



JBoss 企业级 BRMS 平台 5 BRMS 管理员指南

面向 JBoss 管理员
版 5.2.0

Landmann

JBoss 企业级 BRMS 平台 5 BRMS 管理员指南

面向 JBoss 管理员
版 5.2.0

Landmann
rlandmann@redhat.com

法律通告

Copyright © 2010 Red Hat, Inc..

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本文档提供了 JBoss 企业级 BRMS 平台的安装、配置和定制说明。

目录

前言	4
1. 文档约定	4
1.1. 排版约定	4
1.2. 抬升式引用约定	5
1.3. 备注及警告	5
2. 获得帮助并提供反馈信息	6
2.1. 您需要帮助吗？	6
2.2. 我们需要您的反馈！	6
第 1 章 简介	8
1.1. 本版本里的新特征	8
1.2. JBoss 企业级 BRMS 平台	8
第 2 章 安装	9
2.1. 软件包	9
2.2. 安装说明	10
2.2.1. 安装独立软件包	10
2.2.2. 安装可部署软件包	11
2.2.3. 安装和启用 ModeShape	13
2.3. 第一次登录	16
2.4. 登录：故障解除	17
第 3 章 本地化	19
第 4 章 数据库配置	21
4.1. 用于 Apache Jackrabbit 的数据库配置	21
4.1.1. 配置工作区数据库	21
4.1.2. 配置版本数据库	22
4.1.3. 搜索和索引	23
4.2. 用于 ModeShape 的数据库配置	23
4.3. 库配置	23
4.3.1. 修改 JackRabbit 库的位置	23
4.3.2. 配置 BRMS 使用外部的关系型数据库。	25
第 5 章 数据管理	28
5.1. 数据备份	28
5.2. 导入和导出	28
5.3. 在不同的 BRMS 版本间移植数据	28
5.4. 重构二进制软件包	30
第 6 章 安全性	31
6.1. 验证	31
6.1.1. 验证示例：UserRolesLoginModule	32
6.1.2. 验证示例：LDAP	33
6.2. 授权	34
6.3. 规则软件包签名（Rule Package Signing）	38
6.3.1. 设置序列化属性	40
第 7 章 日志	42
第 8 章 自定义	43
8.1. 自定义用户界面	43
8.2. 用于软件包构建的自定义选择器（Customize Selector）	43

第 9 章 监控	45
9.1. 安装 JBoss ON 插件	45
9.2. 发现 BRMS 实例	46
在 EWS 上部署 JBoss BRMS	47
安装 Apache Ant	48
持久化管理者配置示例	49
修订历史记录	51

前言

1. 文档约定

本手册使用几个约定来突出某些用词和短语以及信息的某些片段。

在 PDF 版本以及纸版中，本手册使用在 [Liberation 字体](#) 套件中选出的字体。如果您在您的系统中安装了 Liberation 字体套件，它还可用于 HTML 版本。如果没有安装，则会显示可替换的类似字体。请注意：红帽企业 Linux 5 以及其后的版本默认包含 Liberation 字体套件。

1.1. 排版约定

我们使用四种排版约定突出特定用词和短语。这些约定及其使用环境如下。

单行粗体

用来突出系统输入，其中包括 shell 命令、文件名以及路径。还可用来突出按键以及组合键。例如：

要看到文件您当前工作目录中文件 `my_next_bestselling_novel` 的内容，请在 shell 提示符后输入 `cat my_next_bestselling_novel` 命令并按 **Enter** 键执行该命令。

以上内容包括一个文件名，一个 shell 命令以及一个按键，它们都以固定粗体形式出现，且全部与上下文有所区别。

按键组合与单独按键之间的区别是按键组合是使用加号将各个按键连在一起。例如：

按 **Enter** 执行该命令。

按 **Ctrl+Alt+F2** 切换到虚拟终端。

第一个示例突出的是要按的特定按键。第二个示例突出了按键组合：一组要同时按下的三个按键。

如果讨论的是源码、等级名称、方法、功能、变量名称以及在段落中提到的返回的数值，那么都会以上述形式出现，即**固定粗体**。例如：

与文件相关的等级包括用于文件系统的 **filesystem**、用于文件的 **file** 以及用于目录的 **dir**。每个等级都有其自身相关的权限。

比例粗体

这是指在系统中遇到的文字或者短语，其中包括应用程序名称、对话框文本、标记的按钮、复选框以及单选按钮标签、菜单标题以及子菜单标题。例如：

在主菜单条中选择「系统」→「首选项」→「鼠标」启动 **鼠标首选项**。在「按钮」标签中点击「惯用左手鼠标」复选框并点击 **关闭** 切换到主鼠标按钮从左向右（让鼠标适合左手使用）。

要在 **gedit** 文件中插入特殊字符，请在主菜单栏中选择「应用程序」→「附件」→「字符映射表」。接下来选择从 **Character Map** 菜单中选择 **Search** →「查找.....」，在「搜索」字段输入字符名称并点击「下一个」按钮。此时会在「字符映射表」中突出您搜索的字符。双击突出的字符将其放在「要复制的文本」字段中，然后点击「复制」按钮。现在返回您的文档，并选择 **gedit** 菜单中的「编辑」→「粘贴」。

以上文本包括应用程序名称、系统范围菜单名称及项目、应用程序特定菜单名称以及按钮和 GUI 界面中的文本，所有都以比例粗体出现并与上下文区别。

固定粗斜体 或者 比例粗斜体

无论固定粗体或者比例粗体，附加的斜体表示是可替换或者变量文本。斜体表示那些不直接输入的文本或者那些根据环境改变的文本。例如：

要使用 ssh 连接到远程机器，请在 shell 提示符后输入 **ssh *username@domain.name***。如果远程机器是 **example.com** 且您在该其机器中的用户名为 john，请输入 **ssh john@example.com**。

mount -o remount *file-system* 命令会重新挂载命名的文件系统。例如：要重新挂载 **/home** 文件系统，则命令为 **mount -o remount /home**。

要查看目前安装的软件包版本，请使用 **rpm -q *package*** 命令。它会返回以下结果：***package-version-release***。

请注意上述使用黑斜体的文字 -- *username*、*domain.name*、*file-system*、*package*、*version* 和 *release*。每个字都是一个站位符，可用作您执行命令时输入的文本，也可作为该系统显示的文本。

不考虑工作中显示标题的标准用法，斜体表示第一次使用某个新且重要的用语。例如：

Publican 是一个 *DocBook* 发布系统。

1.2. 抬升式引用约定

终端输出和源代码列表要与周围文本明显分开。

将发送到终端的输出设定为 **Mono-spaced Roman** 并显示为：

```
books      Desktop  documentation  drafts  mss      photos  stuff  svn
books_tests Desktop1  downloads      images  notes    scripts svgs
```

源码列表也设为 **Mono-spaced Roman**，但添加下面突出的语法：

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo            echo    = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

1.3. 备注及警告

最后，我们使用三种视觉形式来突出那些可能被忽视的信息。



注意

备注是对手头任务的提示、捷径或者备选解决方法。忽略提示不会造成负面后果，但您可能会错过一个更省事的诀窍。



重要

重要框中的内容是那些容易错过的事情：配置更改只可用于当前会话，或者在应用更新前要重启的服务。忽略‘重要’框中的内容不会造成数据丢失但可能会让您抓狂。



警告

警告是不应被忽略的。忽略警告信息很可能导致数据丢失。

2. 获得帮助并提供反馈信息

2.1. 您需要帮助吗？

如果您对本文档论述的步骤有疑问，请访问红帽客户门户网站 <http://access.redhat.com>。通过客户门户网站，您可以：

- 搜索或者浏览有关红帽产品技术支持文章的知识库。
- 向红帽全球支持服务（GSS）提交支持案例。
- 访问其它红帽文档。

红帽还托管了大量讨论红帽软件和技术的电子邮件列表。公开列表位于 <https://www.redhat.com/mailman/listinfo>。点击任意列表名称即可订阅该列表或者访问列表归档。

2.2. 我们需要您的反馈！

如果你在本指南里发现了印刷错误，或者你有改进该手册的建议，我们希望听到你的声音！请提交报告到 Bugzilla 并指明产品 **JBoss Enterprise BRMS Platform 5** 和组件 **doc-BRMS_Administrator_Guide**。下面的链接是针对本产品 <http://bugzilla.redhat.com/> 预先填写的错误报告。

请填写 Bugzilla 的 **Description** 字段。请尽量将问题具体化，这样我们就可以尽快改正。

Document URL:

Section Number and Name:

Describe the issue:

Suggestions for improvement:

Additional information:

请提供你的名字以获得关于报告该问题的 credit。

第 1 章 简介

1.1. 本版本里的新特征

表 1.1. 本版本里的新特征

功能	修改
第 2.2.3 节 “安装和启用 ModeShape”	关于将 Modeshape 作为库安装的新章节
第 4.2 节 “用于 ModeShape 的数据库配置”	关于配置用于 Modeshape 的数据源的新章节
第 4.3.2 节 “配置 BRMS 使用外部的关系型数据库”	关于配置 BRMS 以使用外部关系型数据库的新章节
第 9 章 监控	关于用 JON 插件监控 BRMS 的新章节
附录 A. 在 EWS 上部署 JBoss BRMS	关于安装 JBoss BRMS 到 JBoss 企业级 Web 服务器里的说明的新附录

1.2. JBoss 企业级 BRMS 平台

JBoss 企业级 BRMS 平台是一个基于 JBoss Rules 的**商业规则管理系统** (*Business Rules Management System*)。它提供服务器端的、多用户管理的解决方案、存储、编辑和规则部署以及其他 JBoss Rules 资产。它和 JBoss Developer Studio 4 以提供用户友好的、基于 web 的用户界面。



重要

在一些地方，JBoss 企业级 BRMS 平台及其文档都引用了 Guvnor。Drools Guvnor 是一个构建 JBoss 企业级 BRMS 平台的开源项目。API、URL 和 JBoss Developer Studio 工具里都有对 Guvnor 的引用。

第 2 章 安装

阅读本章以学习如何安装 JBoss 企业级 BRMS 平台。

有多个系统配置支持 JBoss 企业级 BRMS 平台。关于它支持的 Java 应用服务器、虚拟机和数据库服务器的完整列表，请参考『<http://www.jboss.com/products/platforms/brms/supportedconfigurations/>』。

除了它支持的 Java 应用服务器、虚拟机和操作系统，部署 BRMS 到产品环境还需要一个数据库服务器以存储规则资产。

2.1. 软件包

JBoss 企业级 BRMS 平台可以从红帽客户支持入口 <https://access.redhat.com> 里下载，你可以进入 **Downloads、BRMS Platform**，然后选择版本。当前的版本是 **5.2.0**。下面所有的说明都采用 5.2.0 软件包名，但除非特别注明，它也可应用于早期版本。

JBoss 企业级 BRMS 平台有两种不同的软件包。

独立软件包

独立软件包包含 JBoss 企业级 Web 平台以及已部署的 JBoss 企业级 BRMS 平台，它不要求现有的应用服务器。

5.1.0 和更高版本里都有独立软件包。

如果下载 5.1.0 版本，独立软件包包含：

- JBoss 企业级 Web 平台 5.1
- 以 **default** 和 **production** 配置部署的 JBoss 企业级 BRMS 平台 5.1.0 web 应用程序。

如果下载 5.2.0 版本，独立软件包包含：

- JBoss 企业级 Web 平台 5.1.1。
- 以 **default** 和 **production** 配置部署的 JBoss 企业级 BRMS 平台 5.2.0 web 应用程序。

关于安装这些软件包的说明，请参考『[第 2.2.1 节 “安装独立软件包”](#)』。

可部署软件包

可部署软件包是一个 ZIP 文件，它包含打包为 WAR 文件的 JBoss 企业级 BRMS 平台的 web 应用程序，这个文件必须部署到被支持的应用服务器上。ZIP 软件包也包含可选的组件。

可部署软件包包含下列文件：

- **jboss-brms-manager.zip** - JBoss 企业级 BRMS 平台 web 应用程序。
- **jboss-brms-engine.zip** - JBoss 企业级 BRMS 平台的客户端库。
- **modeshape.zip** - 安装 ModeShape 库所需的文件。



注意

在 5.2 版本里，modeshape.zip 只是一个技术预览。

请参考『[第 2.2.2 节 “安装可部署软件包”](#)』里关于安装这个软件包的说明。

2.2. 安装说明

阅读本节来学习如何安装 JBoss 企业级 BRMS 平台。请遵循独立和可部署软件包的安装说明。

可选的安装步骤都有标签，关于进一步的信息请参考本指南的相关章节。

2.2.1. 安装独立软件包

独立软件包包含带有已作为展开归档部署的 JBoss 企业级 BRMS 平台的 JBoss 企业级 Web 平台。请按照下面的步骤来进行安装和配置。

注意

关于针对特殊环境的其他配置，请参考 JBoss 企业级 Web 平台 5 的文档：
: http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Web_Platform/。

