



Red Hat JBoss BRMS 6.3

Installation Guide

Red Hat JBoss BRMS 6.3 Installation Guide For Red Hat JBoss
Administrators

Red Hat JBoss BRMS 6.3 Installation Guide

Red Hat JBoss BRMS 6.3 Installation Guide For Red Hat JBoss Administrators

Red Content Services

Gemma Sheldon
gsheldon@redhat.com

Klara Kufova
kkufova@redhat.com

Marek Czernek
mczernek@redhat.com

Tomas Radej
tradej@redhat.com

Vidya Iyengar
viyengar@redhat.com

Legal Notice

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution-Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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, OpenShift, 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.

Abstract

This guide provides the steps necessary for administrators to install Red Hat JBoss BRMS, the plug-ins for Red Hat JBoss Developer Studio, and provides instructions for running example projects.

Table of Contents

CHAPTER 1. INTRODUCTION	4
1.1. ABOUT RED HAT JBOSS BRMS	4
1.2. SUPPORTED PLATFORMS	4
1.3. SUPPORTED COMPONENT VERSIONS	4
CHAPTER 2. INSTALLATION	6
2.1. INSTALLATION OPTIONS	6
2.2. DOWNLOADING RED HAT JBOSS BRMS FOR RED HAT JBOSS EAP	6
2.3. INSTALLING RED HAT JBOSS BRMS USING INSTALLER	6
2.4. INSTALLING RED HAT JBOSS BRMS USING THE INSTALLER IN CLI MODE	17
2.5. TROUBLESHOOTING RED HAT JBOSS BRMS INSTALLER	21
2.6. INSTALLING RED HAT JBOSS BRMS ON RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM	22
2.7. INSTALLING RED HAT JBOSS BRMS ON RED HAT JBOSS WEB SERVER	26
2.8. DEFINING ROLES	32
2.9. CREATING USERS	33
CHAPTER 3. PERSISTENCE SETUPS	35
3.1. CONFIGURING PERSISTENCE FOR BUSINESS CENTRAL	35
3.2. CONFIGURING PERSISTENCE FOR THE REALTIME DECISION SERVER	37
3.3. TROUBLESHOOTING	37
CHAPTER 4. GIT	39
4.1. CLONING EXISTING REPOSITORY	40
4.2. MIGRATING REPOSITORY FROM RED HAT JBOSS BRMS 5.3	41
CHAPTER 5. AUTHENTICATION	43
CHAPTER 6. TESTING INSTALLATION	44
6.1. STARTING SERVER	44
6.2. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT	46
6.3. LOGGING INTO BUSINESS CENTRAL	46
6.4. GIT REPOSITORY CLUSTERING MECHANISM	47
6.5. CLUSTERING ON RED HAT JBOSS EAP	49
6.6. GENERIC BUNDLE CLUSTERING	61
CHAPTER 7. MAVEN REPOSITORIES	64
7.1. ABOUT MAVEN	64
7.2. ABOUT PROVIDED MAVEN REPOSITORIES	64
7.3. CONFIGURING MAVEN TO USE FILE SYSTEM REPOSITORIES	65
7.4. CONFIGURING MAVEN TO USE ONLINE REPOSITORIES	68
7.5. DEPENDENCY MANAGEMENT	72
CHAPTER 8. RED HAT JBOSS DEVELOPER STUDIO	73
8.1. RED HAT JBOSS DEVELOPER STUDIO	73
8.2. INSTALLING THE RED HAT JBOSS DEVELOPER STUDIO PLUG-INS	73
8.3. SETTING THE DROOLS AND JBPM RUNTIME ENVIRONMENTS	74
8.4. CONFIGURING RED HAT JBOSS BRMS SERVER	74
8.5. IMPORTING PROJECTS FROM GIT REPOSITORY INTO RED HAT JBOSS DEVELOPER STUDIO	75
CHAPTER 9. BUSINESS RESOURCE PLANNER	79
9.1. INSTALLING BUSINESS RESOURCE PLANNER	79
9.2. RUNNING BUSINESS RESOURCE PLANNER EXAMPLES	79

CHAPTER 10. PATCHING AND UPGRADING RED HAT JBOSS BRMS 80

10.1. ABOUT PATCHES AND UPGRADES 80

10.2. APPLYING PATCHES IN RED HAT JBOSS BRMS 6.3 80

10.3. PATCHING OTHER PLATFORMS AND APPLICATIONS 82

10.4. UPGRADING TO LATEST MINOR RELEASE 83

APPENDIX A. REVISION HISTORY 86

CHAPTER 1. INTRODUCTION

1.1. ABOUT RED HAT JBOSS BRMS

Red Hat JBoss BRMS is an open source decision management platform that combines Business Rules Management and Complex Event Processing. It automates business decisions and makes that logic available to the entire business.

Red Hat JBoss BRMS uses a centralized repository where all resources are stored. This ensures consistency, transparency, and the ability to audit across the business. Business users can modify business logic without requiring assistance from IT personnel.

Business Resource Planner is included with this release.

Red Hat JBoss BRMS is supported for use with Red Hat Enterprise Linux 7 (RHEL7).

1.2. SUPPORTED PLATFORMS

Red Hat JBoss BPM Suite and Red Hat JBoss BRMS are supported on the following containers:

- Red Hat JBoss Enterprise Application Platform 6.4.(7+) *
- Apache Tomcat 6.0.(37+)
- Apache Tomcat 7.0.(59+)
- Apache Tomcat 8.0.(18+)
- Red Hat JBoss Fuse 6.2.0, 6.2.1 *
- Red Hat JBoss Web Server 2.1 (Tomcat 7) on JDK 1.7 *
- Red Hat JBoss Web Server 3.0 (Tomcat 8) *
- IBM WebSphere Application Server 8.5.(5+) *
- Oracle WebLogic Server 12.1.(3+) *



NOTE

Only Drools, Planner, and jBPM engine artifacts can be deployed on Red Hat JBoss Fuse.

Containers marked with an asterisk (*) are fully supported and tested. For more information, see a list of [Red Hat JBoss BPM Suite 6 Supported Configurations](#) at Red Hat Knowledgebase.

