图报告格式

描述

折线图显示数据随时间推移的进展情况或一系列事件的进展情况。缺省分组字段为 x 轴报告字段。如果您选择了多个分组字段，报告中将显示多条线。

样本

要查看样本折线图，请在快速报告中选择 **SampleResource-DurationOverTime** 报告。

要查看显示选项，请选择 SampleResource-DurationOverTime 报告的**编辑**图标



垂直图表元素（y 轴）

选择一个字段作为您要为一组数据元素（x 轴）进行比较的数据值（y 轴）。字段名称和数据值单位会在图表的 y 轴上显示。只能选择一个字段作为垂直图表元素。

水平图表元素（x 轴）

选择一个字段作为您要使用数据值（y 轴）进行比较的数据元素集合（x 轴）。字段名称显示为图表的 x 轴标签。

x 轴数据元素的数据值在 y 轴上与其值相等的点旁边显示。一条连续的线将这些点连接就形成了折线图。只能选择一个字段作为水平图表元素。

分组字段

您可以通过选择分组字段对图表的水平（或称为 x 轴）上的数据元素进行分组。缺省分组字段为 x 轴报告字段。如果您选择了多个分组字段，报告中将显示多条线。

要求

对于图表，请遵循以下准则：

- 选择 x 轴和 y 轴报告字段。
- 为 x 轴和 y 轴选择不同的报告字段。
- 为 x 轴数据选择分组字段。
- 如果您选择了聚集报告字段，必须同时选择分组字段。请参阅第 392 页的『报告格式和聚集报告字段的选择需求』。

示例

要试验折线图，请尝试创建以下报告：

- 对于“步骤”报告类型的每个“构建标记”（x 轴），显示“步骤持续时间”（y 轴）并按“步骤名称”对结果进行分组。
- 对于“构建”报告类型的每个“构建开始时间”（x 轴），显示“持续时间”（y 轴）并按“项目名称”对结果进行分组。
- 对于“容量”报告类型的每个“构建标记”（x 轴），显示“构建持续时间”（y 轴）并按“服务器名称”对结果进行分组。

饼图报告格式

Y-series 元素

选择一个字段作为饼图的数据值。楔形的大小代表数据值。字段名称和数据值单位会在饼图上显示。只能选择一个字段作为 y-series 元素。

X-series 元素

选择一个字段作为您要使用数据值（y-series）进行求值的数据元素集合（x-series）。楔形数代表数据元素数。只能选择一个字段作为 x-series 元素。

分组字段

组对于饼图是必需的。X-series 元素用作缺省组。选择不同的组可生成数据值（y-series）的另一个饼图。

要求

对于图表，请遵循以下准则：

- 选择 x-series 和 y-series 报告字段。
- 为 x-series 和 y-series 选择不同的报告字段。
- 为 x 轴数据选择分组字段。
- 如果您选择了聚集报告字段，必须同时选择分组字段。请参阅『报告格式和聚集报告字段的选择需求』。

示例

要试验饼图，请尝试创建以下报告：

- 对于“项目”报告类型，选择“失败步骤计数”（y-series）和“步骤名称”（x-series）。
- 对于“容量”报告类型，选择“构建平均持续时间”（y-series）和“项目名称”（x-series）。
- 对于“分析”报告类型，选择“步骤持续时间”（y-series）和“步骤名称”（x-series），并按“步骤名称”分组。

报告格式和聚集报告字段的选择需求

要生成有意义的报告，需要遵循表和图表以及聚集报告字段的一些基本需求。

聚集报告字段

要求

如果在表或图表中包含了聚集报告字段，那么必须也包含分组字段。

定义

聚集报告字段包含从 Build Forge 数据库中的一个或多个原始数据字段中派生的数据。聚集报告字段中的数据不存储在数据库中。

下列报告字段为聚集字段。

分析：上次构建持续时间	项目：失败步骤计数	项目：通过步骤计数
项目：步骤计数	步骤度量值：失败步骤百分比	步骤度量值：通过步骤百分比
步骤度量值：步骤平均持续时间	步骤度量值：步骤计数	质量：构建计数
质量：失败构建百分比	质量：通过构建百分比	构建：构建计数

表报告格式

要求

对于表，请遵循以下需求：

- 选择至少一个报告字段。
- 如果您选择了聚集报告字段，必须同时也选择分组字段。

图表报告格式

要求

对于条形图、折线图和饼图，请遵循以下需求：

- 选择 x-series 和 y-series 报告字段。
- 为 x-series 和 y-series 选择不同的报告字段。


- 为 x 轴数据选择分组字段。
- 如果您选择了聚集报告字段，必须同时也选择分组字段。


样本报告引用

样本报告是可以使用提供的报告类型创建的报告示例。

注：样本报告使用 Rational Build Forge 数据库中的数据来创建报告输出。仅当您已经创建了项目并运行作业时，结果才显示。

要显示样本报告的列表，请单击**快速报告**。

要查看样本报告的字段和格式选项，请选择报告的**编辑**图标 。

要复制样本报告，请选择报告的**编辑**图标 ，然后单击**复制报告**。

要运行样本报告，请选择报告名称。

下表中提供了样本报告定义。

样本报告名称	描述
SampleAnalytic-StepDuration	描述: 创建一个报告构建步骤详细信息的表。 报告类型: 分析 报告格式: 表 报告字段: 构建标记、项目名称、步骤名称、步骤持续时间 报告选项: 无分组或排序选项
SampleBuild-BuildsByState	描述: 创建一个按项目列出构建和构建状态的表 报告类型: 构建 报告格式: 表 报告字段: 构建标记、项目名称、构建计数、构建状态 报告选项: 按项目名称分组
SampleCapacity- RuntimeByProject	描述: 创建一个按项目显示作业的平均构建持续时间的条形图 报告类型: 容量 报告格式: 条形图 报告字段: 构建平均持续时间 (y 轴) 和项目名称 (x 轴) 报告选项: 按项目名称分组

样本报告名称	描述
SampleProject-TotalsByProject	<p>描述: 创建一个按项目报告通过和失败步骤数量的表。</p> <p>报告类型: 项目</p> <p>报告格式: 表</p> <p>报告字段: 项目名称、通过步骤计数、失败步骤计数、步骤计数</p> <p>报告选项: 按项目名称分组</p>
SampleQuality-PercentSuccess	<p>描述: 创建一个按项目报告通过和失败作业百分比的表。</p> <p>报告类型: 质量</p> <p>报告格式: 表</p> <p>报告字段: 项目名称、构建计数、失败构建百分比、通过构建百分比</p> <p>报告选项: 按项目名称分组</p>
SampleResource-DurationOverTime	<p>描述: 创建一个显示每个构建的构建开始时间和构建持续时间的图表</p> <p>报告类型: 资源</p> <p>报告格式: 表</p> <p>报告字段: 构建持续时间 (y 轴) 和构建开始时间 (x 轴)</p> <p>报告选项: 升序排序顺序</p>

使用提供的报告类型创建报告

只需选择您想要使用的报告类型、报告格式和报告字段即可创建您自己的报告。

过程

1. 选择**报告**选项卡，然后选择**快速报告**。
2. 在**报告名称**中，输入唯一的名称。报告名称是必需的。报告名称可用于将报告保存在数据库中，它必须是唯一的。
3. 在**报告标题**中，输入报告的描述性标题。报告标题是必需的。标题在报告的顶部显示。
4. 在**可视性**中，选择“公用”或“专用”。专用报告不能共享。公用报告可以与具备所需权限的用户共享。关于详细信息，请参阅第 383 页的『快速报告的报告组许可权』。
5. 在**报告格式**中，选择表或以下图表格式之一：条形图、折线图或饼图。有关报告格式的详细信息，请参阅第 389 页的『快速报告的报告格式和显示引用』。
6. 在**报告类型**中，选择一个报告类型。该报告类型决定您在报告中包含的内容。有关详细信息和示例，请参阅第 384 页的『快速报告的报告类型引用』。
 - 选择要在表或图表报告中显示的报告字段。
 - 选择用于控制报告显示的分组和排序选项。

要点: 如果您选择了 BOM 报告类型, 将显示“项目”框。要创建特定项目的 BOM 报告, 首先请选择项目以显示报告字段的完整列表, 然后选择要在报告中包含的字段。

7. (仅适用于 BOM 报告类型) 在**项目**中, 选择以下选项之一:

- 选择“所有项目”以仅显示标准 BOM 报告字段。(“所有项目”是缺省设置。)

报告输出包含数据库中所有项目的 BOM 信息。

- 如果选择一个项目, 那么除显示标准 BOM 报告字段之外, 还显示其项目特定的 BOM 字段。

报告输出仅包含单个项目的 BOM 信息。

8. 单击**保存报告**以保存您的选择。该报告将在报告列表中显示。

将报告输出添加到作业 BOM

可选择使用 `.report` 命令将报告结果添加到作业 BOM。

过程

1. 请使用“快速报告”工具来创建报告。请参阅第 394 页的『使用提供的报告类型创建报告』。

注: BOM 报告类型或专用报告不支持此功能。

2. 可使用 `.report` 命令将报告添加到作业 BOM。请参阅第 347 页的『`.report`』的参考信息。

在快速报告中修改和管理报告

在“快速报告”中, 您可以运行报告、查看报告结果和编辑报告设计。

可通过编辑报告来更改报告设计。过滤器是报告定义的一部分, 并可添加到任意报告以过滤输出。

运行报告

要运行您在“快速报告”中创建的报告, 请单击报告名称。

关于此任务

报告结果将在“快速报告”结果视图中显示。

结果

要返回报告列表, 请单击 Web 浏览器的“后退”箭头。

编辑报告

要编辑您在“快速报告”中创建的报告, 请单击报告名称旁边的**编辑**图标。

关于此任务

报告选择会在“报告”详细信息中显示。完成更改后, 请单击**保存报告**来保存更改。

复制报告

复制样本报告或其他报告会将其报告字段和格式选项复制到新的报告设计，并为该报告指定唯一的名称。

开始之前

可通过向报告名称中添加“Copy”来创建唯一的报告名称，例如，`<Report_Name> Copy`。

如果要制作报告的多个副本，可以使用以下语法追加编号：`<Report_Name> Copy <Copy_Number>`。例如，`<Report_Name> Copy 2`。

过程

1. 单击要复制的报告的**编辑**图标。
2. 单击**复制报告**。

复制的报告将在报告列表中显示。

为报告输出创建过滤器

您可以创建一个报告过滤器来控制报告输出中显示的信息。

开始之前

在创建过滤器之前，请查看以下需求和限制：

- 首先创建一个报告；过滤使用报告定义来提供过滤器选项。
- 报告过滤器可作为报告定义的一部分来保存，并仅适用于单个报告。
- 创建报告过滤器后，它将适用于该报告的所有运行，直到您更改或删除了过滤器。
- 对于 BOM 报告类型，您可以按项目过滤，按构建过滤，以及为特定报告字段指定过滤条件。
- 对于除 BOM 以外的报告类型，必须为特定报告字段指定过滤条件。

过程

1. 在报告列表中，选择一个报告并选择该报告的**编辑**图标。
2. （仅对 BOM 报告类型为必需）在**项目**中，选择以下选项之一：
 - 选择“所有项目”以仅显示标准 BOM 报告字段。（“所有项目”是缺省设置。）

报告输出包含数据库中所有项目的 BOM 信息。

 - 如果选择一个项目，那么除显示标准 BOM 报告字段之外，还显示其项目特定的 BOM 字段。

报告输出仅包含单个项目的 BOM 信息。
3. 选择**过滤器**选项卡。
4. （可选，仅针对 BOM 报告类型）单击**显示构建过滤器**，从以下选项中进行选择以指定报告中要包含的构建信息：
 - 要包含所有当前和未来构建的数据，请不作任何选择。
 - 要只包含所有当前构建的数据，请单击**仅所有当前构建**。
 - 要只包含选定构建的所有当前和未来构建数据，请选择列表中的构建标记。

5. （对于所有其他报告类型都为必需）单击**添加过滤器**以在从**报告详细信息**选项卡中选择的报告类型中选择一个用作过滤器的报告字段。

要创建过滤表达式：

- a. 在**过滤字段**中，选择要用于过滤报告数据的报告字段。
 - b. 在**过滤运算符**中，选择关系运算符。
 - c. 在**过滤值**中，输入报告字段值。
6. 单击**保存报告**以保存您的报告过滤器选择。

对快速报告常见问题进行故障诊断

如果您在使用 Rational Software Analyzer 时遇到问题，请复审本主题中的信息，以查看是否存在可接受的变通方法或解决方案。

端口冲突

快速报告使用您在安装过程中指定的应用程序 Web 服务器来显示报告。如果您遇到端口冲突，必须为该应用程序 Web 服务器配置未分配的端口。

第 23 章 使用实用程序

本主题描述如何设置和运行 Rational Build Forge 提供的命令行实用程序。

访问和运行命令行实用程序

命令行实用程序位于 Build Forge 安装目录中，即 `<bfinstall>`（在 Windows 上）和 `<bfinstall>/Platform`（在 UNIX 和 Linux 上）。

您必须正确配置您的环境（或系统级别的环境），引擎级别的 shell 命令才能工作。例如，在 Oracle 和 UNIX 上，必须手动设置 ORACLE_HOME、TNS_ADMIN 和 LD_LIBRARY_PATH，否则实用程序将无法运行。

使用命令行实用程序（如 `bfexport` 或 `bfimport`）时，该命令需要能够找到 `buildforge.conf` 文件才能访问数据库，因此您必须从安装目录运行该命令，或将环境变量 `BF_CONFIG_FILE` 设置为 `buildforge.conf` 文件的完整路径。

导出项目

可以使用 `bfexport` 命令或 `.export` 点命令将项目和其他 Build Forge 对象导出到 XML 文件。

导出的项目会存储在 XML 文件中。导出的项目可以重新导入 Build Forge 中。

示例：您可以在项目末尾添加一个运行 `bfexport` 的步骤以保存项目配置数据。您可以将它用作项目定义的备份。还可以使用它来将项目移到其他 Build Forge 安装。

bfexport 参考

使用 `bfexport` 命令可以将项目数据导出到命名的 XML 文件或将项目数据发送到显示终端（`stdout`）以供查看。导出文件包含单个项目或项目快照的项目配置数据。

语法

`bfexport`

`bfexport [-l]`

`bfexport [-l] <project_name>`

`bfexport [-c "<comment>"] [-f <file_name>] [-g] [-s] [-C] [-L] [-n]`
`<project_name> | <project_name> <snapshot_name> | <project_id>`

用法

要完成常见的项目导出任务，需要使用以下命令选项：

- 要显示命令语法，请使用不带任何选项的 `bfexport`。
- 要列出 Build Forge 数据库中存储的项目名称和项目标识，请使用 `bfexport -l`。

在命令输出中快照名称会追加到项目名称后面：`<project_ID>: <project_name> - <snapshot_name>`

- 要将项目数据发送到 XML 文件，请使用 `bfexport -f <file_name>`。必须指定 `-f <file_name>` 选项以生成可用于导入项目数据的文件。
- 在 z/Linux 平台上，您必须将该命令作为 `bfexport.pl` 运行。在所有其他平台上，该命令都不需要扩展名。

先决条件和限制

在 Build Forge 安装目录中查找 `bfexport` 实用程序。

服务器授权密码未包含在导出文件中；导入后，您必须在 UI 中手动输入服务器授权密码。

此 `bfexport` 命令必须能够找到 `buildforge.conf` 文件并访问 Build Forge 数据库。从 `buildforge.conf` 所在的目录运行 `bfexport`，即 `<bfinstall>`（在 Windows 上）和 `<bfinstall>/Platform`（在 UNIX 和 Linux 上）。

示例

要将输出写入文件，请使用 `-f <file_name>` 选项。在下面的示例中，`helloworld` 为输出文件名称，并使用项目标识代替项目名称。

```
bfexport -c "Saving a copy of project before making changes"
-f helloworld 675B57CC-8366-11DD-B2E0-043C04E44E1A
```

要只导出缺省项目快照，请使用 `<project_name>`。

```
bfexport -f helloworld test_project
```

要导出项目的一个快照，请使用 `<project_name> <snapshot_name>`。

```
bfexport -f helloworld test_project snapshot_1
```

如果父项目快照不是缺省项目，那么必须指定后跟父关键字的 `<project_name>` 以导出父项目快照。

```
bfexport -f helloworld test_project parent
```

选项描述

选项	描述
<code><project_name></code>	要导出的项目的名称。项目名称或项目标识是必需的。如果项目名称包含空格，必须为名称加引号。 在命令选项后指定项目名称。
<code><snapshot_name></code>	要导出的项目快照的名称；项目名称是必需的，如下面的语法中所示： <code><project_name> <snapshot_name></code> 在命令选项后指定项目名称和快照名称。如果项目名称或快照名称包含空格，必须为名称加引号。
<code><project_id></code>	要导出的项目的标识。项目标识是一个 UUID。项目标识或项目名称是必需的。 在命令选项后指定项目标识。

选项	描述
-f <file_name>	<p>bfexport 输出的 XML 文件名。如果没有提供路径名，将使用当前工作目录。如果文件名包含空格，必须为名称加引号。</p> <p>如果您没有提供文件名，会将 bfexport 输出发送到 stdout。</p> <p>注：stdout 仅用于查看。请勿将 stdout 重定向至文件；结果文件包含日志记录消息，且不能用作 bfimport 命令或 UI 导入实用程序的导入文件。</p>
parent	<p>必需的关键字，用于在父级不是缺省项目快照时导出父级项目快照。</p> <p>在项目快照名称后指定父级关键字：</p> <p>bfexport -f helloworld test_project parent</p>
-l	按名称和项目标识列出数据库中的项目。不能将 -l 选项与其他选项一起使用。
-c "<comment>"	包含一段注释。必须为注释加上引号（例如，"my project version 50"）。注释将添加到 <buildforge> XML 元素。
-g	将指定要接收通知的访问组的成员用户保存到 XML 文件。用户及其属性列于 <user> XML 元素中。需要 -s。
-s	将管理控制台中定义的服务器保存到 XML 文件。服务器及其属性列于 <server> XML 元素中，而且还包含所有关联的 <auth> 和 <collector> 信息。
-L	将 UI 中定义的 LDAP 域控制器保存到 XML 文件。LDAP 域控制器及其属性列于 <ldap> XML 元素中。
-n	将为项目及步骤指定的通知模板保存到 XML 文件。通知模板及其属性列于 <mail-template> XML 元素中。
-C	将为项目服务器指定的收集器保存到 XML 文件。收集器及其属性列于 <collector> XML 元素中。需要 -s。

故障诊断

如果您遇到有关无法找到报告文件或库的错误，请确保库路径设置正确。

- *LD_LIBRARY_PATH*（在 Linux 系统和 UNIX 系统上）。
- *LIBPATH*（在 AIX 系统上）。

正确的设置在 <bfinstall>/rc/buildforge 中。检查在该文件中设置的库路径变量和其他变量。根据需要重置，然后将其导出。

使用 .export

可以使用 .export 从项目中的步骤导出项目。

此 .export 命令不会提供导出任何其他对象数据的选项。要导出其他 Build Forge 对象，必须使用 bfexport 命令。

请参阅第 341 页的『.export』。

导入项目

可以使用 `bfimport` 命令或控制台中的“导入”设施导入先前导出的项目和其他 Build Forge 对象。

有一组选项可供您选择要安装的对象。

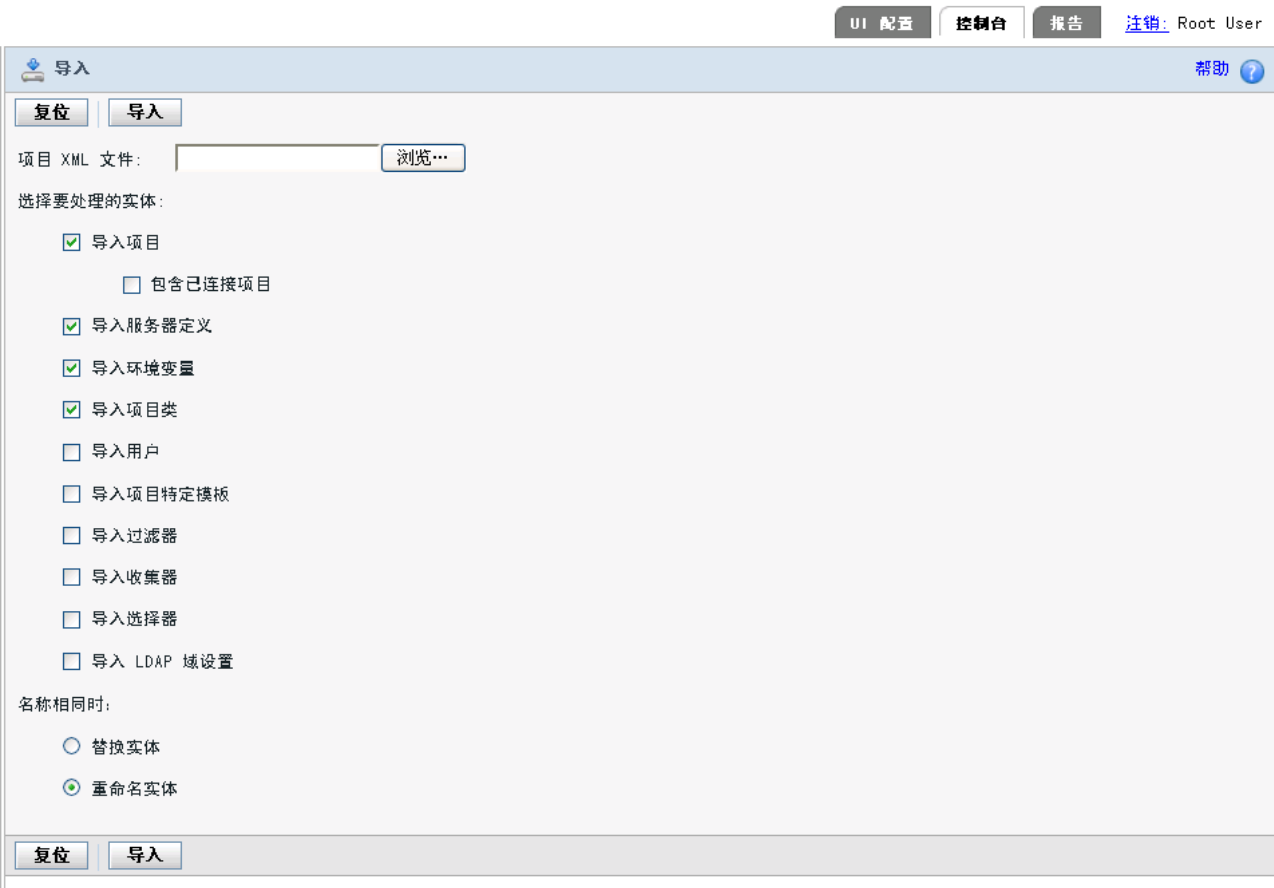
您可以选择如何将访问组应用到导入的对象。它们通过以安全访问导入系统设置进行设置。

注: Build Forge 用户界面可能会拒绝导入大于 2M 的文件。在此情况下, 请直接使用 `bfimport` 命令行工具, 或者调整“`upload_max_filesize`”PHP 环境变量。

使用“导入”实用程序导入项目和其他对象

可以使用“UI 导入”实用程序来导入项目的对象定义和已保存到导出文件的其他对象。您可使用该导入实用程序来选择要从导出文件中导入的对象。

1. 选择管理 > 导入。



2. 单击浏览以找到项目的导出 XML 文件。

您必须使用 `bfexport` 命令或 `.export` 点命令来创建导出文件。

3. 从导出 XML 文件中选择要导入到 UI 的项目和其他对象。

注: 如果您要导入服务器对象, 必须在导入它们后在 UI 中手动输入其服务器授权密码。bfexport 和 .export 命令不会将服务器授权密码保存到导出文件。

4. 选择**替换实体或重命名实体** (缺省选项) 来指定导入实用程序是用相同的名称来替换或覆盖对象还是为这些对象重命名。

要点: 要理解重命名和替换的工作原理, 请参阅第 406 页的『导入时重命名和替换对象』。

bfimport 参考

使用 bfimport 实用程序可将先导出到 XML 文件的项目和其他对象的定义导入 UI。您也可以使用导入实用程序导入 XML 文件中的选定对象。

本主题描述 bfimport 命令的语法并提供用法详细信息。

语法

bfimport

bfimport [-L] <file_name>

bfimport [-p -I -s -S -e -c -C -u -T -f -d -r] <file_name.xml>

bfimport [-L | [-p -I -s -S -e -c -C -u -T -f -d -r]] <file_name.xml>

限制和注意事项

服务器授权密码未包含在导出文件中; 您必须手动输入服务器授权密码。

缺省情况下, 导入时如果存在与导入对象名称相同的对象, 将重命名正在导入的对象以防止数据库对象被覆盖。或者, 您也可以选择用 -r 选项在具有同名对象的情况下替换对象。对于快照对象, 这些对象必须按 UUID 而不是名称进行比较。

缺省情况下, bfimport 会使用下面的命名约定来重命名对象:

<object_name>_IMPORT_<number>

对于重命名, 快照对象会失去它们的快照名称并作为新的基本快照或父级快照导入, 即使该快照对象是父级快照的子代。

有关重命名的详细信息, 请参阅第 406 页的『导入时重命名和替换对象』。

要替换对象, 必须指定 -r 选项。替换选项会覆盖现有的对象。对于 7.0.2 及更早的导出文件, 快照对象是不可替换的, 而是使用 <object_name>_IMPORT_<number> 约定来重命名。对于 7.1 导出文件, 如果快照对象的 UUID 匹配, 那么会替换这些对象。

先决条件

bfexport 命令或 .export 点命令创建的导出 XML 文件。

在 Build Forge 安装目录中查找 bfimport 实用程序。

此 bfimport 命令必须能够找到 buildforge.conf 文件并访问 Build Forge 数据库。从 buildforge.conf 所在的目录运行 bfimport, 即 <bfinstall> (在 Windows 上) 和 <bfinstall>/Platform (在 UNIX 和 Linux 上)。

用法

要完成常见的项目导入任务，需要使用以下命令选项：

- 要显示命令语法，请使用不带任何选项的 `bfimport`。
- 要显示 XML 文件中的 Build Forge 对象及其名称的摘要列表，请使用 `bfimport -L <file_name.xml>`
- 在 zLinux 平台上，您必须将该命令作为 `bfexport.pl` 运行。在所有其他平台上，该命令都不需要扩展名。
- 如果您未指定选项，将不会导入任何对象。必须指定选项才能导入单个对象。
- 如果您指定了不存在的对象的选项，导入实用程序将跳过不在 XML 文件中的对象并导入该文件中存在的对象。

示例

要列出 XML 文件中的 Build Forge 对象，请仅指定 `-L` 选项和 XML 文件名。下面的示例显示了部分命令输出。

```
C:\Program Files\IBM\Build Forge>bfimport -L samples\projects\basic.xml
10/07/2008 5:31:55 PM: Import: 7624: CRRBF200811: Importing export
file from a 7.0.10025 version console.
Project : [Basic Sample]
Tag Variable : [MAJ]
Tag Variable : [MIN]
Step : [Checkout Source]
Step : [Update Applet Version]
Step : [Create Baseline]
Environment : [Basic Environment]
Class : [Production]
Filter : []
Selector : [Web Server]
Selector : [Local Server]
```

要导入 XML 文件中的所有对象，请指定要导入对象的选项，如下面示例中所示。导入时重命名对象。成功语句显示导入是否成功。

```
C:\Program Files\IBM\Build Forge>bfimport -p -I -s -S -e -c -C -u -T -d -f
"samples\projects\basic.xml"
```

选项描述

选项	描述
<code><file_name.xml></code>	包含要导入的 Build Forge 对象的导出 XML 文件的名称。必须使用 <code>bfexport</code> 命令或 <code>.export</code> 点命令来创建 XML 文件。XML 文件名是必需的，如果该 XML 文件不在当前目录（从中发出 <code>bfexport</code> 命令的目录）中，您必须提供路径名。 如果文件名包含空格，必须为名称加引号。
<code>-L</code>	列出导出 XML 文件中的对象及其对象名。单独使用此选项，不要与它一起指定任何其他 <code>bfimport</code> 选项。来自 <code>-L</code> 选项的输出可被发送至 <code>stdout</code> 或重定向至 XML 文件或文本文件。