过程 2.1. 安装独立软件包

1. 下载

从红帽客户支持入口 <https://access.redhat.com> 下载独立软件包 ZIP 文件。进入 **Downloads - BRMS Platform**，然后选择版本。当前的版本是 **5.2.0**。下面所有的说明都采用 5.2.0 软件包名，但除非特别注明，它也可应用于早期版本。

2. 解压

独立软件包包含带有已作为展开归档部署的 JBoss 企业级 BRMS 平台的 JBoss 企业级 Web 平台。解压 ZIP 文件到这个安装目录。

```
[localhost]$ unzip brms-standalone-5.2.0.zip
```

JBoss 企业级 BRMS 平台部署在 **production** 以及 **default** 服务器配置里：**brms-standalone-5.2.0/jboss-as-web/server/PROFILE/deploy/jboss-brms.war/**。

注意

如上所示，本指南在目录路径里使用 **PROFILE** 来指定 **production** 或 **default** 配置。

3. 启用用户

缺省的配置使用 **brms** JAAS 应用程序配置。这个配置在 **PROFILE/conf/props/brms-users.properties** 文件里存储用户名和密码。

你可以通过在这个文件里添加条目来增加用户，其格式是 **username=password**。在一行的开始使用井字符（#）可以注释该行的内容。

```
#admin=password  
jsmith=s@r@hSm1th  
tandrews=pp3rrss0nn311
```

关于修改 JAAS 配置以支持不同的验证系统（如 LDAP），请参考『[第 6.1 节“验证”](#)』。

4. 可选：安装 ModeShape 库

ModeShape 是一个 Java Content Repository (JCR)，它可以替代 Apache Jackrabbit。更多信息请参考『[第 2.2.3 节“安装和启用 ModeShape”](#)』。

**注意**

Modeshape 是 BRMS 5.2 里的技术预览。

5. 可选：启用规则软件包签名

规则软件包签名（Rule Package Signing）使用公共密钥加密（Public Key Cryptography）来确保软件包在从 BRMS 平台服务器下载到客户端应用程序时不会被损坏或改动。红帽推荐你在产品环境里启用这个功能。

请参考『[第 6.3 节“规则软件包签名（Rule Package Signing）”](#)』以学习这个主题。

6. 数据库配置

缺省配置使用了不适合产品环境的数据库。在部署到产品环境之前，你必须修改这个配置以使用被支持的数据库服务器。

请参考『[第 4 章 数据库配置](#)』。

7. 启动服务器

要从命令行启动服务器，请使用 **run.sh**（UNIX 或 Linux 系统），或 **run.bat**（Windows 系统）。**-c** 参数应用来指定所使用的服务器配置。

```
[localhost]$ ./run.sh -c production
```

run.sh 和 **run.bat** 脚本位于 **bin** 目录里。

例如，BRMS 5.2.0 独立服务器的 bin 目录的完整路径是 **brms-standalone-5.2.0/jboss-as-web/bin/**。

关于启动服务器的其他参数，请参考 JBoss 企业级 Web 平台 5 的文档：
：http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Web_Platform/。

8. 测试登录

登录到 JBoss 企业级 BRMS 平台 web 应用程序来确认它已经成功安装和运行。详情请参考『[第 2.3 节“第一次登录”](#)』。

2.2.2. 安装可部署软件包

可部署软件包包含『[第 2.1 节“软件包”](#)』里描述的软件。请按照下面的步骤来进行安装和配置。

过程 2.2. 安装可部署软件包**1. 下载**

从红帽客户支持入口 <https://access.redhat.com> 下载独立软件包 ZIP 文件。进入 **Downloads - BRMS Platform**，然后选择版本。当前的版本是 **5.2.0**。下面所有的说明都采用 5.2.0 软件包名，但除非特别注明，它也可应用于早期版本。

2. 解压软件

从 ZIP 文件里解压 BRMS 可部署文件。

```
[localhost]$ unzip brms-deployable-5.2.0.zip
```

这会创建下列文件：

- **jboss-brms.engine.zip**
- **jboss-brms.manager.zip**
- **modeshape.zip**

**注意**

在 BRMS 5.2 版本可部署软件包里，modeshape.zip 只是一个技术预览。

从 **jboss-brms-manager.zip** 里解压 **jboss-brms.war**。例如：

```
[localhost]$ unzip brms-manager.zip
```

这会创建一个 **jboss-brms.war** 目录，它包含展开归档形式的 JBoss 企业级 BRMS 平台 web 应用程序。

从 **jboss-brms-engine.zip** 里解压客户端库到一个目录里。例如：

```
[localhost]$ unzip brms-engine.zip -d client
```

这会创建一个包含客户端库 Jar 文件的目录。

3. 复制到应用服务器

复制展开的 web 应用程序归档到应用服务器的 deploy 目录。请确保这个归档的文件名是 **jboss-brms.war**。下面的例子演示了在 Linux 命令行下执行这个操作。

```
[localhost]$ cp -rf ~/temp/jboss-brms.war /opt/brms-server/jboss-as/server/production/deploy/
```

直到应用服务器启动后，这个应用程序才会被部署。

从前一步骤创建的 client 目录里复制 JAR 文件到应用服务器。详情请参考应用服务器的文档。

4. 启用用户

缺省的验证配置使用 **jmx-console** JAAS 配置。如果部署到一个现有的应用服务器，这个配置可能已经被修改过了。此时，请参考应用服务器的文档中关于添加新用户的说明。

**注意**

应用服务器里定义的任何用户都能够登录到 JBoss 企业级 BRMS 平台安装里。

红帽推荐对 JBoss 企业级 BRMS 平台用户使用单独的验证系统。请参考『[第 6.1 节“验证”](#)』以学习如何修改 JAAS 配置。

JBoss 产品里的 **jmx-console** 配置的缺省配置使用了 **UsersRolesLoginModule** 登录模块在 **PROFILE/conf/props/jmx-console-users.properties** 文件里存储用户名和密码。

你可以通过在这个文件里添加条目来增加用户，其格式是 **username=password**。通过井字符 (#) 可以注释一行的内容。

```
#admin=sekretz
jsmith=s@r@hSm1th
tandrews=p3r50n311
```

**注意**

如上所示，本指南在目录路径里使用 **PROFILE** 来指定 **production** 或 **default** 配置。

5. 可选：启用规则软件包签名

规则软件包签名（Rule Package Signing）使用公共密钥加密（Public Key Cryptography）来确保软件包在从 BRMS 平台服务器下载到客户端应用程序时不会被损坏或改动。红帽推荐你在产品环境里启用这个功能。

请参考『[第 6.3 节 “规则软件包签名（Rule Package Signing）”](#)』以学习这个主题。

6. 数据库配置

缺省配置使用了不适合产品环境的数据库。在部署到产品环境之前，你必须修改这个配置以使用被支持的数据库服务器。

请参考『[第 4 章 数据库配置](#)』。

7. 启动服务器

使用 **run.sh**（Linux）或 **run.bat**（Windows）来启动服务器。**-c** 参数可用来指定所使用的服务器配置集。

```
[localhost]$ ./run.sh -c production
```

run.sh 和 **run.bat** 脚本位于 **bin** 目录里。

例如，BRMS 5.2.0 独立服务器的 bin 目录的完整路径是 **brms-standalone-5.2.0/jboss-as-web/bin/**。

关于启动服务器的其他参数，请参考 JBoss 企业级 Web 平台 5 的文档：
：http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Web_Platform/。

8. 登录

登录到 JBoss 企业级 BRMS 平台 web 应用程序来确认它已经成功安装和运行。详情请参考『[第 2.3 节 “第一次登录”](#)』。

请参考『[附录 A, 在 EWS 上部署 JBoss BRMS](#)』。

2.2.3. 安装和启用 ModeShape



重要

JBoss 企业级 BRMS 平台以技术预览的方式提供了 ModeShape。

阅读本节来学习对于 BRMS 独立实例如何用 ModeShape 替换缺省的库 Apache Jackrabbit。

ModeShape 是一个 Java Content Repository (JCR)，它可以替代 Apache Jackrabbit。[过程 2.3, “配置 ModeShape”](#) 里介绍了安装 ModeShape 服务以及配置 JBoss 企业级 BRMS 平台使用 ModeShape。

安装 ModeShape 之前，下列要求必须被满足：

- BRMS 独立软件包已经被下载并解压，详情请参考『[第 2.2.1 节 “安装独立软件包”](#)』。



注意

ModeShape 是作为技术预览提供的，因此不支持移植现有的安装到 ModeShape。

- 安装了 Apache Ant，详情请参考『[附录 B, 安装 Apache Ant](#)』。
- 产品部署所要求的数据库服务器。
 - 已经创建了要使用的数据库实例。
 - 存在有修改数据库权限的数据库用户。

- 用于数据库的 JDBC 驱动 JAR 文件位于服务器配置的 **lib/** 目录。

关于被支持的数据库服务器的完整列表，请参考

『<http://www.jboss.com/products/platforms/brms/supportedconfigurations/>』。

过程 2.3. 配置 ModeShape

1. 下载并解压



注意

只有 5.1.0 版本需要这个步骤。5.2.0 独立软件包附带了 ModeShape。

从红帽支持入口 <https://access.redhat.com> 下载 ModeShape 软件包 (modeshape.zip)。

从 ZIP 文件里解压 **modeshape** 目录到 BRMS 的安装目录里。对于 JBoss 企业级 BRMS 平台的独立版本而言，它是 **brms-standalone-5.1.0**。

```
[localhost]$ unzip modeshape.zip
```

解压后，其目录具有下列的结构：

```
- brms-standalone-5.2.0
  | - jboss-as-web
  | - mod_cluster
  | - modeshape
  | - picketlink
  | - resteasy
  | - seam
```

2. 运行安装程序



注意

5.1.0 和 5.2.0 版本都需要下列步骤，从现在开始这些说明都指的是 5.2.0。

要在所需的服务器配置集里安装 ModeShape，请用命令行运行 **ant**，在提示时指定服务器配置集。

```
[localhost modeshape]$ ant
Buildfile: /opt/BRMS/5.1.0/brms-standalone-5.1.0/modeshape/build.xml

determine_home:

set-web-home:

set-as-home:

set-soa-home:

init:
    [echo] JBoss Home is ../jboss-as-web

prompt:
    [input] Enter profile to install ModeShape to: [default]
```

按照脚本将 ModeShape 服务安装到指定的服务器配置集里。

3. 设置用户和角色

添加用户帐号 **admin** 和 **mailman**，并分配下面指定的角色。缺省的配置使用文本文件来存储用户名、密码以及分配的角色。这里的指令假定授权配置未被修改。如果进行了修改，请按照这种配置添加用户。详情请参考『[第 6 章 安全性](#)』。

a. 添加用户 **admin** 和 **mailman**。

在文本编辑器里打开 **PROFILE/conf/props/brms-users.properties** 并添加用户 **admin** 和 **mailman** 到一新行里，其语法是：**username=password**。

```
admin=s3kr3t5
mailman=53cur3m@1l
```

b. 分配角色

用文本编辑器打开 **PROFILE/conf/props/brms-roles.properties**，为每个用户添加一行来分配下面指定的角色到这两个用户：**username=role1, role2, role3**。

admin 用户必须分配 **JBossAdmin**、**HttpInvoker**、**user** 和 **admin** 角色。

mailman 用户必须分配 **JBossAdmin**、**readwrite** 角色。

```
admin=JBossAdmin, HttpInvoker, user, admin
mailman=JBossAdmin, readwrite
```

c. 可选：加密密码

密码也可以以加密方式而不是明文进行存储。

```
[localhost]$ java -cp client/jboss-logging-spi.jar:lib/jbosssx.jar
org.jboss.resource.security.SecureIdentityLoginModule PASSWORD
Encoded password: 5f78dc15b9a559cbdf8592078de921bc
```