1.3. SUPPORTED COMPONENT VERSIONS

Red Hat JBoss BPM Suite and Red Hat JBoss BRMS 6.3 support the following component versions:

Table 1.1. Supported Maven Artifact Versions

Red Hat JBoss BPM Suite and Red Hat JBoss BRMS Version	Maven Artifact Version
6.3.0	6.4.0.Final-redhat-3
6.3.1	6.4.0.Final-redhat-6
6.3.2	6.4.0.Final-redhat-8
6.3.3	6.4.0.Final-redhat-10

Table 1.2. Supported Bill of Material Versions

Red Hat JBoss BPM Suite and Red Hat JBoss BRMS Version	BOM Version
6.3.0	6.3.0.GA-redhat-3
6.3.1	6.3.1.GA-redhat-2
6.3.2	6.3.2.GA-redhat-2
6.3.3	6.3.3.GA-redhat-2

Use one of the following Bill of Materials (BOM):

- **org.jboss.bom.brms:jboss-brms-bpmsuite-platform-bom:\$VERSION**
- **org.jboss.bom.brms:jboss-brms-bpmsuite-bom:\$VERSION**

For further information about BOM, see the [Dependency Management](#) chapter of *Red Hat JBoss BPM Suite Development Guide*.

CHAPTER 2. INSTALLATION

2.1. INSTALLATION OPTIONS

Red Hat JBoss BRMS comes in two versions:

- Executable JAR installer for installation on Red Hat JBoss Enterprise Application Platform (EAP) 6.4.
- ZIP file install which itself comes in two versions:
 - **jboss-brms-6.3-deployable-eap6.x.zip**: version adapted for deployment on Red Hat JBoss Enterprise Application Platform (EAP 6).
 - **jboss-brms-6.3-deployable-generic.zip**: the deployable version with additional libraries adapted for deployment on Red Hat JBoss Web Server (EWS) and other supported containers.

Depending on your environment, you may choose the installation option best suited for your project needs.



NOTE

Red Hat JBoss BRMS is designed to work with UTF-8 encoding. If a different encoding system is used by the underlying JVM, unexpected errors might occur. To ensure UTF-8 is used by the JVM, use the following system property: `"-Dfile.encoding=UTF-8"`.



IMPORTANT

From Red Hat JBoss BRMS 6.1 onwards, you must have JBoss EAP 6.4 or better already installed before attempting to install Red Hat JBoss BRMS.

2.2. DOWNLOADING RED HAT JBOSS BRMS FOR RED HAT JBOSS EAP

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **DOWNLOADS** at the top of the page.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BRMS**.
4. From the **Version** drop-down menu, select version **6.3**.
5. Select **Red Hat JBoss BRMS 6.3.0 Deployable for EAP 6** and then click **Download**.

2.3. INSTALLING RED HAT JBOSS BRMS USING INSTALLER

The installer for Red Hat JBoss BRMS is an executable Java JAR file. You can use it to install Red Hat JBoss BRMS on an existing Red Hat JBoss EAP 6.4 installation.



WARNING

Note that the provided Red Hat JBoss BRMS JAR file installer does *not* support the Red Hat JBoss EAP distribution installed by yum or RPM Package Manager. In this case, download the **Red Hat JBoss BRMS 6.3.0 Deployable for EAP 6** file and follow the steps described in [Section 2.6, “Installing Red Hat JBoss BRMS on Red Hat JBoss Enterprise Application Platform”](#).



NOTE

For security reasons, you should run the installer as a non-root user.

Prerequisite

Before attempting to install Red Hat JBoss BRMS, ensure you have already installed Red Hat JBoss EAP 6.4.7 or better, and create a back up. Ensure that your user has sufficient rights to complete the installation.

1. Set up location and users.

Navigate to the folder where you downloaded the installer file in a command prompt and execute the following command:

```
java -jar jboss-brms-6.3.0.GA-installer.jar
```



NOTE

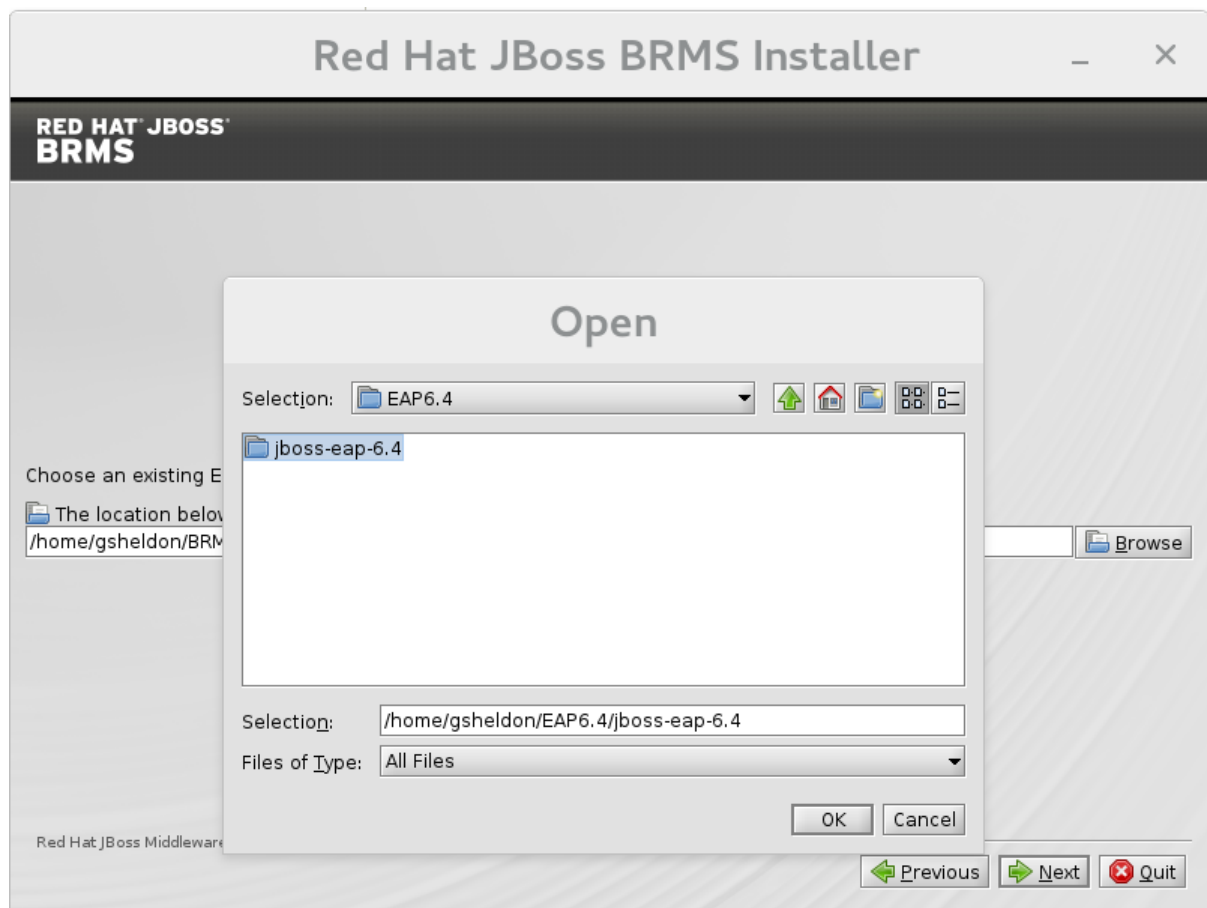
When running the installer on Windows, you may be prompted to provide administrator credentials during the installation. To prevent this, add the **izpack.mode=privileged** option to the installation command:

```
java -Dizpack.mode=privileged -jar jboss-brms-6.3.0.GA-installer.jar
```