选项	描述
-p	<p>从 XML 文件导入项目配置数据。项目配置数据包含步骤和项目定义数据，同时包含标记变量。</p> <p>在重命名时，会按照 <code><project_name>_IMPORT_<number></code> 将项目名称导入 UI。</p>
-I	<p>导入在项目级别或步骤级别引用的链式项目或库。</p> <p>在重命名时，会按照 <code><project_or_library_name>_IMPORT_<number></code>，将链式项目或库导入 UI。</p>
-S	<p>导入 UI 中定义的选择器对象。</p> <p>在重命名时，会按照 <code><selector_name>_IMPORT_<number></code>，将选择器导入 UI。</p>
-s	<p>导入管理控制台中定义的服务器对象（如果已为 bfexport 指定了 -s 选项）。</p> <p>在重命名时，会按照 <code><server_name>_IMPORT_<number></code>，将服务器导入 UI。</p> <p>服务器授权密码未包含在导出 XML 文件中；您必须手动输入服务器授权密码。</p>
-e	<p>导入在项目级别或步骤级别引用的环境及其变量。</p> <p>在重命名时，会按照 <code><environment_name>_IMPORT_<number></code> 导入环境。</p>
-c	<p>导入项目引用的类。</p> <p>在重命名时，会按照 <code><class_name>_IMPORT_<number></code> 导入类。</p>
-C	<p>导入为项目的服务器指定的收集器（如果已为 bfexport 指定了 -C 选项）。</p> <p>在命名时，会按照 <code><collector_name>_IMPORT_<number></code> 导入收集器。</p>
-u	<p>导入指定要接收电子邮件通知的访问组的成员用户（如果已为 bfexport 指定了 -g 选项）。</p> <p>在重命名时，会按照 <code><users>_IMPORT_<number></code> 导入用户信息</p>
-T	<p>导入为项目和步骤指定的用户创建的通知模板（如果已为 bfexport 指定了 -n 选项）。</p> <p>在重命名时，会按照 <code><template_name>_IMPORT_<number></code> 导入通知模板。</p>
-f	<p>导入为项目步骤指定的日志过滤器（如果已为 bfexport 指定了 -n 选项）。</p> <p>在重命名时，会按照 <code><filter_name>_IMPORT_<number></code> 导入日志过滤器。</p>

选项	描述
-d	导入 UI 中定义的 LDAP 域控制器（如果已为 bfexport 指定了 -L 选项）。 在重命名时，会按照 <LDAP_domain_controller>_IMPORT_<number> 导入日志过滤器。
-r	替换导入的对象而不是为它们重命名。缺省情况下，将使用下面的命名约定来重命名导入的对象： <object_name>_IMPORT_<number> 对于 7.1 对象，如果您指定了替换选项，bfimport 命令将覆盖 UI 中的对象。 对于 7.0.2 和更早的对象，不会替换快照对象。将使用以下命名约定对它们重命名： <object_name>_IMPORT_<number> 要点： 快照对象按 UUID 而非名称进行替换评估。 关于详细信息，请参阅『导入时重命名和替换对象』。

访问组如何指定给导入的对象

带安全访问权导入系统设置控制如何将访问组指定给导入的对象。

- 如果带安全访问权导入设置为“是”（缺省值），那么导入缺省安全访问组设置将指定访问组。“构建工程师”组为缺省访问组。
- 如果带安全访问权导入设置为“否”，那么导入不安全缺省访问组设置将指定访问组。开发者为缺省值。

如果未对导入不安全缺省访问组指定访问组，那么将使用最近创建的访问组。如果没有创建访问组，将使用“Default”访问组。

访问组指定和安全性

作为一种安全措施，Build Forge 不允许直接导出或导入访问组。如果直接导出或导入了访问组，那么有权访问控制台主机的用户可以直接操作导出的对象的 XML 文件，然后将其导入。导入时指定访问组可以确保访问组仅由授权用户指定。

导入时重命名和替换对象

如果数据库中已存在相同名称的对象，bfimport 命令和“导入”实用程序将重命名导入的对象。导入时重命名对象是缺省行为。

要更改此行为并在导入时替换现有对象，您必须指定 bfimport 的 -r 选项或在 UI 中选择“替换实体”选项。

下面的主题描述了 bfimport 命令和“导入”实用程序在重命名和替换导入的对象时使用的命名约定。

导入时，快照对象（项目、选择器和环境）会保留它们的快照名称（如有），或者被指定缺省的快照名称。

重命名 7.1 或更早导出文件中的对象

对于 7.1 和更早导出文件中的对象，bfimport 命令和“导入”实用程序会使用以下命名约定重命名 UI 中的对象。

对象状态	UI 对象名	UI 快照名称（仅适用于快照对象）
新名称，不在数据库内	<new_object_name>	<snapshot_name>
存在于数据库中	<existing_object_name>_IMPORT_<n>	<snapshot_name> Base Snapshot

快照对象将作为新的父级快照导入，即使它是父级快照的子代也是如此。只有项目、选择器和环境可以作为快照对象。如果快照名称已存在，将保留该名称。否则，将指定缺省的 Base Snapshot 名称，如表中所示。

替换 7.1 之前的导出文件中的对象

对于 7.1 之前的导出文件中的对象，bfimport 命令和“导入”实用程序会使用以下命名约定来替换 UI 中的对象。请注意，快照对象按 UUID 而非名称进行替换评估。

对象状态	UI 对象名	UI 快照名称（仅适用于快照对象）
新名称，不在数据库内	<new_object_name>	Base Snapshot
存在于数据库中，非快照对象	<existing_object_name	不适用
存在于数据库中，快照对象	<existing_object_name>_IMPORT_<n>	Base Snapshot

对于快照对象，会将快照对象作为新的父级快照导入，即使它是父级快照的子代。只有项目、选择器和环境可以作为快照对象。7.1 之前的导出文件不能包含带有快照名称的对象，因此将指定缺省的 Base Snapshot 名称，如表中所示。

替换 7.1 导出文件中的对象

对于 7.1 导出文件中的对象，bfimport 命令和“导入”实用程序会使用以下命名约定来替换 UI 中的对象。

对象状态	UI 对象名	UI 快照名称（仅适用于快照对象）
新名称，不在数据库内	<new_object_name>	<snapshot_name> Base Snapshot
存在于数据库中	<new_object_name>	<snapshot_name> Base Snapshot

快照对象按 UUID 而非名称进行替换评估。快照对象将作为新的父级快照导入，即使它是父级快照的子代也是如此。只有项目、选择器和环境可以作为快照对象。所有 7.1 对象都具有唯一的快照名称或使用缺省的 Base Snapshot 名称。

快照导入和替换方案

快照对象使用 UUID 而非名称进行替换评估。多次导入快照对象时请多加小心。

以下方案演示了该行为。

1. 从系统 A 导出快照对象并将其导入空的系统 B。系统 A 上导入的对象的 UUID 与系统 B 上相同。
2. 在系统 A 上修改快照对象属性，包括名称。
3. 再次从系统 A 导出快照对象并使用替换选项将其导入系统 B。UUID 匹配，因此会替换该快照对象。系统 B 上替换的对象现在承担系统 A 上的对象中已更改的所有属性，包括更改的名称。

第 24 章 在“UI 配置”选项卡中链接到 Web 资源

使用 **UI 配置**选项卡可对 Build Forge UI 添加选项卡。

可使用 **UI 配置**选项卡向 Build Forge 用户界面添加选项卡。每个新选项卡都包含一个 URL。可以使用这些选项卡链接到外部资源，如关于您的应用程序、操作系统、服务器或用户的信息。

注： 不要创建链接到内部 Build Forge URL 的选项卡。

要添加选项卡，请执行以下操作：

1. 选择 **UI 配置**选项卡。
2. 单击**添加选项卡**。
3. 在**名称**中，输入选项卡的标题或名称。
4. 在**链接**中，可输入以下任何信息：

选项	用法	示例
包含协议的 URL	<ul style="list-style-type: none">• 如果协议不是 http，那么是必需的• 如果协议是 http，那么是可选的	http://www.ibm.com
不包含协议的 URL	使用缺省 http 协议	www.ibm.com

5. 在**目标**中，选择一个选项：
 - 内部（缺省值）：选择此选项可在现有浏览器窗口中打开链接。
 - 外部：选择此选项可在新浏览器窗口中打开链接。
6. 在**已启用**中，选择一个选项：
 - 已启用（缺省值）：选择此选项可启用该选项卡。如果已启用，那么当您选择该选项卡时将尝试连接到该 URL。
 - 已禁用：选择此选项可禁用该选项卡。如果已禁用，那么将隐藏该选项卡。
7. 在**可见**中，选择一个选项：
 - True（缺省值）：选择此选项可在 UI 中显示该选项卡。
 - False：选择此选项可在 UI 中隐藏该选项卡。
8. 在**仅限 Root 用户**中，选择一个选项：
 - True（缺省值）：只允许 root 用户或具有 root 用户访问权的 Build Forge 用户选择该选项卡和连接到该 URL 链接。
 - False：允许任何 Build Forge 用户选择该选项卡和连接到该 URL 链接。

注： 保存新选项卡后，该选项卡标题将出现在列表中。如果选项卡标题没出现，请刷新该页面。

第 25 章 Build Catalyst

Build Catalyst 加速基于 make 的 C 和 C++ 软件构建。它说明并分析现有 make 文件并运行加速的构建。加速的构建可以使用诸如并行构建和分布式构建之类的技术来减少总体构建时间。

Build Catalyst 是 IBM® Rational® Build Forge® 随附的，但是需要额外安装。要将 Build Catalyst 与 Rational Build Forge 结合使用，请将 Build Catalyst 安装在安装了 Rational Build Forge 代理程序的主机中。安装并配置后，Build Catalyst 加速构建可以由 Rational Build Forge 步骤调用。与 Rational Build Forge 项目的集成允许对 Rational Build Forge 提供的更广范围构建自动化工具的访问。例如，调用重构的基于 make 的构建可能是涉及其他构建的更大构建方案的一部分。Rational Build Forge 项目还可以运行预构建和构建后过程。

Build Catalyst 与为 GNU Make 3.80 编写的 makefile 高度兼容。如果为 GNU Make 3.80 编写了 Makefile，那么可以使用 Build Catalyst 来构建您的源代码，而在大多数情况下不需要更改 Makefile。一些情况下可能需要微小的更改。Build Catalyst 支持以下构建：

- **并行构建：**Build Catalyst 可以标识非关联目标并将其并行构建。并行构建将更好地利用构建计算机中的资源，例如多核处理器。Build Catalyst 通过高效使用资源来加速构建并减少构建时间。
- **分布式构建：**分布式构建类似并行构建；但是，构建活动分布于多台计算机之间。

Build Catalyst 文档是针对 Build Catalyst 的新用户和老用户（他们熟悉软件构建概念）。

受支持的操作系统

可以在若干 Linux®、Solaris 和 Microsoft® Windows® 操作系统上使用 Build Catalyst。

这些操作系统支持 Build Catalyst：

- Red Hat Enterprise Linux 4 和 5
- Solaris 9 和 10 (SPARC)
- Microsoft Windows XP Professional
- Microsoft Windows Server 2003

注：只有 Linux 和 Solaris 操作系统支持分布式构建。

安装概述

从归档文件将 Build Catalyst 安装在 Linux®、Solaris 或 Microsoft® Windows® 操作系统上。

有关更多信息，请参阅『受支持的操作系统』。

Build Catalyst 具有独立于 IBM® Rational® Build Forge® 安装程序的安装程序。要将 Build Catalyst 与 Rational Build Forge 结合使用，请将 Build Catalyst 与 Rational Build Forge

代理程序安装在相同计算机上。您可以将 Build Catalyst 安装在计划用作构建主机的任何计算机上，前提是计算机运行受支持的操作系统之一。如果您没有计划拥有在给定计算机上使用 Build Catalyst 的 Rational Build Forge 作业，那么该计算机不必安装 Rational Build Forge 代理程序。

IBM 将 Build Catalyst 分发为包含以下项的归档文件：

- “转储文件”，该文件包含所有可安装文件（二进制文件、各种内置 makefile 等等）
- 安装脚本

在 Linux 和 Solaris 操作系统上安装 Build Catalyst

可以通过使用 Build Catalyst 归档文件在 Linux[®] 和 Solaris 操作系统上安装 Build Catalyst。

关于此任务

要针对分布式构建使用 Build Catalyst，请将其安装在分布式构建中使用的所有计算机上。

过程

1. 从 <http://hostname/buildcatalyst> 下载 Build Catalyst 归档文件，其中 *hostname* 是您的“管理控制台”主机。
2. 将归档文件解压到任何目录。
3. 如果您不是超级用户，通过输入 `su` 来成为超级用户。
4. 转至将安装程序解压到的目录，然后转至包含 `install.sh` 文件的新子目录。
5. 在该目录中运行 `./install.sh`。
 - a. 指定安装目录。（缺省目录为 `/opt/rational/buildforge/buildcatalyst`。）

如果安装已经存在（即使安装在所提供位置以外的位置），安装程序警告如果您继续进行安装，现有安装将被覆盖或表现为不可用。（不在缺省目录中的现有安装可能呈现为不可用，因为当前安装进程从缺省安装目录创建到新指定的安装目录的符号链接。）然后安装程序提示再次输入安装目录。提供安装位置后，安装将文件写入提供的位置。

- b. 指定哪个 `rsh` 或 `ssh` 二进制文件用于分布式构建（如果在路径中找到，缺省程序为 `rsh`。）

Build Catalyst 二进制文件在安装目录中的 `bin` 目录中安装为 `rafmake`。安装将创建以下项：

- 从 `/opt/rational/buildforge/buildcatalyst` 到安装目录的符号链接。
- 安装目录中从 `etc/rafbe_starter` 到 `shell` 二进制文件的符号链接。`shell` 二进制文件是 `rsh` 或 `ssh`。

结果

现在，您可以从 IBM[®] Rational[®] Build Forge[®] 作业启动 `rafmake` 来构建 C 或 C++ 软件。还可以独立启动 Rational Build Forge 作业的 `rafmake`。

在 Windows 上安装 Build Catalyst

可以通过使用 Build Catalyst 归档文件在 Microsoft® Windows® 上安装 Build Catalyst。

开始之前

请验证 Microsoft Visual C++ 2005 SP1 Redistributable Package V8.0.50727.762 是否安装在您的计算机上。您可以通过使用“控制面板”中的**添加或删除程序**来验证安装了哪个软件包。

要点: Build Catalyst 没有使用此软件包的任何其他版本。您必须安装了 V8.0.50727.762。

如果需要，可以从 www.microsoft.com/downloads/ 下载正确的版本。

要获取关于 Windows 支持的更多信息，请参阅第 411 页的『受支持的操作系统』。

过程

1. 从 <http://hostname/buildcatalyst> 下载 Build Catalyst 归档文件，其中 *hostname* 是您的“管理控制台”主机。
2. 将归档文件解压到任何文件夹。
3. 打开该文件夹，然后打开包含 `install.bat` 文件的已创建文件夹。
4. 运行 `install.bat`。已将 Build Catalyst 二进制文件安装为 `rafmake.exe` 文件。在下表中指示保存该文件的 `bin` 文件夹。

表 12. 安装文件夹

操作系统类型	位置
32 位操作系统	C:\Program Files\IBM\Build Forge\buildcatalyst
64 位操作系统	C:\Program Files (x86)\IBM\Build Forge\buildcatalyst

结果

现在，您可以从 IBM® Rational® Build Forge® 作业启动 `rafmake` 来构建 C 或 C++ 软件。还可以独立启动 Rational Build Forge 作业的 `rafmake`。

Build Catalyst 示例

使用这些示例来更好地理解如何将 Build Catalyst 用于简单构建、并行构建和分布式构建。

简单构建

对于简单构建（非并行或分布式的构建），请按如下方式运行 **rafmake** 命令：

```
rafmake [ -f makefile ] [ options ] ... [ targets ] ...
```

其中

- *makefile* 是要使用的 makefile
- *options* 包含第 416 页的『rafmake 实用程序引用』中说明的选项

- *targets* 是要构建的目标文件

以下列表按操作系统提供了示例。

- Linux 和 Solaris
 - 在具有其所有依赖性的特定 *makefile* 中无条件地构建缺省目标:
`% rafmake -u -f project.mk`
 - 在特定 *makefile* 中构建特定目标, 并带有详细输出:
`% rafmake -v -f project.mk a.out`
- Windows
 - 以 *INCL_DIR* 宏设置的特定值在缺省 *makefile* 中构建缺省目标:
`Y:\> rafmake INCL_DIR=C:\src\include`

并行构建

rafmake 命令支持并行构建, 其中将构建分离为独立的组成部分, 这些组成部分可以同时构建来减少总体构建时间。

用于运行并行构建的语法如下:

```
rafmake -J num [ -f makefile ] [ options ] ... [ targets ] ...
```

其中

- *num* 是同时构建独立目标的并行构建作业的数量
- *makefile* 是要使用的 *makefile*
- *options* 包含第 416 页的『**rafmake** 实用程序引用』中说明的选项
- *targets* 是要构建的目标文件

运行并行构建可实现高效的资源利用, 尤其是使用多核处理器的情况下。从并行构建实现的性能提高程度取决于构建的类型。例如, 在 CPU 密集型构建的情况下, 并行构建作业的最佳数量将与处理器 (和核心) 的数量相同。如果指定数量超过此数字, 那么可能无法实现显著的性能提高。但是, 对于 IO 密集型构建, 增加并行作业的数量将减少构建时间。以不同的数量尝试一些构建以确定您资源的并行构建作业的最佳数量。

以下列表按操作系统提供了并行构建的示例。

- Linux 和 Solaris
 - 以三个并行构建作业并行地在特定 *makefile* 中构建缺省目标:
`% rafmake -J 3 -f project.mk`
- Windows
 - 以 *INCL_DIR* 宏设置的值使用五个并行作业在缺省 *makefile* 中并行地构建特定目标:
`Y:\> rafmake -J 5 INCL_DIR=C:\src\include software.exe`

分布式构建

并行构建只能使用一台计算机上的资源。分布构建通过跨多台计算机分布构建负载来将并行构建推进一步。

注：要针对分布式构建使用 Build Catalyst，请将其安装在分布式构建中使用的所有计算机上。

要分布构建，您的环境必须符合以下需求：

- 在不同计算机之间共享源代码。

对于在其上分布构建的所有计算机，必须能够以相同路径访问相同源代码。通常，在 NFS 安装的共享上具有源代码（在所有计算机上将其安装在相同路径中）将确保相同源目录在所有构建计算机上相同路径中可用。

- 通过所选远程 shell 的登录（rsh 或 ssh）配置为不需要密码。

用户从主计算机启动分布式构建时，**rafmake** 命令在要分布构建的所有计算机上从内部启动构建引擎（rafbe）。使用 **ssh** 或 **rsh** 命令来启动构建引擎，该引擎在安装 Build Catalyst 时已配置。`/opt/rational/buildforge/buildcatalyst/etc/rafbe_starter` 文件是安装期间用户选择的远程 shell 的符号链接。启动构建的用户必须无需密码就可以登录使用所选远程 shell 的所有计算机。通过无密码登录，可以在各种计算机上静默启动构建引擎。参考 **rsh** 或 **ssh** 命令手册以设置无需密码的登录。

- 构建主机文件描述如何在计算机上分布构建。

构建主机文件一般包含带有许多计算机的主机名或 IP 地址的若干行。这些计算机是要分布构建的位置。该文件还包含描述其他选项的行，例如 `-idle num`。该类型的行指定某台计算机必须达到何等空闲程度，才能在该计算机下方列出的各计算机上调度构建。如果没有明确提及，空闲度假定为 50。以下是构建主机文件的示例：

```
### Contents of build hosts file
host1.mydomain.com
-idle 25
host2.mydomain.com
host2.mydomain.com
192.168.0.200
-idle 75
host3.mydomain.com
### End of build hosts file
```

在该示例中，将构建分布到四台计算机：host1、host2、host3 和 192.168.0.200。host1.mydomain.com 计算机的空闲度假定为 50，因为没有明确提及该空闲度。对于 host2 和 192.168.0.200 计算机，将空闲度指定为 25。这意味着如果其中一台计算机的负载或忙碌度低于 75%（或空闲度至少为 25%），那么在该计算机上调度构建。如果计算机空闲度低于 25%，那么不会在这些计算机上安排构建。而且，该示例两次列出 host2 计算机。因此，可以在 host2 计算机上启动两个并行构建线程。根据计算机配置，您可以调节构建主机文件以最佳地将资源用于构建。

创建构建主机文件后，您可以通过使用以下命令语法来启动分布式构建：

```
rafmake -J num -B build_hosts_file [ -f makefile ] [ options ] ... [ targets ] ...
```

`-J` 选项的 `num` 值指定所有计算机的构建线程的总数。

不使用 `-B` 选项，**rafmake** 命令将执行并行构建而不是分布式构建。指定 `-B` 选项和有效构建主机文件时，构建引擎在构建主机文件中指定的不同计算机上调度分布式构建。

rafmake 实用程序引用

Build Catalyst 的接口是称为 rafmake 的 make 实用程序。该接口具有控制其行为的各种选项。

命令的语法取决于操作系统，如下表中所示。

表 13. 选项（按操作系统）

操作系统	选项
Linux 和 Solaris（构建目标。）	rafmake [-f <i>makefile</i>] ... [-ukinservwdpqUN] [-J <i>num</i>] [-B <i>bldhost-file</i>] [-c <i>compat-mode</i>] [-C <i>change-directory</i>] [-A <i>BOS-file</i>] ... [<i>macro=value</i> ...] [<i>target-name</i> ...]
Windows（构建目标。）	rafmake [-f <i>makefile</i>] ... [-ukinservwdpqUN] [-J <i>num</i>] [-c <i>compat-mode</i>] [-C <i>change-directory</i>] [-A <i>BOS-file</i>] ... [<i>macro=value</i> ...] [<i>target-name</i> ...]
所有（显示 rafmake 实用程序的版本信息。）	rafmake { -ver/sion -VerAll }

选项描述

rafmake 命令支持 GNU V3.80 make 命令的大部分公共选项。**rafmake** 命令还提供附加选项。

提示：您可以合并不带参数的选项，例如 -rNi。

表 14. 选项描述

选项	描述
-f <i>makefile</i>	将 <i>makefile</i> 用作输入文件。如果您省略此选项， rafmake 命令将在当前工作目录中查找名为 <i>makefile</i> 和 <i>Makefile</i> （以该顺序）的输入文件。可以指定多个 -f <i>makefile</i> 参数对。将有效并置多个输入文件。
-u	（无条件）重新构建所有指定的目标及其所有依赖性，无论是否需要重新构建。（另请参阅 -U。）
-k	如果当前条目上的工作失败，请将其放弃，但是在不依赖于该条目的其他目标上继续处理。
-i	忽略命令返回的错误代码。
-n	（不执行）从需要重新构建的目标的 <i>makefile</i> 列出命令行（包括以 at 符号 (@) 开头的命令行），但是不运行这些命令行。异常：始终运行包含字符串 \$(MAKE) 的命令。
-s	（静默）在运行命令行之前请勿将其列出。
-e	环境变量覆盖 <i>makefile</i> 文件中的宏分配。（但是，命令行或构建选项规范中的宏=值分配将覆盖环境变量。）
-r	（无规则）请勿使用 buildcatalyst-home-dir/etc/builtin.mk 文件（Linux 和 Solaris）或 buildcatalyst-home-dir\etc\builtin.mk 文件（Windows）中的内置规则。与 -c 选项一起使用时，-r 选项也禁止读取特定于平台的启动文件。请参阅 -c 选项以了解更多信息。
-v	（详细）比缺省输出方式稍微更详细些。
-w	（工作目录）在运行 <i>makefile</i> 之前和之后打印包含工作目录的消息。

表 14. 选项描述 (续)

选项	描述
-d	(调试) 相当详细, 并包含构建期间 rafmake 命令读取的环境变量的列表。仅在调试 makefile 时使用此选项。
-p	(打印) 列出所有目标描述和所有宏定义, 包括特定于目标的宏定义和隐式规则。请勿运行任何对象。
-q	(查询) 评估 makefile 目标, 但是为运行构建脚本。 rafmake 命令返回以下响应之一: <ul style="list-style-type: none"> • 如果目标为最新, 那么返回 0 • 如果需要重新构建任何目标, 那么返回 1
-U	仅无条件地构建目的目标。请勿构建子目标。如果没有在命令行上指定任何目标, 那么缺省目标为目的。(-u 选项无条件地构建目的目标和构建依赖性。)
-N	禁用用于读取一个或多个构建选项规范 (BOS) 文件的缺省过程。
-J <i>num</i>	启用并行构建能力。并行目标重新构建的最大数量设置为整数 <i>num</i> 。如果 <i>num</i> =0, 那么禁用并行构建。(这等价于不指定 -J 选项。)或者, 您可以将 <i>num</i> 指定为 RAFMAKE_CONC 环境变量的值。
-B <i>bldhost-file</i>	将 <i>bldhost-file</i> 文件用作并行构建的构建主机文件。如果没有指定 -B 选项, 那么 rafmake 命令使用主目录中的 .bldhost.\$RAFMAKE_HOST_TYPE 文件。使用 -B 选项时, 您还必须使用 -J 选项或设置 RAFMAKE_CONC 环境变量。
-c <i>compat-mode</i>	<p>(兼容性) 以兼容性方式调用 rafmake 命令。</p> <p>(或者, 可以使用 RAFMAKE_COMPAT 环境变量来指定兼容性方式。)</p> <p>您可以将 <i>compat-mode</i> 变量设置为以下值之一:</p> <p>gnu</p> <p>模拟“自由软件基金会”的 Gnu make 程序。要定义内置 make 规则, rafmake 命令读取 gnubuiltin.mk 而不是 builtin.mk。</p> <p>该方式是缺省兼容性方式。</p> <p>std 在启用 IBM Rational® ClearCase® clearmake 命令兼容性方式的情况下, 运行 rafmake 命令。</p> <p>仅在 Linux 和 Solaris 系统上, <i>compat-mode</i> 变量还可以具有以下值之一。-c 选项独立于 Linux 和 Solaris 平台。但是, 一些方式尝试读取特定于系统的文件; 如果这些文件不存在, 那么命令失败。</p> <p>sun 模拟 SunOS 系统提供的标准 make(1) 命令。</p> <p>aix 模拟 IBM AIX® 系统提供的标准 make(1) 命令。</p>
-C <i>dir</i>	更改目录。 rafmake 命令切换到 <i>dir</i> 目录, 然后启动构建。

表 14. 选项描述 (续)

选项	描述
<code>-A BOS-file</code>	<p>指定在缺省读取的 BOS 文件后立即读取的 BOS 文件。多次使用该选项来指定多个 BOS 文件。</p> <p>将 <code>-N</code> 与该选项一起使用时, rafmake 读取指定的 BOS 文件, 而不是缺省的 BOS 文件。</p> <p>或者, 您可以指定 BOS 文件路径名的冒号分隔的列表 (Linux 和 Solaris) 或者此类路径名的分号分隔的列表作为环境变量 <code>RAFMAKE_OPTS_SPECS</code> 的值。</p>
<code>-version</code>	打印关于 rafmake 命令的版本信息。
<code>-VerAll</code>	打印关于 rafmake 命令以及 rafmake 命令使用的库 (Linux 和 Solaris) 或 DLL (Windows) 的版本信息。

Build Catalyst 环境变量

Build Catalyst 支持若干环境变量来简化用法。

在以下部分中这些变量是按操作系统分组的:

- 『Linux、Solaris 和 Microsoft Windows 的公共环境变量』
- 第 419 页的『仅限于 Linux 和 Solaris 的环境变量』
- 第 420 页的『仅限于 Windows 的环境变量』

Linux、Solaris 和 Microsoft Windows 的公共环境变量

RAFMAKE_CONC

设置 **rafmake** 构建中的并行级别。该变量采用与 `-J` 选项相同的值。使用 **rafmake** 命令时指定 `-J` 选项将覆盖该变量的设置。

RAFMAKE_COMPAT

指定 **rafmake** 命令兼容性方式。该变量采用与 `-c` 选项相同的值。使用 **rafmake** 命令时指定 `-c` 选项将覆盖该变量的设置。

缺省值: 无

RAFMAKE_MAKEFLAGS

提供用于指定 **rafmake** 命令选项的备用或补充机制。RAFMAKE_MAKEFLAGS 环境变量可以包含用于命令行选项 (除了使用无效参数的选项) 的相同键字母字符串。如果存在冲突, 那么在 **rafmake** 命令行上指定的选项将覆盖此环境变量的设置。

缺省值: 无

RAFMAKE_OPTS_SPECS

提供路径名的列表, 以冒号 (Linux 和 Solaris) 或分号 (Windows) 分隔, 每个路径名指定 **rafmake** 命令读取的一个构建选项规范 (BOS) 文件。可以使用此变量, 而不是通过一个和多个 `-A` 选项在 **rafmake** 命令行上指定 BOS 文件。

缺省值: 未定义

RAFMAKE_SHELL_FLAGS

指定 **rafmake** 命令选项来传递到运行构建脚本命令的子 shell 程序。

缺省值:

- Linux 和 Solaris: `-e`
- Windows: 无

RAFMAKE_SHELL_REQUIRED

强制执行 **rafmake** 命令以在通过 SHELL 宏指定的 shell 程序中运行构建脚本。要使 **rafmake** 命令在 shell 程序中运行构建脚本, 请将此变量设置为 TRUE。要配置 **rafmake** 命令来直接运行构建脚本, 请将变量取消设置。