然后用这个加密的版本替换 **PROFILE/conf/props/brms-users.properties** 里的密码。

4. 可选：数据库配置

Modeshape 的缺省配置使用了不适合产品环境的数据库。在部署到产品环境之前，你必须修改这个配置以使用被支持的数据库服务器。

请参考『[第 4 章 数据库配置](#)』里的说明。

5. 删除未使用的 JAR 文件

删除 **server/PROFILE/deploy/jboss-brms.war/WEB-INF/lib/** 里的下列 JAR 文件。

- **jackrabbit-api-2.1.0.jar**
- **jackrabbit-core-2.1.0.jar**
- **jackrabbit-jcr-commons-2.1.0.jar**
- **jackrabbit-spi-2.1.0.jar**
- **hibernate-commons-annotations-3.1.0.GA.jar**
- **hibernate-core-3.3.2.GA.jar**
- **hibernate-validator-3.1.0.GA.jar**
- **lucene-core-2.4.1.jar**
- **jcr-2.0.jar**

6. 在 WAR 里启用

编辑 **jboss-brms.war/WEB-INF/components.xml** 文件，删除 Apache Jackrabbit 配置并添加

ModeShape 配置。

要删除 Apache Jackrabbit 配置，可以删除或注释下面的内容：

```
<property name="properties">
<key>org.drools.repository.configurator</key>
<value>org.drools.repository.jackrabbit.JackrabbitRepositoryConfigurator</value>
<!-- the root directory for the repo storage the directory must exist. -->
<!-- <key>repository.root.directory</key><value>/opt/yourpath</value> -->
</property>
```

缺省配置里是注释 ModeShape 配置的，你可以取消其注释：

```
<property name="properties">
<key>org.drools.repository.configurator</key>
<value>org.drools.repository.modeshape.ModeShapeRepositoryConfigurator</value>
<key>org.modeshape.jcr.URL</key><value>jndi:jcr/local?
repositoryName=brms</value>
<key>org.drools.repository.secure.passwords</key><value>>false</value>
<key>org.drools.repository.admin.password</key><value>admin</value>
<key>org.drools.repository.mailman.password</key><value>mailman</value>
</property>
```

7. 设置密码

ModeShape 要求两个用户帐号的存在：**admin** 和 **mailman**。这些帐号的密码必须在前一步骤的 ModeShape 配置里进行指定。

```
<key>org.drools.repository.secure.passwords</key><value>>false</value>
<key>org.drools.repository.admin.password</key><value>password</value>
<key>org.drools.repository.mailman.password</key><value>password</value>
```

8. 启动服务器

使用 **run.sh** (Linux) 或 **run.bat** (Windows) 来启动服务器。**-c** 参数可用来指定所使用的服务器配置集。

```
[localhost]$ ./run.sh -c production
```

run.sh 和 **run.bat** 脚本位于 **bin** 目录里。

例如，BRMS 5.2.0 独立服务器的 bin 目录的完整路径是 **brms-standalone-5.2.0/jboss-as-web/bin/**。

关于启动服务器的其他参数，请参考 JBoss 企业级 Web 平台 5 的文档：
http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Web_Platform/。

9. 登录

登录到 JBoss 企业级 BRMS 平台 web 应用程序来确认它已经成功安装和运行。详情请参考『第 2.3 节“第一次登录”』。

2.3. 第一次登录

JBoss 企业版 BRMS 平台为规则开发和管理提供了一个 web 用户界面。在成功安装后，你可以使用 web 浏览器来登录这个界面。

过程 2.4. 登录 JBoss 企业级 BRMS 平台

1. 我在浏览器里打开这个 URL

打开一个 web 浏览器并输入 URL <http://localhost:8080/jboss-brms/>，请用访问登录对话框的服务器的主机名代替 **localhost**。

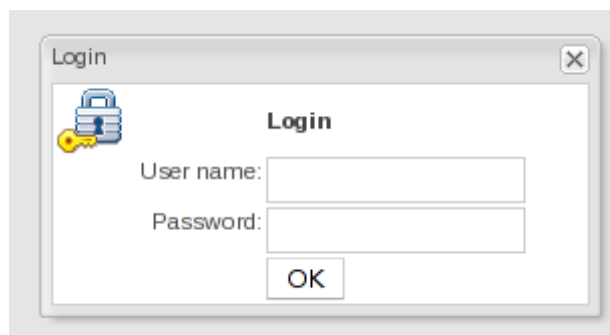


图 2.1. 登录对话框

2. 登录

输入用户凭证

如果这是全新的安装，系统会提示你是否创建一系列规则和资产示例。这些例子对于培训、测试和演示都是很有用的。

在安装了示例规则后，系统会加载 **Find** 页面。

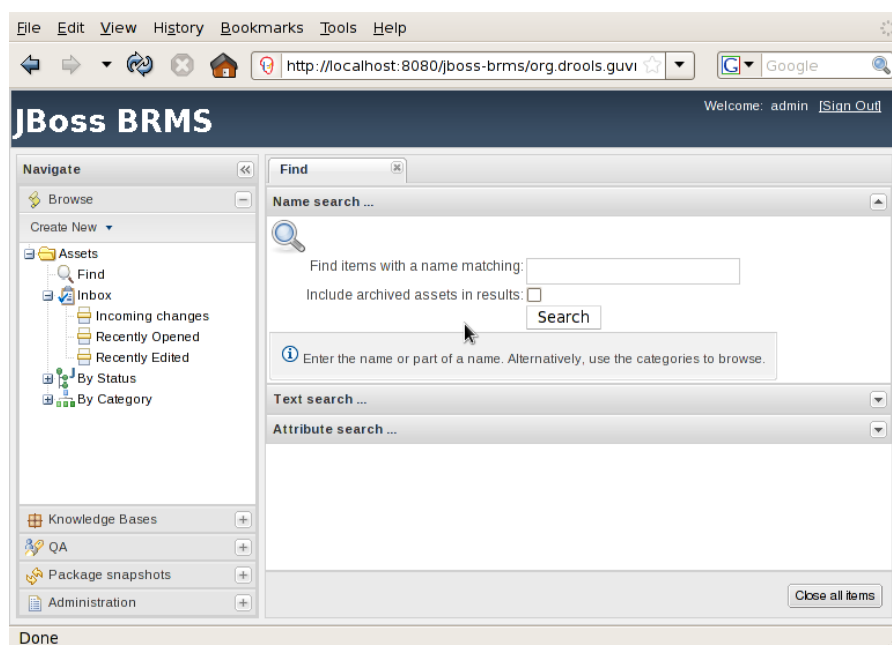


图 2.2. Find 页面

如果这个过程不是象预期那样进行，请查看『[第 2.4 节“登录：故障解除”](#)』。

2.4. 登录：故障解除

下面是第一次登录到 JBoss 企业级 BRMS 平台时会遇到的常见问题：

我在浏览器里打开 URL 后没有出现登录页面：

- ▶ 检查应用服务器是否正在运行，且是否已经完成了启动。根据你的服务器的负载，启动时间会

有所不同。

- 检查服务器配置的 URL 是否正确。应用服务器可能使用不同于缺省配置的主机名或端口来进行配置。
- 检测防火墙是否阻隔了访问。如果防火墙阻隔了访问，你需要配置它使其允许对服务器运行的端口进行访问。
- 检查是否所有的安装和配置步骤已经正确地完成。

在登录时我遇到错误信息 "You must configure a key store before proceeding"。

规则软件包签名虽被启用，但没有正确地进行配置。

请参考『[第 6.3 节 “规则软件包签名 \(Rule Package Signing\)”](#)』。

在登录时我遇到这样的信息：“Wrong username or password”。

登录用户名或密码不正确，或者是没有正确地配置验证系统。

- 检查用户名和密码。这些凭证是区分大小写的。
- 检查对验证配置的修改是否正确。

关于验证的配置，请参考『[第 6.1 节 “验证”](#)』。

如果这个问题仍然存在，请检查服务器日志。在缺省的情况下，出于性能考虑，**production** 服务器配置的日志有限。你需要配置日志以提供更多细节来帮助调试，但红帽推荐在解决了问题后恢复原来的日志配置。关于日志配置的详情，请参考『[第 7 章 日志](#)』。

如果这个问题仍然存在，请联系红帽支持。

第 3 章 本地化

JBoss 企业级应用程序平台的 web 用户界面可以以多种语言查看：

- 美式英语 (**en-US**)
- 日语 (**ja-JP**)
- 简体中文 (**zh-CN**)

如果未指定语言，缺省设置将是美式英语。

过程 3.1. 改变 Web 界面语言

1. 找到 index.jsp

index.jsp 位于 **jboss-brms.war** 目录里。

2. 编辑 index.jsp

编辑 **index.jsp**，指定另外的区域设置。缺省的 **index.jsp** 文件使用美式英语。

例 3.1. 缺省的 index.jsp

```
<%
String redirectURL = "org.drools.guvnor.Guvnor/Guvnor.html";
response.sendRedirect(redirectURL);
%>
```

要指定日语为缺省语言，可以编辑这个文件：

例 3.2. 以日语为缺省语言的 index.jsp

```
<%
String redirectURL = "org.drools.guvnor.Guvnor/Guvnor.html?locale=ja_JP";
response.sendRedirect(redirectURL);
%>
```

要指定简体中文为缺省语言，可以编辑这个文件：

例 3.3. 以简体中文为缺省语言的 index.jsp

```
<%
String redirectURL = "org.drools.guvnor.Guvnor/Guvnor.html?locale=zh_CN";
response.sendRedirect(redirectURL);
%>
```

『[表 3.1 “所支持的语言的 URL”](#)』里列出了所支持的语言。

表 3.1. 所支持的语言的 URL

语言	URL 路径
日语	<code>org.drools.guvnor.Guvnor/Guvnor.html?locale=ja_JP</code>
简体中文	<code>org.drools.guvnor.Guvnor/Guvnor.html?locale=zh_CN</code>
美式英语	<code>org.drools.guvnor.Guvnor/Guvnor.html?locale=en_US</code>

手动输入带有 locale 参数的 URL 可以对每个会话指定使用的语言。

所支持的语言的完整 URL

日语

http://<localhost>:8080/jboss-brms/org.drools.guvnor.Guvnor/Guvnor.html?locale=ja_JP

简体中文

http://<localhost>:8080/jboss-brms/org.drools.guvnor.Guvnor/Guvnor.html?locale=zh_CN

美式英语

http://<localhost>:8080/jboss-brms/org.drools.guvnor.Guvnor/Guvnor.html?locale=en_US

第 4 章 数据库配置

阅读本章来学习如何配置 JBoss 企业级 BRMS 平台库所使用的数据库、修改库的位置以及配置外部的关系型数据库。

JBoss 企业级 BRMS 平台可以使用 Apache Jackrabbit（缺省设置）或 Modeshape 作为其内容库。两者都使用数据库来存储数据，但配置不一样。下面的章节包含配置每个库的数据库的指令。



警告

Apache Jackrabbit 和 Modeshape 使用的缺省配置包括不适用于产品环境的数据库，分别是 Derby 和 Hypersonic。这个配置只是用于评估和演示目的，在产品环境里不被支持。产品部署要求被支持的数据库配置。请参考

『<http://www.jboss.com/products/platforms/brms/supportedconfigurations/>』里被支持的完整数据库系统列表。

4.1. 用于 Apache Jackrabbit 的数据库配置

Apache Jackrabbit 库存储两种不同的信息：*工作区 (Workspace)* 和 *版本 (Version)*。工作区是通用目的的存储系统，它保存规则资产和其他数据。版本保存所有对工作区的修改。工作区和版本可以保存在分开的数据库里，但红帽推荐你使用相同的数据库以方便管理。

JBoss 企业级 BRMS 平台库用 *持久化管理者 (persistence manager)* 管理对这些数据库的访问。系统提供了用于 JDBC 兼容数据库的通用持久化管理者，以及用于一些特定数据库的管理者。关于持久化管理者的更多信息，请参考『<http://wiki.apache.org/jackrabbit/PersistenceManagerFAQ>』，并请参考『附录 C, *持久化管理者配置示例*』里的示例。



警告

在修改数据库配置之前，请使用导入和导出功能（请参考『第 5.2 节“导入和导出”』）来保留任何存储在现有数据库里的资产。

请确保在处理之前准备好了下列事项：

- 一个 JBoss 企业级 BRMS 平台使用的空数据库。
- 一个 JBoss 企业级 BRMS 平台使用的数据库服务器用户帐号。JBoss 企业级 BRMS 平台将在指定的数据库里生成数据库表（如果不存在的话）。这一步完成后，红帽推荐你删除这个数据库用户创建表的权限。
- 用于数据库服务器的 JDBC 驱动 JAR。

4.1.1. 配置工作区数据库

现有的工作区数据库配置位于 `Workspaces/Default/workspace.xml` 文件里。`repository.xml` 文件包含用来创建新工作区的缺省设置。比更新现有缺省工作区更简单的方法是删除它并用 `repository.xml` 里的模板来更新。