Furthermore, when running the installer with a 32-bit Java Virtual Machine, you can encounter memory limitations. To solve the issue, run

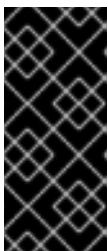
```
java -XX:MaxHeapSize=4g -jar jboss-brms-6.3.0.GA-installer.jar
```

2. The graphical installer will execute and display a splash screen and a license agreement page. Read and accept the license to proceed.
3. In the next screen, provide the parent location of an existing Red Hat JBoss EAP where Red Hat JBoss BRMS must be installed. The screenshot below depicts an example directory path:

Figure 2.1. Red Hat JBoss BRMS Installer for EAP Directory Path

4. In the next two screens, create two users: the first one for the management console of the Red Hat JBoss EAP (ManagementRealm) and the second one for managing Red Hat JBoss BRMS itself (ApplicationRealm).
Creation of the first user for the management console of Red Hat JBoss EAP is optional and you may skip it if it is not required.

Make a note of these user names and passwords as you will need them to access the Red Hat JBoss EAP server (if you do decide to create it) and the Red Hat JBoss BRMS application respectively.



IMPORTANT

Make sure that the selected user name does *not* conflict with any known title of a role or a group.

For example, if there is a role called **admin**, you should*not* create a user with the user name **admin**.



NOTE

The passwords that you create must have at least 8 characters and must contain at least one number and one non-alphanumeric character (not including the character &).

**NOTE**

The application role assigned to the second user that you create is the **admin** role. This is the only role that can be assigned to this newly created user. You can create more users with narrow roles afterwards by using the command line.

5. Set up security environment.

Next, you will set up the security environment of your new Red Hat JBoss BRMS install. Decide to enable or disable the Java Security Manager in this step by clicking on the check box. The Java Security Manager makes your system more secure but may downgrade performance. You need to make a decision based on your environment.

6. Choose whether you want to set up pure IPv6 configuration on the server that the installation is taking place. This will allow you to set up runtime IPv6 specific configurations later.

7. Configure runtime environment.

This step provides the option of using a default configuration or specifying an advanced configuration.

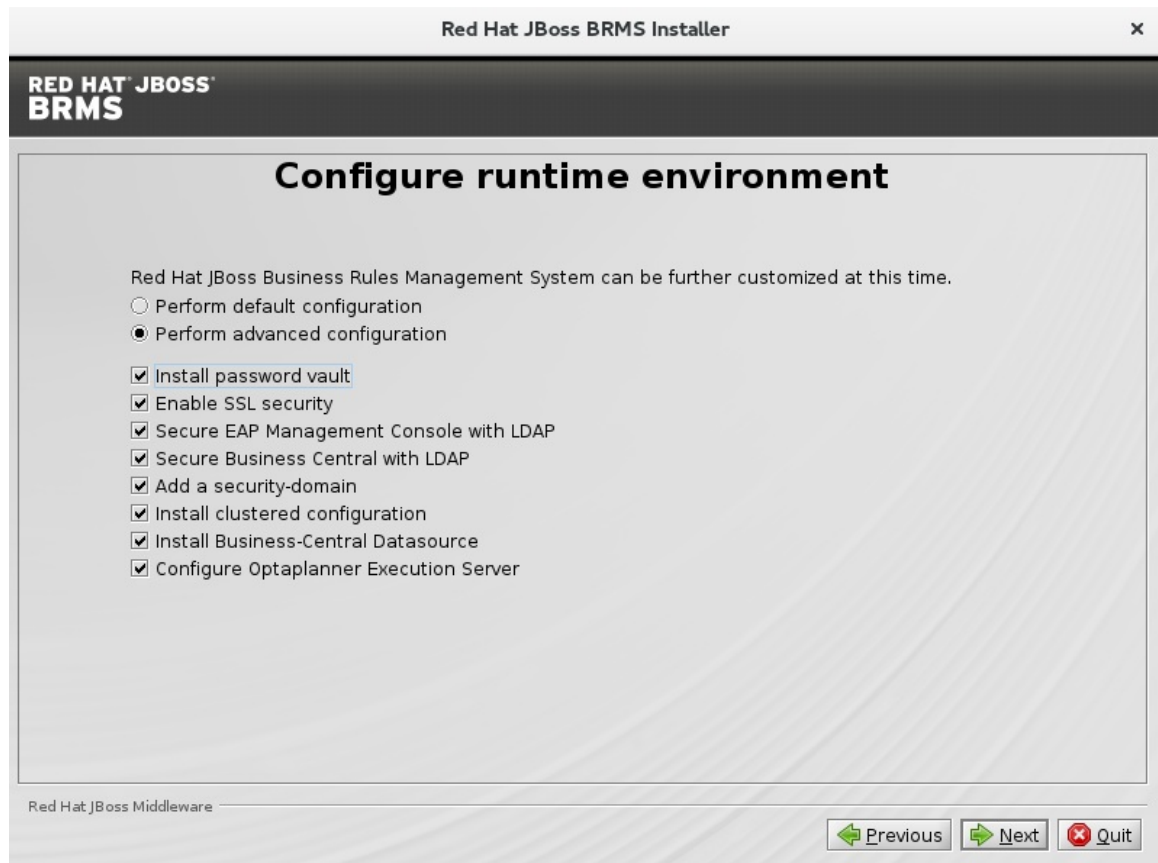
a. *Default Configuration*

Choose **Perform default configuration** for the runtime environment in the next step and click **Next** to review the installation details. If you are happy with the details, click **Next** to start the actual installation or click **Previous** to go back and make changes.

b. *Advanced Configuration*

Choose to enable advanced configuration options. Select **Perform advanced configuration** and choose the advanced configuration options you want to enable for your environment using the check boxes.