缺省值: **rafmake** 命令直接运行构建脚本

RAFMAKE_VERBOSITY

以如下方式指定 **rafmake** 命令消息记录级别:

1 在命令行上等价于 `-v` (详细)

2 在命令行上等价于 `-d` (调试)

0 或未定义

等价于标准消息记录级别

如果在命令行上也指定 `-v` 或 `-d`, 那么以较大值为准。

缺省值: 0

MAKEFLAGS

列出要传递到 **make** 命令的一个或多个标志。 **rafmake** 命令在启动时读取 MAKEFLAGS 环境变量的内容。然后该命令修补变量以包含在命令行上传递的未特定于 Build Catalyst 的标志。通过 RAFMAKE_MAKEFLAGS 环境变量传递特定于 Build Catalyst 的标志, 如果 **rafmake** 命令检测到 MAKEFLAGS 变量中的这些标志, 那么该命令将其移到 RAFMAKE_MAKEFLAGS 变量。

通过 MAKEFLAGS 变量传递的标志如下: `-I`、`-p`、`-N`、`-w`、`-e`、`-r`、`-i`、`-k`、`-n`、`-q` 和 `-s`

通过 RAFMAKE_MAKEFLAGS 变量传递的标志如下: `-A`、`-B`、`-N`、`-v`、`-c`、`-U`、`-M`、`-u` 和 `-d`

缺省值: 无

仅限于 Linux 和 Solaris 的环境变量

RAFMAKE_BLD_HOSTS

指定在其上 **rafmake** 命令可以构建目标的一个和多个构建主机。

缺省值: 未定义

RAFMAKE_HOST_TYPE

确定并行构建期间要使用的构建主机文件的名称 (`-J` 选项): 主目录中的 `.bldhost.$RAFMAKE_HOST_TYPE`。(通过检查密码数据库来确定您的主目录。)

在命令行上指定 `-B` 选项将覆盖此变量的设置。

C Shell 用户: 在 `.cshrc` 文件 (而不是 `.login` 文件) 中设置此变量。并行构建设施调用远程 shell, 该 shell 不会读取 `.login` 文件。

您还可以将 RAFMAKE_HOST_TYPE 编码为 **make** 宏。

缺省值: 无

仅限于 Windows 的环境变量

RAFMAKE_NO_ESC_PATT_CHARS

以 **rafmake** 命令的 GNU 兼容方式覆盖转义字符 (\)。例如, **rafmake** 和 GNU **make** 命令假定 \% 字符串指示字面值字符 %。这些命令不会将该规则视为模式规则。要阻止 **rafmake** 命令使用转义字符来指示字面值字符, 请将此环境变量设置为任何非空值。

RAFMAKE_PNAME_SEP

为 **rafmake** 命令构造的路径名设置路径名分隔符。该变量可以设置在 makefile 中、在 BOS 文件中、在命令行上或设置为环境变量。

缺省值: 如果该变量未设置或者设置为正斜杠 (/) 或反斜杠 (\) 以外的任何值, 那么 **rafmake** 命令将反斜杠 (\) 用作路径名分隔符。

第 26 章 Rational Build Forge 集成方法

Build Forge 使用以下方法与其他产品集成:

- 使用命令行进行的集成。Rational Build Forge 步骤在代理程序主机上运行命令。
- 使用适配器进行的集成, 它是基于 XML 的程序, 具有特殊功能用于帮助与源控制管理系统集成。
- 使用 IDE 插件进行的集成, 它允许您通过 IDE 用户界面访问控制台。
- 与 Websphere 产品进行的集成。可以使用 Websphere Application Server 代替 Apache Tomcat 作为 Web 应用程序服务器。可以使用 IBM HTTP Server 代替 Apache HTTP Server。

使用命令行界面集成

Build Forge 可轻松与具有命令行界面的应用程序集成。

要设置这种类型的集成, 您需要执行以下操作:

1. 在主机上安装 Build Forge 控制台
2. 在应用程序主机 (或可以访问该应用程序的主机) 上安装 Build Forge 代理程序
3. 在控制台中, 创建服务器资源和服务器认证。配置服务器资源以访问您已安装的 Build Forge 代理程序。
4. 配置要在应用程序上运行命令所需的代理程序和主机环境。这可能非常简单, 就像创建用户帐户供 Build Forge 使用一样, 然后保证正在运行的代理程序正确设置了其路径。它可能还要求您安装并配置用于在应用程序中运行命令的客户机。例如, Rational ClearCase 和 ClearQuest 要求使用客户机应用程序才能运行命令。

设置完成后, 在 Build Forge 中创建的项目可能包含运行应用程序命令的步骤。您可以根据步骤的通过或失败状态控制项目执行, 或设置日志过滤器以在 Build Forge 日志中扫描命令返回的输出的模式。应用程序可能需要在项目逻辑中进行其他设置。例如, 与 ClearCase 集成将要求包括视图的创建、启动和填充。

您可以使用条件步骤和循环步骤来控制发生的情况, 以响应特定命令的成功或失败。

源控制应用程序的典型用途是避免构建。将运行命令以查询自上次运行构建以来是否已检入源代码更新:

- 如果没有更新, 那么不会运行构建。
- 如果已有更新, 那么会运行构建

模块依赖性可以通过项目步骤的执行逻辑来表示。您可以控制特定模块的构建是否触发其他模块的构建或整个软件项目。

使用通知模板以控制如何通知项目成员组以响应构建成功或失败。在连续集成环境中, 针对失败进行通知很重要, 这样可以修订有问题的代码。使用命令行集成时, 可以只通知整个项目成员组。项目成员将需要检查项目的步骤日志以确定哪些代码导致失败。

使用适配器进行集成

可以使用适配器将 Rational Build Forge 与外部应用程序进行集成。

对于 IBM Rational ClearCase 和 IBM Rational ClearQuest 集成，提供了样本适配器模板。适配器工具包是一个单独许可的功能，您可用来编写定制适配器。

适配器的行为由 XML 文件来定义，并且该 XML 文件的元素由包含的 DTD 来指定。Build Forge 执行与项目步骤相关联的适配器。

适配器为您提供了其他工具来进行基于集成的构建：

- 内部条件逻辑：基于内部变量的设置
- 命令定义：可以通过命令语句和变量来构建应用程序命令。
- 响应扫描：您可以定义模式以扫描对每个命令的响应。
- 动态通知组：通知可以基于从应用程序收集的数据。例如，可以仅通过检入代码更改的成员来构建要通知的团队成員组。
- 脚本执行：适配器可以在 Build Forge 主机上运行脚本。这可独立于应用程序主机上运行的脚本和命令。

适配器需求

本主题描述 Build Forge 适配器的安装与配置需求以及软件需求。

请验证您的环境是否满足使用 Build Forge 产品随附的任何适配器的需求：

- 在运行外部应用程序的计算机上安装 Build Forge 代理程序。某些产品具有其他安装需求。Rational Clear Case 和 Rational ClearQuest 需要其他步骤。请参阅这些产品的“集成”部分中的信息。
- 如果希望使用 ClearCase 或 ClearQuest 之外的应用程序模板，请安装 Build Forge 适配器工具包的许可证密钥。
- 运行 Build Forge 产品支持的外部应用程序版本。

表 15. 适配器支持的应用程序版本

应用程序	版本
Rational ClearCase	6.0 或更高版本
Rational ClearQuest	7.0 或更高版本
CVS	1.1, 1.2
Microsoft Visual SourceSafe	6.0
Perforce	2005.1
StarTeam	2005 发行版 2
Subversion	1.3.1 或更高版本

适配器

本主题提供有关适配器的一些一般信息。还描述了适配器如何与 Build Forge 对象和功能交互。

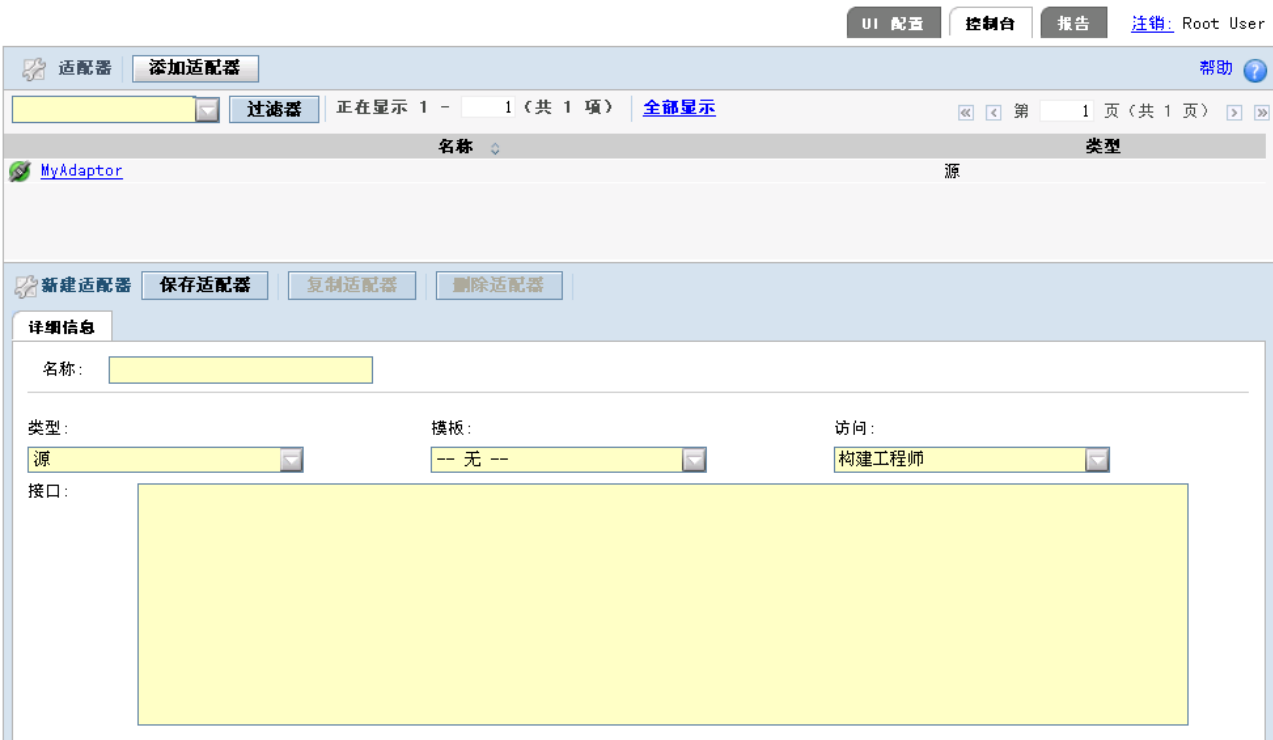
要着手使用适配器，请先阅读本部分中提供的信息。

关于适配器

适配器是到外部应用程序的接口。Build Forge 项目可通过适配器与外部应用程序交换信息来达到某个目的。

例如，源代码应用程序的适配器检查存储库以查找源代码更改，这是运行 Build Forge 项目的先决条件。如果源代码文件已更改，项目将运行。如果未更改，那么项目不运行。

关于“适配器”面板： 可以使用“适配器”面板来创建和编辑适配器。要查看该面板，请选择项目 > 适配器。



该面板具有以下字段：

- 名称** 适配器的唯一名称，此名称不包含空格
- 类型** 适配器类型
- 模板** Rational Build Forge 中无适配器模板或随附了一个适配器模板
注： 更新 Rational Build Forge 版本可清除此设置。
- 访问** 用于将针对适配器的查看和编辑访问权限于组成员的访问组
- 接口** 您或模板提供的 XML 接口

适配器模板

适配器是适配器模板的一个实例。创建适配器时，您将为其指定一个唯一的名称并将其与模板关联。

模板是一个 XML 文件。这个 XML 文件包含用于收集信息的应用程序命令、用于分析信息的指示信息和用于在 BOM 报告中显示结果的格式详细信息。

Build Forge 提供的模板不经修改即可使用。但是，您可以修改模板或将其用作新建适配器模板的模型。

适配器模板安装在以下目录中：

bfinstall\interface（针对运行 Microsoft Windows 操作系统的计算机）

bfinstall/Platform/interface（针对运行 UNIX 或 Linux 操作系统的计算机）

样本适配器模板

针对以下几种应用程序提供了样本适配器模板：源代码、缺陷跟踪、测试和打包。适配器按照支持的应用程序类型分类。

Build Forge 产品为应用程序提供了下表中列出的样本适配器模板。ClearCase 和 ClearQuest 的模板无需单独的许可证密钥。其他应用程序模板通过 Build Forge 适配器工具包进行许可。

适配器模板 XML 文件安装在以下目录中，其文件扩展名为 .xml。

bfinstall\interface（针对运行 Microsoft Windows 操作系统的计算机）

bfinstall/Platform/interface（针对运行 UNIX 或 Linux 操作系统的计算机）

注：

如果使用通过适配器链接连接的适配器运行项目，该适配器未处于调试方式并且适配器步骤失败，那么该适配器将除去作业。

表 16. 适配器模板描述

适配器模板名称	描述	类型
ClearCaseBaseline	扫描 ClearCase 视图中的目录。 将 ClearCase 报告的分支和版本信息写入 BOM 报告。	源
ClearCaseByBaselineActivities	从 ClearCase 视图的内容创建新的基线。 将新基线与来自上次执行适配器的基线进行比较以识别更改活动。 为每个更改活动将以下信息写入 BOM 报告：活动、已更改文件、用户、日期、注释和版本。 为更改的每个文件将更改详细信息（来自 diff 命令输出）写入 BOM 报告。	源
ClearCaseByBaselineVersions	从 ClearCase 视图的内容创建新的基线。 将新基线与来自上次执行适配器的基线进行比较以识别已更改文件。 为每个已更改文件将以下信息写入 BOM 报告：文件名、版本、日期、用户和注释。 为更改的每个文件将更改详细信息（来自 diff 命令输出）写入 BOM 报告。	源

表 16. 适配器模板描述 (续)

适配器模板名称	描述	类型
ClearCaseByDate	<p>查询 ClearCase 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。</p> <p>为每个已更改文件将以下信息写入 BOM 报告: 文件名、版本、日期、用户和注释。</p> <p>为更改的每个文件将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源
ClearCaseByLabel	<p>新建标签并将其应用于 ClearCase 视图的内容。</p> <p>将新标签与来自上次执行适配器的标签进行比较以识别已更改文件。</p> <p>为每个已更改文件将以下信息写入 BOM 报告: 文件名、版本、日期、用户和注释。</p> <p>为更改的每个文件将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源
ClearQuestBaseClearCaseByDate	<p>查询 ClearCase 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。</p> <p>为每个已更改文件查找识别 ClearQuest 更改标识的 ClearQuest 超链接属性。通过添加作业信息以解析 ClearQuest 中的缺陷记录来尝试解析更改标识, 前提是 ClearQuest 状态允许解析这些缺陷记录。</p> <p>为每个已更改文件将以下信息写入 BOM 报告: 文件名、缺陷标识、缺陷状态和任何 ClearQuest 错误。</p>	缺陷
ClearQuestClearCaseByActivity	<p>查找与 ClearCase 活动列表关联的 ClearQuest 缺陷记录。它为发现的每个缺陷记录添加作业信息以解析 ClearQuest 内的缺陷记录, 前提是 ClearQuest 状态允许解析这些缺陷记录。</p> <p>将以下信息写入 BOM 报告: 与 ClearCase 活动标识和 ClearQuest 缺陷状态关联的文件。</p>	缺陷
ClearQuestUCMClearCaseByDate	<p>查询 ClearCase 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。</p> <p>使用 Rational 统一变更管理 (UCM) 来生成结果。</p> <p>为每个已更改文件将以下信息写入 BOM 报告: 文件名、缺陷标识、缺陷状态和任何 ClearQuest 错误。</p>	缺陷
CVSv1Baseline	<p>扫描 Build Forge 代理程序上的 CVS 目录以查找已更改文件。</p> <p>将以下信息写入 BOM 报告: 已更改文件名、状态、工作版本、存储库版本和固定标记。</p>	源

表 16. 适配器模板描述 (续)

适配器模板名称	描述	类型
CVSv1ByDate	<p>查询 CVS 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。</p> <p>将以下信息写入 BOM 报告: 更改类型、日期、用户名、版本和文件名。</p> <p>为更改的每个文件将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源
CVSv1ByTag	<p>将新标记应用于 CVS 模块。</p> <p>比较新标记的模块与上次执行适配器期间标记的模块之间的差别。</p> <p>将以下信息写入 BOM 报告: 文件名、修订、状态、日期、时间、更改作者和落实注释。</p> <p>为更改的每个文件将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源
CVSv2ByDate	<p>查询 CVS 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。</p> <p>将以下信息写入 BOM 报告: 更改类型、日期、用户名、版本和文件名。</p> <p>为更改的每个文件将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源
JazzJBE	<p>针对 Rational Team Concert 3.x 及更高版本的用户。要了解相关信息, 请参阅 Rational Team Concert 文档。</p>	源
JazzSCM	<p>查询 Rational Team Concert 中是否有更改过的源文件。</p> <p>会将以下信息写入 BOM 报告: 更改的源文件、已更改文件的大小、更改的组件以及更改集。</p>	源
JazzSCMv2	<p>查询 Rational Team Concert V2.x, 以查找更改过的源文件。</p> <p>以下信息将写入 BOM 报告: 更改的源文件、更改的流、更改的组件、变更集、执行更改的用户、检入注释和 Jazz 工作项。</p>	源
JUnit	<p>运行 JUnit 测试。</p> <p>以下信息将写入 BOM 报告: 运行的测试、故障、错误和耗用时间。</p>	测试
PerforceByDate	<p>查询 Perforce 客户机以查找执行适配器以来发生的更改。</p> <p>将以下信息写入 BOM 报告: 更改、日期、时间、用户、Perforce 客户机和注释。</p> <p>将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源

表 16. 适配器模板描述 (续)

适配器模板名称	描述	类型
PerforceByRev	<p>查询 Perforce 客户机以查找上次存储库修订以来发生的更改。</p> <p>将以下信息写入 BOM 报告: 更改、日期、时间、用户、Perforce 客户机和注释。</p> <p>将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源
Quota	<p>查询 Windows 文件夹以确定它的任何子文件夹是否超过了指定的阈值大小。</p> <p>为每个子文件夹将以下信息写入 BOM 报告: 文件夹大小、所有者和上次修改日期。</p> <p>将 BOM 报告写入超出了阈值大小的子文件夹列表。</p> <p>要点: 如果目录超出阈值, 那么会按以下顺序向作为该目录所有者的用户发送通知: 首先按 Build Forge 系统中的名称; 然后按用户名。如果通知的用户不存在, 那么 <notify> 将失败。</p>	源
StarTeamBaseline	<p>查询文件夹以查找 StarTeam 视图来收集有关文件的信息。</p> <p>将以下信息写入 BOM 报告: 文件名、状态、修订和分支。</p>	源
StarTeamByDate	<p>使用 StarTeam API 查询 StarTeam 视图以识别当前日期与上次执行适配器之间的更改。</p> <p>将以下信息写入 BOM 报告: 已更改的文件与目录、用户、版本、日期和更改注释。</p> <p>将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源
SubversionByDate	<p>查询 Subversion 以查找过去的日期与当前日期之间发生的存储库更改。</p> <p>将以下信息写入 BOM 报告: 更改类型、修订、用户、文件或目录和更改日期。</p> <p>将以下信息写入 BOM 报告: 文件名、状态、修订和分支。</p>	源
SubversionByRev	<p>查询 Subversion 以查找当前修订与早期修订之间对存储库的更改。</p> <p>为每个更改将以下信息写入 BOM 报告: 修订、用户、更改类型、文件或目录路径和更改日期。</p> <p>将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源
SynergyByDate	<p>查询 Rational Synergy 以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。</p> <p>为每个已更改文件将以下信息写入 BOM 报告: 文件名、版本、用户、项目和日期。</p> <p>为更改的每个文件将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。</p>	源

表 16. 适配器模板描述 (续)

适配器模板名称	描述	类型
VSSByDate	查询 Visual Source Safe 目录以查找早期日期与当前日期之间的更改。 将项目和文件的更改信息写入 BOM 报告: 项目或文件、版本、用户、日期、时间、项目活动、文件项目 and 操作。 将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。	源

适配器与项目

要运行适配器代码并与外部应用程序交互，必须将适配器添加到 Build Forge 项目。

请使用点命令或适配器链接将适配器添加到项目。

任何适配器都可以使用适合应用程序类型的以下点命令添加至项目：
.source、.defect、.test 或 .pack。

只有源代码适配器可以添加到带有适配器链接的项目。使用适配器链接，而不是 .source 命令将该适配器添加到项目。

适配器与环境变量

适配器需要环境变量才能运行应用程序命令。在适配器模板中，XML 文件的 <template> 节的 <env> 元素中列出了环境变量。

例如，对于 ClearCaseBaseline 适配器，ClearCaseBaseline.xml 文件中列出了下列环境变量：

```
<template>
<!-- Template section, these are parsed out of the final xml.
Use the list below to help identify the variables needed to run this interface
if you are integrating it during a regular BuildForge step.
-->
<env name="VIEW" value="my_adaptor_view" />
<env name="VOB_PATH" value="\AdaptorVob" />
<env name="CCSERVER" value="BFServerName" /></template>
```

在 Build Forge 中，环境变量存储在环境内。创建适配器之前，请为应用程序环境变量创建一个环境。

适配器与通知

大多数适配器模板都会向用户发送电子邮件通知。例如，ClearCaseByDate 适配器在运行时将向更改了源代码文件的用户发送已通过电子邮件通知。如果未更改文件，那么它将发送失败电子邮件通知。

可以选择为适配器修改通知：

- 在适配器模板中，复制 <adduser> 元素以将用户添加到适配器通知组。
- 在适配器模板中，使用 <notify> 元素来添加或删除通知消息。
- 为适配器项目设置项目级别的通知。
- 为适配器点命令步骤设置步骤级别的通知。

要点: 如果 <notify> 伪指令失败（例如，作为电子邮件收件人的用户不存在），那么 XML 也将失败，并且所有的后续通知都将失败。

适配器与作业执行

可以调度作业的形式或使用项目的任何手动启动选项来启动使用适配器点命令的适配器项目。

适配器链接的项目通常按调度运行。但是，如果完成了某些额外的设置，那么可以手动启动适配器链接的项目。请参阅第 437 页的『手动启动适配器链接的项目』。

适配器与作业结果

对于适配器点命令项目，请按照下面的步骤在步骤日志或 BOM 报告中查看作业的结果：

- 选择**作业 > 已完成**。选择要查看其步骤日志的作业的标记。
- 选择**作业 > 已完成**。选择 BOM 选项卡以显示 BOM 报告并按类别查看作业结果。

对于适配器链接的项目，请按照下面的步骤在 BOM 报告的“源更改”类别中查看作业结果：

- 选择**作业 > 已完成**。选择 BOM 选项卡。在 BOM 报告中，找到“源更改”类别。

关于适配器链接

适配器链接用于将适配器连接到项目并将环境与适配器关联。

适配器链接只能处理源代码适配器。可以使用点命令将任何适配器类型连接到项目。

适配器链接具有下列特征：

- 将适配器代码添加到项目以作为步骤 0（在运行其他项目步骤之前测试源代码更改）。
- 自动使用应用程序环境变量填充环境。
- 用于通过选择状态（活动、不活动和调试）来控制是否运行适配器代码。
- 用于以手动启动的作业的形式运行适配器。

适配器链接具有以下限制：适配器链接仅针对一个适配器和一个项目定义。要将同一个适配器与另一个项目一起使用，必须创建该适配器的另一个实例。

在重新启动作业时，作业中任何适配器链接步骤的重新启动行为如下所述：

- 如果适配器链接的状态为不活动，那么适配器链接步骤失败并且导致重新启动运行失败。
- 如果适配器链接的状态为活动并且链接失败，那么适配器链接步骤失败并且导致重新启动运行失败。

在以上两种情况中，都不会除去构建。但是，构建状态会设置为“失败”。

关于“适配器链接”面板： 可以使用“适配器链接”面板将适配器连接到项目，以及将环境与适配器关联。要查看该面板，请选择**项目 > 适配器链接**。



该面板具有以下字段:

状态 “活动”、“调试”或“不活动”之一

注: 这些状态仅适用于一个适配器链接。您可以通过使用“链接调试方式”系统设置来将所有适配器链接设置为“调试”状态。该状态优先于系统配置设置。

适配器 链接到项目的适配器

入口点 XML 接口的入口点

项目 要使用该适配器的项目

环境 供项目使用的环境

适配器任务概述

本主题涵盖创建源代码适配器，使用适配器链接将其连接到项目和以测试方式运行适配器链接的项目所需的全部任务。

通过选择模板创建适配器

要通过选择模板创建适配器，请执行以下操作:

1. 选择项目 → 适配器。
2. 单击**添加适配器**。
3. 在“名称”字段中，为适配器输入唯一名称。适配器名称在整个适配器集合中必须唯一并且不能包含空格。
4. 在“类型”部分，选择适配器类型。
5. 在“模板”部分，选择模板。此列表包含随 Build Forge 产品安装的适配器模板。ClearCase 和 ClearQuest 适配器无需单独的许可证密钥。其他适配器则需要通过适配器工具包 (Adaptor Toolkit) 单独进行许可。
6. 在“访问”部分，选择一个访问组。只有这些组成员才能查看或编辑该适配器。
7. 单击**保存适配器**。

创建空环境

要创建空环境，请执行以下操作：

1. 选择项目 → 环境。
2. 单击添加环境。
3. 在“名称”字段中，输入环境名称。指定一个描述环境的用途的名称。
4. 在“访问”部分，选择一个访问组。只有这些组成员才能查看或编辑该环境。
5. 单击保存环境。


将适配器添加到项目

要将适配器添加到项目，请执行以下操作：

1. 选择项目 → 适配器链接。
2. 单击添加适配器链接。
3. 在“适配器”中，选择创建的适配器（以及适配器模板）。
4. 在“项目”部分，选中此项目。此列表显示尚未链接到适配器的项目。
5. 在“状态”中，选择活动。
6. 在“环境”中，选择为适配器链接创建的空环境。
7. 在“填充环境”部分，选择是。适配器模板中的应用程序环境变量已添加到环境中。
8. 单击保存可将适配器链接到项目。适配器和项目将添加到适配器链接列表中。

编辑环境变量

要编辑环境变量，请执行以下操作：

1. 选择环境。
2. 对于所创建的环境，单击编辑图标 。该面板会显示自动添加到环境中的适配器环境变量。
3. 复审适配器模板提供的环境变量的缺省值。
4. 根据需要更改您的源代码应用程序的缺省值，以便运行适配器项目。

条件属性

condition 属性允许通过使用 Perl 比较运算符将条件应用于某些适配器属性。字符串文字、数字或变量都可以用于比较。

condition 属性的语法为：

```
condition="true(<lvalue> <operator> <rvalue>)"
condition="false(<lvalue> <operator> <rvalue>)"
condition="hastext(variable)"
condition="isempty(variable)"
```

指定以下四种类型之一：

类型	描述
true	如果 lvalue 和 rvalue 相等，那么求值成功。
false	如果 lvalue 和 rvalue 不相等，那么求值成功。
hastext	如果值长度大于 0，那么求值成功。

类型	描述
isempty	如果值长度为 0，那么求值成功。

lvalue 和 rvalue 可以是字符串、数字或包含字符串或数字的变量。条件运算符是任何 Perl 兼容的条件运算符。包括字符串条件运算符和数字条件运算符。您必须使用恰当的运算符，否则您会接收到不可预测的结果。

字符串运算符	数字运算符
eq	==
ne	!=
gt	>
lt	<
ge	>=
le	<=

对字符串使用数字运算符不会返回正确的结果，对数值使用字符串比较运算符也是如此。

例如: `condition="true("PASS"=="FAIL")`

上面的条件总是返回 `true`，这是错误的。

条件示例:

- `condition="true($BF_SERVER eq "TEST_BOX")` - 仅当构建服务器变量包含 `TEST_BOX` 时，才运行该项。
- `condition="false($BF_BID <=141)` - 仅当构建标记大于 141 或不小于 141 时，才运行该构建。

支持条件的适配器属性:

- adduser
- bom
- run
- setenv

仔细检查 Build Forge 当前安装的 DTD 中是否有关于哪些属性支持该条件属性的最新信息。

对于支持条件的属性，会看到类似于下面的条目:

```
<!ATTLIST adduser condition CDATA #IMPLIED>
```

适配器 DTD 作为 `interface.dtd` 文件位于 `%BF_HOME%\interface`（针对 Windows）和 `$BF_HOME/Platform/interface`（针对 UNIX）中。

以测试方式运行适配器

要以测试方式运行适配器:

1. 选择管理 → 系统。

2. 在系统配置参数列表中，选择**链接调试方式**。
3. 在“链接调试方式”中，选择**是**。
4. 单击**保存**。
5. 选择**作业 → 启动**。
6. 在项目列表中，从“启动项目”页面选择创建的，适配器链接的项目。
7. 单击**执行**。