过程 4.1. 修改库数据库

1. 关闭应用服务器
在进行这些修改前关闭应用服务器。
2. 安装 JDBC 驱动 JAR

复制包含 JDBC 驱动的 JAR 文件到服务器配置的 lib 目录，如 `jboss-as-web/PROFILE/production/lib/`。

3. 在文本编辑器里打开 `repository.xml`

在文本编辑器里打开 `repository.xml`。使用 JBoss 时，这个文件将驻留在应用服务器的 `bin` 目录里，除非指定了其他位置（如『[第 4.3.1 节 “修改 JackRabbit 库的位置”](#)』里所描述的）。

4. 更新 `PersistenceManager` 配置

找到 `<Workspace>` 里的 `<PersistenceManager>` 元素。

例如：

```
<PersistenceManager class=
  "org.apache.jackrabbit.core.persistence.bundle.DerbyPersistenceManager">
  <param name="url" value="jdbc:derby:${wsp.home}/db;create=true"/>
  <param name="schemaObjectPrefix" value="${wsp.name}_"/>
</PersistenceManager>
```

用合适的数据库配置替换这些内容。请参考『[附录 C, 持久化管理者配置示例](#)』里的例子。

5. 设置 `schemaObjectPrefix` 参数

这个值是通过持久化管理者管理的每个数据库表的名称前缀来指定的（对于每个数据库它必须是唯一的）。缺省配置使用包含当前工作区的 `${wsp.name}` 变量。

```
<param name="schemaObjectPrefix" value="${wsp.name}_"/>
```

6. 删除旧的工作区

删除 `workspaces/Default/` 目录。

7. 重新启动

重启应用服务器。

4.1.2. 配置版本数据库

JBoss 企业级 BRMS 平台维护着所有对数据库的修改的历史记录。为了在版本增多时防止性能降低，这些记录保存在单独的存储空间里。

这个存储空间具有自己的持久化管理配置，这些配置保存在 `repository.xml` 文件里。不象工作区配置，版本存储配置并不仅仅是一个模板，它是在使用的实际配置。

过程 4.2. 修改版本数据库

1. 关闭应用服务器

在进行这些修改前关闭应用服务器。

2. 安装 JDBC 驱动 JAR

复制包含 JDBC 驱动的 JAR 文件到服务器配置的 lib 目录，如 `jboss-as/server/PROFILE/lib/`。

3. 在文本编辑器里打开 `repository.xml`

在文本编辑器里打开 `repository.xml`。使用 JBoss 时，这个文件将驻留在应用服务器的 `bin` 目录里，除非指定了其他位置（如『[第 4.3.1 节 “修改 JackRabbit 库的位置”](#)』里所描述的）。

4. 更新 `PersistenceManager` 配置

找到 `<Workspace>` 里的 `<PersistenceManager>` 元素。

例如：

```
<PersistenceManager class=
  "org.apache.jackrabbit.core.persistence.bundle.DerbyPersistenceManager">
  <param name="url" value="jdbc:derby:${rep.home}/version/db;create=true"/>
  <param name="schemaObjectPrefix" value="version_"/>
</PersistenceManager>
```

用合适的数据库配置替换这些内容。请参考『附录 C. 持久化管理者配置示例』里的例子。

5. 设置 schemaObjectPrefix 参数

这个值是通过持久化管理者管理的每个数据库表的名称前缀来指定的（对于每个数据库它必须是唯一的）。如果相同的数据库用于版本以及数据存储，请确保没有使用相同的 schemaObjectPrefix（缺省配置使用 **version_**）。

```
<param name="schemaObjectPrefix" value="version_"/>
```

6. 重新启动

重启应用服务器。

4.1.3. 搜索和索引

JBoss 企业级 BRMS 平台的索引和搜索功能由 *Apache Lucene* (<http://lucene.apache.org/>) 提供。

在缺省的情况下，搜索索引保存在本地的文件系统里。这是因为它提供更快的性能。红帽不推荐修改这个缺省设置，除非你有其他的需要。

要配置搜索索引的位置，修改 **repository.xml** 文件的 **<SearchIndex>** 元素：

```
<SearchIndex class="org.apache.jackrabbit.core.query.lucene.SearchIndex">
  <param name="path" value="${wsp.home}/index"/>
  <param name="extractorPoolSize" value="2"/>
  <param name="supportHighlighting" value="true"/>
</SearchIndex>
```

4.2. 用于 ModeShape 的数据库配置

缺省的 Modeshape 库配置使用一个 JNDI 数据源 — **ModeShapeBRMSRepo** — for data storage which is configured in the **PROFILE/deploy/modeshape-brms-store-ds.xml** 文件。编辑这个文件来修改缺省的数据源配置 Hypersonic 为其他的数据库。这个文件是一个标准的 JBoss 数据源配置文件。

关于配置 JBoss 数据源的细节，请参考应用服务器文档的数据源配置章节。

4.3. 库配置

阅读本节来学习库配置以及配置外部的关系型数据库。

JBoss 企业级 BRMS 平台使用 JCR（Content Repository API for Java）规格的 API 来存储和跟踪资产（例如规则）。Apache Jackrabbit 是它附带的 JCR 实现。

4.3.1. 修改 JackRabbit 库的位置

阅读本节来学习如何修改库的位置。

当 JBoss 企业级 BRMS 平台应用程序第一次启动时，它会创建一个库（repository）。除非你修改缺省的配置，否则这个库将在应用程序的 **bin** 目录里生成。

通过编辑 JBoss Seam 的 **components.xml** 配置文件指定不同的位置。请选择一个经常进行备份的安全位置。

过程 4.3. 修改库位置

1. 关闭应用服务器

在进行这些修改之前先请停止应用服务器。

2. 找到 **components.xml** 文件

components.xml 文件位于已部署的 BRMS web 归档的 **WEB-INF/** 目录下 (**deploy/jboss-brms.war/WEB-INF/**)。它是一个标准的 JBoss Seam 配置文件，它可以对应用程序的不同方面进行定制。

3. 找到 **repository.root.directory** 键-值属性。

在文本编辑器里打开 **components.xml** 并找到 **repository.root.directory** 属性。它包含在 **repositoryConfiguration** 组件里且缺省是被注释的。

```
<property name="properties">
  <key>org.drools.repository.configurator</key>

  <value>org.drools.repository.jackrabbit.JackrabbitRepositoryConfigurator</value>
</property>

<!-- the root directory for the repo storage the directory must exist. -->
<!-- <key>repository.root.directory</key><value>/opt/yourpath</value> -->
```



注意

对于 BRMS 5.1 之前的版本，请遵循下面的步骤：

- 找到 **homeDirectory** 属性
- 在文本编辑器里打开 **components.xml** 并找到 **homeDirectory** 属性。它包含在 **repositoryConfiguration** 组件里且缺省是被注释的。

```
<component name="repositoryConfiguration">
  <!--
  *** This is for configuring the "home" directory for the repository
  storage. the directory must exist. ***
  <property name="homeDirectory">
    /home/michael/RulesRepository_001
  </property>
  -->
</component>
```

4. 更新 **repository.root.directory** 属性。

删除 **repository.root.directory** 属性周围的注释标记并将其值修改为数据库所在的路径（这个目录必须已经存在）。

```
<property name="properties">
  <key>org.drools.repository.configurator</key>

  <value>org.drools.repository.jackrabbit.JackrabbitRepositoryConfigurator</value>
  <!-- the root directory for the repo storage the directory must exist. -->
  <key>repository.root.directory</key>
  <value>/opt/brms-standalone-5.2.0/BRMSRulesRepository</value>
</property>
```



注意

对于 BRMS 5.1 之前的版本，其代码是：

```
<component name="repositoryConfiguration">
  <property name="homeDirectory">/opt/jboss-soa-
platform/BRMSRulesRepository</property>
</component>
```

5. 移动现有的数据库（可选）

JBoss 企业级 BRMS 平台将在这个位置创建一个新的数据库（如果之前不存在）。要保留现有的数据库，在重启服务器之前复制现有的文件到新的位置。

6. 重启应用服务器

重启服务器。如果没有移动现有的数据库到新的位置，新的数据库将被创建。

4.3.2. 配置 BRMS 使用外部的关系型数据库。

你可以配置 JBoss 企业级 BRMS 平台使用外部的数据库。**repository.xml** 包含关于库数据存储位置的信息。你必须修改这个文件以设立外部的 RDBMS。你可以手动编辑这个文件或通过 BRMS 用户界面里的 **Repository Configuration** 进行编辑。

要手动编辑 **repository.xml** 文件，找到这个文件并用文本编辑器打开它。这个文件用描述选项的注释进行了注解。

除非指定不同的主目录（如 [第 4.3.1 节“修改 JackRabbit 库的位置”](#) 所描述的），**repository.xml** 文件位于应用服务器的 **bin** 目录里。

关于配置示例，请参考『[附录 C, 持久化管理者配置示例](#)』。

过程 4.4. 用 Repository Configuration 工具配置外部 RDBMS

1. 登录用户界面

打开一个 web 浏览器并输入 URL <http://localhost:8080/jboss-brms/>，请用访问登录对话框的服务器的主机名代替 **localhost**。

2. 选择 Repository Configuration 工具

在导航面板里，选择 **Administration**，然后再选择 **Repository Configuration**。

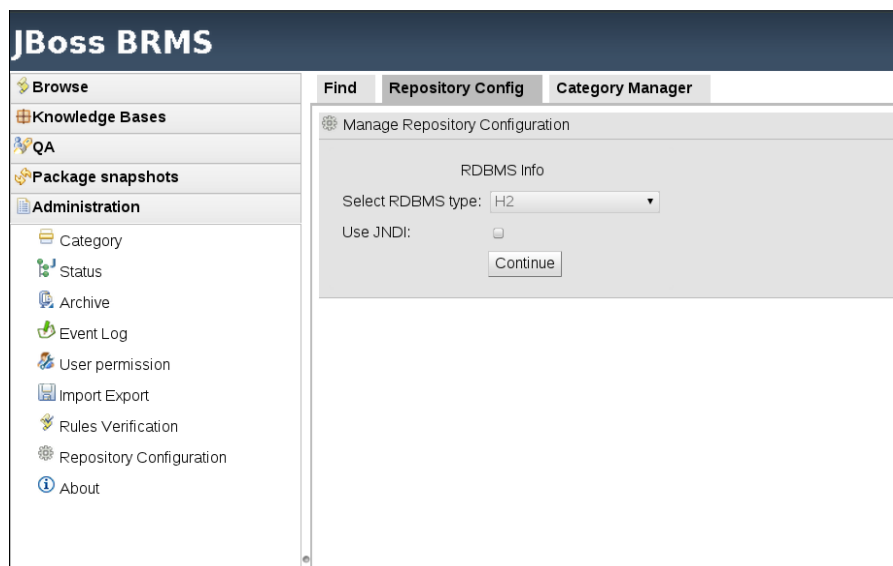


图 4.1. 库配置（Repository Configuration）工具

3. 选择 RDBMS 类型

从 **Select RDBMS type:** 下拉菜单里，选择 RDBMS 类型。

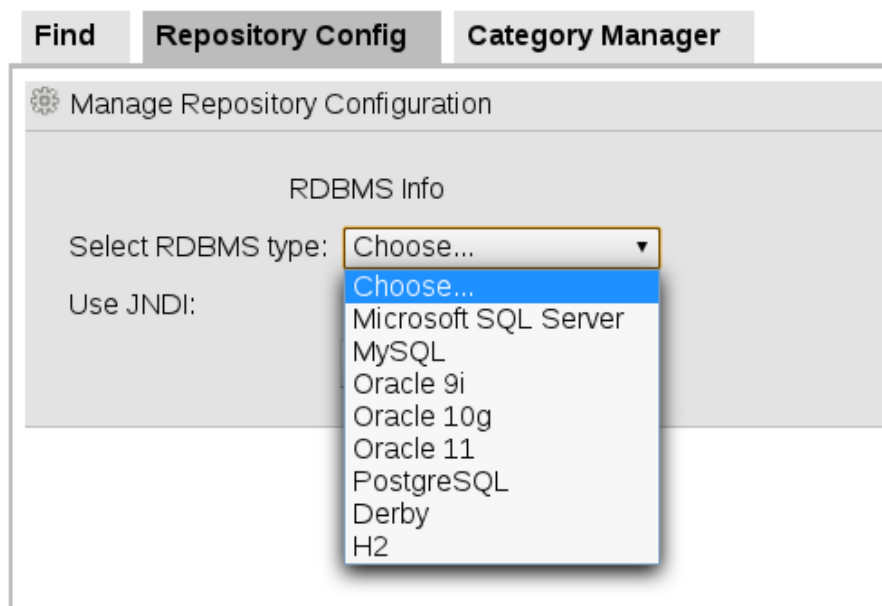


图 4.2. RDBMS 类型

4. 输入 RDBMS 细节

如果配置了 JNDI，选中 **USE JNDI** 复选框。

点击 **Continue** 并输入在部署的数据源里配置的 JNDI 名称。

如果没有使用 JNDI，输入所需的 RDBMS 信息：

- 驱动
- URL
- 用户
- 密码

The screenshot shows a software configuration window titled "Manage Repository Configuration". It has three tabs: "Find", "Repository Config" (which is selected), and "Category Manager". The "Repository Config" tab contains two sections. The first section, "RDBMS Info", has a dropdown menu for "Select RDBMS type:" currently showing "MySQL", an unchecked checkbox for "Use JNDI:", and a "Continue" button. The second section, "MySQL Info", contains four text input fields labeled "Driver:", "URL:", "User:", and "Password:". Below these fields is a button labeled "Generate repository config".