Figure 2.2. Advanced Configuration Options



i. *Configure Password Vault*

Vault passwords are used to obfuscate passwords in the various server descriptors using a Java secret key generated during the installation process, or manually using the `keytool`. This prevents passwords from being stored as plain text in the descriptors. The **Iteration count** and **Salt** are both parameters to the encryption process.

For more information about vault passwords, see the *Red Hat JBoss EAP Security Guide*.

Figure 2.3. Configure Password Vault

Red Hat JBoss BRMS Installer

RED HAT JBOSS BRMS

Configure password vault

A password vault encrypts sensitive strings and stores them in an encrypted keystore. The vault mechanism manages decrypting the strings for use with security domains, security realms, or other verification systems. Note that the keystore must be of type "JCEKS".

Please make note of your entry below in order to mask any subsequent passwords. See JBoss EAP 6 documentation for further details.

The password must have no fewer than 6 characters.

Vault alias:

Salt (8-chars)

Iteration count

Vault keystore password:

Re-enter vault keystore password:

New Keystore location:

Encrypted vault directory:

Red Hat JBoss Middleware

ii. *SSL Security*

This screen allows you to add the `<ssl>` and `<truststore>` elements to the ManagementRealm security realm using the provided keystore.

- The `<ssl>` element causes the server to present the certificate within the keystore as its identity, which allows the user to apply their official certificate.
- The `<truststore>` element enables Client-Cert authentication. This means that, if a remote client attempts to connect to any resource managed by the ManagementRealm, the client can present a certificate, and if an entry in the truststore matches, will be authenticated without needing to provide a user name/password.

The end result is an encrypted connection that is secure between the client and the server for the ManagementRealm.

Figure 2.4. SSL Security Configuration

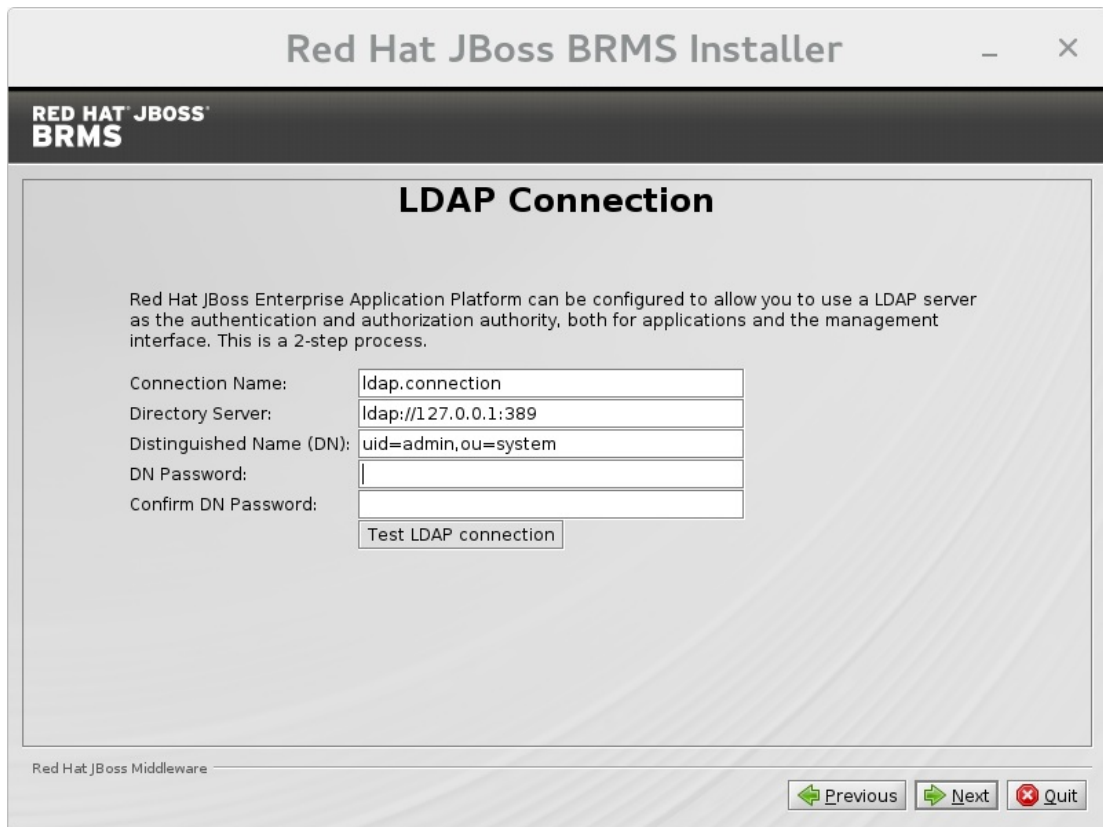
The screenshot shows the 'Red Hat JBoss BRMS Installer' window with the 'SSL Security' tab selected. The window title bar includes standard minimize, maximize, and close buttons. The main content area has a header 'SSL Security' and a descriptive text: 'SSL security can be enabled for the management interfaces by specifying the location of a keystore and the password for it.' Below this, there are three input fields: 'SSL keystore password:', 'Confirm SSL keystore password:', and 'Keystore location:'. The 'Keystore location' field contains the path '/home/gsheldon/EAP6.4/jboss-eap-6.4' and has a 'Browse' button next to it. At the bottom of the window, there is a status bar with the text 'Red Hat JBoss Middleware' and three navigation buttons: 'Previous', 'Next', and 'Quit'.

iii. *LDAP Connection*

This step in the installer allows the user to define an LDAP server, which in turn defines users which should be allowed to authenticate with the ManagementRealm. This replaces the default configuration.

The **LDAP Connection** screen allows users to define how to connect to the LDAP server.

- **Distinguished Name (DN):** the user that can connect to the LDAP server. Typically the DN will uniquely define a special user for this purpose.

Figure 2.5. LDAP Connection Configuration


Red Hat JBoss BRMS

LDAP Connection

Red Hat JBoss Enterprise Application Platform can be configured to allow you to use a LDAP server as the authentication and authorization authority, both for applications and the management interface. This is a 2-step process.