查看作业状态和日志

要查看适配器项目的作业状态和日志信息：

1. 打开**作业**。
2. 在项目列表中找到适配器链接的项目以查看作业的通过/失败状态。
3. 要查看作业日志：
 - 选择适配器项目的“标记名称”以访问作业日志信息。
 - 选择“材料清单”以访问 BOM 报告。

核心适配器任务

本主题提供有关创建和配置适配器的信息：

- **选择模板**：描述每个适配器应用程序模板。
- **创建环境**：描述用于将环境与适配器相关联的选项。
- **创建适配器**：描述如何创建适配器并将其与模板关联。
- **将适配器添加到项目**：描述用于将适配器添加到项目的选项。
- **测试适配器**：仅描述如何测试适配器配置。

本主题还提供有关其他适配器任务的信息：

- **设置适配器日志级别**：描述如何控制为适配器记录的信息量。
- **快速启动适配器链接的项目**：描述手动启动所需的设置。
- **重置适配器模板**：描述何时需要重置来更新模板信息。

为适配器创建环境

适配器需要环境变量才能运行应用程序命令。在适配器模板中，XML 文件的 `<template>` 节的 `<env>` 元素中列出了环境变量。

切勿编辑适配器模板文件中的环境变量。在 Build Forge 产品中，请使用环境定义适配器的环境变量。

使用现有环境或创建一个适配器专用的环境。建议创建一个环境，这样可以为该环境指定特定用途和描述性名称；这样还可以简化故障诊断。

本部分描述如何将环境与适配器关联：

- 通过适配器链接
- 在与适配器点命令关联的步骤中

为适配器链接创建环境： 如果要使用适配器链接将源代码适配器链接到项目，请使用此方法。

1. 选择项目 > 环境。
2. 单击添加环境。
3. 在“名称”字段中，输入环境名称。指定一个描述环境的用途的名称。
4. 在“访问”部分，选择一个访问组。只有组成员才能查看或编辑此环境。
5. 单击保存环境。切勿在此时向该组添加环境变量。这些环境变量在您创建适配器链接时自动从适配器模板填充。

为适配器点命令创建环境： 如果正在使用适配器点命令将适配器添加到项目，请使用此方法。

对于此任务，需要针对您的适配器的环境变量。

在 `<adaptor_name>.xml` 文件中，`<template>/<env>` 元素内列出了外部环境变量。

在以下目录中查找适配器模板：

`<bfinstall>\interface`

1. 选择项目 > 环境。
2. 单击添加环境。
3. 在“名称”字段中，输入环境名称。指定一个描述环境的用途的名称。
4. 在“访问”部分，选择一个访问组。只有组成员才能查看或编辑此环境。
5. 单击保存环境。
6. 单击添加环境变量。
7. 在“名称”中，输入在 XML `<env>` 元素中显示的环境变量名称。
8. 在“值”中，将 XML `<env>` 元素中的替换变量更改为您应用程序的实际值。（如果不知道正确的值，可以稍后输入。）
9. 在“操作”中，选择设置。
10. “在项目中”下，选择普通。

从模板创建适配器

每个适配器都基于一个适配器模板。请在创建适配器之前，确定要使用哪个模板。

要创建适配器，请执行下列步骤：

1. 选择项目 > 适配器。
2. 单击添加适配器。
3. 在“名称”字段中，为适配器输入唯一名称。适配器名称在整个适配器集合中必须唯一并且不能包含空格。
4. 在“类型”部分，选择适配器类型。
5. 在“模板”部分，选择模板。此列表包含随 Build Forge 产品安装的适配器模板。ClearCase 和 ClearQuest 适配器无需单独的许可证密钥。其他适配器则需要通过适配器工具包（Adaptor Toolkit）单独进行许可。
6. 在“访问”部分，选择一个访问组。只有这些组成员才能查看或编辑该适配器。
7. 单击保存适配器。

将适配器添加到项目

要运行适配器代码，必须将适配器添加到项目。新建项目或将适配器添加到现有项目。

此选择将通过使用下列方法向您显示如何将适配器添加到项目：

- 可以使用适配器链接将源代码适配器添加到项目。
- 任何适配器（包括源代码适配器）都可以通过使用适配器点命令添加到项目。

使用适配器链接添加源代码适配器： 适配器链接用于将适配器连接到项目，并将应用程序环境变量连接到该适配器。

开始此任务之前，请为适配器创建项目和环境。

完成此任务之后，请打开环境并为应用程序环境变量提供实际值（如果尚未执行此操作）。

1. 选择项目 > 适配器链接。
2. 单击添加适配器链接。
3. 在“状态”部分，选择状态：

状态	描述
活动	项目运行时运行适配器。
非活动	项目运行时跳过适配器代码。
调试	仅运行适配器代码；项目运行时跳过其他步骤。

4. 在“适配器”部分，选择适配器模板。此列表显示随 Build Forge 产品安装的适配器模板。
5. 在“项目”部分，选中此项目。此列表显示尚未链接到适配器的项目。
6. 单击保存可将适配器链接到项目。适配器名称已添加到列表。
7. 在“环境”中，为适配器链接选择环境。
8. 在“填充环境”部分，选择是。适配器模板中的应用程序环境变量已添加到环境中。
9. 单击保存以保存该适配器链接。

使用点命令添加适配器： 可以使用适配器点命令将适配器添加到项目。该步骤运行时，这些点命令会调用 <adaptor_name>.xml 文件。

开始此任务之前，请为适配器创建项目和环境。

完成此任务之后，打开环境并为应用程序环境变量提供真正的值（如果尚未执行此操作）。

要将适配器点命令添加到项目以充当步骤，请执行下列步骤：

1. 选择库。
2. 在列表中选中该项目。
3. 单击添加步骤。
4. 在“名称”字段中，输入步骤名称。
5. 在“命令”中，输入适合应用程序类型的适配器点命令：.source、.defect、.test 和 .pack。

6. 在“环境”中，选择为适配器创建的环境。
7. 单击**保存步骤**。

测试适配器

运行适配器项目以测试适配器配置。要验证适配器是否可与外部应用程序交互并返回预期的结果，请以与其他项目步骤隔离的方式运行适配器。

本主题介绍如何测试下列适配器：

- 通过适配器链接添加到项目的适配器
- 使用适配器点命令添加到项目的适配器

测试链接的适配器： 对于通过适配器链接链接到项目的源代码适配器，可以使用此过程测试适配器配置。

常规过程如下：

1. 更改源文件
2. 使用链接的适配器运行 Build Forge 项目
3. 查看 BOM 报告以获取有关已更改的源文件的信息
4. 查看电子邮件以获取通过或失败通知

要测试链接的适配器：

1. 选择**项目 > 适配器链接**。
2. 在列表中选中链接的适配器和项目。
3. 在“状态”中，选择**调试**。
4. 单击**保存**。
5. 在源代码应用程序中，更改一个或多个源文件。提交更改以更新源代码库。
6. 如下所示运行适配器链接的项目：
 - a. 选择**管理 > 系统**。
 - b. 在列表中，选择**链接调试方式**。
 - c. 在“链接调试方式”中，选择**是**。
 - d. 单击**保存**。
 - e. 选择**作业 > 启动**。
 - f. 在项目列表中，从“启动项目”页面选择适配器链接的项目。
 - g. 单击**执行**。
7. 复审作业的 BOM 报告：
 - a. 打开**作业**。
 - b. 选择**已完成**选项卡，然后选择 **BOM** 选项卡。

测试适配器点命令： 为使用点命令添加到项目的适配器使用此可选过程来测试适配器配置。

常规过程如下：

1. 更改源文件
2. 使用适配器点命令运行 Build Forge 项目

3. 查看 BOM 报告以获取有关已更改的源文件的信息
4. 查看电子邮件以获取通过或失败通知

要测试适配器点命令，请执行以下操作：

1. 选择**作业 > 启动**。
2. 在列表中选中该项目。
3. 打开**作业步骤**选项卡。
4. 使用“步骤名称”复选框取消选中适配器点命令之外的所有项。
5. 单击**执行**以运行此项目。
6. 复审作业的 BOM 报告：
 - a. 选择**作业**。
 - b. 选择**已完成**选项卡，然后选择 **BOM** 选项卡。

设置适配器日志级别

要控制写入适配器的步骤日志的信息量，请使用 `_InterfaceLoggingLevel` 环境变量。

1. 将 `_InterfaceLoggingLevel` 添加到适配器的环境中。
 - 级别 8 记录的信息最多，而级别 0 记录的信息最少。
 - 其中包含下级日志的信息。例如，级别 2 包含级别 1 到 0 的信息。
 - 级别 7 为缺省的日志级别。
2. 指定一个日志级别来充当 `_InterfaceLoggingLevel` 变量的值：
 - 0: Exec 行加上服务器连接错误或取消通知；不包含其他内容
 - 1: 语法分析的命令（因为这些命令将发送到服务器）
 - 2: 未语法分析的命令（具有自己的局部变量集之前的命令）
 - 3: 构建和环境变量 SET 行
 - 4: Temp 和内部变量 SET 行
 - 5: 环境求值，电子邮件组添加，BOM 文本记录行
 - 6: 块与子块开始/结束行
 - 7: （缺省的日志级别）将针对匹配模式检查的代理程序输出，以及匹配这些模式的行
 - 8: 所有代理程序输出

手动启动适配器链接的项目

适配器链接的项目可以按照调度运行、使用快速启动选项运行，或者在选中了“运行链接”复选框时可手动启动。如果未选中“运行链接”，项目将在没有适配器步骤的情况下运行。


1. 选择**作业 > 启动**。
2. 在项目列表中，选择适配器链接的项目的名称。

“启动项目”页面将打开。
3. 选中**运行链接**复选框。

4. 从“启动项目”页面单击**执行**。

快速启动适配器链接的项目

适配器链接的项目可按计划运行，通过从“启动项目”页面中选择项目名称来手工启动（作业 → 启动），也可以在完成“启用快速启动”系统配置设置后使用“快速启动”图标来运行：以下步骤假定已设置“启用快速启动”。

1. 选择**管理 > 系统**。
2. 在列表中选择**链接手动作业**。
3. 在“链接手动作业”部分，选择**是**。
4. 单击**保存**。
5. 选择**作业 > 启动**。
6. 在项目列表中，选择适配器链接的项目的**快速启动**图标 ，以立即运行该项目。
要验证作业是否正在运行，请选择**作业 > 正在运行**选项卡。

复位适配器模板

关于此任务

复位适配器模板的操作会将最新的适配器模板从 `<bfinstall>\interface` 目录复制到 Build Forge 数据库。

随时都可以复位适配器模板：

- 安装维护版或新的产品版本
- 通过编辑接口目录中（不是管理控制台中）的版本来修改模板
- 创建新的适配器模板

过程

1. 选择**管理 > 系统**。
2. 在列表中，选择**复位适配器模板**。
3. 在“复位适配器模板”中，选择**是**。
4. 单击**保存**。

更新 ClearQuest 构建记录

您可以将系统与 Rational ClearQuest 集成，以自动创建和更新 ClearQuest 数据库内的构建记录。

系统可以在您的 IBM Rational® ClearQuest® 数据库中自动创建构建记录，并附带指向日志数据的链接。再则，当作业通过时，系统可以更新 ClearQuest 数据库，标注作业已完成、记录结束时间以及完成的步骤汇总。该功能需要 Rational ClearQuest V7.0 或更高版本。

注：要成功创建构建记录并用构建相关信息填充该记录，您必须先打开 ClearQuest 维护工具并建立与数据库的连接。

当您将项目配置为更新 ClearQuest 数据库时，系统执行以下步骤：

作业启动

当系统启动一个作业时，系统将创建一个 ClearQuest 构建记录。构建记录处于

“已提交”状态，其中包含作业日志 URL、开始时间、发行版名称、标识以及表明“Build XYZ started.”的日志条目。如果源控制适配器取消了作业（例如，由于找不到任何源更改），那么不会创建任何 ClearQuest 构建记录。

注：如果某项目链向其他项目，那么新项目将获取其自己的唯一 ClearQuest 构建标识。

作业通过/失败

当作业通过或失败时，系统会将 ClearQuest 中的构建状态更改为“已完成”或“失败”，设置构建结束时间，并在 ClearQuest 构建日志中存储作业步骤的摘要。该摘要包含每个步骤的名称、结果状态和服务器。

作业重新启动

当作业重新启动时，系统会将 ClearQuest 中的构建状态更改为“已提交”，并创建一个 ClearQuest 构建日志条目指出“已重新启动构建 XYZ”。

您可以通过特殊环境变量配置 ClearQuest 集成。要将项目链接到 ClearQuest 数据库，请确保在项目的环境中包含下表中的变量。

注：项目环境中必须有这些变量。将它们添加至某个步骤还不够。但可以使用设置为“包含”类型的变量，这一类型通过另一个环境来包含这些变量。此外，由于 CQ_RELEASE_NAME 值是唯一可能随项目而变化的值，因此您可能希望创建一个环境来包括其他变量，并使用“包含”类型的变量将该环境包括在项目环境中，您还可在其中将 CQ_RELEASE_NAME 指定为特定于项目的环境变量。

表 17. Rational ClearQuest 集成所需的环境变量

变量	描述
CQ_DBNAME	要更新的 ClearQuest 数据库的名称。
CQ_INTERACTION	如果您的项目环境已定义用于启用 ClearQuest 构建记录创建的相应环境变量，而您不希望创建构建记录，请将此变量设置为 OFF 来禁用构建记录创建。 要启用创建构建记录，将此环境变量设置为 ON。 注： 如果要使用 ClearQuest 的任一适配器，将此环境变量设置为 OFF。
CQ_USER	用于登录到 ClearQuest 数据库的用户名。
CQ_RELEASE_NAME	ClearQuest 数据库中要更新的发行版的名称。
CQ_PASSWORD	登录到 ClearQuest 数据库时使用的密码。不是必需的；缺省为空。
CQ_DBSET	ClearQuest 数据库设置值。不是必需的；缺省为空。

高级适配器任务

可以为要与 Build Forge 项目交互的外部应用程序修改 Build Forge 适配器模板或新建适配器模板。

修改和创建模板是高级任务,需要初步了解:

- XML 语言

- 外部应用程序的命令语言
- 正则表达式

执行高级任务之前,请阅读以下部分中的信息:

- 第 433 页的『核心适配器任务』
- 第 446 页的『适配器模板结构』
- 第 444 页的『适配器参考』

此部分提供有关下列任务的信息:

- **修改适配器模板:** 描述如何修改针对所有适配器的模板。
- **修改针对单个适配器的模板:** 描述如何修改针对一个适配器的模板。
- **新建适配器模板:** 描述新建模板的一般过程。
- **示例: 向适配器通知添加用户:** 描述如何向电子邮件通知添加访问组。
- **示例: 从 BOM 报告除去更改详细信息:** 描述如何从 BOM 报告除去更改详细信息。

修改适配器模板

如果希望从适配器模板创建的所有将来的适配器挑选模板修改, 请使用此过程。

首先, 请了解要修改的内容。例如, 您可能希望更改通知或修改 BOM 报告格式。

1. 使用 XML 编辑器打开要修改的适配器模板。适配器模板位于以下目录中:

bfinstall\interface (针对运行 Microsoft Windows 操作系统的计算机)

bfinstall/Platform/interface (针对运行 UNIX 或 Linux 操作系统的计算机)

2. 输入模板修改的内容。
3. 保存适配器模板。切勿更改模板名称。适配器和适配器模板都必须有唯一名称。

下次使用修改的适配器模板创建适配器时, 新的适配器将包含您的模板修改内容。

修改针对单个适配器的模板

使用此过程可修改仅针对适配器的单个实例的模板。

首先, 请了解要修改的内容。例如, 您可能希望更改通知或修改 BOM 报告格式。

如果要修改的适配器已创建, 请如下所示通过管理控制台修改模板:

1. 选择项目 > 适配器。
2. 在文本框中输入对模板的修改。
3. 单击保存适配器。

注: 您的更改将仅保存到与 Build Forge 数据库中的适配器关联的适配器模板的实例中。更改不会保存到 *interface* 目录中的适配器模板文件内。

如果要修改的适配器尚未创建, 请如下所示在创建适配器之前修改模板:

1. 使用 XML 编辑器打开适配器模板。适配器模板位于以下目录中:

bfinstall\interface (针对运行 Microsoft Windows 操作系统的计算机)

bfinstall/Platform/interface (针对运行 UNIX 或 Linux 操作系统的计算机)

2. 输入对模板的修改。
3. 更改适配器名称，然后保存适配器模板。

注：适配器和适配器模板都必须有唯一名称。

4. 重置适配器模板以挑选已修改模板，然后将其添加到管理控制台中的可用模板列表中。请参阅第 438 页的『复位适配器模板』。

创建新的适配器模板

使用此过程可以为要与 Build Forge 项目交互的外部应用程序新建适配器模板。

1. 仔细查看 Build Forge 产品提供的适配器模板中的 XML 结构和元素。适配器模板位于以下目录中：

`bfinstall\interface`（针对运行 Microsoft Windows 操作系统的计算机）

`bfinstall/Platform/interface`（针对运行 UNIX 或 Linux 操作系统的计算机）

2. 规划希望新适配器执行的操作：
 - 了解该适配器将运行哪些命令
 - 确定如何对这些命令进行语法分析
 - 确定如何处理从语法分析结果收集的数据
 - 了解需要哪些外部环境变量
3. 在接口目录中选择一个适配器模板来用作模型。如果可以，请根据同一个外部应用程序选择适配器。或者，选择具有与要创建的适配器类似功能的适配器。使用模型中的 XML 层次结构、元素和元素属性作为指导创建新的模板。
4. 使用 XML 编辑器打开模型适配器模板。使用新名称将模板保存到接口目录中。
5. 根据您的计划为新适配器编写 XML 代码。
6. 保存这个新的适配器模板。
7. 使用接口目录中的 `interface.dtd` 文件验证适配器模板。
8. 重置适配器模板以挑选新模板，然后将其添加到管理控制台中的可用模板列表中。请参阅第 438 页的『复位适配器模板』。
9. 为适配器创建一个项目。
10. 为适配器创建一个环境。请参阅第 433 页的『为适配器创建环境』。
11. 使用这个新的适配器模板创建一个适配器。请参阅第 434 页的『从模板创建适配器』。
12. 将适配器添加到项目。请参阅第 435 页的『将适配器添加到项目』。
13. 运行适配器项目以测试该适配器。请参阅第 436 页的『测试适配器』。

创建多入口点适配器

由 Build Forge 提供的适配器模板是单入口点适配器。

对于单入口点适配器，在管理控制台（**项目 > 适配器**）中，选择想要运行的应用程序和函数的模板名称。例如，`ClearCaseBaseline` 或 `ClearCaseByDate`。

如果希望，您可以创建一个包含 `ClearCase` 多个接口或函数的 `ClearCase` 适配器模板。对于多入口点适配器，可以通过名称（称为入口点）来识别每个接口。

要创建多入口点适配器：

1. 创建适配器模板。要创建模板，您可以选择将提供的某个模板用作模型，或者根据需要修改 XML。在该模板中，必须将 `name` 属性添加至 `<interface>` 元素，以标识您添加到模板的每个接口的入口点。`<interface>` 元素的相关语法如下示例所示：

```
<interface name="By Date" default="true">
</interface>
```

2. 使用不包含空格的唯一名称创建适配器，并将该适配器与适配器模板相关联。请参阅第 434 页的『从模板创建适配器』。
3. 使用适配器点命令或适配器链接将适配器添加至项目。

- 以下示例使用 `.source` 适配器命令将适配器添加至在 `ClearCase` 适配器中调用“`By Date`”接口函数的项目步骤：

```
.source ClearCase "By Date"
```

- 要使用适配器链接来调用多入口点适配器，请在适配器模板中执行以下某个操作以指定项目运行时运行哪个接口函数：
 - 在模板文件中放入要作为第一个 `<interface>` 元素运行的函数的 `<interface>` 元素定义
 - 针对要运行的函数，将 `<interface>` 元素的 `default` 属性设置为 `true` (`default="true"`)

示例：启用电子邮件通知

可以配置适配器模板来向在外部应用程序中引起更改的用户发送电子邮件通知。下面的示例显示如何设置两种类型的通知：

- 通知为当前构建检入了文件的所有用户
- 通知 `Build Forge` 访问组的所有成员

下列过程引用了 `ClearCaseByDate` 模板中的元素。任何适配器模板都可以添加自己的元素来启用通知。

通知为当前构建检入了文件的所有用户

可以使用 `<adduser>` 命令动态构建检入该构建代码的用户组，然后使用 `<notify>` 命令向该组发送通知。

`ClearCaseByDate` 模板查询 `ClearCase` 以了解两个时间戳记之间的所有更改。缺省时间戳记用于当前适配器运行和上次适配器运行。实际上，它将转换为自上次构建以来为当前构建检入的所有更改的列表。

假定：视图中所有用户名对您用于按该名称通知的 `SMTP` 服务器均为已知。这就意味着 `ClearCase` 用户名需要与电子邮件用户名保持一致。

要启用此通知：

1. 在 XML 编辑器中打开 `ClearCaseByDate` 适配器模板。
2. 找到并编辑 `<adduser>` 以创建用户组，如下所示：

```
<adduser group="MyChangers" user="$4">
```

位置参数 `$4` 引用 `ClearCaseByDate` 模板生成的 `ClearCase` 视图中显示的用户名字段。

3. 设置通知，以向此组发送电子邮件。下列设置将在项目失败以及项目成功时均发送电子邮件。在某些环境中，您可能希望仅当构建失败时才通知。

```
<!-- Set some notifications for when the build completes -->
<onproject result="fail">
  <notify group="MyChangers" subject="Build $BF_TAG ($CurDate) Failed." message="{Changing}{Changes}"/>
</onproject>
<onproject result="pass">
  <notify group="MyChangers" subject="Build $BF_TAG ($CurDate) Passed." message="{Changing}{Changes}"/>
</onproject>
```

4. 保存适配器模板。

适配器运行时，将从视图中的用户名构建 MyChangers 组。构建项目完成时，将向该组发送电子邮件通知。

通知属于一个 Build Forge 访问组的所有用户

在本示例中，您希望通知某个 Build Forge 访问组的所有成员。ClearCaseByDate 适配器模板用于此示例。假定：Build Forge 中的所有用户名与 SMTP 服务器中的电子邮件用户名相对应。

1. 在 XML 编辑器中打开 ClearCaseByDate 适配器模板。
2. 找到并编辑 <adduser> 以从 Build Forge 访问组创建用户组，如下所示：

```
<adduser group="Developer_Access_Group" user="Developer">
```

3. 设置通知，以向此组发送电子邮件。下列设置将在项目失败以及项目成功时均发送电子邮件。在某些环境中，您可能希望仅当构建失败时才通知。

```
<!-- Set some notifications for when the build completes -->
<onproject result="fail">
  <notify group="Developer_Access_Group"
subject="Build $BF_TAG ($CurDate) Failed." message="{Changing}{Changes}"/>
</onproject>
<onproject result="pass">
  <notify group="Developer_Access_Group"
subject="Build $BF_TAG ($CurDate) Passed." message="{Changing}{Changes}"/>
</onproject>
```

4. 保存适配器模板。

适配器运行时，将从属于 Developers 访问组的用户名构建 Developer_Access_Group 组。构建项目完成时，将向该组发送电子邮件通知。

要点：如果 <notify> 伪指令失败（例如，作为电子邮件收件人的用户不存在），那么 .xml 也将失败，并且所有的后续通知都将失败。

示例：从 BOM 报告除去更改详细信息

大多数适配器模板都将更改详细信息记录到 BOM 报告中。（diff 命令用于记录更改详细信息。）

下列步骤引用 ClearCaseByDate 模板中的元素，但可用于除去任何适配器模板的更改详细信息。

要除去 BOM 报告中的更改详细信息：

1. 在 XML 编辑器中打开适配器模板。
2. 找到调用 diff 命令的 <run> 元素。除去下面的行：

```
<run command="cc_diff" params="$VIEW $1 $2" server="$CCSERVER" dir="/" timeout="360"/>
```

3. 找到 diff 命令的 <command> 元素。除去下列行：

```

<!-- The cc_diff command does a generic clearcase diff, logging the full output
of the diff in the BuildForge BOM -->
<command name="cc_diff">
<execute>
pushd \\view\%1 && cleartool diff -pred -diff_format "$2@@$3"
</execute>
<resultsblock>
<match pattern=".">
<bom category="Source" section="diff">
<field name="diff" text="$_" />
</bom>
</match>
</resultsblock/>
</command>

```

4. 找到 <bomformat> 节，然后找到 diff 命令输出的 <section> 元素。除去下列行：

```

<section name="diff">
<field name="diff" title="Change Details"/>
</section>

```

5. 保存适配器模板。

适配器参考

适配器设计为添加到 Build Forge 项目且不经修改即运行。要修改或新建适配器，需了解 Build Forge 适配器模板中使用的 XML 模板结构和元素。

注：本部分不描述 Build Forge 适配器模板中使用的外部应用程序命令。有关这些命令的信息，请查阅外部应用程序的文档。

随 Build Forge 产品安装的适配器模板位于以下目录中：

bfinstall\interface（针对运行 Microsoft Windows 操作系统的计算机）

bfinstall/Platform/interface（针对运行 UNIX 或 Linux 操作系统的计算机）

本部分描述以下参考信息：

- **适配器需求：**描述使用适配器的一般需求以及使用 ClearQuest 适配器模板的特殊需求。
- **适配器点命令：**描述适配器点命令的语法。
- **ClearCase 和 ClearQuest 环境变量：**描述 ClearCase 和 ClearQuest 适配器使用的环境变量。
- **Perforce 环境变量：**描述 Perforce 所需的某些其他环境变量。
- **适配器模板结构：**描述 Build Forge 适配器模板的一般结构。
- **适配器 XML 参考：**描述 Build Forge 适配器模板中使用的 XML 元素。

适配器点命令

部分点命令可用于将外部应用程序的适配器作为项目步骤添加到 Build Forge 项目。

- **.source：**将源代码应用程序的适配器添加到项目步骤。
- **.defect：**将缺陷跟踪应用程序的适配器添加到项目步骤。
- **.test：**将测试应用程序的适配器添加到项目步骤。
- **.pack：**将打包应用程序的适配器添加到项目步骤。

另请参阅第 333 页的『点命令参考』。

Rational ClearCase 与 Rational ClearQuest 环境变量

Rational ClearCase 与 Rational ClearQuest 适配器模板使用下表中的环境变量来运行 cleartool 命令。

下表中每一个环境变量并非都是每一个 ClearCase 或 ClearQuest 适配器模板所必需的。

在每个适配器模板中，所需的环境变量在 <template> 节的 <env> 元素中列出。

运行 ClearCase 或 ClearQuest 适配器项目之前，请为必需的变量提供实际值，或者接受缺省值。编辑环境中对适配器指定的变量值。

变量必须出现在项目的环境中。它们在步骤级别不起作用。

适配器模板位于以下目录中：

bfinstall\interface（针对运行 Microsoft Windows 操作系统的计算机）

bfinstall/Platform/interface（针对运行 UNIX 或 Linux 操作系统的计算机）

表 18. ClearCase 和 ClearQuest 的环境变量

环境变量名	替换变量	描述
VIEW	value=my_adaptor_view	将此变量设置为要用于适配器的 ClearCase 视图的名称。
VOB_PATH	value=\c_vob	将此值设置为组件 VOB 的名称及其子目录（子目录可选）。
PROJECT_VOB	value=\ProjectVob	使用 ByBaseline 适配器时，将此变量设置为项目 VOB 的名称（仅用于 UCM ClearCase）。
CCSERVER	value=BFServerName	将此变量设置为安装了 ClearCase 客户机，并且正在运行该客户机的 Build Forge 服务器的名称。
CurDate	value=.date %d-%b-%y.%H:%M:%S	向适配器提供当前日期，使用 .date 命令以 ClearCase 期望的格式生成该日期。请不要更改此值。
LAST_RUN	value=1-Jan-05.00:00:00	对于 ByDate 适配器，系统使用该值来确定是否发生了任何更改；该值是上一次成功运行的日期。可在测试适配器时操作此值，通过选择在某些更改之前的某个已知日期可强制适配器运行。如果适配器允许运行继续，它将自动把此值更新为当前日期。缺省值为 1-Jan-05.00:00:00。
LABEL	value=BUILD_1	对于 ByLabel 适配器，当您使用您的适配器来按标签生成差异时（通过 ByLabel 适配器），系统将使用此值作为标签。
BASELINE	value=BUILD_1	对于 ByBaseline 适配器，当您使用您的适配器来按基线生成差异时，系统将使用此值作为基线。
ACTIVITIES	value=SAMPL0001@\ProjectVob	对于 ClearQuestClearCaseByActivity 适配器，这是一组用空格定界的活动标识。