图 4.3. RDBMS 信息

5. 生成库的配置文件

选择 **Generate repository config** 以生成 **repository.xml** 文件。

你可以剪切或粘贴生成的文本到一个文件里并保存为 **repository.xml**，或者是选择 **Save Configuration** 将这些文本作为文件下载。用新的 **repository.xml** 替换现有的 **repository.xml**。除非指定了不同的主目录（如 [第 4.3.1 节“修改 JackRabbit 库的位置”](#) 里所描述的），**repository.xml** 将位于应用服务器的 **bin** 目录里。

第 5 章 数据管理

5.1. 数据备份

JBoss 企业级 BRMS 平台没有提供备份方案。我们应该使用数据库供应商提供的工具来执行数据库备份。

当恢复备份时，红帽推荐首先清除数据库索引，然后可为新数据重建和优化。



警告

请不要使用库导入和导出功能而不是数据库供应商提供的工具来进行库备份。如『[第 5.2 节“导入和导出”](#)』里所描述的，它有一些限制，所以不适合作为备份工具，且红帽不为这种功能提供支持。

5.2. 导入和导出

JBoss 企业级 BRMS 平台提供了 JCAR 标准的导出和导入功能，用于数据库之间的移植。导出的库安装 JCR 标准写入到 XML 文件里。通过导航面板里管理部分可以执行这个功能。



警告

导入/导出功能的目的是不是用作备份方案。请使用数据库供应商提供的备份工具。

当执行导入/导出操作时请考虑下列要点：

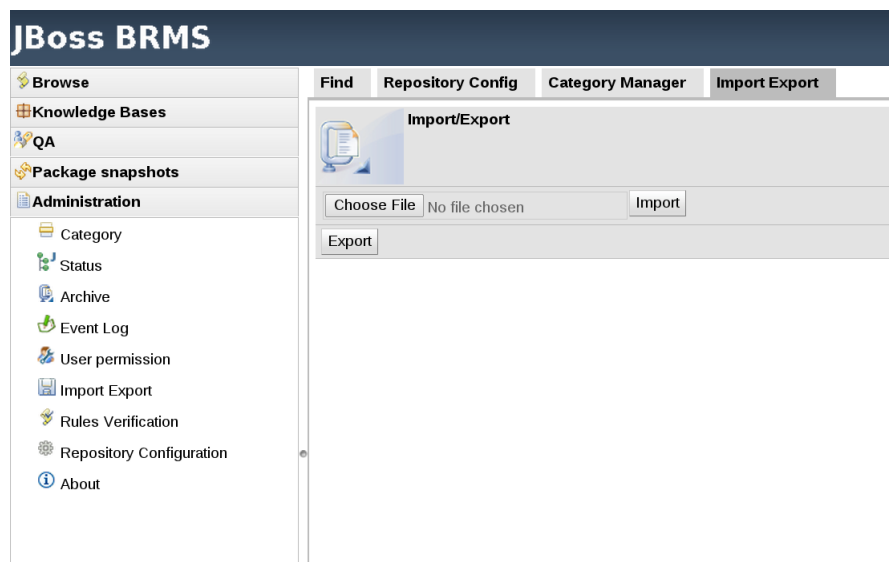
- 导入时，数据库里现有的内容将被删除。
- 当没有使用数据库时不应该执行导出或导入操作。
- 性能非常依赖于数据库的大小和服务器里可用的内存。导入是一个非常消耗内存的过程。
- 库版本的历史不会被导出，规则的 Creation Date 属性将重设为它们被导入的日期。

5.3. 在不同的 BRMS 版本间移植数据

要从旧版本的 JBoss 商业规则管理系统里移植库到当前的版本，你需要从旧的 JBoss 商业规则管理系统实例导出库（如『[过程 5.1, “导出库”](#)』里所描述的），然后再导入库到当前的版本（如『[过程 5.2, “导入库”](#)』里所描述的）。

过程 5.1. 导出库

1. 从左侧的导航面板里选择 **Administration**。
2. 点击 **Import Export**。
3. 点击 **Export**。



浏览器将下载一个包含 XML 文件里的库的 ZIP 文件。
根据库的大小，这个操作可能需要一些时间。

注意

当导入库时，请注意库里的任何内容都将被删除并用导入的 XML 文件里的内容替代。

过程 5.2. 导入库

1. 从左侧的导航面板里选择 **Administration**。
2. 点击 **Choose File** 从本地机器里选择将要上载的 XML 文件，然后点击 **Import**。
3. 点击 **Import**。

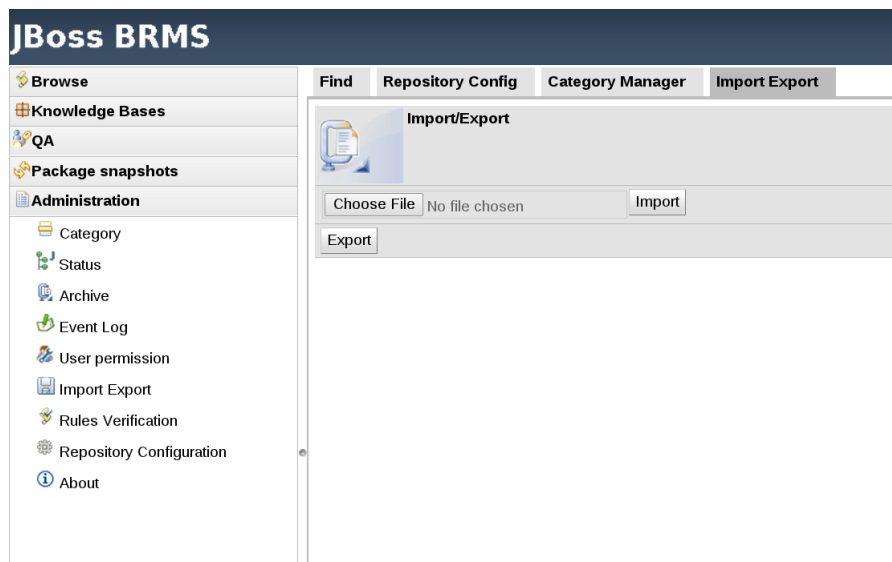


图 5.1. 导入和导出库

当导入结束后，将出现一个对话框并显示 "Rules repository imported successfully. Browser will now refresh to show the new content."。

5.4. 重构二进制软件包

在升级了 JBoss 商业规则管理系统并导入现有的库后，我们有必要重构二进制软件包。

要重构二进制软件包，从 **Navigation Menu** 里选择 **Knowledge Bases**，然后点击 **Rebuild all package binaries**。

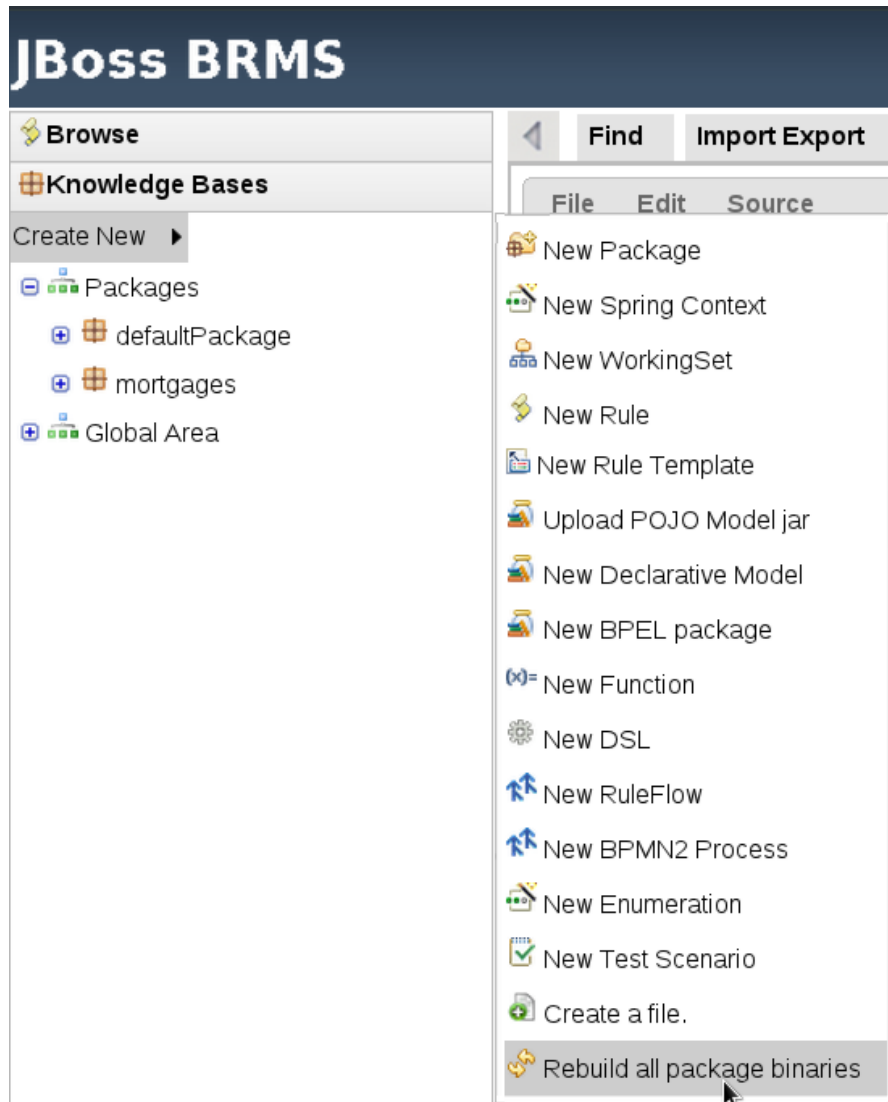


图 5.2. 重构软件包

第 6 章 安全性

阅读本章来学习如何启用和配置 JBoss 企业级 BRMS 平台里的不同安全性系统。这里涵盖的内容有：

- 验证
- 授权
- 规则软件包签名 (Rule Package Signing)

6.1. 验证

JBoss 企业级 BRMS 平台使用 *Java Authentication and Authorization Service (JAAS)* 来检验用户凭证。这个服务通过应用服务器来提供且用于访问单独的验证系统。单独的系统可以是轻量级目录访问协议 (LDAP)、活动目录服务器 (Active Directory server) 或 JDBC 数据库。



重要

当用户被赋予访问 JBoss 企业级 BRMS 平台的权限时，它们被潜在地给予了影响其他应用程序的商业逻辑的能力。使用基于角色的授权来定义每个用户能做和不能做的事情。关于这个主题的更多内容，请参考『[第 6.2 节“授权”](#)』。

你可以通过 `jboss-brms.war/WEB-INF/components.xml` 文件配置使用哪种验证方法。缺省的配置会有多个“被注释”的选项，但实际的设置类似于：

```
<security:identity authenticate-method="#{authenticator.authenticate}" jaas-config-
name="jmx-console"/>
<component name="org.jboss.seam.security.roleBasedPermissionResolver">
  <property name="enableRoleBasedAuthorization">false</property>
</component>
```



注意

`components.xml` 文件在 BRMS 5.2 里进行了修改。在 5.1 和更早版本里，这个文件类似于：

```
<security:identity authenticate-method="#{authenticator.authenticate}" jaas-
config-name="jmx-console"/>
<security:role-based-permission-resolver enable-role-based-
authorization="false"/>
```