Connection Name:

Directory Server:

Distinguished Name (DN):

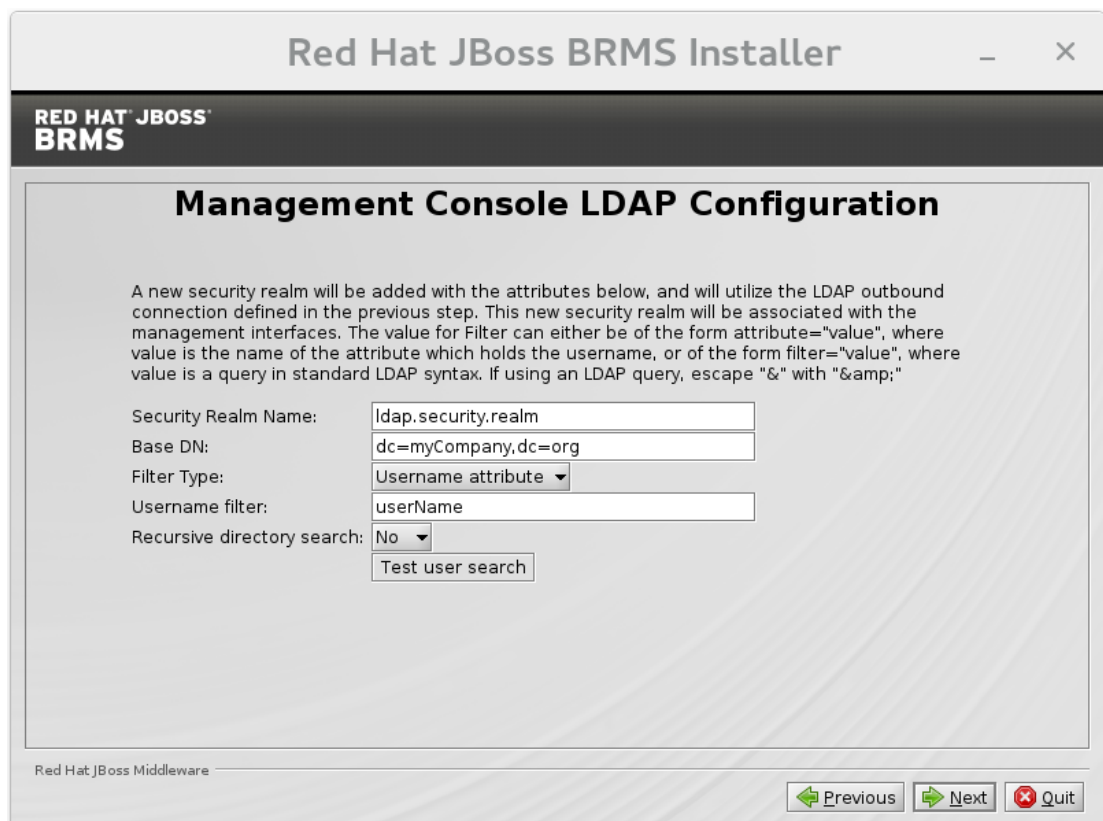
DN Password:

Confirm DN Password:

Red Hat JBoss Middleware

LDAP Security (Management Console)

The **Management Console LDAP Configuration** screen allows you to set up a security realm. This defines the `<security-realm>` element to be added to the descriptors, and utilizes the connection defined previously.

Figure 2.6. Management Console LDAP Configuration


Red Hat JBoss BRMS

Management Console LDAP Configuration

A new security realm will be added with the attributes below, and will utilize the LDAP outbound connection defined in the previous step. This new security realm will be associated with the management interfaces. The value for Filter can either be of the form `attribute="value"`, where value is the name of the attribute which holds the username, or of the form `filter="value"`, where value is a query in standard LDAP syntax. If using an LDAP query, escape "&" with "&".

Security Realm Name:

Base DN:

Filter Type:

Username filter:

Recursive directory search:

Red Hat JBoss Middleware

- **Base DN:** Will typically define a 'base search' or 'root context' to begin searching for users.
- **Filter Type:** Tells Red Hat JBoss EAP how to find the LDAP attribute that defines a user; it can be a simple attribute, but can also be a complex LDAP filter.
- **Username filter:** The LDAP attribute which holds the user name values. A user name entered in this field is used for search queries as a value of the **uid** attribute. If a user chooses LDAP Syntax Query as a filter type, this query must be specified in this field.
- **Recursive directory search:** If enabled, Red Hat JBoss EAP will traverse the LDAP tree recursively, starting at Base DN. Otherwise, the search will be limited to Base DN.

LDAP Security (Business Central)

Most of the following fields are similar to the Base DN. Contexts are used to search for roles, which allows it to perform authorization in addition to authentication. Otherwise, the context fields are analogous to the Base DN from the previous, and attribute fields are analogous to user name attribute. The filters allow fine grained control over which values of the given attribute will be accepted.

Input values from **Business Central LDAP Configuration** page are used to configure a new security domain, which make use of **LdapExtended** login module. This security domain is set as default for Business Central web application. For more information about security domains and login modules, see the *Red Hat JBoss EAP Security Guide*

Figure 2.7. Business Central LDAP Configuration

The screenshot shows a window titled "Red Hat JBoss BRMS Installer". Inside, there's a section titled "Business Central LDAP Configuration". Below the title, there's a paragraph of instructions: "Enter the details below to configure LDAP authentication on Business Central. The LDAP server URL, bind DN and bind password previously defined will be used. The contexts below define the users and roles that should be allowed access to Business Central. A security domain will be defined using these details." Below this, there are several input fields: "User context:", "User filter:", "User roles context:", "User roles filter:", "Role context:", "Role filter:", "Role attribute id:", and "Role name id:". Each field has a corresponding text box. At the bottom left of the form area, there is a checkbox labeled "Role attribute is distinguished name". At the bottom right of the window, there are three buttons: "Previous", "Next", and "Quit".

The **Security Domain** screen allows you to configure all of the elements of the `<security-domain>` security subsystem for managing security information, including JSSE configuration. For more detailed information about configuring security domains, see the *Red Hat JBoss EAP Security Guide*.

Figure 2.8. Security Domain

Red Hat JBoss BRMS Installer

RED HAT JBOSS BRMS

Security Domain

Configure a security-domain using the fields below. The only required field is the security-domain name. The rest of the fields are optional. Most fields have a predefined list of valid values.