Perforce 环境变量

将下列 Perforce 环境变量添加到指定给 Perforce 适配器的 Build Forge 环境组：

- P4USER
- P4PASSWD

要访问 Perforce 服务器，Build Forge 需要有效的用户名和密码。在步骤日志中，Perforce 用户名和密码以明文的格式编写。

不能使用环境变量的“指定隐藏”属性来将 Perforce 认证信息进行加密。

适配器模板结构

本主题描述了 Build Forge 适配器模板中的常规 XML 结构或元素层次结构。

适配器模板由以下节元素组成: <template>、<interface>、<command> 和 <bomformat>。其中的每个部分都包含子元素。

要获取元素描述, 请参阅『适配器 XML 引用』。

```
<PROJECT_INTERFACE>
<template>
<env/>
</template>

<interface>
<setenv/>
<run/>
<ontempenv>
<step/>
</ontempenv>
<onproject>
<notify/>
</onproject>
</interface>

<command>
<execute> 或 <command>
command line
</execute> 或 </command>
<resultsblock>
<match>
<bom>
<field/>
</bom>
<adduser/>
<setenv/>
<run/>
</match>
</resultsblock>
</command>

<bomformat>
<section>
<field/>
</section>
</bomformat>
</PROJECT_INTERFACE>
```

适配器 XML 引用

本部分按字母顺序列出适配器 XML 元素。它是适配器 XML 语言中使用的元素的参考。还描述了一些示例和伪码。

adduser

在 <interface> 元素或 <match> 元素中使用 <adduser> 元素, 以根据更改命令的输出将用户添加到临时组, 以便适配器可以将通知发送到引起更改的用户。如果用户已经是组的成员, 系统不会将用户添加到组, 以防多个通知。<adduser>元素是一个空元素。组属性指定在适配器逻辑运行期间创建的临时组; 必须在 <notify>元素中引用同一个组, 才能发送实际的通知。

```
<adduser group="MyChangers" user="$4"/>
```

使用 `condition="function"` 属性控制 `<adduser>` 元素是否将导致更改的用户添加到临时访问组中。`function` 的值是一个表达式，其求值结果为 `true` 或 `false`。如果表达式求值的结果为 `true`，那么将用户添加到临时组；否则，不添加用户。请参阅第 323 页的『条件函数』。

支持以下电子邮件地址格式。`Username` 是电子邮件用户名。`Name` 是任意字符串，通常是某种形式的给定名称。

```
username@host.com
name username@host.com
name "<username@host.com>"
name <username@host.com>
name "username@host.com"
```

bom

`<bom>` 元素定义要记录到作业的材料清单（BOM）中的信息，应以 `<match>` 元素围起来。`<bom>` 元素必须在 BOM 内指定一个类别和节，并定义应该将 `<match>` 元素收集的已编号变量中的哪些（\$1...\$n）转换为 BOM 数据的字段。

```
<bom category="Source" section="changesets" >
  <field name="Change" text="$1"/>
  <field name="Date" text="$2"/>
  <field name="User" text="$4"/>
</bom>
```

使用 `condition="function"` 属性控制是否将 `<bom>` 元素写入 BOM 报告。`function` 的值是一个表达式，其求值结果为 `true` 或 `false`。如果表达式求值的结果为 `true`，那么 `<bom>` 元素中的信息将写入 BOM；否则不会写入。请参阅第 323 页的『条件函数』。

bomformat

使用 `<bomformat>` 元素可定义较早 `<bom>` 元素中收集的数据的显示方式。它使用指定 BOM 类别逻辑名的 `category` 属性以及指定类别显示名称的 `title` 属性。包含 `<field>` 元素的 `<section>` 元素的结构定义了布局，如下面的示例所示：

```
<bomformat category="Detail" title="Change Details">
  <section name="descriptions" parent="section name" expandable="yes">
    <field name="Description" title="Change Description"/>
  </section>
  <section name="diff">
    <field name="Diff" title="Differences"/>
  </section>
```

command

一个适配器 XML 文件可包含多个 `<command>` 元素；每个元素定义一个可由 `<interface>` 元素内的 `<run>` 元素引用的指定命令。`<command>` 元素在 `<interface>` 元素外指定，这样能使 XML 文件内的多个接口复用同一命令。

通过在 `<resultsblock>` 元素的 `<match>` 元素中嵌入 `<run>` 命令，命令可以调用其他命令。

`<command>` 元素如下所示合并了 `<execute>` 和 `<resultsblock>` 元素的结构：

```
<command name="p4_changes">
  <execute>
    command line
  </execute>
```

```

    <resultsblock>
      Has its own structure.
    </resultsblock>
  </command>

```

或者，也可以使用 `<integrate>` 元素替换块中的 `<execute>` 元素。

使用 `mode` 属性来指定 `<command>` 元素的方式。 `mode` 属性接受以下值：

- **conjoined**: 所有对该命令的调用都分组为一个调用以供服务器进行处理。
- **parallel**: 但服务器槽变为可用时，分别处理各个调用。
- **exec**: 命令启动并立即由服务器处理。

env

`<env>` 元素在元素内部使用，用于定义可复制到项目链接所用环境中的环境变量（具有初始值）。每个 `<env>` 元素应包含名称和值属性。值为变量提供了初始值。

```
<env name="FILESPEC" value="//depot..." />
```

execute

在 `<command>` 元素内使用 `<execute>` 元素来指定命令。元素的内容是要发送至服务器（由适配器使用）的一行或多行文本。不能在 `<execute>` 元素中使用点命令。`<run>` 元素调用 `<command>` 元素时，系统会将 `<execute>` 元素内容中的所有位置参数替换为所调用的 `<run>` 元素中指定的参数。`<execute>` 元素内容中的 `$1` 参数将由首个参数替换，`$2` 参数将由第二个参数替换，依此类推。

使用 `condition="function"` 属性控制 `<execute>` 元素中命令的执行。`function` 的值是一个表达式，其求值结果为 `true` 或 `false`。如果表达式求值结果为 `true`，那么系统执行命令；如果表达式求值结果为 `false`，那么不会执行命令。请参阅第 323 页的『条件函数』。

```

<execute>
p4 changes -s submitted -t -i $2@$1,@now
</execute>

```

field

在 `<bom>` 或 `<section>` 元素中使用 `<field>` 元素来指定字段。

当在 `<bom>` 元素中使用时，指定名称和文本；文本属性定义使用哪个变量在数据字段中填充数据。

当在 `<bomformat>` `<section>` 元素中使用时，指定名称和标题。名称指定逻辑名，而标题用于显示。如果在 `<section>` 中有多个字段，那么包含一个顺序属性。

```

<section name="changesets">
  <field order="1" name="Change" title="Change ID"/>
  <field order="2" name="Date" title="Date"/>
  <field order="3" name="Time" title="Time"/>
  <field order="4" name="User" title="User ID"/>
  <field order="5" name="Client" title="Client"/>
  <field order="6" name="Comment" title="Comment"/>
</section>

```

使用 `condition="function"` 属性控制是否将 `<field>` 元素写入 BOM 报告。`function` 的值是一个表达式，其求值结果为 `true` 或 `false`。如果表达式求值结果为 `true`，那么 `<field>` 元素中的信息将写入 BOM。如果表达式求值结果为 `false`，那么该信息不会写入 BOM。请参阅第 323 页的『条件函数』。

使用 `template` 属性定义 `<field>` 元素的文本格式。例如，如果文本为字符串，那么模板值可能为“Hello \$VALUE”。当字段写入 BOM 报告时，\$VALUE 会替换为字段文本。

integrate

`<integrate>` 元素类似于 `<execute>` 元素。可以使用 `<integrate>` 元素来替换 `<execute>` 元素。和 `<execute>` 元素一样，`<integrate>` 元素也指定要运行的命令行。它具有以下区别：

- 命令行在管理控制台系统上执行，而不是在执行适配器的服务器上执行。
- 命令行使用 `\integration` 目录（安装目录的子目录）作为其当前目录。

`<integrate>` 元素对于执行位于管理控制台计算机（特别是位于 `\integration` 目录）中的应用程序或脚本非常有用。

`<run>` 元素调用包含 `<integrate>` 元素的 `<command>` 元素时，系统会将 `<integrate>` 元素中的所有位置变量替换为所调用的 `<run>` 元素中指定的参数。`<integrate>` 元素中的 `$1` 会替换为第一个参数，`$2` 替换为第二个参数，依此类推。

和 `<execute>` 元素一样，在 `<integrate>` 元素中不能使用点命令。

以下示例来自 IBM Rational ClearQuest 适配器，它通过运行 `CQperl` 命令（用于执行 Perl 代码的 ClearQuest 程序）并为其提供 `\integration` 目录中的 Perl 脚本名称，以向 ClearQues 发送数据。该示例假定 ClearQuest 安装在管理控制台系统上。

```
<integrate>
cqperl bfcqresolve.pl $2 Fixed "Fixed in build $BF_TAG"
</integrate>
```

interface

`<interface>` 元素是适配器入口点的容器。其中的元素定义了适配器的程序逻辑。它包含 `<setenv>`、`<run>`、`<ontempenv>`、`<onproject>` 和 `<adduser>` 元素。

一个适配器模板可以有多个入口点。如果创建具有多个入口点的适配器模板，请执行以下操作：

- 使用 `name` 属性来标识每个入口点。
- 如果在不指定名称的情况下调用适配器模板，请使用 `default` 属性来标识要运行的入口点。

要指定入口点，请在对模板的 `.source` 调用中将入口点名称用作参数。

示例：创建名为 `MyAdaptorTemplate` 的适配器模板。在该模板中，放入以下代码以定义指定的入口点并将它设置为缺省值：

```
<interface name="By Date" default="true">
</interface>
```

要通过名称调用此接口，请在项目步骤中使用以下命令：

```
.source MyAdaptorTemplate "By Date"
```

属性：

name 可选。此接口的名称。将名称用作参数调用适配器时，将使用名称与该参数匹配的接口。

default

可选。设置为“是”或“否”。如果为“是”，那么不用名称参数调用适配器时，将使用该接口。

match

<resultsblock> 元素中使用 <match> 元素来处理输出行。<match> 元素采用定义匹配行的模式属性。模式是一种 Perl 正则表达式。

匹配模式可以包含夹注表达式，这些表达式存储在变量 \$1...\$n 中。

```
<match pattern="^Change (\d+) on (.*) (.*) by (.*)@(.*)'(.*)'$">
```

<match> 元素使用 <adduser>、<setenv>、<bom> 和 <field> 作为子元素。有关更详细的示例，请参阅第 451 页的『resultsblock』的参考信息。

notify

<notify> 元素指定电子邮件和分发列表。它通常在 <onproject> 元素中使用，以根据执行结果指定通知。

属性:

group 电子邮件的收件人。这是在适配器中定义的一组用户。

message

电子邮件的正文内容。您可指定该文本。

subject

电子邮件的主题。您可指定该文本。

onproject

<onproject> 元素定义在系统运行项目步骤之后，由系统执行的通知操作。该元素带有一个必需的 result 属性，该属性指定这些操作是针对已通过的作业还是失败的作业执行。通常，适配器 XML 文件中含有两个 <onproject> 元素：一个用于通过的情况，一个用于失败的情况。以下示例显示了一对 <onproject> 元素，它们使用 notify 元素，根据项目是通过还是失败来发送不同的消息：

```
<onproject result="fail">
  <notify group="MyChangers"
    subject="Run $BF_TAG ($CurDate) Failed." message="$Changing$Changes"/>
</onproject>

<onproject result="pass">
  <notify group="MyChangers"
    subject="Run $BF_TAG ($CurDate) Passed." message="$Changing$Changes"/>
</onproject>
```

ontempenv

<ontempenv> 元素在 <interface> 元素内使用，其作用类似于 if-then 语句。使用该元素可以向项目返回一个通过或失败值；通过值指示系统应继续并运行项目的剩余部分，而失败值指示系统应停止。这通常用于指示接口是否找到了应重新运行项目的相关更改。

在系统运行 <run> 元素中指定的所有命令之后，它会处理 <ontempenv> 元素。使用该元素的 name 属性可指定一个临时环境变量，而使用 state 属性可指定一个值。

<ontempenv> 包含 <step> 元素，后者仅当 <run> 元素命令运行后存在临时环境变量名称和状态时才会运行。

```
<ontempenv name="Changes" state="empty">
  <step result="FAIL"/>
</ontempenv>
```

PROJECT_INTERFACE

<PROJECT_INTERFACE> 元素包含适配器模板中的所有其他标记。它带有一个 IFTYPE 属性，该属性指示适配器类型。有效类型包括 Source、Test 和 Defect。

```
<PROJECT_INTERFACE IFTYPE="Source">
...all other elements...
</PROJECT_INTERFACE>
```

relate

<relate> 元素指定工件与用户之间的关系。它与类型为“通知更改人”的日志过滤器结合使用，后者在项目中定义并在一个步骤的 Result 属性中使用，而该步骤在其调用适配器之前那一刻运行。“通知更改人”过滤器通常会指定一个预期的日志行，该行指出是成功还是失败。该步骤运行时，过滤器会与 <relate> 元素中指定为工件的文本进行比较。如果两者匹配，那么将向与该工件（artifact）关联的用户发送电子邮件。

另请参阅第 293 页的『过滤器操作』中“通知更改人”过滤器的描述。

属性:

artifact

与日志过滤器搜索的文本匹配的文本。

user 与工件相关联的用户。过滤器与文本匹配时，系统向此用户发送电子邮件。

text 当日志过滤器与此关系之间存在匹配时要记录到 BOM 中的文本。

resultsblock

<resultsblock> 元素定义系统应如何处理从相关的 <execute> 元素执行的命令行的结果。<resultsblock> 元素只能在 <command> 元素中使用。<resultsblock> 元素可嵌套在部分结果中。

<resultsblock> 元素可以有可选的 beginpattern 和 endpattern 属性，这两个属性使用 Perl 正则表达式来定义要一系列要处理的输出行。这样，可以使用不同 <resultsblock> 元素处理不同的范围。以下伪码显示了 <resultsblock> 元素的结构。

```
<resultsblock startpattern="" endpattern="" >
  <match>
    <bom>
      <field/>
    </bom>
    <adduser/>
    <setenv/>
    <run/> ( <run> 元素可用来运行其他命令中的命令 )
  </match>
  <setenv/>
</resultsblock>
```

下面的示例演示了 <resultsblock>、<match> 和 <bom> 元素如何协同工作:

```
<resultsblock
beginpattern="^Change (\d+) by (.*)@(.*) on (.*) (.*)$"
endpattern="^Differences ...$">
```

```

<match pattern="(?(?:(!Differences ...).)*$).?">
<bom category="Detail" section="descriptions">
<field name="Description" text="$_" />
</bom>
</match>
</resultsblock>

```

run

<run> 元素在 **<interface>** 元素使用，用来指定要运行的指定命令。命令在同一个 XML 文件的后面定义。**<run>** 元素是一个空元素。

属性:

condition

或者，其形式为 **condition="function"**。*function* 的值是一个表达式，其求值结果为 true 或 false。如果它的求值结果为 true，那么运行命令。如果它的求值结果为 false，那么不运行命令。请参阅第 323 页的『条件函数』。

command

必需。它指定要运行的已定义命令的名称。该命令在 **<command>** 元素中指定和定义。

dir 必需。它指定要运行命令的目录。**dir** 解释为运行命令的服务器资源中所设置路径的扩展。

mode 可选。它指定运行命令的运行方式。它可以是下面的一种方式:

- **conjoined:** 所有对该命令的调用都分组为一个调用以供服务器进行处理。
- **parallel:** 但服务器槽变为可用时，分别处理各个调用。
- **exec:** 命令启动并立即由服务器处理。

params

必需。它指定传递给命令的参数。使用空格来分隔参数。

server 必需。它指定要用于运行命令的服务器资源。如果它设置为空，那么该命令在运行适配器的步骤中所用的服务器上运行。使用 **server=""** 可将服务器设置为空。

timeout

必需。它指定在命令超时前所经过的秒数。

显示必需属性的示例:

```
<run command="UpdateEnv" params="" server="" dir="/" timeout="360"/>
```

示例:

```
<run command="p4_changes" params="$LAST_RUN $FILESPEC $LAST_VER"
server="$P4CLIENT" dir="/" timeout="360"/>
```

section

使用 **<section>** 元素可定义如何显示部分 BOM 类别。它接受一个 **name** 属性。您只能在 **<bomformat>** 元素内使用 **<section>** 元素。

setenv

使用 **<setenv>** 元素可初始化 **<interface>** 或 **<match>** 元素中的环境变量值。**<setenv>** 元素不包含其他元素。

可以三种不同的方式使用该变量:

- 如果您指定组名, 它的工作方式与 `.set` 命令类似。它设置数据库中主记录中的变量值而不设置当前步骤使用的副本。当前步骤中运行的适配器看不到更改。您不能以这种方式创建新变量。
- 如果未指定组名, 它的工作方式类似于 `.bset` 命令。它设置在运行的作业的环境中的变量值。更改对作业中的所有步骤都可用。可以这种方式创建新变量。
- 如果不指定组名而指定临时变量 (`type="temp"`), 它将设置一个仅供适配器逻辑使用的临时变量。该变量在适配器步骤运行后不会持久保存。可以这种方式创建新变量。

属性:

condition

或者, 其形式为 `condition="funtion"`。 *funtion* 的值是一个表达式, 其求值结果为 `true` 或 `false`。如果它的求值结果为 `true`, 那么运行命令。如果它的求值结果为 `false`, 那么不运行命令。请参阅第 323 页的『条件函数』。

eval 可选。设置为 `True` 或 `False`。如果为 `true`, 适配器将尝试对 `value` 属性表达式求值并存储结果。

group 可选。它指定其中定义该变量的 Build Forge 环境。如果指定环境名称, 必须引用指定环境中的现有变量。

如果将 `[ADAPTOR]` 指定为值, 那么该值在运行时设置。它设置为调用适配器的步骤或适配器链接的环境。

Build Forge 允许多个环境中具有相同名称的变量。环境继承的优先顺序以及环境包含会影响确定在运行时指定给变量的值的方式。

name 必需。它指定要设置的变量的名称。该值可以是一个变量。在这种情况下, 在运行时才会设置变量名称。

type 可选。它指定设置变量的方法。它使用下面的一个值:

- `append text`: 将指定值放在任何现有值之后。如果指定了可选的 `text`, 该文本将放在两个值之间。
- `once`: 仅当尚未设置时才应设置该变量。
- `prepend text`: 将值放在任何现有值之前。
- `temp`: 仅应在适配器上下文中设置变量。如果指定了可选的 `text`, 该文本将放在两个值之间。请参阅下面的示例。

value 必需。它是要在变量中设置的值。如果还指定了 `eval` 属性, 那么它可以是要求值的表达式。求值结果以值的形式存储。

示例:

以下示例对 `value` 属性中的表达式求值并将结果存储在变量 `LAST_VER` 中。它将设置为 `$LAST_VER` 和 `$1` 变量值之间的较大者。

```
<setenv group="Adaptor" name="LAST_VER"
value="$LAST_VER>$1?$LAST_VER:$1" eval="true" />
```

以下示例在向变量 `Changes` 追加数据之前插入换行符 (`\n`):

```
<setenv name="Changes" value="$4 - $1 - $6" type="temp append\n" />
```

以下示例在向变量 `INFOPATH` 追加的值之后插入冒号:

```
<setenv name="INFOPATH" value="/usr/local" type="temp prepend:" />
```

step

`<step>` 元素仅在 `<ontempenv>` 元素中使用。它指定特殊适配器步骤的结果。它是空元素。以下示例显示了 `<step>` 元素的两种形式。

```
<step result="FAIL"/>
```

```
<step result="pass"/>
```

第 27 章 Rational Build Forge 和 Rational ClearCase

BuildForge 能够与 Rational ClearCase 协作以进行源代码管理。

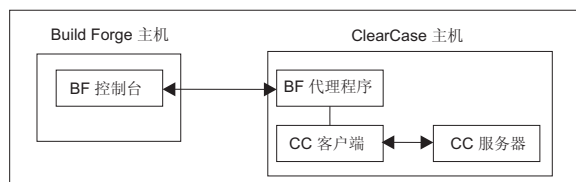
本主题描述与 Rational ClearCase 集成的其他设置需求，并概述了所提供样本适配器模板的特征。

ClearCase 适配器模板样本提供了几种方法来分析对基线的变更。通常，变更分析用于避免构建：如果基线组件尚未更改，那么不会重新构建

ClearCase 适配器的其他设置需求

请参阅第 422 页的『适配器需求』以了解一般需求。除了实现适配器的一般需求外，还要执行以下操作：

1. 在可与 ClearCase 服务器连接的主机上安装 Build Forge 代理程序。
2. 在代理程序主机上安装 ClearCase 完整客户机。
3. 设置代理程序的环境以便能够通过 ClearCase 客户机来运行命令。
4. 确定如何实施以及如何、何时启动所需的 ClearCase 视图。提供的模板假定使用动态视图，并包括在运行 cleartool 命令时启动该视图。在代理程序主机上安装 ClearCase 完整客户机。



ClearCase 视图

ClearCase 将视图用作您访问数据的方式。可以使用两种类型的视图：

- 动态视图：动态视图允许您查看指定的 VOB 中最新版本的元素，而无需将它们复制到本地系统中。它们必须启动。指定要在视图中安装的 VOB。
- 快照视图：快照视图是存储库的副本。您可以指定要使用的对象的版本。必须手动更新快照视图，才能获取已检入的最新项。

动态视图

动态视图显示每当在您选择查看的视图上下文中更改任何内容时，不断更新的工作空间。仅将您检出的元素复制到您的工作空间，并且必须将它们检入以供其他用户使用。使用动态视图时，需要使用三种规范来访问您所需的数据：

- 视图：定义要显示的版本的列表（通过配置规范）。必须使用命令来启动该视图。它在您的本地磁盘上表示为内容的根目录。也称为视图根。常用的目录如下所示：
 - Windows: 驱动器 M:（共享驱动器）
 - UNIX 或 Linux: /view（安装的文件系统）

- 视图上下文：定义要通过视图标记使用的内容的目录。视图标记在 ClearCase 中定义。启动视图后，目录将可供您使用。
- VOB（版本化对象库）：定义视图上下文的子目录。VOB 在 ClearCase 中定义。需要在视图上下文中显式安装 VOB。该子目录包含版本化工件（源文件和其他工件）。

本地机器上 VOB 的路径是从以下三项构建的：

- Windows 系统的示例路径，使用缺省值：M:\view_tag\vob_name
- UNIX 和 Linux 系统的示例路径，使用缺省值：/view/view_tag/vob_name

快照视图

使用快照视图时，需要指定视图上下文，并且还必须装入数据。这样您就拥有指定的所有数据的本地副本。必须重新装入快照视图，才能获取自上次装入数据以来检入的更改。

支持动态视图的 Rational Build Forge 功能部件

您可以使用特殊环境变量来定义和启动 Build Forge 作业的动态视图。

CLEARCASE_VIEW

启动指定的 ClearCase 视图。此变量中指定的视图必须存在，而且必须将使用此变量的步骤设置为“absolute”。在运行 Microsoft Windows 的系统上，该变量必须与 bfaagent.conf 中代理程序的 cc_suppress_server_root 参数一起使用。

_CLEARCASE_VIEWS

指定要在命令执行之前启动的 ClearCase 视图的列表。请将其值设置为视图的逗号分隔列表；例如，“View1,View2,View3”。

_CLEARCASE_VOBS

指定要在命令执行之前挂装的 ClearCase VOB 的列表。请将其值设置为 VOB 的逗号分隔列表；例如，“\Vob1,\Vob2,\Vob3”。

您还需要使用代理程序参数来管理代理程序主机上的视图。如果您要与 ClearCase 一起使用的代理程序运行的是 Windows 系统，那么必须设置代理程序参数。如果您要通过该代理程序使用的视图根不同于作业设置的视图根，那么这也非常有用。如果将作业用于多个代理程序并且需要在每个代理程序主机系统上使用不同的规范时，就会出现这种情况。

ccviewroot root-path

指定该主机的缺省视图根。请参阅关于 init 的 ClearCase 文档以了解更多信息。内部缺省值如下所示：

- Windows 系统：ccviewroot M:
- UNIX 或 Linux 系统：ccviewroot /view

cc_suppress_server_root

如果设置，那么查看路径为 ccviewroot 设置的路径。如果不设置，那么服务器定义中设置的路径将附加到 ccviewroot 设置的路径。此设置不需要值。如果 bfaagent.conf 中存在此设置，将设置该值。

ClearCase 适配器模板样本

提供了以下适配器模板样本。

ClearCaseBaseline

1. 扫描 ClearCase 视图中的目录。
2. 将 ClearCase 报告的分支和版本信息写入 BOM 报告。

所用的变量:

- INT_STREAM
- VIEW
- PROJECT_VOB
- CCSERVER
- UNIXCLIENT

ClearCaseByBaselineActivities

1. 从 ClearCase 视图的内容创建新的基线。
2. 将新基线与来自上次执行适配器的基线进行比较以识别更改活动。
3. 为每个更改活动将以下信息写入 BOM 报告: 活动、已更改文件、用户、日期、注释和版本。
4. 为更改的每个文件将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。

所用的变量:

- CurDate
- LAST_RUN
- BASELINE
- VIEW
- VOB_PATH
- PROJECT_VOB
- CCSERVER
- UNIXCLIENT

ClearCaseByBaselineVersions

1. 从 ClearCase 视图的内容创建新的基线。
2. 将新基线与来自上次执行适配器的基线进行比较以识别已更改文件。
3. 为每个已更改文件将以下信息写入 BOM 报告: 文件名、版本、日期、用户和注释。
4. 为更改的每个文件将更改详细信息 (来自 diff 命令输出) 写入 BOM 报告。

所用的变量:

- CurDate
- LAST_RUN
- LABEL
- BASELINE
- VIEW
- VOB_PATH
- PROJECT_VOB
- CCSERVER
- UNIXCLIENT

- _CHAR_NATIVE

ClearCaseByDate

1. 查询 ClearCase 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。
2. 为每个已更改文件将以下信息写入 BOM 报告：文件名、版本、日期、用户和注释。
3. 为更改的每个文件将更改详细信息（来自 diff 命令输出）写入 BOM 报告。

所用的变量：

- CurDate
- LAST_RUN
- LABEL
- BASELINE
- VIEW
- VOB_PATH
- PROJECT_VOB
- CCSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearCaseByLabel

1. 新建标签并将其应用于 ClearCase 视图的内容。
2. 将新标签与来自上次执行适配器的标签进行比较以识别已更改文件。
3. 为每个已更改文件将以下信息写入 BOM 报告：文件名、版本、日期、用户和注释。
4. 为更改的每个文件将更改详细信息（来自 diff 命令输出）写入 BOM 报告。

所用的变量：

- CurDate
- LAST_RUN
- LABEL
- BASELINE
- VIEW
- VOB_PATH
- PROJECT_VOB
- CCSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearCase 适配器变量

此表引用了适配器模板的变量列表。

环境变量名	描述
BASELINE	对于 ByBaseline 适配器，当您使用您的适配器来按基线生成差异时，系统将使用此值作为基线。
CCSERVER	将此变量设置为安装了 ClearCase 客户机和 Build Forge 代理程序的主机的名称。
CurDate	向适配器提供当前日期，使用 .date 命令以 ClearCase 期望的格式生成该日期。请不要更改此值。
LABEL	对于 ByLabel 适配器，当您使用您的适配器来按标签生成差异时（通过 ByLabel 适配器），系统将使用此值作为标签。
LAST_RUN	对于 ByDate 适配器，系统使用该值来确定是否发生了任何更改；该值是上一次成功运行的日期。可在测试适配器时操作此值，通过选择在某些更改之前的某个已知日期可强制适配器运行。如果适配器允许运行继续，它将自动把此值更新为当前日期。缺省值为 1-Jan-05.00:00:00。
PROJECT_VOB	使用 ByBaseline 适配器时，将此变量设置为项目 VOB 的名称（仅用于 UCM ClearCase）。示例: \ProjectVob
UNIXCLIENT	用于设置特定于平台的信息。如果客户机运行在 Windows 平台上，请设置为 0。如果客户机运行在 UNIX 或 Linux 平台上，请设置为 1。
VIEW	将此变量设置为要用于适配器的 ClearCase 视图的名称。
VOB_PATH	将此值设置为组件 VOB 的名称及其子目录（子目录可选）。对于多个名称，请使用以逗号分隔的列表。
_CHAR_NATIVE	内部使用并且始终设置为 1。

以下触发器变量还可用于启动视图和安装 VOB。但是，它们独立于在适配器变量中设置的视图和 VOB。

- CLEARCASE_VIEW
- _CLEARCASE_VIEWS
- _CLEARCASE_VOBS

请参阅第 270 页的『触发器变量引用』。

第 28 章 Rational Build Forge 和 Rational ClearQuest

BuildForge 能够与 Rational ClearQuest 协作以更新构建记录。

Build Forge 与 ClearQuest 具有两种集成方式:

- 根据作业状态来自动创建或更新构建记录。在项目环境中设置了必需的环境变量时, 将自动激活此功能。
- ClearQuest 适配器

这些功能是完全相互独立的。特别是, 适配器与为其创建的环境相关联。该环境中的变量独立于为激活自动构建记录而设置的变量。

设置自动生成构建记录

系统可以在您的 IBM Rational® ClearQuest® 数据库中自动创建构建记录, 并附带指向日志数据的链接。再则, 当作业通过时, 系统可以更新 ClearQuest 数据库, 标注作业已完成、记录结束时间以及完成的步骤汇总。该功能需要 Rational ClearQuest V7.0 或更高版本。

当您项目配置为更新 ClearQuest 数据库时, 系统执行创建或更新构建记录操作, 如下所述:

作业启动

当系统启动一个作业时, 系统将创建一个 ClearQuest 构建记录。构建记录处于“已提交”状态, 其中包含作业日志 URL、开始时间、发行版名称、标识以及表明“Build XYZ started.”的日志条目。如果源控制适配器取消了作业(例如, 由于找不到任何源更改), 那么不会创建任何 ClearQuest 构建记录。

注: 如果某项目链向其他项目, 那么新项目将获取其自己的唯一 ClearQuest 构建标识。

作业通过/失败

当作业通过或失败时, 系统会将 ClearQuest 中的构建状态更改为“已完成”或“失败”, 设置构建结束时间, 并在 ClearQuest 构建日志中存储作业步骤的摘要。该摘要包含每个步骤的名称、结果状态和服务器。

作业重新启动

当作业重新启动时, 系统会将 ClearQuest 中的构建状态更改为“已提交”, 并创建一个 ClearQuest 构建日志条目指出“已重新启动构建 XYZ”。

您可以通过特殊环境变量来配置自动更新构建记录。要将项目链接到 ClearQuest 数据库, 请确保在项目的环境中包含下表中的变量。

注: 项目环境中必须有这些变量。将它们添加至某个步骤还不够。但可以使用设置为“包含”类型的变量, 这一类型通过另一个环境来包含这些变量。此外, 由于 CQ_RELEASE_NAME 值是唯一可能随项目而变化的值, 因此您可能希望创建一个环境来包括其他变量, 并使用“包含”类型的变量将该环境包括在项目环境中, 您还可在其中将 CQ_RELEASE_NAME 指定为特定于项目的环境变量。

要通过 Build Forge 作业来激活自动更新构建记录，必须为项目设置以下环境变量。它们在步骤级别不起作用。

变量	描述
CQ_DBNAME	必需。要更新的 ClearQuest 数据库的名称。
CQ_DBSET	ClearQuest 数据库设置值。不是必需的。缺省为空。
CQ_INTERACTION	如果您的项目环境已定义用于启用 ClearQuest 构建记录创建的相应环境变量，而您不希望创建构建记录，请将此变量设置为 OFF 来禁用构建记录创建。 要启用创建构建记录，将此环境变量设置为 ON。 注： 如果要使用 ClearQuest 的任一适配器，将此环境变量设置为 OFF。适配器直接与构建记录交互。
CQ_PASSWORD	必需。登录到 ClearQuest 数据库时使用的密码。不是必需的；缺省为空。
CQ_RELEASE_NAME	必需。ClearQuest 数据库中要更新的发行版的名称。
CQ_USER	必需。用于登录到 ClearQuest 数据库的用户名。

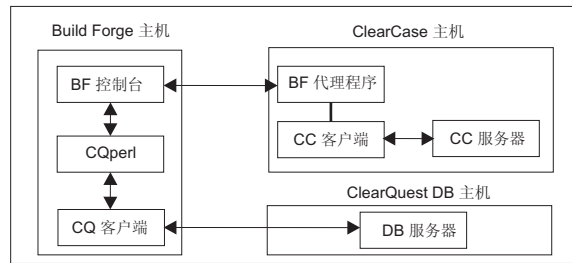
ClearQuest 适配器的其他设置需求

ClearQuest 适配器模板样本提供了几种方法来扫描 ClearCase 和更新 ClearQuest 中的构建记录。这通常与 Build Forge 中运行的构建的成功或失败有关。请参阅第 422 页的『适配器需求』以了解一般需求。另外，需要执行以下配置。

- 访问 ClearCase 以扫描源。作业运行期间，适配器通过代理程序和 ClearCase 客户机运行 cleartool 命令。适配器使用 ClearQuest Perl API (cqperl) 运行命令。
- 访问 ClearQuest 以更新构建记录。作业运行期间，适配器直接在控制台主机上运行 cqperl 脚本。它们由 Cqperl 实用程序解释，并通过 ClearQuest 客户机运行，后两者都安装在控制台主机上。

执行以下操作：

1. 在可与 ClearCase 服务器连接的主机上安装 Build Forge 代理程序。
2. 在代理程序主机上安装 ClearCase 完整客户机。
3. 设置代理程序的环境以便能够通过 ClearCase 客户机来运行命令。
4. 在 Build Forge 控制台主机上安装 ClearQuest 完整客户机。
5. 将 cqperl (ClearQuest Perl API) 目录添加到系统路径。
6. 定义 Build Forge 主机上 ClearQuest 客户机可用于访问 ClearQuest 数据库的连接。在 ClearQuest 客户机主机上执行这些操作。
 - a. 使用 cqreg 命令来添加数据库集 (cqreg add_dbset)。
 - b. 使用 CQ 维护工具来设置与 ClearQuest 数据库的连接。
7. 确定如何实施以及如何、何时启动所需的 ClearCase 视图。



您无需安装代理程序。ClearQuest 适配器使用 ClearQuest Perl API，通过客户机直接与 ClearQuest 进行通信。

要点：仅通过步骤中的点命令便可调用 ClearQuest 适配器。它不是源适配器，因此无法使用适配器链接。

ClearQuest 适配器模板样本

提供了以下适配器模板样本。

ClearQuestBaseClearCaseByDate

1. 查询 ClearCase 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。
2. 为每个已更改文件查找识别 ClearQuest 更改标识的 ClearQuest 超链接属性。通过添加作业信息以解析 ClearQuest 中的缺陷记录来尝试解析更改标识，前提是 ClearQuest 状态允许解析这些缺陷记录。
3. 为每个已更改文件将以下信息写入 BOM 报告：文件名、缺陷标识、缺陷状态和任何 ClearQuest 错误。

适配器模板中定义的变量：

- CurDate
- LAST_RUN
- VIEW
- VOB_PATH
- CQ_USER
- CQ_PASSWORD
- BFSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearQuestClearCaseByActivity

1. 查找与 ClearCase 活动列表关联的 ClearQuest 缺陷记录。
2. 它为发现的每个缺陷记录添加作业信息以解析 ClearQuest 内的缺陷记录，前提是 ClearQuest 状态允许解析这些缺陷记录。
3. 将以下信息写入 BOM 报告：与 ClearCase 活动标识和 ClearQuest 缺陷状态关联的文件。

适配器模板中定义的变量：

- CurDate
- VIEW

- VOB_PATH
- ACTIVITIES
- CQ_USER
- CQ_PASSWORD
- PROJECT_VOB
- BFSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearQuestUCMClearCaseByDate

1. 查询 ClearCase 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。它使用 Rational 统一变更管理 (UCM) 来生成结果。
2. 为每个已更改文件将以下信息写入 BOM 报告：文件名、缺陷标识、缺陷状态和任何 ClearQuest 错误。

适配器模板中定义的变量:

- CurDate
- LAST_RUN
- VIEW
- VOB_PATH
- CQ_USER
- CQ_PASSWORD
- BFSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearQuest 适配器变量

此表引用了适配器模板的变量列表。

表 19. Rational ClearQuest 集成所需的环境变量

变量	描述
ACTIVITIES	对于 ClearQuestClearCaseByActivity 适配器，这是一组用空格定界的活动标识。示例：SAMPL0001@\ProjectVob
BFSERVER	将此变量设置为 Build Forge 控制台的主机名。
CQ_PASSWORD	必需。登录到 ClearQuest 数据库时使用的密码。不是必需的；缺省为空。
CQ_USER	必需。用于登录到 ClearQuest 数据库的用户名。
CurDate	通过使用 .date 命令生成当前日期，以提供给适配器。请不要更改此值。
LAST_RUN	对于 ByDate 适配器，系统使用该值来确定是否发生了任何更改；该值是上一次成功运行的日期。可在测试适配器时操作此值，通过选择在某些更改之前的某个已知日期可强制适配器运行。如果适配器允许运行继续，它将自动把此值更新为当前日期。缺省值为 1-Jan-05.00:00:00。

表 19. Rational ClearQuest 集成所需的环境变量 (续)

变量	描述
UNIXCLIENT	用于设置特定于平台的信息。如果客户机运行在 Windows 平台上，请设置为 0。如果客户机运行在 UNIX 或 Linux 平台上，请设置为 1。
VIEW	将此变量设置为要用于适配器的 ClearCase 视图的名称。
VOB_PATH	将此值设置为组件 VOB 的名称及其子目录（子目录可选）。对于多个名称，请使用以逗号分隔的列表。
_CHAR_NATIVE	内部使用并且始终设置为 1。

重新启动由 ClearQuest 集成的作业

完成由 ClearQuest 集成的作业后，通常无法在 Rational Build Forge 中重新启动。作为简单的变通方法，您可以将此作业作为新作业启动。

要能够重新启动，您必须使用 ClearQuest 设计器工具编辑 ClearQuest 模式。必须修改 Build 记录的工作流，以允许状态从“已完成”转变为“提交”。

第 29 章 Rational Build Forge 和 CVS

BuildForge 能够与 CVS 协作以进行源代码管理。

本主题描述与 CVS 集成的其他设置需求，并概述了所提供样本适配器模板的特征。

CVS 适配器模板样本提供了几种方法来分析对基线的变更。通常，变更分析用于避免构建：如果基线组件尚未更改，那么不会重新构建

CVS 适配器的其他设置需求

请参阅第 422 页的『适配器需求』以了解一般需求。对于 CVS，没有任何其他需求。

CVS 适配器模板样本

提供了以下适配器模板样本。

CVSv1Baseline

1. 扫描 Build Forge 代理程序上的 CVS 目录以查找已更改文件。
2. 将以下信息写入 BOM 报告：已更改文件名、状态、工作版本、存储库版本和固定标记。扫描 ClearCase 视图中的目录。

所用的变量：

- CVSROOT
- MODULE
- CVSCLIENT

CVSv1ByDate

1. 查询 CVS 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。
2. 将以下信息写入 BOM 报告：更改类型、日期、用户名、版本和文件名。
3. 为更改的每个文件将更改详细信息（来自 diff 命令输出）写入 BOM 报告。

所用的变量：

- LAST_RUN
- CurDate
- CVSROOT
- MODULE
- BRANCH
- CVSCLIENT

CVSv1ByTag

1. 将新标记应用于 CVS 模块。
2. 比较新标记的模块与上次执行适配器期间标记的模块之间的差别。
3. 将以下信息写入 BOM 报告：文件名、修订、状态、日期、时间、更改作者和提交注释。

4. 为更改的每个文件将更改详细信息（来自 diff 命令输出）写入 BOM 报告。

所用的变量:

- LAST_RUN
- CurDate
- CVSROOT
- MODULE
- BRANCH
- LAST_TAG
- CVSCLIENT

CVSv2ByDate

1. 查询 CVS 视图以查找两个日期之间的更改。缺省日期是当前时间戳记和上次执行适配器的时间戳记。

2. 将以下信息写入 BOM 报告：更改类型、日期、用户名、版本和文件名。

3. 为更改的每个文件将更改详细信息（来自 diff 命令输出）写入 BOM 报告。

所用的变量:

- LAST_RUN
- CurDate
- CVSROOT
- MODULE
- CVSCLIENT

第 30 章 IDE 集成

可以使用插件将 Rational Build Forge 与集成开发环境 (IDE) 进行集成。

本部分描述如何安装和使用可用于从集成开发环境 (IDE) 访问管理控制台功能的插件。

关于 IDE 集成

开发者可以使用集成开发环境 (IDE) 的插件连接到管理控制台。

使用提供的插件，开发者可以直接从其 IDE 完成以下任务：

- 查看项目
- 运行作业
- 检查作业结果

其他能力视每个插件而定。

每个开发者使用 Build Forge 从用户列表中可以识别的用户名连接到控制台。对项目的访问权受用户名在访问组中的成员资格控制。对项目步骤的访问权也受访问组控制。在步骤中，可以明确指定访问组。如果未指定，那么继承项目访问组。

这些插件不提供编辑或删除项目和步骤的功能。

每个通过 IDE 访问 Build Forge 的用户使用一个许可证，这正如用户通过浏览器客户机会话使用许可证一样。

为以下 IDE 提供插件：

- Eclipse[™]
- Rational[®] Application Developer，它是基于 Eclipse[™] 的 IBM IDE
- Rational[®] Team Concert，它是 IBM 分发的 Jazz.net

测试项目的特殊变量

在使用插件运行项目的测试构建时，可以通过一些特殊的环境变量，指定要在将文件从系统复制到服务器之前和之后运行的命令。

所有这些命令都在项目目录中运行：

- PRECMD 变量用于对从开发者的计算机复制到正在运行构建的服务器的目录和文件运行命令。该命令在项目步骤之前运行。示例：该命令可用于在复制文件之前将文件从源控制系统中检出。
- POSTCMD 变量用于在项目步骤运行之后对目录和文件运行命令。示例：该命令可以用于（在使用此概念的源控制系统中，如 Rational ClearCase）释放检出的虚拟目录。

对于在 Reflector 插件中标记为 Build Forge 项目工件的目录和文件，要运行一些命令。将在遍历 Reflector 插件的目录树时应用这些命令。

注：目录树的遍历对于 PRECMD 命令是广度优先向下进行，对于 POSTCMD 命令是逆向进行。在遍历期间，将根据情况运行针对目录的命令和针对文件的命令。

_PRISM_DIR_PRECMD

指定在树遍历期间遇到目录时对目录运行的命令。该命令对每个至少包含一个文件的目录运行一次。系统会将命令中的第一个 \$1 替换为目录名称。

_PRISM_FILE_PRECMD

指定在树遍历期间遇到文件时对文件运行的命令。该命令对每个文件运行一次。系统会将命令中的第一个 \$1 替换为文件名。

_PRISM_DIR_POSTCMD

指定在树遍历期间遇到目录时对目录运行的命令。该命令对每个至少包含一个文件的目录运行一次。系统会将命令中的第一个 \$1 替换为目录名称。

_PRISM_FILE_POSTCMD

指定在树遍历期间遇到文件时对文件运行的命令。该命令对每个文件运行一次。系统会将命令中的第一个 \$1 替换为文件名。

Eclipse 和 Rational Application Developer 的插件

插件用于从 Eclipse™ 和 Rational® Application Developer IDE 内访问管理控制台功能。

Eclipse 和 Rational Application Developer 环境可以使用以下插件：

Frequency

使用 Frequency 插件，开发者可以完成以下任务：

- 访问一个或多个管理控制台以查看项目
- 启动作业
- 查看作业状态
- 查看正在运行和已完成作业的构建日志

Reflector

Reflector 插件使用本地环境中的文件运行作业。运行这些作业通常是为了测试新代码，通过测试后再将其检入到源控制中供其他开发者或生产构建使用。

Eclipse 插件用户可以选择覆盖项目环境变量的值。启动 Build Forge 项目时，会显示“作业设置”弹出窗口。对环境变量的更改将仅应用于此作业。不会更改项目的缺少变量值。

在 Eclipse 或 Rational Application Developer 中使用插件

安装这些插件之后，可以以下列方式将其激活：

- 要访问管理控制台以启动作业和查看项目日志，请使用 Frequency 插件。在 IDE 内，选择窗口 > 打开透视图 > 其他。IDE 将显示一个对话框，其中包含透视图类型列表；请选择 Build Forge 透视图。系统将显示“控制台浏览器”、“构建信息”和“构建日志”窗口。右键单击控制台浏览器并选择新建控制器以配置管理控制台连接。有关使用 Frequency 的更多信息，请参阅插件随附的联机帮助。

注：如果需要配置对 LDAP/Active Directory 域的访问，请务必为该 LDAP 域使用 Build Forge 系统名称，而不是域的实际名称。

- 要运行测试构建，请使用 Reflector 插件。在 IDE 内，通过选择您的项目并右键单击来配置 Reflector。从弹出菜单选择**属性**。在**属性**对话框的属性选项列表中，选择 **Build Forge 项目工件**。使用希望您的项目处理的 Build Forge 项目配置该对话框，然后选择要上传到系统的文件。有关更多信息，请参阅插件随附的联机帮助。

注：Reflector 插件需要 Frequency 插件。

安装 Eclipse 或 Rational Application Developer 插件

从 Build Forge 服务器安装 IDE 环境插件。

开始之前

先决条件：

- Eclipse V3.0.2 或更高版本或 Rational Application Developer V7.0 或更高版本
- Java 2 SE V5.0
- Build Forge 系统必须运行

关于此任务

要安装插件，请从 IDE 内完成此过程。

过程

1. 选择**帮助 > 软件更新 > 查找和安装**。
2. 单击**搜索要安装的新功能**单选按钮，然后单击**下一步**。系统将显示**要访问的更新站点**对话框。
3. 单击**新建远程站点**按钮。系统将显示**新建远程站点**对话框。
 - a. 在名称字段中输入“Build Forge 更新站点”。
 - b. 在 **URL** 字段中，使用您的管理控制台计算机的主机名，输入以下更新站点 URL: `http://<console_host_name>/prism/eclipse/updateSite/site.xml`。
 - 如果运行 Eclipse 的系统与运行 Build Forge 的系统相同，可以使用 localhost 作为主机名。
 - 如果控制台不在 80 端口上运行，请包括端口号。例如：`http://myhostname:11812/prism/eclipse/updateSite/site.xml`
 - c. 单击**确定**。
4. 在**要访问的更新站点**对话框中，选中 **Build Forge 更新站点**复选框，然后单击**完成**。
5. 系统在**搜索结果**对话框中显示可用插件列表。选择提供的所有插件，然后单击**下一步**。

注：Reflector 插件需要 Frequency 插件。如果单独安装，那么无法运行。

6. 阅读许可证协议，选择**我接受许可证协议中的条款**，然后单击**下一步**。
7. 选择要安装功能的位置。要添加新位置，单击**新位置**，并浏览至所需位置。
8. 单击**完成**。
9. 如果显示**功能验证**对话框，请单击**安装**。出现对话框是因为插件是未签署的功能。如果没有选择**全部安装**，每安装一种功能就显示一次该对话框。
10. 要求您重新启动 Eclipse 才能使更改生效。单击**是**。

SSL 启用时的备选安装

如果 Build Forge 系统上启用了 SSL，请使用备选安装方法。

开始之前

当前版本的 Eclipse 和 Rational Application Developer 中未启用 SSL。因此，当 Build Forge 系统中启用了 SSL 时，从 Build Forge 系统无法安装 Eclipse 和 Rational Application Developer 插件。作为变通方法，使插件安装文件可在非安全 Web 服务器上获取，或者将这些文件手动分发给用户。

要将文件打包：

1. 创建名为 prism 的目录。
2. 在 prism 中，创建名为 eclipse 的目录。
3. 在 eclipse 中，创建名为 updateSite 的目录。

将以下文件从 `<bfinstall>/webroot/public/prism/eclipse/updateSite` 复制到您创建的 updateSite 目录：

- features 目录
- plugins 目录
- site.xml 文件

关于此任务

提供 prism 目录后，用户就可以从其 IDE 内执行以下步骤：

过程

1. 选择帮助 > 软件更新 > 查找和安装。
2. 单击搜索要安装的新功能单选按钮，然后单击下一步。系统将显示要访问的更新站点对话框。
3. 新建站点。选择以下过程之一。

- 从远程服务器获取文件：

- a. 单击新建远程站点按钮。系统将显示新建远程站点对话框。
- b. 在名称字段中输入“Build Forge 更新站点”。
- c. 输入文件位置：

`http://host/path/prism/eclipse/updateSite/site.xml`

host 是 Web 服务器的主机名或 IP 地址。

path 是您放置文件的服务器根路径。

- d. 单击确定。

- 从本地主机获取文件：

- a. 单击新建本地站点按钮。系统显示新建本地站点对话框。
- b. 在名称字段中输入“Build Forge 更新站点”。
- c. 输入文件位置：

`file://path/prism/eclipse/updateSite/site.xml`

path 指定文件的位置。

- d. 单击**确定**。
 4. 在**要访问的更新站点**对话框中，选中 **Build Forge** 更新站点复选框，然后单击**完成**。
 5. 系统在**搜索结果**对话框中显示可用插件列表。选择提供的所有插件，然后单击**下一步**。
- 注：Reflector 插件需要 Frequency 插件。如果单独安装，那么无法运行。
6. 阅读许可证协议，选择**我接受许可证协议中的条款**，然后单击**下一步**。
 7. 选择要安装功能的位置。要添加新位置，请单击**新位置**，然后浏览至所需的位置。
 8. 单击**完成**。
 9. 如果显示**功能验证**对话框，请单击**安装**。出现对话框是因为插件是未签署的功能。如果没有选择**全部安装**，每安装一种功能就显示一次该对话框。
 10. 要求您重新启动 Eclipse 才能使更改生效。单击**是**。

使用 Eclipse 和 Rational Application Developer 的插件

要访问启动作业并查看项目日志（Frequency 插件）：

1. 在 IDE 内，选择**窗口 > 打开透视图 > 其他**。
2. 选择 **Build Forge** 透视图。透视图包括以下窗口：
 - 控制台浏览器
 - 构建信息
 - 构建日志
3. 右键单击**控制台浏览器**并选择**新建控制台**以配置管理控制台连接。
4. 在 **Build Forge** 服务层主机名中输入主机名或 IP 地址，验证或编辑其他字段，然后单击**确定**。

建立连接后，“构建信息”窗口用您可以运行的作业进行填充。要使用本地文件运行作业，请将作业配置为使用应使用的文件。

1. 在您的 IDE 内，连接到 Build Forge。
2. 在**控制台浏览器**窗口中，右键单击某个作业并选择**属性**。
3. 在**属性**对话框中，选择 **Build Forge** 项目工件。
4. 在 **Build Forge** 项目工件中，选择要处理的项目并选择要使用的本地文件。

有关更多信息，请参阅插件随附的联机帮助。

Rational Team Concert 插件

Rational Team Concert 客户机插件是集成 Rational Team Concert 和 Build Forge 的组件。还需要服务器扩展和适配器模板。如果设置了与 Rational Team Concert 集成，Rational Team Concert 用户可以完成以下任务：

- 将 Build Forge 设置为 RTC 构建服务器
- 将 Build Forge 项目设置为 RTC 构建定义
- 从 RTC 客户机查看项目、运行作业和查看作业结果

有关设置集成的指示信息，请参阅 第 481 页的第 33 章，『Rational Build Forge 和 Rational Team Concert』。

使用 Rational Team Concert 插件

以下说明假设 Rational Team Concert 集成已经设置，且插件已经安装在 Rational Team Concert 客户机。

有关设置集成的指示信息，请参阅 第 481 页的第 33 章，『Rational Build Forge 和 Rational Team Concert』。

要设置构建定义并运行构建：

1. 设置构建定义。
 - a. 在“团队工件”视图中，展开项目文件夹。
 - b. 右键单击**构建**，然后单击**新建构建定义**。
 - c. 在“新建构建定义”中，选择**新建构建**，然后单击**下一步**。
 - d. 在“常规信息”中，输入构建标识和描述。在“可用模板”列表中，选择 **Rational Build Forge**。单击**下一步**。
 - e. 在“更多配置”中，选择“常规”和“属性”，并单击**完成**。此时创建了一个选项卡，标有您输入的构建标识。
 - f. 单击 **Build Forge** 选项卡。
2. 为构建定义选择项目。
 - a. 单击 **Build Forge** 选项卡。输入连接到 Build Forge 所需的信息：
 - 主机名 - 运行 Build Forge 的主机的主机名。这必须与**控制台 URL** 系统设置（如果已设置）相匹配。如果无法访问**管理 > 系统**中的设置，请联系管理员。
 - 端口 - 用于与 Build Forge 通信的端口。3966 是缺省端口。如果选择了**安全连接 Build Forge**，缺省情况下会显示端口 49150。如果安装时使用其他端口，请输入您的端口号。这必须与**控制台 URL** 系统设置中的端口设置（如果已设置）相匹配。如果无法访问**管理 > 系统**中的设置，请联系管理员。
 - 用户名 - 连接 Build Forge 所需的用户名。该用户名必须已经存在于 Build Forge 中。
 - 密码 - 用户名的密码
 - 确认密码 - 重新输入用户名的密码
 - b. 单击**获取项目**
 - c. 在显示的 **Build Forge** 项目列表中，为此构建定义选择项目。
3. 请求构建。
 - a. 右键单击构建定义，然后选择**请求构建**。
 - b. 指定所需的构建选项和构建属性，然后单击**提交**。
4. 检查构建结果。
 - a. 构建完成后，在**构建**选项卡下的列表中选择该构建。随即显示一个窗口。
 - b. 在“外部链接”下，单击 **Build Forge 结果链接**。
 - c. 此时显示 Build Forge 登录面板。登录。
 - d. 转至**主页 > 已完成的运行**，然后从列表中选择您的构建。

- e. 此时显示构建步骤和结果。单击步骤链接查看步骤日志。
- f. 完成后，单击**注销**并关闭窗口。

对 Rational Team Concert 插件进行故障诊断

本部分描述了已知问题和如何解决这些问题。

具有“过期”状态的作业

如果作业保持在“过期”状态超过几分钟时间，那么它可能被挂起。

变通方法

停止并重新启动 Build Forge 引擎。请参阅第 229 页的『启动和停止引擎』。

第 31 章 Rational Build Forge 和 Perforce

BuildForge 能够与 Perforce 协作以进行源代码管理。

本主题描述与 Perforce 集成的其他设置需求，并概述了所提供样本适配器模板的特征。

Perforce 适配器模板样本提供了几种方法来分析对基线的变更。通常，变更分析用于避免构建：如果基线组件尚未更改，那么不会重新构建

Perforce 适配器的其他设置需求

请参阅第 422 页的『适配器需求』以了解一般需求。

1. 在与代理程序相同的主机上安装 P4 命令行客户机。
2. 设置 Perforce 用户帐户以供 Build Forge 使用。
3. 设置代理程序环境，以便其可以访问客户机。

Perforce 适配器模板样本

提供了以下适配器模板样本。

PerforceByDate

1. 查询 Perforce 客户机以查找执行适配器以来发生的更改。
2. 将以下信息写入 BOM 报告：更改、日期、时间、用户、Perforce 客户机和注释。
3. 将更改详细信息（来自 diff 命令输出）写入 BOM 报告。

适配器中设置的变量：

- LAST_RUN
- CurDate
- P4PORT
- BFCLIENT
- P4CLIENT
- FILESPEC

环境所需的其他变量：

- P4USER
- P4PASSWD

不能对这些变量使用“指定隐藏”属性。用户和密码将以明文形式写入到步骤日志中。

PerforceByRev

1. 查询 Perforce 客户机以查找自上次存储库修订以来发生的更改。
2. 将以下信息写入 BOM 报告：更改、日期、时间、用户、Perforce 客户机和注释。
3. 将更改详细信息（来自 diff 命令输出）写入 BOM 报告。

适配器中设置的变量:

- LAST_RUN
- CurDate
- P4PORT
- BFCLIENT
- P4CLIENT
- FILESPEC

环境所需的其他变量:

- P4USER
- P4PASSWD

不能对这些变量使用“指定隐藏”属性。用户和密码将以明文形式写入到步骤日志中。

第 32 章 Rational Build Forge 和 Rational Quality Manager

Rational Build Forge 可以与 Rational Quality Manager 协作

您可以将 Rational® Build Forge® 与 Rational Quality Manager 集成以定义和运行通过 Rational Quality Manager 对软件进行构建和打包的项目。

集成在 Rational Quality Manager 中进行。请参阅信息中心以获取您的 Rational Quality Manager 版本。请参阅 Rational Quality Manager 信息页面以获取链接。

- 有关 Rational Quality Manager V2，请参阅集成 > **Rational Quality Manager** 以及软件库存和自动化工具 > 与 **Rational Build Forge** 集成
- 有关 Rational Quality Manager V1，请参阅集成 > 与软件库存和供应工具集成 > 与 **Rational Build Forge** 集成

第 33 章 Rational Build Forge 和 Rational Team Concert

Build Forge 可以与 Rational Team Concert 集成。

设置集成后，Rational Team Concert 的用户可以执行以下操作：

- 将 Build Forge 设置为 RTC 构建服务器
- 从 RTC 客户机查看项目、运行作业和查看作业结果
- 将 Build Forge 项目设置为 RTC 构建定义