重要

这个缺省配置使用 `jmx-console` 验证策略里定义的帐号名、密码和角色。红帽推荐你编辑这个策略以满足特定环境的需要。

要配置验证，请按照下列步骤：

1. 编辑应用服务器的合适的 JBoss 登录模块。
2. 配置 JBoss 企业级 BRMS 平台来使用这个模块。



注意

许多 JBoss 登录模块都提供为每个用户指定多个角色的方法。JBoss 企业级 BRMS 平台有自己的管理用户角色的机制。



警告

如果禁用了 *基于角色的授权 (role-based authorization)*，所有用户都事实上具有了 admin 权限。这使它们具有对 JBoss 企业级 BRMS 平台的完全访问权限。



重要

红帽推荐你启用基于角色的授权。在此之前，在 **BRMS Permissions** 界面里为至少一个用户分配 admin 角色，只有具有 admin 权限的用户可以执行许多管理性的任务，如管理用户角色（『[第 6.2 节 “授权”](#)』里会进一步进行解释）。

6.1.1. 验证示例：UserRolesLoginModule

这个例子解释了使用 `org.jboss.security.auth.spi.UsersRolesLoginModule` 登录模块来访问存储在 `props/brms-users.properties` 和 `props/brms-roles.properties` 文件里的一系列用户帐号。

过程 6.1. 验证示例：UserRolesLoginModule

1. 确保已经正确配置了验证系统

登录模块使用两个文件来存储登录名、密码以及分配给用户的角色。在 `jboss-as-web/server/PROFILE/conf/props/` 目录里创建 `brms-users.properties` 和 `brms-roles.properties` 文件并用这个格式在 `brms-users.properties` 里指定至少一个用户：`username=password`。（`brms-roles.properties` 可以为空）

2. 关闭应用服务器

在进行这些修改前关闭应用服务器。

3. 配置 JBoss 登录模块

要配置 JBoss 登录模块，打开 `jboss-as-web/server/PROFILE/conf/login-config.xml` 文件。它是一个包含 `<policy>` 元素以及几个 `<application-policy>` 子元素的 XML 文件。每个 `<application-policy>` 元素都定义了一个不同的验证模式。添加下列 `<application-policy>` XML 片段作为 `<policy>` 元素的新的子元素。

```

<!--BRMS Platform Security Domain-->
<application-policy name="brms">
  <authentication>
    <login-module
      code="org.jboss.security.auth.spi.UsersRolesLoginModule"
      flag="required">
      <module-option name="usersProperties">
        props/brms-users.properties
      </module-option>
      <module-option name="rolesProperties">
        props/brms-roles.properties
      </module-option>
    </login-module>
  </authentication>
</application-policy>

```

4. 配置 BRMS 平台以使用登录模块

打开 `jboss-as-web/server/PROFILE/deploy/jboss-brms.war/WEB-INF/components.xml` 为。它包含一个 `<components>` 元素并带有几个子元素，其中包括 `<security:identity>`。

注释现有的 `<security:identity>` 元素以防止冲突。添加下列 `<security:identity>` 元素。

```

<security:identity authenticate-
method="#{authenticator.authenticate}" jaas-config-name="brms"/>

```

`jaas-config-name` 属性必须和 `application-policy` 相同。如果在前一步骤修改了 `application-policy`，也请修改这里的 `jaas-config-name` 属性进行匹配。

5. 重新启动

重启应用服务器

6.1.2. 验证示例：LDAP

LDAP 是大型企业的流行的选项。基本的配置步骤和前面例子里的相同，但配置细节有所不同。

过程 6.2. 验证示例 2：LDAP

1. 确保已经正确配置了 LDAP 服务器

确定防火墙和网络配置没有阻断应用服务器和 LDAP 服务器间的通讯。

2. 关闭应用服务器

在进行这些修改前关闭应用服务器。

3. 配置 JBoss 登录模块

要配置 JBoss 登录模块，打开 `jboss-as-web/server/PROFILE/conf/login-config.xml` 文件。它是一个包含 `<policy>` 元素以及几个 `<application-policy>` 子元素的 XML 文件。每个 `<application-policy>` 元素都定义了一个不同的验证模式。添加下列 `<application-policy>` XML 片段作为 `<policy>` 元素的新的子元素。

```
<application-policy name="brms">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.LdapExtLoginModule"
      flag="required" >
      <module-option name="java.naming.provider.url">
        ldap://ldap.company.com:389
      </module-option>
      <module-option name="bindDN">DEPARTMENT\someadmin</module-option>
      <module-option name="bindCredential">password</module-option>
      <module-option name="baseCtxDN">cn=Users,dc=company,dc=com
      </module-option>
      <module-option name="baseFilter">(sAMAccountName={0})</module-option>
      <module-option name="rolesCtxDN">cn=Users,dc=company,dc=com
      </module-option>
      <module-option name="roleFilter">(sAMAccountName={0})</module-option>
      <module-option name="roleAttributeID">memberOf</module-option>
      <module-option name="roleAttributeIsDN">true</module-option>
      <module-option name="roleNameAttributeID">cn</module-option>
      <module-option name="roleRecursion">-1</module-option>
      <module-option name="searchScope">ONELEVEL_SCOPE</module-option>
    </login-module>
  </authentication>
</application-policy>
```

用适合你的 LDAP 服务器的值更新这个配置文件。

4. 配置 BRMS 平台以使用登录模块

打开 `jboss-as-web/server/PROFILE/deploy/jboss-brms.war/WEB-INF/components.xml` 为。它包含一个 `<components>` 元素并带有几个子元素，其中包括 `<security:identity>`。

注释现有的 `<security:identity>` 元素以防止冲突。添加下列 `<security:identity>` 元素。

```
<security:identity authenticate-method="#{authenticator.authenticate}" jaas-
  config-name="brms"/>
```

jaas-config-name 属性必须和 application-policy 相同。如果在前一步骤修改了 application-policy，也请修改这里的 jaas-config-name 属性进行匹配。

5. 重新启动

重启应用服务器

注意

要学习不同 LDAP 配置场景的更多细节，请阅读这两个网页：
<http://www.jboss.org/community/wiki/LdapLoginModule> 和
<http://www.jboss.org/community/wiki/LdapExtLoginModule>。

6.2. 授权

BRMS 使用基于角色的授权来为用户分配权限。在缺省的情况下，基于角色的授权是被禁用的，你必须启用它。类似地，你必须分配角色给用户。如果没有启用基于角色的授权，所有用户都会具有管理权限。

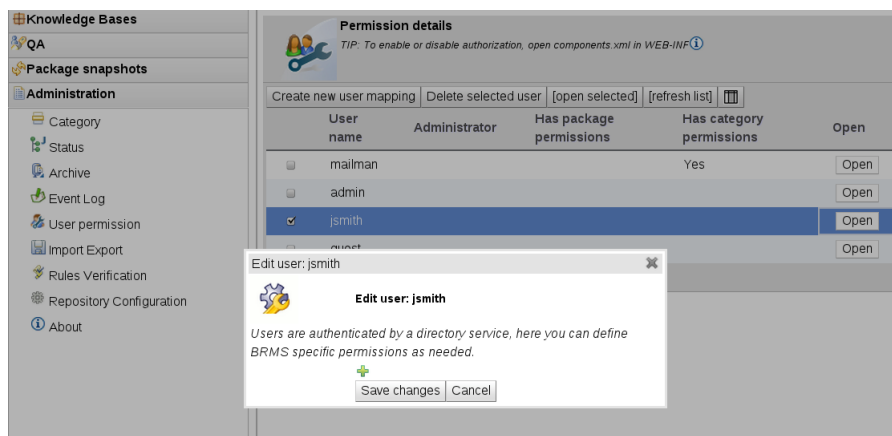


图 6.1. 管理用户权限

过程 6.3. 启用基于角色的授权

1. 分配管理员权限给用户

在启用基于角色的授权之前，我们有必要为至少一个信任的用户分配管理员角色。

详情请参考『[过程 6.5, “管理用户权限”](#)』。

2. 关闭应用服务器

在进行这些修改前关闭应用服务器。

3. 打开 components.xml

在文本编辑器里打开 `jboss-as-web/server/PROFILE/deploy/jboss-brms.war/WEB-INF/components.xml` 文件。

4. 找到 `<property name="enableRoleBasedAuthorization">` 元素

在缺省的 `components.xml` 文件里，这个 XML 元素是 `<components>` 的子元素。

```
<component name="org.jboss.seam.security.roleBasedPermissionResolver">
  <property name="enableRoleBasedAuthorization">false</property>
</component>
```



注意

在 5.1 和更早的版本里，找到下列 XML

```
<security:role-based-permission-resolver
  enable-role-based-authorization="false"/>
```

5. 更新属性值为 "True"

更新 `enable-role-based-authorization` 属性为 `true` 并保存文件。

```
<component name="org.jboss.seam.security.roleBasedPermissionResolver">
  <property name="enableRoleBasedAuthorization">true</property>
</component>
```

注意

在 5.1 和更早的版本里，找到下列 XML

```
<security:role-based-permission-resolver
    enable-role-based-authorization="true"/>
```

6. 重新启动

重启应用服务器

JBoss 企业级 BRMS 平台不管理用户的身份。只有分配为 BRMS 用户的用户才会在 JBoss 企业级 BRMS 平台的用户权限里可见。

过程 6.4. 添加新用户到 BRMS 里**1. 选择权限细节**

从 **navigation pane** 里选择 **Administration**，然后选择 **User Permissions**。

2. 添加用户映射

点击 **Create new user mapping** 按钮。在对话框里输入用户名，并点击 **OK**。

注意

为该角色指定的用户名必须匹配验证服务里的一个用户名，否则它无法使用。

3. 添加权限

详情请参考『[过程 6.5, “管理用户权限”](#)』。

过程 6.5. 管理用户权限**1. 选择用户权限**

从 **navigation pane** 里选择 **Administration**，然后选择 **User Permissions**。

2. 选择用户

点击用户名旁边的 **Open**

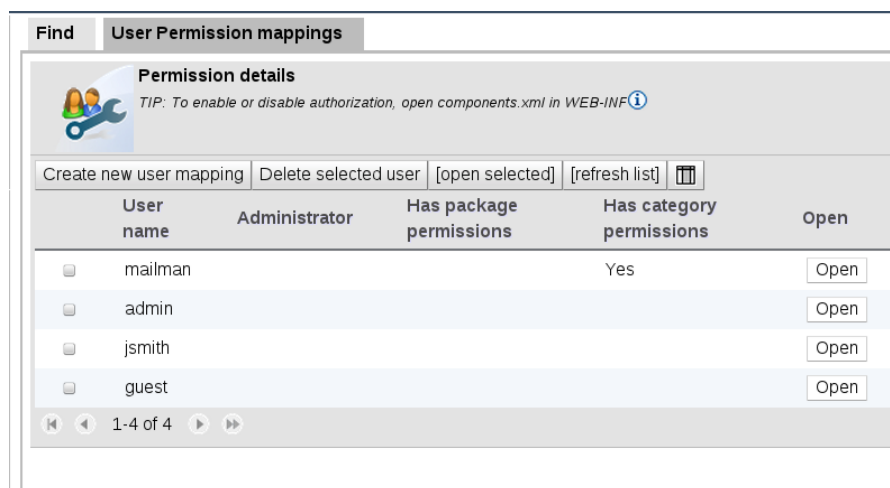


图 6.2. 权限细节

3. 分配用户权限

点击加号图标来添加权限，然后从 **Permission type** 下列菜单里选择合适的权限。点击 **OK** 进行确认。

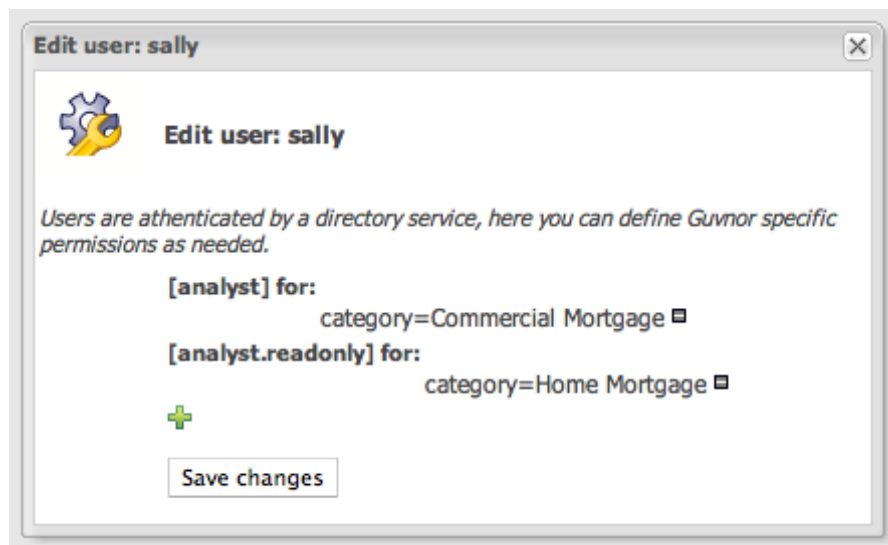


图 6.3. 编辑用户权限

4. 删除用户权限

点击要删除的权限旁边的减号图标，然后点击 **OK** 来确认。

分配了 **admin** 权限的用户可以修改其他用户的角色和权限。

用户可以被分配下列三种角色：

- admin
- analyst
- package

分配了 **admin** 角色的用户具有对 JBoss 企业级 BRMS 平台所有方面的访问权限。

Analyst 权限适用于负责维护规则资源的用户。开发人员和商业分析师应该具有这个级别的权限。

注意

当为用户分配 **analyst** 权限时，你会被提示输入 *category*。*Category* 是一种将规则分组的方法，它独立于这些规则所属的知识软件包。