Security-Domain Name:

Security-Domain Cache-type:

Add authentication element:

Authentication code: Authentication flag: Authentication options:

Add authorization element:

Authorization code: Authorization flag: Authorization options:

Add mapping element:

Mapping code: Mapping type: Mapping options:

Red Hat JBoss Middleware

Figure 2.9. JSSE Configuration

Red Hat JBoss BRMS Installer

RED HAT JBOSS BRMS

JSSE Configuration

Configure a JSSE element. The JSSE element requires either a keystore or a truststore to be configured.

☐ Add jsse element

Cipher suites used by SSLContext:

Protocols used by SSLContext:

Alias of client-side keystore:

Alias of server-side keystore:

Third party validation token:

☐ Add keystore element

JSSE keystore password:

Confirm JSSE keystore password:

Keystore Provider:

Provider argument:

Keystore type:

Keystore URL:

☐ Add keystore manager element

KeyManagerFactory algorithm:

KeyManagerFactory Provider:

Red Hat JBoss Middleware

v. *Configure Clustering*

Selecting this option installs Red Hat JBoss BRMS ready for clustered operation. For more information, see [Section 6.5, “Clustering on Red Hat JBoss EAP”](#).

vi. *Business Central Datasource Setup*

After cluster configuration, the next screen allows you to configure the Business Central data source.

Figure 2.10. Business Central Datasource Setup

The screenshot shows the 'Business Central Datasource Setup' window within the 'Red Hat JBoss BRMS Installer'. The window has a title bar with the text 'Red Hat JBoss BRMS Installer' and a close button. Below the title bar is a dark header with the 'RED HAT JBOSS BRMS' logo. The main content area is titled 'Business Central Datasource Setup' and contains the instruction: 'Configure the datasource for Red Hat JBoss Enterprise Application Platform below.' Below this instruction are several input fields and dropdown menus:

Name:	myNewBusinessCentralDatasource
JNDI Name:	java:/DB2BusinessCentralDS
Min Pool Size:	0
Max Pool Size:	20
Security Type:	Security-domain
Security-domain:	mySecurityDomain
Datasource type:	ibmdb2
Connection URL:	jdbc:db2://SERVER_NAME:PORT/DATABASE_NAME

At the bottom right of the form is a button labeled 'Test Datasource Connection'. At the bottom of the window, there is a status bar with the text 'Red Hat JBoss Middleware' and three navigation buttons: 'Previous' (with a left arrow), 'Next' (with a right arrow), and 'Quit' (with a red X icon).

vii. *Configure Business Resource Planner*

This screen allows you to configure Business Resource Planner.

Figure 2.11. Configure Business Resource Planner

8. The installer will go through the steps to install Red Hat JBoss BRMS and will perform post installation configuration steps when you click **Next**. The installer will also start the Red Hat JBoss BRMS server and connect to it to validate the installation. Click **Next** to get to the last screen where you can generate the installation script and properties file. Click **Done** to quit the installer.

You have successfully installed Red Hat JBoss BRMS using the installer.

2.4. INSTALLING RED HAT JBOSS BRMS USING THE INSTALLER IN CLI MODE

The installer for Red Hat JBoss BRMS can also be executed through the command-line interface (CLI). The procedure below demonstrates the steps that you are likely to encounter using this option to install Red Hat JBoss BRMS.

Prerequisite

Before attempting to install Red Hat JBoss BRMS, ensure you have already installed Red Hat JBoss EAP 6.4.7 or better, and create a back up. Ensure that your user has sufficient rights to complete the installation.

1. Navigate to the folder where you downloaded the installer file in a command prompt and execute the following command.

```
java -jar jboss-brms-6.3.0.GA-installer.jar -console
```

2. The command-line interactive process will start and display the End-User license agreement. You will be prompted to select an option at the end of this license:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

-
3. Enter **1** to begin the installation and type in the parent directory of an existing Red Hat JBoss EAP installation.

The location below must specify the JBOSS_HOME of an existing EAP installation.

```
[/home/user/BRMS-6.3.0/jboss-eap-6.4]
```

The installer will verify the location of the Red Hat JBoss EAP installation at the provided location.

press 1 to continue, 2 to quit, 3 to redisplay.

Enter **1** to confirm and continue.

4. Create an administrative user.

Create an administrative user

This user will be added to the host container's management realm for administrative purposes. It can be used to access the management console, the management CLI or other applications secured in this realm.

The password must be at least eight characters long, with one alphabetic character, one digit, and one non-alphanumeric character not including &.

Create an administrative user.

0 [x] Skip new administrative user creation.

1 [] Create a new administrative user.

5. Create and confirm a password for the user of the EAP management console:

Admin password: []

Confirm admin password: [*****]

6. After the passwords have been entered, choose an option from the prompt below:

press 1 to continue, 2 to quit, 3 to redisplay.

7. Enter **1** and create a Red Hat JBoss BRMS user:

Create a Business Rules Management System Admin User

Create a BRMS admin user. The user will be added to the ApplicationRealm, and can be used to access the Business Central Console. The User will be assigned the 'admin' application role. The BRMS username cannot be any of the following: 'admin', 'analyst', 'user', 'manager' or 'developer'.

The password must be at least eight characters long, with one alphabetic character, one digit, and one non-alphanumeric character

not including &.

BRMS username: [brmsAdmin]



IMPORTANT

Make sure that the selected user name does *not* conflict with any known title of a role or a group.

For example, if there is a role called **admin**, you should *not* create a user with the user name **admin**.

8. Create and confirm a password for the Red Hat JBoss BRMS user:

```
BRMS password: []
*****
Confirm BRMS password: [*****]
*****
```

9. After the passwords have been entered, choose an option from the prompt below:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

10. Configure the Java Security Manager.

A Java security manager offers JVM level security beyond what is provided by the application container. It enforces access rules at the JVM runtime based on one or more security policies.

This installer will place two security policies in the installation directory with the filenames 'security.policy' and 'kie.policy' regardless of choice. Those policies will be enabled at runtime if the option below is selected.

Please note that a security manager imposes a significant performance overhead when enabled. It is suggested the included policies be applied in production if user requirements call for a stronger measure than what is already provided by the application container's authentication and authorization mechanism.