Rational Team Concert V3.x

集成组件随 Rational Team Concert 安装提供。

Rational Team Concert 3.x 和 Rational Build Forge V7.1.1.3 及更高版本之间的集成相对于与先前版本的 Rational Team Concert 之间的集成而言，具有明显的优势。Rational Team Concert 3.x 是与 Rational Build Forge 集成的首选平台。

安装 Rational Build Forge 和 Rational Team Concert 之后，在 Rational Team Concert 中配置集成。通常，将两个产品安装在不同的主机上。

要配置集成并对其进行故障诊断，请参阅 jazz.net 上的 Rational Build Forge wiki 以获取指示信息。

Rational Team Concert V1.x 和 V2.x

您需要安装随 Rational Build Forge 提供的组件，以便与 Rational Team Concert V1.x 和 V2.x 集成

集成要求您执行以下操作：

- 安装 Rational Build Forge 中的组件
- 配置 Rational Team Concert 以与 Rational Build Forge 通信

安装组件

安装以下组件：

- Rational Team Concert 服务器的服务器扩展：该组件支持 Rational Team Concert 与 Build Forge 之间的通信。服务器扩展安装在 RTC 服务器上。第 482 页的『安装 Rational Team Concert 服务器扩展』中提供了安装指示信息。将 Build Forge 用户分配到 Rational Team Concert 服务器。
- 客户机插件：此组件是用于 Rational Team Concert 客户机的插件。它允许用户访问 Build Forge 以查看项目、运行作业和查看作业结果。必须在每个将访问 Build Forge 的 RTC 客户机上安装客户机插件。第 483 页的『为 Rational Team Concert 安装客户机插件』中提供了安装指示信息。
- 适配器：此组件是允许项目访问 Rational Team Concert 存储库中文件的源类型适配器。RTC V1.x 和 RTC V2.x 使用不同的适配器。请参阅第 485 页的『配置 Rational Team Concert 适配器』。

配置 Rational Team Concert

要配置集成并对其进行故障诊断，请参阅 jazz.net 上的 Rational Build Forge wiki 以获取指示信息。

安装 Rational Team Concert 服务器扩展

开始之前

先决条件:

- Rational Team Concert 的版本必须是 1.x 或 2.x。要注意过程中由于版本差异导致的指令和文件的不同。
- Build Forge 系统必须正在运行。

关于此任务

从 Rational Team Concert 服务器主机执行以下步骤:

过程

1. 登录到 Rational Team Concert 服务器主机。
2. 使用适用于您的 Rational Team Concert 版本的 URL 浏览到 Build Forge 服务器，将服务器扩展下载到某个临时位置。
 - Rational Team Concert version 1.x:
`http://<bf_console_hostname>/rtc-server/BuildForgeConnectorServer.zip`
 - Rational Team Concert version 2.x:
`http://<bf_console_hostname>/rtc2-server/BuildForgeConnectorServer.zip`
 - 如果您正在运行 Build Forge 的同一系统上运行 Rational Team Concert，那么可以将 localhost 用作主机名。
 - 如果控制台不在 80 端口上运行，请包括端口号。示例：`http://myhostname:11812/clients`。
3. 将文件内容抽取到 `RTC_install/jazz/server`。如果您下载了 Rational TeamConcert V1.x 的扩展，那么增加了以下文件:

```
RTC_install/jazz/server/buildforgeconnector-update-site (directory)
RTC_install/jazz/server/provision_profiles/buildforgeconnector-profile.ini
```

如果您下载了 Rational Team Concert V2.x 的扩展，那么增加了以下文件:

```
RTC_install/jazz/server/buildforgeconnector-update-site (directory)
RTC_install/jazz/server/conf/provision_profiles/buildforgeconnector-profile.ini
```

要在 WebSphere Application Server 中运行 RTC 服务器，您必须编辑 `buildforgeconnector-profile.ini` 文件，以设置 `buildforgeconnector-update-site` 目录的标准路径。

4. 重新启动 Rational Team Concert 服务器。

结果

Rational Team Concert 服务器使用 Build Forge 用户许可证与控制台通信。

为 Rational Team Concert 安装客户机插件

从 Build Forge 服务器为 Rational Team Concert 安装客户机插件。

开始之前

先决条件:

- Rational Team Concert V1.x 或 V2.x - 每个都需要不同的插件。
- Build Forge 连接器服务器扩展必须安装在 Rational Team Concert 服务器上。
- Build Forge 系统必须正在运行。
- 每个用户都应用两个用于控制台的登录标识。Build Forge 将每个用户的登录会话限制为一个。同一用户名不能同时通过管理控制台和 RTC 客户机登录。新会话启动时，任何现有会话都会终止。RTC 客户机用户应该有两个标识：
 - UserRTC - 当他们配置构建定义来运行构建时使用。此用户名用于访问 Build Forge 控制台以运行构建。
 - UserBF - 当他们检查构建结果时使用。需要能够在 RTC 客户机中的窗口中直接登录到 Build Forge 管理控制台。

关于此任务

必须在每个将访问 Build Forge 的 Rational Team Concert 客户机上安装客户机插件。

要安装这些插件，请从 Rational Team Concert 客户机执行以下步骤。

过程

1. 选择帮助 > 软件更新 > 查找和安装。
2. 选择搜索要安装的新功能，然后单击下一步。系统将显示要访问的更新站点对话框。
3. 单击新建远程站点。系统将显示新建远程站点对话框。
 - a. 在名称字段中输入 Build Forge Connector Update Site。
 - b. 在 **URL** 字段输入更新站点的 URL。该 URL 随着您使用的 Rational Team Concert 版本不同而不同。

Rational Team Concert version 1.x:
`http://<console_host_name>/rtc/site.xml`

Rational Team Concert version 2.x:
`http://<console_host_name>/rtc2/site.xml`

 - 如果您正在运行 Build Forge 的同一系统上运行 Rational Team Concert，那么可以将 localhost 用作主机名。
 - 如果控制台不在 80 端口上运行，请包括端口号。如果您使用的是 Rational Team Concert V1.x，且 Build Forge 运行于端口 11812 上，那么以下示例适用。`http://myhostname:11812/rtc/site.xml`
 - c. 单击确定。
4. 在更新要访问的站点对话框中，选中 **Build Forge 连接器更新站点** 复选框，然后单击完成。
5. 系统将在搜索结果对话框中显示更新站点上可用功能的列表。选择提供的所有功能，然后单击下一步。
6. 阅读许可证协议，选择我接受许可证协议条款，再单击下一步。

7. 选择要安装功能的位置。 要添加新的位置，请单击**更改位置**，然后浏览到您期望的位置。
8. 单击**完成**。
9. 如果显示**功能验证**对话框，请单击**安装**。 出现对话框是因为插件是未签署的功能。如果没有选择**全部安装**，每安装一种功能就显示一次该对话框。
10. 系统会询问您是否要重新启动 Rational Team Concert 以使更改生效。单击**是**。
11. 重新启动 Build Forge。

结果

完成集成后，就可以使用该插件运行作业和检查作业结果了。必须指定 RationalBuildForgeConnector 作为您构件的构件引擎。

SSL 启用时的备选安装

如果 Build Forge 系统上启用了 SSL，请使用备选安装方法。

开始之前

Rational Team Concert 的当前版本不支持 SSL。因此，如果 Build Forge 系统已启用 SSL，那么无法从 Build Forge 系统安装插件。作为变通方法，使插件安装文件可在非安全 Web 服务器上获取，或者将这些文件手动分发给用户。

需要以下目录中的文件：<bfinstall>/webroot/public/rtc（针对 RTC V1.x）或 <bfinstall>/webroot/public/rtc2（针对 RTC V2.x）：

- features 目录
- plugins 目录
- site.xml 文件

关于此任务

您使 rtc 或 rtc2 目录可用后，用户即可从其 Rational Team Concert 客户机执行以下步骤：

过程

1. 选择**帮助 > 软件更新 > 查找和安装**。
2. 单击**搜索要安装的新功能**，然后单击**下一步**。 系统将显示**要访问的更新站点**对话框。
3. 新建站点。 选择以下过程之一。
 - 从远程服务器获取文件：
 - a. 单击**新建远程站点**。系统将显示**新建远程站点**对话框。
 - b. 在名称字段中输入“Build Forge 连接器更新站点”。
 - c. 输入文件位置：
`http://host/prism/eclipse/updateSite/site.xml`

host 是 Web 服务器的主机名或 IP 地址。

path 是您放置文件的服务器根路径。
 - d. 单击**确定**。

- 从本地主机获取文件:
 - a. 单击**新建本地站点**。系统显示**新建本地站点**对话框。
 - b. 在名称字段中输入“Build Forge 连接器更新站点”。
 - c. 输入文件位置:

`file://path/prism/eclipse/updateSite/site.xml`

path 指定文件的位置。

- d. 单击**确定**。
 4. 在**更新要访问的站点**对话框中，选中 **Build Forge 连接器更新站点**复选框，然后单击**完成**。
 5. 系统在**搜索结果**对话框中显示可用功能的列表。选择提供的所有功能，然后单击**下一步**。
- 注：** Reflector 插件需要 Frequency 插件。如果单独安装，那么无法运行。
6. 阅读许可证协议，选择**我接受许可证协议中的条款**，然后单击**下一步**。
 7. 选择要安装功能的位置。要添加新的位置，请单击**更改位置**，然后浏览到您期望的位置。
 8. 单击**完成**。
 9. 如果显示**功能验证**对话框，请单击**安装**。出现对话框是因为插件是未签署的功能。如果没有选择**全部安装**，每安装一种功能就显示一次该对话框。
 10. 系统会询问您是否要重新启动 Rational Team Concert 以使更改生效。单击**是**。
 11. 重新启动 Build Forge。

配置 Rational Team Concert 适配器

关于此任务

用于将 Rational Build Forge 连接到 Rational Team Concert 源存储库的适配器是随 Build Forge 一起自动安装的。该适配器在 Rational Build Forge 控制台主机上运行，并作为客户端与 Rational Team Concert 直接通信。您无需在 Rational Build Forge 控制台主机上安装 Rational Team Concert 客户机。您可以按如下所述配置各个 Build Forge 项目以访问 Rational Team Concert 源。

过程

1. 定义服务器资源。
 - 输入此服务器资源的**名称**。
 - 输入**主机**：这是运行 Rational Team Concert 服务器的主机的标准域名。

注： 通常，该属性指的是 Build Forge 代理程序。您无需在 Rational Team Server 主机上安装 Build Forge 代理程序。该属性专门用于直接与 Rational Team Concert 通信。

- 根据需要输入其他属性的值。

注： RTC 工具 scm 必须在 RTC 服务器的用户概要文件或启动概要文件的 PATH 上。适配器使用 scm 来访问源文件。

2. 为使用 Rational Team Concert 存储库的每个项目设置适配器链接。以下指示信息描述了如何为项目设置适配器链接。

- a. 在控制台中，转至项目 > 适配器链接。
 - b. 单击**新建适配器链接**。在详细信息选项卡中设置其属性，然后单击**保存**。
 - 状态 - 选择**活动**。
 - 名称 - 输入适配器链接的名称。
 - 适配器
 - 对于 Rational Team Concert V1.x，请选择 JazzSCM
 - 对于 Rational Team Concert V2.x，请选择 JazSCMv2
 - 项目 - 选择要应用适配器链接的项目。
 - 环境 - 选择要用于此适配器链接的环境。它必须是现有的环境。
 - c. 单击刚创建的适配器链接。
 - d. 单击**填充环境**。此步骤用适配器（JazzSCM.xml）中定义的变量来填充指定环境。
 - e. 单击**保存**。
3. 更新适配器提供的变量。在您设置并填充的环境中，编辑适配器提供的变量。

适配器中设置了四个环境变量：

- **Current_Date** - 设置当前日期。它用于应用时间戳记。不要更改此定义。
- **Last_Run** - 由系统自动更新。不要更改此定义。
- **Directory_Path** - 设置从存储库检索到的源文件的位置。缺省情况下它设置为 C:\temp。将此目录更改为您希望使用的临时目录。

缺省情况下，运行作业后并不清除该目录。从已运行的作业删除旧目录。

- **Jazz_Server** - 设置 Rational Team Concert 服务器的位置。缺省情况下它设置为 \$BFServer。必须将此设置更改为指向 Rational Team Concert 服务器的服务器资源的**名称**属性。
- a. 在控制台中，转至**环境**。
 - b. 选择适配器链接使用的环境。
 - c. 选择并编辑您希望更改的变量。
 - d. 单击**保存**。

结果

您现在可以运行该项目。每次运行时，项目都会连接到 Rational Team Concert 存储库。它更新自从上次项目运行以来在 RTC 服务器上已更改的源文件。

更多资源：

- Rational Team Concert V1.x: JazzSCM 适配器位于 `<bfinstall>/interface/JazzSCM.xml` 中。您可以用文本编辑器或 XML 阅读器打开该适配器。
- Rational Team Concert V2.x: JazSCMv2 适配器位于 `<bfinstall>/interface/JazSCMv2.xml` 中。您可以用文本编辑器或 XML 阅读器打开该适配器。
- Rational Team Concert 文档中记载了适配器使用的 SCM 命令。测试服务器连接和适配器所用命令时可以使用这些命令及其他命令。

第 34 章 Rational Build Forge 和 StarTeam

BuildForge 能够与 StarTeam 协作以进行源代码管理。

本主题描述与 StarTeam 集成的其他设置需求，并概述了所提供样本适配器模板的特征。

StarTeam 适配器模板样本提供了几种方法来分析对基线的变更。通常，变更分析用于避免构建：如果基线组件尚未更改，那么不会重新构建

StarTeam 适配器的其他设置需求

请参阅第 422 页的『适配器需求』以了解一般需求。

1. 在与代理程序相同的主机上安装 StarTeam 命令行客户机。
2. 设置 StarTeam 用户帐户以供 Build Forge 使用。
3. 设置代理程序环境，以便其可以访问客户机。

StarTeam 适配器模板样本

提供了以下适配器模板样本。

StarTeamBaseline

1. 查询文件夹以获取 StarTeam 视图来收集有关文件的信息。
2. 将以下信息写入 BOM 报告：文件名、状态、修订和分支。

适配器中设置的变量：

- USER
- PASS
- HOST
- PORT
- PROJECT
- VIEW
- DIR
- BFCLIENT

环境所需的其他变量：

- P4USER
- P4PASSWD

不能对这些变量使用“指定隐藏”属性。用户和密码将以明文形式写入到步骤日志中。

StarTeamByDate

1. 使用 StarTeam API 查询 StarTeam 视图以识别当前日期与上次执行适配器之间的更改。
2. 将以下信息写入 BOM 报告：已更改的文件与目录、用户、版本、日期和更改注释。

3. 将更改详细信息（来自 diff 命令输出）写入 BOM 报告

适配器中设置的变量:

- STA_USER
- STA_PASS
- STA_HOST
- STA_PORT
- STA_PROJECT
- STA_VIEW
- STA_DIR
- BFCLIENT
- LASTRUN
- CURDATE
- EMAILCHANGES
- STARTEAM80JAR

环境所需的其他变量:

- P4USER
- P4PASSWD

不能对这些变量使用“指定隐藏”属性。用户和密码将以明文形式写入到步骤日志中。

第 35 章 Rational Build Forge 和 Subversion

BuildForge 能够与 Subversion (SVN) 协作以进行源代码管理。

本主题描述与 SVN 集成的其他设置需求，并概述了所提供样本适配器模板的特征。

SVN 适配器模板样本提供了几种方法来分析对基线的变更。通常，变更分析用于避免构建：如果基线组件尚未更改，那么不会重新构建

SVN 适配器的其他设置需求

请参阅第 422 页的『适配器需求』以了解一般需求。对于 SVN 没有任何其他需求。

SVN 适配器模板样本

提供了以下适配器模板样本。

SubversionByDate

1. 查询 Subversion 以查找过去的日期与当前日期之间发生的存储库更改。
2. 将以下信息写入 BOM 报告：更改类型、修订、用户、文件或目录和更改日期。
3. 将以下信息写入 BOM 报告：文件名、状态、修订和分支。

所用的变量：

- SVN_CLIENT
- SVN_OPTS
- SVN_REPOSITORY
- SVN_LAST_REV
- SVN_LAST_DATE

SubversionByRev

在 [jazz.net](https://jazz.net/wiki/bin/view/Main/RationalBuildForge/SVNAdaptorExample) 上有一篇描述如何设置该适配器的文章：<https://jazz.net/wiki/bin/view/Main/RationalBuildForge/SVNAdaptorExample>。

1. 查询 Subversion 以查找当前修订与早期修订之间对存储库的更改。
2. 为每个更改将以下信息写入 BOM 报告：修订、用户、更改类型、文件或目录路径和更改日期。
3. 将更改详细信息（来自 diff 命令输出）写入 BOM 报告。

所用的变量：

- SVN_CLIENT
- SVN_OPTS
- SVN_REPOSITORY
- SVN_LAST_REV
- SVN_LAST_DATE

第 36 章 Rational Build Forge 和 WebSphere 产品

本部分描述将 Build Forge® 与 WebSphere 产品集成的方法:

- 使用 WebSphere Application Server 而非 Apache Tomcat 来运行 Build Forge 服务和“快速报告”
- 将 IBM HTTP Server (IHS) 而不是 Apache 用作 Web 应用程序服务器

使用 WebSphere Application Server 而非 Apache Tomcat

您可以使用 WebSphere Application Server 作为应用程序服务器，用于取代所提供的 Apache Tomcat 应用程序服务器。您可以将其配置为运行 Build Forge 服务和 Build Forge 帮助应用程序。

以下是 Websphere Application Server 中配置的三个方面:

- 将其配置为运行 Build Forge 服务和帮助应用程序。这是必需配置。
- 为 Build Forge 服务应用程序配置 Java 2 安全性。如果在 WebSphere Application Server 上运行 Java 2 安全性，那么这是必需配置。
- 在 Build Forge 中配置对 SSL 和密码加密的支持。如果已在 Build Forge 中配置这些功能部件，那么这是必需配置。

先决条件

- 查看特定产品的详细系统需求站点。针对您的 Rational Build Forge 修订版和版本运行查询以查看支持哪些版本的 WebSphere Application Server。
- 由于许可证服务器中的限制，Build Forge 控制台和 WAS 必须运行在同一操作系统和硬件平台上。
- 在尝试部署 Build Forge 服务层之前，请确保已对 WebSphere Application Server 应用最新修订包。

要点: 使用该配置时，必须在启动 Build Forge 之前，在 WebSphere Application Server 中启动 Build Forge 服务和 Build Forge 帮助应用程序。这些服务和应用程序不会自动启动。

配置 Websphere Application Server 以运行 Build Forge 应用程序

配置 WebSphere Application Server 以运行服务应用程序和帮助应用程序。

关于此任务

在 WebSphere 管理控制台中执行以下过程中的步骤。

自 WebSphere V8.0 开始，WebSphere 管理控制台的用户界面稍有变化。以下指示信息适用于 V7 和较低版本。V8.0 中的以下差异影响该过程:

- 要访问应用程序，请使用应用程序 > 应用程序类型 > 企业应用程序。
- 设置上下文根的步骤移到了选择应用程序之后。

- 您不会看到为资源创建 **MBean** 复选框。

过程

1. 打开 WebSphere 管理控制台。 以下是控制台的 URL:
 - `http://<was_host>:<was_port>/ibm/console`; 缺省端口为 9060。
 - `https://<was_host>:<was_port>/ibm/console`; 缺省端口为 9043。如果已启用 WebSphere 管理安全性, 请使用此 URL。
2. 为 JDBC 驱动程序配置支持。
 - a. 新建变量 `RBFIJDBC_DRIVER_PATH`。在 **环境 > WebSphere 变量** 中创建变量。其作用域应该为 WAS 节点和服务端。将其值设置为包含数据库驱动程序 JAR 文件的目录。
 - b. 将更改保存到主配置。
 - c. 停止并重新启动 WebSphere 服务器以使新变量可用。
 - d. 新建共享库 `RBFIJDBC_LIBRARY`。

在 **环境 > 共享库** 中创建库。使用刚创建的 `RBFIJDBC_DRIVER_PATH`, 添加 JDBC 设备驱动程序的 JAR 文件名称。如果有多个 jar 文件需要列出, 请将每个 jar 文件都以单独的一行列出 (如同 DB2 中的情形)。

以下示例针对 MySQL 数据库驱动程序:

```
${RBFIJDBC_DRIVER_PATH}\mysql-connector-java-5.0.5-bin.jar
```

下一个示例针对 DB2:

```
${RBFIJDBC_DRIVER_PATH}\db2jcc.jar  
${RBFIJDBC_DRIVER_PATH}\db2jcc_license_cu.jar
```

在这些示例中, 对于 UNIX 或 Linux, 使用正斜杠 (/) 而非反斜杠 (\)。

- e. 将更改保存到主配置。
3. 安装 Build Forge 应用程序 WAR 文件。
 - a. 打开 **应用程序 > 企业应用程序**。
 - b. 单击 **安装**。
 - c. 浏览到 `rbf-services.war` 文件。使用 `rbf-services` 作为上下文根。
 - 对于 Windows 系统:


```
<bfinstall>\PrepForExternal\rbf-services.war
```
 - UNIX 和 Linux 系统:


```
<bfinstall>/Platform/PrepForExternal/rbf-services.war
```
 - d. 单击下一步, 如果选中了以下复选框, 那么取消选中:
 - 为资源创建 **MBean**
 - e. 单击下一步, 直到您看到 **完成** 按钮, 然后单击 **完成**。
 - f. 单击安装文本底部的 **保存**。
 4. 安装 Build Forge 帮助 WAR 文件。
 - a. 打开 **应用程序 > 企业应用程序**。
 - b. 单击 **安装**。

- c. 浏览到 Build Forge 安装目录中的 BuildForgeHelp.war 文件。使用 /BuildForgeHelp 作为上下文根。
 - d. 单击下一步，如果选中了以下复选框，那么取消选中：
 - 为资源创建 MBean
 - e. 单击下一步，直到您看到完成按钮，然后单击完成。
 - f. 单击安装文本底部的保存。
5. 将 RBF_JDBC_LIBRARY 设置为共享库引用。
 - a. 打开应用程序 > 企业应用程序。
 - b. 单击 **rbf-services.war** 链接。
 - c. 单击共享库引用。
 - d. 选中服务层登录 **Servlet** 框。
 - e. 单击引用共享库。
 - f. 将 RBF_JDBC_LIBRARY 添加到列表中。
 6. 设置类装入器顺序。在管理模块下，选择服务层登录 **Servlet**，并在下拉框中找到类装入器顺序。将该值更改为首先是使用本地类装入器装入的类（最后是使用父类装入器装入的类）。
 7. 将更改保存到主配置。
 8. 检查 Build Forge 配置文件。确保文件中的 services_url 条目指向正确的 URL。如果使用 WAS，那么 buildforge.conf 中的该属性值应类似于 https://<hostname>:9443/rbf-services 或 https://<hostname>:9080/rbf-services。buildforge.conf 文件位于以下位置：
 - Windows 系统
`<bfinstall>\`
 - UNIX 和 Linux 系统
`<bfinstall>/Platform/`
 9. 添加对 rbf-services 的 bin 目录中所有文件的执行许可权。（UNIX 和 Linux 系统）。在该示例中，使用概要文件 AppSrv01。如果不是 AppSrv01，请使用您自己的概要文件名称。
`chmod -R 755 <WAS-install>/AppServer/profiles/AppSrv01/installedApps/<node>/rbf-services_war.`
 10. 更新 PATH 变量以包含令牌库目录。
 - Windows 系统
`<bfinstall>\token_libs`
 - UNIX 和 Linux 系统
`<bfinstall>/Platform/token_libs`
 11. 如果要使用 64 位 Linux 或 WAS，那么需要复制所提供的 64 位库。
 - a. 重命名 32 位库文件。
 - Windows 系统：将 `<bfinstall>\token_libs\rcl_rational.dll` 重命名为 `<bfinstall>\token_libs\rcl_rational.dll.old`。
 - UNIX 和 Linux 系统：将 `<bfinstall>/Platform/token_libs/librcl_rational.so` 重命名为 `<bfinstall>/Platform/token_libs/librcl_rational.so.old`。

- b. 将 64 位库文件从 token_libs_64 复制到 token_libs。请勿在命令中使用换行符。此处使用换行符仅为保证可读性。

- Windows 系统

```
copy <bfinstall>\token_libs_64\rcl_rational.dll  
<bfinstall>\token_libs\rcl_rational.dll
```

- UNIX 和 Linux 系统

```
cp <bfinstall>/Platform/token_libs_64/librcl_rational.so  
<bfinstall>/Platform/token_libs/librcl_rational.so
```

在 Websphere Application Server 中为 Build Forge 配置 Java 2 安全性

如果正在运行已启用 Java 2 安全性的 WebSphere Application Server，那么必须配置 Build Forge 服务层以使用该安全性。

关于此任务

在 WebSphere 管理控制台中执行以下步骤进行配置。

过程

1. 打开应用程序 > 企业应用程序。
2. 选择 **rbf-services_war**，然后单击更新。
3. 选择替换或添加单个文件。
4. 在指定从“安装的应用程序归档文件”开始到“要替换或添加的文件”的路径中，输入 META-INF/was.policy。
5. 选择本地文件系统，浏览至针对您操作系统的 was.policy 文件，然后单击下一步。
 - 对于 Windows 系统：
`<bfinstall>/samples/projects/was.policy`
 - UNIX 和 Linux 系统：
`<bfinstall>Platform/samples/projects/was.policy`
6. 单击确定。
7. 将更改保存到主配置，然后停止并启动 Build Forge 服务应用程序。

在 Websphere Application Server 中为 Build Forge 配置 SSL 和密码加密支持

在 WebSphere Application Server 中为 Build Forge 应用程序启用 SSL 和/或密码加密。

关于此任务

启用对 SSL 和密码加密的支持需要额外配置。您可以配置其中任意一项或同时配置。必须首先在 WebSphere Application Server 上配置 SSL，然后才能在 Build Forge 上启用 SSL。

过程

1. 将 `<bfinstall>\keystore` 目录复制到 `<WAS_INSTALL_DIR>\AppServer\profiles\AppSrv01`。如果使用 AppSrv01 之外的概要文件，请将密钥库目录复制到此概要文件目录。

该路径在 UNIX 和 Linux 系统上等效。该目录包含应用程序服务器概要文件。该目录可能因您的 WebSphere Application Server 版本而异。

2. 登录到 WebSphere 控制台。 使用 WebSphere 管理控制台。
3. 转至“定制属性”页面。 打开**服务器 > 服务器类型 > WebSphere Application Server > server1 > Java 和流程管理 > 流程定义 > Java 虚拟机 > 定制属性**

缺省服务器名称是 **server1**。如果您的服务器具有不同名称，请使用该名称。

4. 启用 SSL。 设置定制属性以指向 `bfclient.conf` 文件。

- Windows

```
com.buildforge.client.config=<bfinstall>\bfclient.conf
```

- UNIX 或 Linux

```
com.buildforge.client.config=<bfinstall>/Platform/bfclient.conf
```

5. 启用密码加密。 设置定制属性以指向 `bfpassword.conf` 文件。

- Windows

```
com.buildforge.password.encryption.file=<bfinstall>\bfpassword.conf
```

- UNIX 或 Linux

```
com.buildforge.password.encryption.file=<bfinstall>/Platform/bfpassword.conf
```

6. 重新启动 WebSphere Application Server。 更改将在您重新启动后生效。

使用 Websphere Application Server 运行 Build Forge 应用程序的启动顺序

在启动 Build Forge 之前，首先在 WebSphere Application Server 中启动各个 Build Forge 应用程序。

关于此任务

在启动 Build Forge 之前从 WebSphere Application Server 启动服务和帮助应用程序。

过程

1. 打开 WebSphere 管理控制台。
2. 选择**应用程序 > 企业应用程序**。
3. 选择 **rbf-services.war** 和 **BuildForgeHelp.war**。
4. 单击**启动**。
5. 启动 Build Forge。

使用 IBM HTTP Server 而不是 Apache HTTP Server

可以配置 IBM HTTP Server (IHS) 而不是缺省情况下安装的 Apache HTTP，以与管理控制台一起使用。

开始之前

- 由于许可证服务器的限制，Build Forge 控制台和 IBM HTTP Server 必须在同一个操作系统和硬件平台上运行。

- 您必须使用提供的预编译 **PHP** 模块。请勿尝试使用您自己编译的 PHP。请查看发行说明以确保确实包含了操作系统的模块。如果未包含，那么您需要与支持人员合作来编译该模块。
- 如果您完成该设置有困难，请与 IBM 支持联系。
- 在示例中，以下目录用作 IBM HTTP Server 安装目录。如果您的安装目录有所不同，请替换为实际的安装目录。
 - Windows: C:\Program Files\IBM\HTTPServer
 - Unix 和 Linux: /opt/IBM/HTTPServer