注意

当 **analyst** 负责维护某个类别里的规则且需要检查另外一个类别里的规则时，你可以使用只读 **analyst** 权限。

**重要**

只赋予了 category 权限的用户将不能查看任何软件包视图细节：它们只能查看 **Simple Categories** 视图。

当分配角色时，你会被提示选择软件包权限。有三种不同的软件包权限可分配给用户：

软件包管理员 (Package Administrator)

软件包管理员权限赋予对指定软件包的完全控制，包括部署的权利。但软件包管理员权限不赋予 JBoss 企业级 BRMS 平台其他部分的任何管理权限。

软件包开发人员 (Package Developer)

软件包开发人员权限允许用户在指定的软件包内创建和编辑条目。如创建和运行测试用例，但不包括部署软件包的权利。

只读软件包 (Package Read-only)

只读软件包权限和 analyst read-only 权限类似，但它分配给软件包而不是类别。

6.3. 规则软件包签名 (Rule Package Signing)

阅读本节来学习规则软件包签名以及密钥库的配置。

规则软件包签名 (Rule Package Signing) 确保了软件包在从 BRMS 平台服务器下载到客户端应用程序时不会被损坏或改动。在缺省情况下它是禁用的。

**重要**

红帽强烈推荐在产品环境里启用规则软件签名。

规则软件包签名是用公共密钥加密 (Public Key Cryptography) 来实现的。JDK 命令 **keytool** 用来创建私有密钥和对应的公共数字证书。用私有密钥签名的软件包只可以用匹配的证书来检验。私有密钥存储在名为 **keystore** 的文件里，服务器用它来自动为每个软件包签名。被称为 **信任库 (truststore)** 里的公共证书对于每个客户端应用程序可用。信任库里的证书用来检验签名软件包的真实性。在下载过程中已损坏或修改规则软件包将被客户端拒绝，因为其签名和证书不再匹配。

下面的过程描述了配置用于规则软件包签名的服务器的过程。

在这个过程中，你必须：

- 创建一个私有签名密钥和对应的公共数字证书。
- 使私有签名密钥和公共数字证书对于密钥库里的服务器可用。
- 配置服务器以使用密钥库。

过程 6.6. 配置规则软件包签名**1. 创建私有密钥**

使用 **keytool** 创建私有密钥：

```
keytool -genkey -alias ALIAS -keyalg RSA -keystore PRIVATE.keystore
```

-alias 参数指定了用来链接密钥库里相关实体的名称。对于每个步骤都使用相同的别名。别名是不区分大小写的。**-keystore** 提供了将被创建以保存私有密钥的文件名称。

keytool 将提示你确定一些信息以及两个密码。第一个密码是**密钥库密码**。第二个密码是密钥密码，用于正创建的密钥。

```
[localhost]$ keytool -genkey -alias BRMSKey -keyalg RSA -keystore
PrivateBRMS.keystore
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: John Smith
What is the name of your organizational unit?
[Unknown]: Accounts
What is the name of your organization?
[Unknown]: ACME INC
What is the name of your City or Locality?
[Unknown]: Captial City
What is the name of your State or Province?
[Unknown]: CC
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=John Smith, OU=Accounts, O=ACME INC, L=Captial City, ST=CC, C=US
correct?
[no]: yes
Enter key password for <BRMSKey>
(RETURN if same as keystore password):
Re-enter new password:
```

2. 创建数字证书

使用 **keytool** 创建数字证书：

```
keytool -export -alias ALIAS -file CERTIFICATE.crt -keystore
PRIVATE.keystore
```

使用和前一步骤相同的别名和密钥库。**-file** 参数指定要创建的新证书的名称。**-keystore** 参数指定私有密钥库的文件名。

在提示符后输入密钥库的密码。

```
[localhost]$ keytool -export -alias BRMSKey -file BRMSKey.crt -keystore
PrivateBRMS.keystore
Enter keystore password:
Certificate stored in file <BRMSKey.crt>
```

3. 将数字证书导入到信任库

使用 **keytool** 命令将数字证书导入到密钥库里：

```
keytool -import -alias ALIAS -file CERTIFICATE.crt -keystore
PUBLIC.keystore
```

这会创建一个新的密钥库和包含数字证书的信任库。信任库使数字证书可为客户端应用程序所用。

```
[localhost]$ keytool -import -alias BRMSKey -file BRMSKey.crt -keystore
PublicBRMS.keystore
Enter keystore password:
Re-enter new password:
Owner: CN=John Smith, OU=Accounts, O=ACME INC, L=Captial City, ST=CC, C=US
Issuer: CN=John Smith, OU=Accounts, O=ACME INC, L=Captial City, ST=CC, C=US
Serial number: 4ca0021b
Valid from: Sun Sep 26 22:31:55 EDT 2010 until: Sat Dec 25 21:31:55 EST 2010
Certificate fingerprints:
    MD5: 31:1D:1B:98:59:CC:0E:3C:3F:57:01:C2:FE:F2:6D:C9
    SHA1: 4C:26:52:CA:0A:92:CC:7A:86:04:50:53:80:94:2A:4F:82:6F:53:AD
    Signature algorithm name: SHA1withRSA
    Version: 3
Trust this certificate? [no]: yes
Certificate was added to keystore
```

4. 将私有密钥库移至安全位置

私有密钥库需要保存在只有 JBoss 企业级 BRMS 平台服务器能够访问的安全位置。它可以在相同的机器或是安全的网络位置上。



重要

JBoss 企业级 BRMS 平台不能为网络资源提供验证凭证。如果私有密钥库保存在安全的网络位置，那么任何验证过程都必须代替 JBoss 企业级 BRMS 平台服务器来执行，从而使得私有密钥库为之所用。例如，操作系统可以验证并挂载一个保存私有密钥库的共享文件为本地目录，以供 JBoss 企业级 BRMS 平台服务器进行访问。

5. 将信任库移至公共位置

信任库需要可被客户端应用程序访问。这可以通过把信任库放在网络共享或 web 服务器上来实现。

6. 设置 Drools 序列化属性

你需要在服务器上设置 Drools 序列化系统属性。这是存储访问密钥库所需信息的属性。因为 JBoss 企业级 BRMS 平台也包含客户组件，所以私有密钥库和信任库属性都需要进行设置。

关于在哪里设置这些属性的细节，请参考『[第 6.3.1 节“设置序列化属性”](#)』。

需要设置的属性是：

- **drools.serialization.sign** - 指定是否启用签名。它必须为 **true**。
- **drools.serialization.private.keyStoreURL** - 私有密钥库所在的 URL
- **drools.serialization.private.keyStorePwd** - 信任库的密码
- **drools.serialization.private.keyAlias** - 创建密钥库时使用的别名
- **drools.serialization.private.keyPwd** - 密钥密码
- **drools.serialization.public.keyStoreURL** - 信任库所在的 URL
- **drools.serialization.public.keyStorePwd** - 信任库的密码

7. 加密密钥库凭证

密钥库密码目前是以明文进行存储的。

关于加密密钥库凭证的说明，请参考『<https://access.redhat.com/kb/docs/DOC-47247>』。

- 关于如何配置客户端来使用签名的 Rule 软件包，请参考《BRMS 用户指南》。

6.3.1. 设置序列化属性

如下所述，用于密钥库凭证的系统属性可以多种方式进行设置。这些属性只需要在一个地方进行设置，就可

为运行在相同应用服务器上的所有应用程序所用。

JBoss Properties 服务

要用 JBoss Properties 服务设置属性，在 `/server/PROFILE/deploy/properties-service.xml` 文件里添加下列受管 bean 的配置，并用你的系统里的这些属性替换示例里的值。

```
<mbean code="org.jboss.varia.property.SystemPropertiesService"
      name="jboss:type=Service,name=SystemProperties">
  <attribute name="Properties">
    # Drools Security Serialization specific properties
    drools.serialization.sign=true

    drools.serialization.private.keyStoreURL=file:///opt/secure/PrivateBRMS.key
    store
    drools.serialization.private.keyStorePwd=storepassgoeshere
    drools.serialization.private.keyAlias=BRMSKey
    drools.serialization.private.keyPwd=keypassgoeshere

    drools.serialization.public.keyStoreURL=file:///opt/public/PublicBRMS.keyst
    ore
    drools.serialization.public.keyStorePwd=keypassgoeshere
  </attribute>
</mbean>
```

jboss-brm.war 属性文件

要在 jboss-brm.war 属性文件里设置属性，请添加下列代码到 `jboss-brms.war/WEB-INF/classes/preferences.properties` 文件里。

```
drools.serialization.sign=true
drools.serialization.private.keyStoreURL=file:///opt/secure/PrivateBRMS.key
store
drools.serialization.private.keyStorePwd=storepassgoeshere
drools.serialization.private.keyAlias=BRMSKey
drools.serialization.private.keyPwd=keypassgoeshere
drools.serialization.public.keyStoreURL=file:///opt/public/PublicBRMS.keyst
ore
drools.serialization.public.keyStorePwd=keypassgoeshere
```

第 7 章 日志

JBoss 企业级 BRMS 平台 web 应用程序的日志功能由 **log4j** 提供。

WAR 里包括的配置，**WEB-INF/classes/log4j.xml**，发送所有的消息到标准输出（**STDOUT**）里。当以 **production** 服务器配置集部署时，日志消息附加在服务器日志文件 **PROFILE/log/server.log** 里。而以 **default** 服务器配置集部署时，这些消息将显示在服务器控制台上。

要修改日志行为，你可以编辑相关服务器配置集的 **log4j** 配置文件：**jboss-as-web/server/PROFILE/conf/jboss-log4j.xml**。

添加下列 XML 到 **default** 服务器配置集的 **log4j** 配置文件里以创建指引 **STDOUT** 消息到服务器日志文件里的新类别。

```
<category name="STDOUT" additivity="false">
  <priority value="INFO" />
  <appender-ref ref="FILE"/>
</category>
```



注意

关于日志的更多信息，请参考 JBoss 企业级应用程序平台的文档。

第 8 章 自定义

8.1. 自定义用户界面

BRMS 用户界面是由 GWT 框架动态提供的。通过编辑图片和 CSS 样式表单，你可以出于品牌或集成目的定制用户界面的外观。

.css 文件和一些图片可以在 `jboss-brms.war/org.drools.guvnor.Guvnor/` 里找到（剩下的图片位于 `images` 子目录里）。要访问它们，WAR 文件必须以展开归档的方式部署（如『[第 2 章 安装](#)』里所描述的）。

编辑或替换图片以及 CSS 文件且不要改动文件名缓存。如果遇到问题，请从 WAR 归档里恢复原来的版本。



注意

红帽推荐把改动的文件添加到版本控制系统以便于维护。

最常见的修改是替换品牌图片，这是位于屏幕顶部的 Logo 和网页图标（Site Favorite Icon）（分别是 `hdrlogo_brms.gif` 和 `drools.gif`）。^[1]

`Guvnor.css` 控制页面元素的总体风格。



警告

GWT 组件使用了几个额外的 CSS 文件，请勿修改这些文件。

通过编辑部署描述符 `jboss-brms.war/WEB-INF/web.xml` 可以定制 BRMS 平台使用的 URL。对于其他 Java web 应用程序，你可以使用相同的步骤。