Please see the JBoss Business Rules Management System administrative documentation for further details and consideration.

```
[ ] Enable the Java security manager
Input 1 to select, 0 to deselect:
```

11. After the Java Security Manager choice, choose an option from the prompt below:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

12. Next, select whether to enable the IPv6 configuration.

```
IPv6 configuration
```

```
If this computer is using a pure IPv6 configuration, please check
the box below. A pure IPv6 setup requires additional configuration
at runtime to ensure the proper bindings of the management and http
interfaces.
```

```
[ ] Enable pure IPv6 configuration
Input 1 to select, 0 to deselect:
```

13. After the IPv6 configuration choice, choose an option from the prompt below:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

14. Configure the runtime environment by either choosing the default configuration or inputting advanced options.

```
Configure runtime environment
Red Hat JBoss Business Rules Management System can be further
customized at this time.
0 [x] Perform default configuration
1 [ ] Perform advanced configuration
Input Selection:
```

If you select **1, Perform advanced configuration**, complete the following configurations:

- a.

```
[ ] Install password vault
Input 1 to select, 0 to deselect:
```
- b.

```
[ ] Enable SSL security
Input 1 to select, 0 to deselect:
```
- c.

```
[ ] Secure EAP Management Console with LDAP
Input 1 to select, 0 to deselect:
```
- d.

```
[ ] Secure Business Central and Dashbuilder with LDAP
Input 1 to select, 0 to deselect:
```
- e.

```
[ ] Add a security-domain
Input 1 to select, 0 to deselect:
```
- f.

```
[ ] Install Business-Central Datasource
Input 1 to select, 0 to deselect:
```
- g.

```
[ ] Configure Optaplanner Execution Server
Input 1 to select, 0 to deselect:
```

15. Next, choose an option from the prompt below:

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

16. The **.jar** file will begin to unpack and configure.

17. After a successful installation, the command-line will ask you if you would like to generate an automatic installation script and properties file.

```
Installation has completed successfully.
Application installed on /home/user/BRMS-6.3.0/jboss-eap-6.4
Would you like to generate an automatic installation script and
properties file?
(y/n) [n]:
```

18. If you select **y**, provide a path for the automatic installation script:

```
Select path for the automatic installation script: [/home/user/BRMS-
6.3.0/jboss-eap-6.4/AUTO_SCRIPT_FILENAME]
```

This generated script will allow the user to run the installer in the following way for future installations:

```
java -jar jboss-brms-6.3.0.GA-installer.jar AUTO_SCRIPT_FILENAME
```

NOTE

Running the installer in this way will result in an installation identical to the installation from which the auto script was generated. Note that sensitive values, such as passwords, will need to be provided from an external file or provided at auto installation time. The optional argument below allows the user to provide these values automatically:

```
-variablefile VARIABLE_FILENAME
```

Sensitive values can also be provided using the following argument:

```
-variables key1=value1,key2=value2
```

19. The command-line will provide the following message upon a successful auto script creation and/or console installation:

```
XML written successfully.
[ Console installation done ]
[BRMS_Installer]$
```

20. Start Red Hat JBoss EAP by running **standalone.sh** in the **jboss-eap-6.4/bin** directory.

```
./standalone.sh
```

21. Navigate to <http://localhost:8080/business-central> in a web browser.
22. Log in with the correct user name/password as given to the Red Hat JBoss BRMS user in the *Create and confirm a password for the Red Hat JBoss BRMS user* step.

2.5. TROUBLESHOOTING RED HAT JBOSS BRMS INSTALLER

The Red Hat JBoss BRMS installation failed. How do I reinstall Red Hat JBoss BRMS?

If the installer detects Red Hat JBoss BRMS applications, the installation will not continue. In case of a failed installation:

1. Change into ***EAP_HOME/standalone/deployments***.
2. Delete all Red Hat JBoss BRMS deployments, that is:
 - ***business-central.war***
 - ***dashbuilder.war***
 - ***kie-server.war***
3. Start the installer again.

2.6. INSTALLING RED HAT JBOSS BRMS ON RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

To install Red Hat JBoss BRMS 6.3 deployable on Red Hat JBoss EAP:

1. Download the **Red Hat JBoss Enterprise Application Platform 6.4.0** ZIP file from the [Customer Portal](#).
2. Extract the ZIP file. This location is your ***EAP_HOME***.
3. Patch the Red Hat JBoss EAP to the supported version for your Red Hat JBoss BRMS version.
 - See [Red Hat JBoss BPM Suite 6 Supported Configurations](#) to verify which patch is applicable for your Red Hat JBoss BRMS version.
 - See [Patching a Zip/Installer Installation](#) from the *Red Hat JBoss EAP Installation Guide* for further information about applying patches.
4. Download the **Red Hat JBoss BRMS 6.3.0 Deployable for EAP 6** ZIP file.
5. Extract the file and copy ***jboss-eap-6.4/bin/**** into ***EAP_HOME/bin/****. When asked, merge the directories.
 - a. If you want to run Red Hat JBoss BRMS in the standalone mode:
 - Copy ***jboss-eap-6.4/standalone/configuration/**** into ***EAP_HOME/standalone/configuration/***.
 - Copy ***jboss-eap-6.4/standalone/deployments/**** into ***EAP_HOME/standalone/deployments/***.



NOTE

If you already have deployments on your Red Hat JBoss EAP, ensure that your current deployments do not have colliding names with Red Hat JBoss BRMS deployments.

b. If you want to run Red Hat JBoss BRMS in the domain mode:

- Copy **jboss-eap-6.4/domain/configuration/*** into **EAP_HOME/domain/configuration/**.



WARNING

Make sure this step is performed by the same user account that was used to install Red Hat JBoss EAP. This account must not be a superuser account.

6. Add an application user:

```
./EAP_HOME/bin/add-user.sh -a --user bpmAdmin --password password@1
--role kie-server,admin,rest-all,analyst
```

To start Red Hat JBoss BRMS in standalone mode:

1. Change into **EAP_HOME/bin**.
2. Execute:
In a Unix environment:

```
./standalone.sh
```

In a Windows environment:

```
standalone.bat
```

You can now log into Business Central in your web browser: **localhost:8080/business-central**.

Configuring Domain Mode

If you installed Red Hat JBoss BRMS as described in [Section 2.2, “Downloading Red Hat JBoss BRMS for Red Hat JBoss EAP”](#), deploy Red Hat JBoss BRMS web applications manually.