关于此任务

本主题描述如何执行以下任务：

1. 设置预编译的 PHP 文件。
2. 修改 IBM HTTP Server 配置文件以指向 Build Forge Web 应用程序。
3. 如果使用代理服务器来访问数据库，请修改 PHP 以使用代理服务器。
4. 如果使用 SSL，请配置 IHS 以通过 SSL 使用 Build Forge。
5. Windows 系统：替换 IBM HTTP Server 安装目录中的 libeay32.dll。
6. AIX 系统：额外设置

设置预编译的 PHP 文件

关于此任务

该任务描述如何找到提供的 PHP 文件，以及如何将它们复制到指定的位置。这些文件位于以下目录中：

- Windows 系统
`<bfinstall>\ihs_modules`
- UNIX 和 Linux 系统
`<bfinstall>/Platform/ihs_modules`

过程

1. 复制 PHP 模块。如有必要，创建 modules 目录。
 - Windows 系统


```
cd <bfinstall>\ihs_modules\modules\  
copy *.dll C:\Program Files\IBM\HTTPServer\modules
```
 - UNIX 和 Linux 系统


```
cd <bfinstall>/Platform/ihs_modules/modules  
cp libphp5.so /opt/IBM/HTTPServer/modules
```
2. 复制 PHP 扩展。如有必要，创建 extensions 目录。
 - Windows 系统


```
cd <bfinstall>\ihs_modules\lib\php\extensions\  
copy *.dll C:\Program Files\IBM\HTTPServer\extensions
```
 - UNIX 和 Linux 系统


```
cd <bfinstall>/Platform/ihs_modules/lib/php/extensions/no-debug-zts-20090626  
cp *.so /opt/IBM/HTTPServer/extensions
```
3. 复制提供的 php.ini 文件。如有必要，创建 conf 目录。

- Windows 系统


```
cd <bfinstall>\ihs_modules
copy php.ini C:\Program Files\IBM\HTTPServer\conf
```
 - UNIX 和 Linux 系统


```
cd <bfinstall>/Platform/ihs_modules
cp php.ini /opt/IBM/HTTPServer/conf
```
4. 编辑您复制到 conf 目录中的 php.ini 文件，以包含指向 extensions 目录的指针。添加或替换以下行:
- Windows 系统


```
extension_dir = C:\Program Files\IBM\HTTPServer\extensions
```
 - UNIX 和 Linux 系统


```
extension_dir = /opt/IBM/HTTPServer/extensions
```
5. 使用针对您系统的信息编辑 php.ini 文件。您可以使用针对 Apache HTTP Server 提供的 php.ini 文件作为指南。该文件位于以下位置:
- Windows 系统


```
<bfinstall>\Apache\php\php.ini
```
 - UNIX 和 Linux 系统


```
<bfinstall>/server/apache/conf/php.ini
```
6. 添加 php.ini 中使用的所有目录的写访问权。在 UNIX 和 Linux 系统上，对目录运行以下命令:
- ```
chmod -R 777
```

## 编辑 IBM HTTP Server 配置文件 过程

1. 在服务器安装的 conf 目录中找到 IBM HTTP Server (IHS) 的 httpd.conf 文件。
2. 修改 DocumentRoot 设置以指向 Build Forge Web 应用程序，如示例中所示。在该示例中，Build Forge 安装目录为 /opt/buildforge。
 

```
<VirtualHost *:80>
 ServerAdmin build@yourdomain.com
 DocumentRoot /opt/buildforge/webroot/public
 ServerName ausbuild01.yourdomain.com
 ServerAlias build.yourdomain.com mc.yourdomain.com #optional server aliases
 ErrorLog logs/ausbuild.error_log
 CustomLog logs/ausbuild.access_log common
</VirtualHost>
```
3. 添加与 PHP 相关的条目。
  - Windows 系统
 

```
LoadModule php5_module "C:\Program Files\IBM\HTTPServer\modules\php5apache2_2.dll"

<IfModule dir_module>
 DirectoryIndex index.html index.php
</IfModule>

AddType application/x-httpd-php .php

PHPIniDir is the location of the php.ini file
PHPIniDir "C:\Program Files\IBM\HTTPServer\conf"
```
  - UNIX 和 Linux 系统

```
LoadModule php5_module "/opt/IBM/HTTPServer/modules/libphp5.so"
```

```
<IfModule dir_module>
 DirectoryIndex index.html index.php
</IfModule>
```

```
AddType application/x-httpd-php .php
```

```
PHPIniDir is the location of the php.ini file
PHPIniDir "/opt/IBM/HTTPServer/conf"
```

4. 请更改 `User` 和 `Group` 语句以使用 `daemon`，即 `<bfinstall>/webroot/templates_c` 的缺省用户。

```
User daemon
Group daemon
```

5. 如果需要，请更改 `IHS` 端口号。缺省端口号为 80。对 `httpd.conf` 作出其他任何必要的更改。

## 识别 PHP 中的代理服务器

### 关于此任务

可选：仅当管理控制台需要使用代理服务器访问其数据库时，才需要此步骤。

### 过程

编辑 PHP 配置文件 `php.ini`。它位于 `<php-install>/lib` 中，例如 `/usr/local/php-5.3.6`。

添加以下条目：

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

## 为 IHS 配置 SSL

### 关于此任务

除了 `IHS` 通常的 SSL 设置外，用于 `Build Forge` 时还有额外要求。

### 过程

1. 在您的 `PATH` 中包括 `Build Forge` 工具目录。
  - Windows 系统
    - `bfinstall\ibmjdk\bin`
    - `bfinstall\openssl`
  - UNIX 或 Linux 系统
    - `bfinstall/server/ibmjdk/bin`
2. 在库路径中包含 `Build Forge` 工具目录。
  - AIX 系统 (`LIBPATH`)
    - `bfinstall/openssl`
  - UNIX 或 Linux 系统 (`LD_LIBRARY_PATH`)
    - `bfinstall/openssl`
3. 在 `JAVA_HOME` 中包含 `Build Forge` 工具目录。

- Windows
    - *bfinstall\ibmjdk*
  - UNIX 或 Linux
    - *bfinstall/server/ibmjdk*
4. 将 Build Forge 密钥从 PKCS12 转换为 CMS。使用 GSKIT 工具的最新版本。在 *gsk7\bin* (Windows) 或 *bin* (UNIX 或 Linux) 中, 运行以下命令 (分行只是为了更加清晰):
- ```
gsk7cmd -keydb
        -convert
        -db bfinstall\keystore\buildForgeKeyStore.p12
        -pw buildForgeKeyStore_password
        -old_format pkcs12
        -new_format cms
```
5. 将密码保存到隐藏文件中。IHS 使用该文件在启动期间获取密码。若无此文件, IHS 会提示输入密码。使用 GSKIT 工具的最新版本。在 *gsk7\bin* (Windows) 或 *bin* (UNIX 或 Linux) 中, 运行以下命令 (分行只是为了更加清晰):
- ```
gsk7cmd -keydb
 -stashpw
 -db bfinstall\keystore\buildForgeKeyStore.kdb
 -pw buildForgeKeyStore_password
```
6. 修改 *httpd.conf*。包含以下针对 Windows 的条目。请注意, 如果您希望用户仅通过 https 进行访问, 请注释掉 *Listen 80*。
- ```
LoadModule ibm_ssl_module modules/mod_IBM_ssl.so
# Listen 80
Listen 0.0.0.0:443
<VirtualHost *:443>
SSLEnable
SSLClientAuth None
SSLProtocolDisable SSLv2
SSLServerCert buildforge
KeyFile bfinstall\keystore\buildForgeKeyStore.kdb
SSLStashFile bfinstall\keystore\buildForgeKeyStore.sth
ErrorLog bfinstall\Apache\logs\ssl_error.log
TransferLog bfinstall\Apache\logs\transfer.log
</VirtualHost>
```

包含以下针对 UNIX 和 Linux 的条目: 请注意, 如果您希望用户仅通过 https 进行访问, 请注释掉 *Listen 80*。

```
LoadModule ibm_ssl_module modules/mod_IBM_ssl.so
# Listen 80
Listen 0.0.0.0:443
<VirtualHost *:443>
SSLEnable
SSLClientAuth None
SSLProtocolDisable SSLv2
SSLServerCert buildforge
KeyFile bfinstall/Platform/keystore/buildForgeKeyStore.kdb
SSLStashFile bfinstall/Platform/keystore/buildForgeKeyStore.sth
ErrorLog bfinstall/server/apache/logs/ssl_error.log
TransferLog bfinstall/server/apache/logs/transfer.log
</VirtualHost>
```

bfinstall 是安装目录的根目录。对于上述步骤, 在 UNIX 和 Linux 系统上, 通常需将 */Platform* 附加到 *bfinstall* 的末尾, 以便能访问正确的目录。您还需要使用斜杠 (/) 而非反斜杠 (\) 来将目录分开。

结果

有关设置 SSL 的更多信息，请参阅 IHS 文档。

Windows: 替换 DLL

关于此任务

如果您正在使用的 IBM HTTP Server 版本低于 V8.0，那么必须将 DLL 替换为 Rational Build Forge 随附的 DLL。

过程

在 IBM HTTP Server 安装目录的多个位置中找到并替换现有 libeay32.dll 文件。

- 使用 <bfinstall>/openssl 中提供的 libeay32.dll。
- 搜索所有 DLL 实例并将其替换。Windows 2003 中的位置示例:

```
<ihs_install>\Plugins\gsk7\gsk7_32\icc\osslib  
<ihs_install>\gsk7\icc\osslib
```

结果

替换 DLL 后，HTTP Server 可正常启动。如果您不将其替换，那么 HTTP Server 会启动失败。

AIX 系统: 额外设置

AIX 系统需要一个额外设置。

过程

- AIX 6 系统。
 - 将 <bfinstall>/Platform/his_modules/libxml2.a 复制到 /opt/IBM/HTTPServer/modules。
 - 将 /opt/IBM/HTTPServer/modules 添加到 LIBPATH。
- AIX 7 系统。
 - 将 <bfinstall>/Platform/his_modules/libxml2.a 和 <bfinstall>/Platform/his_modules/libxml2.so.2 复制到 /opt/IBM/HTTPServer/modules。
 - 将 /opt/IBM/HTTPServer/modules 添加到 LIBPATH。
 - 将以下行添加到 httpd.conf:
LoadFile "/opt/IBM/HTTPServer/modules/libxml2.so.2"

第 37 章 使用 API

Build Forge 提供了 Java 客户机 API 和 Perl 客户机 API。

客户机文件存储在 `<bfinstall>/webroot/public/clients/` 中。

您可以通过处于运行状态的管理控制台访问文档。客户机下载目录位于以下 URL: `http://<hostname>:<portnumber>/clients/`

API 访问 Build Forge

使用 API 的程序直接与服务层进行通信。服务层是 Apache Tomcat 上的应用程序。安装期间, Apache Tomcat 服务器配置为在特定的端口上侦听。这些端口必须处于打开状态, 以使 API 与 Build Forge 通信。这些端口缺省情况下设置如下:

- 3966 (不安全)
- 49150 (安全 - 启用 SSL)

API 客户机的 SSL 配置在 `bfclient.conf` 中, 该文件必须包含在该客户机中。如果启用了 SSL, 客户机必须要有密钥库和证书才能与 Build Forge 通信。有关更多信息, 请参阅第 106 页的『为 API 客户机 (Perl 或 Java) 启用 SSL』。

为 API 程序创建 Build Forge 用户

在管理控制台上为程序创建一个用于认证的用户。

关于此任务

为 API 程序创建一个用于登录管理控制台的用户。登录以测试该用户, 验证该程序是否在运行。

只要程序访问控制台, 都必须使用用户凭证向控制台认证自身。

认证后, 会为用户会话生成会话标识并存储在数据库中。如果该程序使用与现有用户相同的登录信息, 该用户会话将关闭。

仅一个线程或进程可使用该用户凭证。如果其他线程或进程试图使用同一个用户凭证建立会话, 会导致原先会话终止。

注: 请勿使用 LDAP/Active Directory 认证提供的用户。请在管理控制台中创建用户。

Java 客户机 API

请使用 Java 客户机 API 编写访问管理控制台的 Java 程序。

使用 Java 客户机 API 创建的程序在客户机主机上运行, 并访问管理控制台上的数据。Java 客户机 API 由一个 `.jar` 文件构成, 该文件包含了定义管理控制台对象方法的类, 这些方法提供对这些对象的操作。

要使用 Java 客户机 API, 必须具有 Java SDK 1.5 或 1.6。

文档以 JavaDocs 的形式提供。

注：必须在管理控制台上为程序定义一个用于认证的 Build Forge 用户。

获取 Java 客户机 API 软件包

您可以从管理控制台主机下载 Java 客户机软件包。

过程

要下载 Java API:

1. 访问客户机下载目录。在 Web 浏览器中，访问以下 URL:
`http://<hostname>:<portnumber>/clients/`
2. 保存 JAR 文件。在 Java 客户机下，右键单击 **JAR 文件** 链接并选择 **链接另存为**。指定 JAR 文件的保存位置。
3. 保存 JavaDocs。在 Java 客户机下，右键单击 **JavaDoc 引用 ZIP** 链接并选择 **链接保存为**。指定 JAR 文件的保存位置。

结果

您可以通过管理控制台访问文档。在客户机下载目录页面上的“Java 客户机”下，单击 **JavaDoc 参考**。

安装 Java 客户机 API

将 Java API 放置在客户机主机上，并将 Java SDK 设置为使用该 API。

关于此任务

主机将用作管理控制台主机的客户机。

过程

1. 将 .jar 文件放置在期望位置。
2. 更新 CLASSPATH。将 CLASSPATH 设置为包含您放置 rbf-services-client-java.jar 的目录。

Perl 客户机 API

请使用 Perl 客户机 API 编写访问管理控制台的 Perl 程序。

Perl 客户机是一组 Perl 模块，通过这些模块可以访问管理控制台数据对象和方法的抽象。

Perl 客户机模块的文档以两种格式包含在客户机 API 软件包中:

- 文件: apidoc.txt
- POD (Plain Old Documentation) 格式的 Perl 文档。有关更多信息，请参阅位于 <http://www.perl.org> 的联机文档。

要使用 Perl 客户机，您必须:

- 从管理控制台计算机获取 Perl 客户机软件包。

- 安装此软件包（若尚未安装 Perl，那么还要安装 Perl）。

注：必须在管理控制台上为程序定义一个用于认证的 Build Forge 用户。

获取 Perl 客户机 API 软件包

您可以从管理控制台主机下载 Perl 客户机 API。

过程

要下载 Perl 客户机 API:

1. 访问客户机下载目录。在 Web 浏览器中，访问以下 URL:
`http://<hostname>:<portnumber>/clients/`
2. 保存 ZIP 文件。在 Perl 客户机下，右键单击 **ZIP 文件** 链接并选择**将链接保存为**。指定 ZIP 文件的保存位置。
3. 保存文档。在 Perl 客户机下，右键单击 **PerlDoc 引用 tar.gz**链接并选择**将链接保存为**。指定 ZIP 文件的保存位置。解压缩文件，以访问每个模块的文档。

结果

您可以通过管理控制台访问文档。在客户机下载目录页面上的“Perl 客户机”下，单击 **PerlDoc 参考**。

安装 Perl 客户机 API

要使用 Perl 客户机 API，您必须在计划运行应用程序的主机上安装 Perl 客户机 API。

关于此任务

该主机用作管理控制台主机的客户机。

过程

1. 在客户机主机上安装 Perl 解释器，比如 ActiveState 的 ActivePerl V5.8.4 或更高版本。需要下列 Perl 必备模块（ActivePerl V5.8.8 中包含这些模块）：
 - Exporter
 - LWP::UserAgent
 - HTTP::Request

有关安装 Perl 模块的信息，请参阅 Perl 文档。

2. 将下载的 Perl 客户机 API 软件包解压缩到临时目录。
3. 将 Perl 客户机 API 作为标准的 Perl 分发进行安装，如 apidoc.txt 文件中所述。

在 Windows 上，需要 nmake 1.5，它包含在 Visual Studio 中，也可以从 Microsoft Web 站点下载。nmake 1.5 必须安装在 C:\Windows 之类可通过 PATH 环境变量找到的位置。在解压缩 Perl 客户机软件包的临时目录中，运行以下命令：

```
perl Makefile.PL
nmake
nmake install
```

在 UNIX 或 Linux 系统上（或 Windows 上的 Cygwin 环境中）：

```
perl Makefile.PL  
make  
make install
```

安装后，顶级的 Perl 客户机模块为 `BuildForge::Services::DB0`。详细信息请参阅每个模块的 `PerlDoc`。

第 38 章 确定管理控制台版本号

要查看您所用的管理控制台版本，请将鼠标光标放在页面左上角的 Rational Build Forge 徽标上。系统将在弹出工具提示内显示版本号。

第 39 章 性能和可伸缩性

该系统旨在用于各种广泛的负载。在高负载下，运行系统需要对配置和资源分配进行适当调整。

下文可用作额外资源。另请参阅第 29 页的第 5 章，『规划安装』。

- 部署 IBM Rational Build Forge 管理控制台：获取结果和性能：本文描述了规划和部署的阶段、提供了资源功能的建议以及相关调优。
- Rational Build Forge 性能测试结果：在 7.1.2 中评估相对于 7.1.1.4 得到改善的性能：本文提供了在不断增加负载的情况下各种系统表现的一般描述。它提供了各种操作系统和数据库组合的信息。它还展示了 V7.1.2 所提供的相比于 V7.1.1.4 的性能改进。

第 40 章 随产品安装的可执行命令

下表描述了 Rational® Build Forge® 提供和使用的可执行命令。

在 Windows 上，命令文件位于 Build Forge 安装目录中，缺省为 C:\Program Files\IBM\Build Forge。

在 UNIX/Linux 上，命令文件位于 <bfinstall>/Platform 目录中，其中，<bfinstall> 缺省为 /opt/buildforge。

注：如果在 z/Linux 上运行管理控制台，必须指定 .pl 扩展名才能运行命令。

要显示任何可执行命令的版本号，请使用 -v 选项。必须从安装可执行命令的目录中运行命令。

bfproject -v

任何命令的 - v 选项都可显示命令名称及其版本号，如以下示例中所示：

bfproject.exe 7.0.351

| 可执行文件 | 服务？ | 描述 |
|----------------------------|-----|--|
| bfdbmigrate | N | 使用该迁移脚本可在模式转换为 V7.0 后将缺省数据库从 V3.8 (MySQL) 转换为 DB2®。该脚本应在使用 bfmigrate.exe/bfmigrate.pl 之后再使用。 |
| bfproject | N | buildforge.exe 发出该命令以启动作业。 |
| bfengine | Y | 此命令启动 buildforge.exe 和 Web 服务器。仅适用于 Windows。 |
| bfexport | N | 使用该实用程序可从数据库中导出数据。 |
| bf bomexport | N | 使用该实用程序可从数据库导出 BOM。 |
| bfimport | N | 使用该实用程序可将项目数据导入数据库。 |
| bfmigrate | N | 使用该迁移脚本可将 V3.8 数据库升级至 V7.0。 |
| bfpurge | N | buildforge.exe 发出该命令以清除构建。 |
| bfrefresh | N | 构建系统发出该命令以更新服务器清单。 |
| bf sched.exe 或 bf sched.pl | N | buildforge.exe 发出该命令以在数据库中检查已调度作业，并在适当的时候启动这些作业。 |
| bfstepcmd | N | bfproject 对于长时间运行的步骤发出该命令，为这些步骤创建单独的进程。 |
| buildforge | N | 该命令管理构建、清除和调度进程。 |
| console_uninst | N | 使用此命令可卸载管理控制台。仅适用于 Windows。 |
| bfdispatch | Y | 此命令启动代理程序服务。仅适用于 Windows。 |
| bfagent | N | 代理程序可执行文件 |
| bf pwncrypt | N | 用于将密码加密的实用程序 |

第 41 章 词汇表

本主题提供了系统中使用的概念和术语的定义。

步骤 (step)

步骤是项目或库的组件。它包含一个或多个要运行的命令行。与步骤关联的选择器可确定要使用的服务器。如果未指定任何选择器，那么使用项目选择器。步骤属性可确定步骤的运行方式以及输出的处理方式。步骤可在创建或编辑项目或库时来定义。

步骤日志 (step log)

已完成作业中已完成步骤的数据列表。查看作业时，步骤日志显示在**步骤**选项卡上。每个步骤的信息显示在列中。选择**作业 > 作业名**时，即显示步骤列表。单击步骤以查看其步骤日志。

插件 (plug-in)

可单独安装的软件模块，用于向现有程序、应用程序或接口添加函数。

动态 (dynamic)

与运行时或处理期间发生的事件相关。

代理程序 (agent)

Build Forge[®] 系统的组件。代理程序必须安装在要定义为系统中的服务器资源的任何计算机上。每个代理程序都会与管理控制台进行通信，并运行步骤中定义的命令。代理程序同时还汇编来自步骤执行的输出，并在步骤日志中返回输出。

服务器 (server)

在 Build Forge 中，服务器是与主机关联的对象。它也称为**服务器资源**。项目或步骤在主机上运行。通过与项目或步骤关联的选择器来定义要使用的服务器。

要在 Build Forge 中设置将用作服务器的计算机，必须执行下列操作：

- 在该计算机上安装代理程序（请参阅第 131 页的第 10 章，『安装代理程序』，以获取更多信息）。
- 使用管理控制台创建服务器资源。

服务器资源在**服务器**面板中定义。

服务 (services)

服务是一种系统组件，因为它充当介于客户机和数据库之间的抽象层，所以也称为服务层。客户机包括系统自身以及使用所提供的 Java API 或 Perl API 构造的客户机。

访问组 (access group)

共享许可权、通知和 LDAP 组属性的一组用户。您可将访问组映射到 LDAP 组。可以嵌套组。用户继承其所属组的许可权。

管理控制台 (Management Console)

安装在单台计算机上用于协调系统的系统组件。用户登录到管理控制台上来定义和运行项目以及查看结果和报告。管理控制台向代理程序发出指令以完成作业。

归档 (archive)

输出文件已删除但仍在数据库中保存有数据的作业列表。可以在作业面板中查看该列表。

环境 (environments)

环境是一组变量的容器。环境可以明确地分配给服务器、项目和步骤。步骤环境的构造依次应用服务器环境、项目环境和步骤环境。如果一个变量出现在其中多个环境里，那么采用最后指定的值。

接口 (interface)

接口是适配器模板的实例。若要使用适配器，必须创建接口（并编辑接口）。原始适配器模板将保持不变。另请注意：接口可以包含多个 <interface> 元素，每个元素都是可单独执行的操作。

静态 (static)

与在预先确定的时间或固定时间发生的操作有关。

快照 (snapshot)

某个时间点上的备份数据的记录。

库 (library)

库是可执行的工作定义。它由步骤组成。其行为通过属性来控制。它与项目的不同之处在于它没有选择器来确定运行它的服务器。库从项目内的某个步骤进行调用。

类 (class)

具有全局属性的项目分组。这些属性用于管理已完成的作业，通常定期删除已完成的作业或启动另一个作业执行特定的清除任务。

拦截器 (interceptor)

Web service 用于认证入局消息的处理程序。在 Build Forge 中，提供了拦截器来实施单点登录。

破坏 (clobber)

从数据库删除项目及其所有关联的作业。

清单 (manifest)

通过收集器收集的服务器数据列表。清单数据由选择器用于选择服务器。服务器清单会自动更新。您也可以手动更新服务器清单。在**服务器 > 服务器名称**中查看服务器时使用**队列清单刷新**按钮。

轻量级目录访问协议 (Lightweight Directory Access Protocol)

一种使用 TCP/IP 提供目录访问的开放式协议，它支持 X.500 模型且不需要更复杂的 X.500 目录访问协议 (DAP) 的资源。例如，LDAP 可用于查找因特网或内部网目录中的人员、组织和其他资源。

数据库 (database)

数据库存储输入到管理控制台中的所有信息。另外，数据库还会存储系统在运行项目或记录用户操作时创建的数据。

收集器 (collector)

确定从服务器资源中收集哪些信息或将哪些信息分配给服务器资源的对象。这些信息通过收集器中的属性来指定。分配给服务器的收集器用作服务器清单的规范。在**服务器 > 收集器**面板中定义收集器。

适配器 (adaptor)

适配器是允许 Build Forge 系统与外部系统（如源控制系统、调试数据库或测试系统）进行交互的附件。例如，源代码适配器可使系统监视和跟踪源代码控制系统（如 IBM® Rational® ClearCase®、Perforce、Visual SourceSafe 和 CVS）中的更改，并基于这些更改执行操作。可以将适配器配置为收集要存储在材料清单（BOM）中的信息，或者将信息推送回其他信息系统。

通知模板 (notification templates)

通知模板定义在发生特定事件时发送到访问组的电子邮件的内容和格式。系统附带许多缺省模板。您可以编辑模板，也可以创建特定于项目的新模板。

握手 (handshake)

安全套接字层会话开始时的报文交换，其允许客户机使用公共密钥技术认证服务器（以便服务器认证客户机），并允许客户机和服务器在创建用于加密、解密和篡改删除的对称密钥时协作。

信号量 (semaphore)

系统中用于防止活动同时发生的全局标志。一个信号量就是系统管理的一个标签。通常，需要独占使用资源的项目或步骤会获取信号量以确保此独占使用。

使用 **.semget** 命令可以在步骤中设置信号量。使用 **.semput** 命令可以在另一个步骤中释放信号量。获取信号量之后，其他步骤就不能再获取信号量。尝试获取信号量的步骤要等到信号量被释放为止。

项目完成时，系统自动释放项目使用的全部信号。在某些情况下（例如，在作业因系统错误而结束时）不释放信号量。这时，您可以手动进行释放。

选择器 (selector)

与项目或步骤关联的对象，用于选择运行项目或步骤的服务器。选择器属性确定服务器的选择方式。选择器可以使用静态信息。例如，选择器可以指定服务器名称。服务器也可以使用动态信息。例如，选择器会指定指出了 CPU 类型、磁盘大小或当前负载等属性的服务器。在运行时，系统使用选择器编译匹配的服务器列表，并为其中的某台服务器分配项目或步骤。在**管理 > 选择器**面板中定义选择器。在定义项目前必须定义至少一个选择器。

项目 (project)

项目是可执行的工作定义。它由步骤组成。其行为通过项目属性来控制。项目具有关联的选择器，用于确定项目可以在哪台或哪些服务器上运行。可以向项目分配其自己的环境。正在执行的项目就是一项作业。未分配选择器的项目则是一个库。

线程技术 (threading)

用于并发运行各种相关事务的方式。

引擎 (engine)

系统组件。引擎使用通过管理控制台输入且存储在数据库中的信息，以控制项目执行、发送通知电子邮件，以及与（服务器上运行的）代理程序进行通信。

用户 (user)

系统登录名。系统维护其自己的用户集合和许可权设置。在生产安装中，LDAP 用于用户管理，LDAP 条目和组映射到系统中。用户与访问组关联，访问组授予用户访问系统资源的特定许可权。

作业 (job)

运行中项目的实例。系统存储每个已完成作业的数据，包括步骤日志和 BOM 数据。

BOM

已完成作业的数据列表。BOM 是 Bill of Materials（材料清单）的缩写。查看作业时，作业显示在 BOM 选项卡，而有关各个步骤执行的数据则显示在“步骤”选项卡。BOM 包含作业中的步骤信息以及从中产生的文件更改的信息。一个常用的用途是用于软件构建中需要对源文件的更改进行审计的源代码适配器。可以使用 `.scan` 命令设置源代码更改基线，然后设置检查点，使之汇总自上次执行 `.scan` 命令以来所作的更改。

附录. 声明

本信息是为在美国提供的产品和服务编写的。IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，将由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamoto-shi
Kanagawa, 242-8502 Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区： International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证，因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

Intellectual Property Dept. for Rational Software
IBM Corporation

5 Technology Park Drive
Westford, MA 01886
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息仅用于规划的目的。在所描述的产品上市之前，此处的信息可随时更改。

这些信息中包含日常商业运营中使用的样本数据和报告。为了尽可能完整地说明这些内容，样本中包含个人、公司、品牌和产品的名称。这些名称全部都是虚拟的，如果与实际的企业使用的名称和地址类似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。

凡这些样本程序的每份拷贝或其任何部分或任何演绎作品，都必须包括如下版权声明：

© Copyright IBM Corp. 2003, 2013.

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

商标

IBM、IBM 徽标和 ibm.com 是 International Business Machines Corp. 在全球许多司法辖区注册的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。Web 站点 www.ibm.com/legal/copytrade.shtml 上包含了 IBM 商标的最新列表。

Adobe、Adobe 徽标、PostScript 和 PostScript 徽标是 Adobe Systems Incorporated 在美国和/或其他国家或地区的注册商标或商标。

Intel 是 Intel Corporation 或其附属机构在美国和其他国家或地区的商标或注册商标。

Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的注册商标。

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和/或其他国家或地区的商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的注册商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。