8.2. 用于软件包构建的自定义选择器（Customize Selector）



重要

这只是一个技术预览功能。

选择器是一个可选功能。当用 **Packages** 功能构建知识软件包时，你可以指定 *选择器*。输入选择器的名称到 `selector` 文本框里。



注意

要配置选择器，首先以展开归档的方式部署 BRMS WAR。这对于执行任何自定义或配置任务来说是必要的。详情请参考『[第 2 章 安装](#)』。

找到 `selectors.properties` 文件。这个文件里有解释如何配置自定义选择器的细节。你也可以添加新的 `selectors.properties` 文件到系统的 class-path 里。你可以选择使用 **DRL** (*Drools Rule Language*) 文件，或者实现 `AssetSelector` 接口的类的名称。（`selectors.properties` 文件里已

经有了一个 **DRL** 文件示例。)

确保这个属性文件为每个选择器都赋予了唯一的名称。这是构建知识软件包时使用的名称。

[1] "favorites icon" 显示在不同的位置，这取决于浏览器和操作系统。通常它会显示在浏览器的地址条、书签或喜好菜单、窗口标题以及用于桌面 URL 快捷方式里。

第 9 章 监控

JBoss Operations Network (JBoss ON) 是 JBoss 中间件的系统管理套件，它可以用于监控 **JBoss** 商业规则管理系统。

JBoss Operations Network 可通过 <https://access.redhat.com> 访问。请参考《*JBoss Operations Network 安装指南*》里的安装说明。

9.1. 安装 JBoss ON 插件

BRMS JBoss ON 插件可以通过命令行或 **JBoss ON** web 用户界面进行安装。

过程 9.1. 通过命令行安装 JBoss ON 插件

1. 下载 JBoss ON 插件

从 [Customer Portal](#) 下载 **JBoss ON** 插件。

- 在 Customer Support Portal 里，点击 **Downloads**，然后点击 **JBoss Operations Network**，再从下拉框里选择 **JBoss ON for BRMS**。
- 点击 **Download** 开始下载。

2. 解压插件压缩文件

从下载的 ZIP 文件里解压 JAR 文件：

```
解压 jon-plugin-pack-brms-2.4.1.GA.zip
```

3. 安装插件

插件需要安装在 **JBoss ON** 服务器目录下。将前一步骤里解压的 JAR 文件移到顶层目录里。如带有 **JBoss ON** 的红帽企业版 Linux 里的 **/opt/** 目录。

```
mv rhq-brms-plugin-5.2.0.jar /opt/jon-server-2.4.0.GA1/rhq-brms-plugin-5.2.0.jar
```

4. 重启 JBoss ON 服务器。

过程 9.2. 通过 JBoss ON web 界面安装 JBoss ON 插件。

1. 下载 JBoss ON 插件

从 [Customer Portal](#) 下载 **JBoss ON** 插件。

- 在 Customer Support Portal 里，点击 **Downloads**，然后点击 **JBoss Operations Network**，再从下拉框里选择 **JBoss ON for BRMS**。
- 点击 **Download** 开始下载。

2. 解压插件压缩文件

从下载的 ZIP 文件里解压 JAR 文件：

```
解压 jon-plugin-pack-brms-2.4.1.GA.zip
```

3. 登录

访问 <http://localhost:7080/Login.do> 并登录。

4. 上载插件的 JAR 文件

从菜单里选择 **Administration**、**System Configuration**、**Plugins**。点击 **Add** 按钮并选择前一步骤里解压的 JAR 文件，然后点击 **upload**。

9.2. 发现 BRMS 实例

安装了插件后，JBoss Operations Network 需要发现被监控的 BRMS。

JBoss ON 将自动发现 BRMS。然而，如果没有的话，你可以手动添加资源。相关的细节请参考《JBoss Operations Network 基础管理指南》。

在发现了 BRMS 之后，它必须被导入到库存里。

过程 9.3. 导入 BRMS

1. 从 Discovery Queue 里选择 JMX 进程，然后点击 **Import**。
2. 从 **Resources** 菜单里选择服务器。
3. 选择 JMX 服务器来查看 BRMS 服务。

注意

对于 JBoss ON，Mbean 必须被启用才能监控 BRMS。
Mbean 可以通过传入参数启用：

```
-Ddrools.mbeans = enabled
```

或通过 API：

```
KnowledgeBaseConfiguration conf =  
    KnowledgeBaseFactory.newKnowledgeBaseConfiguration();  
conf.setOption( MBeansOption.ENABLED );
```

在 EWS 上部署 JBoss BRMS

BRMS 可以安装在带有 Tomcat 5 或 6 的 JBoss 企业级 Web 服务器 1.0.1 里。

关于 JBoss 企业级 Web 服务器的安装和配置说明，请参考《JBoss 企业级 Web 服务器安装指南》。

过程 A.1. 在 JBoss 企业级 Web 服务器上安装可部署软件包。

1. 下载 BRMS

从红帽客户支持入口 <https://access.redhat.com> 下载 BRMS 可部署软件包 ZIP 文件。进入 **Downloads - BRMS Platform**，然后选择版本。

2. 解压软件

从 ZIP 文件里解压 BRMS 可部署文件。

```
[localhost]$ unzip brms-deployable-5.2.0.zip
```

这会创建下列 ZIP 文件：

- **jboss-brms.engine.zip**
- **jboss-brms.manager.zip**

从 **jboss-brms-manager.zip** 里解压 **jboss-brms.war**。例如：

```
[localhost]$ unzip brms-manager.zip
```

这会创建一个 **jboss-brms.war** 目录，它包含展开归档形式的 JBoss 企业级 BRMS 平台 web 应用程序。

从 **jboss-brms-engine.zip** 里解压客户端库到一个目录里。例如：

```
[localhost]$ unzip brms-engine.zip -d client
```

这会创建一个包含客户端库 Jar 文件的目录。

3. 将 jboss-brms.war 复制到应用服务器

复制解开的 BRMS 目录 **jboss-brms.war** 到下列 EWS 目录，**tomcatversion/webapps/jboss-brms/WEB-INF/**，其中 **version** 是版本号 5 或 6。

4. 复制 slf4j 库到已部署的 Web 应用程序里。

从前面步骤创建的 **client** 目录复制 slf4j 库复制到已部署的 BRMS web 应用程序里：

```
$ cp client/lib/slf4j* jboss-brms.war/WEB-INF/lib/
```

5. 启用用户

创建一个用于 Tomcat 5 或 6 的自定义登录模块，其详情请参考 Tomcat 的文档。

6. 启动服务器

进入 **tomcatversion/bin/** 目录并使用 **run.sh** (Linux) 或 **run.bat** (Windows) 来启动服务器。

7. 登录

登录到 JBoss 企业级 BRMS 平台 web 应用程序来确认它已经成功安装和运行。详情请参考『第 2.3 节“第一次登录”』。

安装 Apache Ant

JBoss ModeShape 要求安装 Java 构建工具 *Apache Ant* 和 Trax 插件。而对于 JBoss 企业版 BRMS 平台的安装或普通操作，这并不是必需的。Apache Ant 还要求安装 Java Runtime Environment (JRE)。

如果运行开发人员工作站，Apache Ant 可能已经安装了。Apache Ant 文章上的 Apache Ant 软件包包括了 Trax 插件，但红帽企业版 Linux 将其列为单独的软件包。

过程 B.1. 在红帽企业版 Linux 上安装 Apache Ant

- 通过以下命令下载并安装红帽企业版 Linux 库里的 Apache Ant 以及 Trax 插件：

```
[localhost]$ sudo yum install ant-trax
```

过程 B.2. 在其他操作系统上安装 Apache Ant

1. 下载并解压

从 <http://ant.apache.org/bindownload.cgi> 下载 Apache Ant 二进制版本。

下载完毕后，解压到选定的安装位置，如 `c:\Program Files\Apache\Ant\` 或 `/opt/apache-ant-1.8/`。

2. 添加 ANT_HOME 环境变量

创建一个名为 `ANT_HOME` 的变量。这个变量必须包含前一步骤里创建的目录。

- 对于红帽企业版 Linux，在 `~/.bash_profile` 里添加下列内容，并替换前一步骤里创建的路径。

```
export ANT_HOME=/opt/apache-ant-1.8.1
```

- 在 Microsoft Windows 里，点击『启动』，打开『控制面板』并选择『系统』 -> 『高级』 -> 『环境变量』。

创建一个名为 `ANT_HOME` 的新变量并使其指向前一步骤里创建的目录。

3. 在 PATH 里包含 bin 目录

附加 Ant 的 `bin` 目录到环境变量 `PATH` 里。

- 在 Unix/Linux 系统里，你可以在 `~/.bash_profile` 文件里设置 `ANT_HOME` 变量的语句后简单地添加下列行：

```
export PATH=$PATH:$ANT_HOME/bin
```

- 在 Microsoft Windows 里，打开

要测试 Apache Ant 安装，在命令行运行 `ant -version`。其输出应该类似于：

```
[localhost]$ ant -version
Apache Ant version 1.8 compiled on June 27 2008
```

要学习 Apache Ant 的更多知识，请访问其项目主页：<http://ant.apache.org>。

持久化管理者配置示例

下面是用于所支持的数据库的持久化管理者配置示例。当然，你需要根据所使用的数据库更新配置里的版本号，如 JDBC URL 和 schemaObjectPrefix。



重要

下面是 JBoss 企业级 BRMS 平台支持的数据库列表

: <http://www.jboss.com/products/platforms/brms/supportedconfigurations/>

关于 Apache Jackrabbit 持久化管理者的其他细节，请参考

<http://wiki.apache.org/jackrabbit/PersistenceManagerFAQ>。

例 C.1. 使用 BundleDbPersistenceManager 的 MySQL 通用 JDBC 配置

```
<PersistenceManager class=
"org.apache.jackrabbit.core.persistence.bundle.BundleDbPersistenceManager">
  <param name="driver" value="com.mysql.jdbc.Driver"/>
  <param name="url" value="jdbc:mysql://localhost/brms"/>
  <param name="user" value="brms_user"/>
  <param name="password" value="brms_password"/>
  <param name="schema" value="mysql"/>
  <param name="schemaObjectPrefix" value="${wsp.name}_"/>
</PersistenceManager>
```

例 C.2. 使用 MySqlPersistenceManager 的 MySQL 配置

```
<PersistenceManager class=
"org.apache.jackrabbit.core.persistence.bundle.MySqlPersistenceManager">
  <param name="driver" value="com.mysql.jdbc.Driver"/>
  <param name="url" value="jdbc:mysql://localhost:3306/brms"/>
  <param name="user" value="brms_user"/>
  <param name="password" value="brms_password"/>
  <param name="schemaObjectPrefix" value="${wsp.name}_"/>
  <param name="schema" value="mysql"/>
</PersistenceManager>
```

例 C.3. 使用 OraclePersistenceManager 的 Oracle 配置

```
<PersistenceManager class=
"org.apache.jackrabbit.core.persistence.bundle.OraclePersistenceManager">
  <param name="driver" value="oracle.jdbc.OracleDriver"/>
  <param name="url" value="jdbc:oracle:thin:@localhost:1521:brms" />
  <param name="schema" value="oracle"/>
  <param name="user" value="brms_user" />
  <param name="password" value="brms_password" />
  <param name="schemaObjectPrefix" value="${wsp.name}_"/>
</PersistenceManager>
```

例 C.4. 使用 PostgreSQLPersistenceManager 的 PostgreSQL 配置

```
<PersistenceManager class="org.apache.jackrabbit.core.persistence.bundle.
PostgreSQLPersistenceManager">
  <param name="driver" value="org.postgresql.Driver"/>
  <param name="url" value="jdbc:postgresql://localhost:5432/brms" />
  <param name="schema" value="postgresql"/>
  <param name="user" value="brms_user" />
  <param name="password" value="brms_password" />
  <param name="schemaObjectPrefix" value="${wsp.name}_" />
</PersistenceManager>
```

例 C.5. 使用 MSSqlPersistenceManager 的 Microsoft SQL Server 2005 配置

```
<PersistenceManager class="org.apache.jackrabbit.core.persistence.bundle.
MSSqlPersistenceManager">
  <param name="driver"
    value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
  <param name="url"
    value="jdbc:sqlserver://localhost:3918;DatabaseName=brms" />
  <param name="user" value="brms_user" />
  <param name="password" value="brms_password" />
  <param name="schema" value="mssql"/>
  <param name="schemaObjectPrefix" value="${wsp.name}_" />
</PersistenceManager>
```

修订历史记录

修订 5.2.0-2.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
修订 5.2.0-2 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
修订 5.2.0-2 针对 5.2.0 进行更新 添加监控方面的内容 添加 EWS 安装说明	Tue Jul 1 2011	L Carlon
修订 5.1.0-1 添加 ModeShape 说明	Tue Mar 1 2011	Darrin Mison
修订 5.1.0-0 创建 5.1.0	Tue Aug 17 2010	David Le Sage, Darrin Mison