The **business-central.war**, **dashbuilder.war**, and **kie-server.war** applications are supplied in the **Red Hat JBoss BRMS 6.3.0 Deployable for EAP 6** ZIP file as directories. To deploy the applications into domain mode:

1. Package the application directories into archives:
 - a. Extract the following files from the **Red Hat JBoss BRMS 6.3.0 Deployable for EAP 6** ZIP file:
 - **jboss-eap-6.4/standalone/deployments/business-central.war**
 - **jboss-eap-6.4/standalone/deployments/kie-server.war**
 - **jboss-eap-6.4/standalone/deployments/dashbuilder.war**

- b. Create a ZIP file with the content of the **business-central.war**, **kie-server.war**, and **kie-server.war** directories, for example:

- i. Change into the directory:

```
cd business-central.war
```

- ii. Execute **zip -r business-central.war .** to create a ZIP file of the content of the **business-central.war** directory.
 - iii. Repeat this procedure for all the web applications you want to deploy. This ensures that **business-central.war**, **kie-server.war**, and **dashbuilder.war** are archives, not directories.

2. Deploy the archives:

- a. Add a management user:

```
./EAP_HOME/bin/add-user.sh -b --user mgmtAdmin --password  
password@1 --role admin
```

- b. Execute **./EAP_HOME/bin/domain.sh**.
- c. Log into **http://localhost:9990/** using your management user.
- d. Click **Deployments** → **Content Repository** → **Add**.
- e. Select and upload the web archive from the file system.
- f. Select the deployment and click **Assign**.



NOTE

If you want to deploy multiple Red Hat JBoss BRMS nodes on a single machine, set ports and other properties before assigning the deployment to a server. See [the section called “Red Hat JBoss BRMS Settings for Red Hat JBoss EAP”](#) for more information.

- g. Select the server group.

You can now log into Business Central at **localhost:8080/business-central**.



NOTE

To log into Business Central deployed on Host Controller (HC) machines, the user created on the Domain Controller machine has to be created on the Host Controller machines as well, by following the steps in [Section 2.9, “Creating Users”](#).

Red Hat JBoss BRMS Settings for Red Hat JBoss EAP

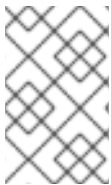
If you want to run multiple instances of Red Hat JBoss EAP with Red Hat JBoss BRMS, the best practice is to meaningfully set the following properties:

- **org.uberfire.nio.git.dir**

- `org.uberfire.metadata.index.dir`
- `org.uberfire.nio.git.ssh.cert.dir`

When multiple Red Hat JBoss BRMS nodes are used on a single machine, the below properties need to be specified:

- `org.uberfire.nio.git.daemon.host`: can be `localhost`.
- `org.uberfire.nio.git.daemon.port`
- `org.uberfire.nio.git.ssh.host`: can be `localhost`.
- `org.uberfire.nio.git.ssh.port`



NOTE

Both the `org.uberfire.nio.git.daemon.port` and the `org.uberfire.nio.git.ssh.port` require different port values in order to avoid port conflicts.

Set the properties in the `EAP_HOME/domain/configuration/host.xml` file:

Node A:

```
<system-properties>
  <property name="org.uberfire.nio.git.dir" value="/valid/path/.." boot-
time="false"/>
  <property name="org.uberfire.metadata.index.dir"
    value="/valid/path/.." boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir"
    value="/valid/path/.." boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.host"
    value="10.10.10.10" boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.port" value="9417" boot-
time="false"/>
  <property name="org.uberfire.nio.git.ssh.host" value="10.10.10.10" boot-
time="false"/>
  <property name="org.uberfire.nio.git.ssh.port" value="8002" boot-
time="false"/>
</system-properties>
```

Node B:

```
<system-properties>
  <property name="org.uberfire.nio.git.dir" value="/valid/path/.." boot-
time="false"/>
  <property name="org.uberfire.metadata.index.dir"
    value="/valid/path/.." boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir"
    value="/valid/path/.." boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.host"
    value="10.10.10.10" boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.port" value="9418" boot-
time="false"/>
```

```
<property name="org.uberfire.nio.git.ssh.host" value="10.10.10.10" boot-  
time="false"/>  
<property name="org.uberfire.nio.git.ssh.port" value="8003" boot-  
time="false"/>  
</system-properties>
```

The system properties depicted above should indicate the host, port, or location of the `.index` or `.nio.git` files. These files, which should be used by a respective node, would then be grouped in a particular domain.

2.7. INSTALLING RED HAT JBOSS BRMS ON RED HAT JBOSS WEB SERVER

The generic deployable package is provided for customers to install Red Hat JBoss BRMS 6.3 to an existing application server. The following procedure provides instructions for installation on an existing Red Hat JBoss Web Server 2.1.0 instance.



IMPORTANT

In a fresh Red Hat JBoss BRMS installation, you can encounter exceptions in the log similar to the following:

WARNING: Unable to instantiate EJB Asynchronous Bean. Falling back to Executors' CachedThreadPool.

The exceptions are caused by the code that failed to look up an Enterprise Java Beans object registered under a JNDI name that does not exist in Apache Tomcat. Therefore, a default implementation is used instead.

These messages are only warnings and do not have any impact on the overall functionality of the system.

Procedure: Installing Generic Deployable Package

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **DOWNLOADS** at the top of the page.
3. In the **Product Downloads** page that opens, choose **Red Hat JBoss BRMS**.
4. From the **Version** drop-down menu, select version **6.3**.
5. Select **Red Hat JBoss BRMS 6.3.0 Deployable for All Supported Containers** package and click **Download**.
Also select and download the **Red Hat JBoss BRMS Core Engine** files (`jboss-brms-VERSION-engine.zip`).
6. Extract `business-central.war` and `kie-server.war` from the generic deployable archive and copy to `tomcat7/webapps/` folder.
7. Remove the `.war` extensions from the `business-central.war` and `kie-server.war` folders.
8. Move the `kie-tomcat-integration-VERSION.jar` file from the Red Hat JBoss BRMS Core Engine distribution to `tomcat7/lib`.

- Define users and roles in **tomcat7/conf/tomcat-users.xml** as shown below.



IMPORTANT

Make sure that the selected user name does *not* conflict with any known title of a role or a group. See [Section 2.8, “Defining Roles”](#) for a list of defined roles.

For example, if there is a role called **admin**, you should*not* create a user with the user name **admin**.

```
<role rolename="admin"/>
<role rolename="analyst"/>
<user username="user" password="password" roles="admin,analyst"/>
```

- Install the transaction manager.



WARNING

Please note that the following section describes the setup of a transaction manager, Bitronix, that is not officially supported by Red Hat.

Copy the following transaction manager JAR libraries from the **lib** folder to **\$TOMCAT_DIR/lib/** directory. These JARs are available in **jboss-brms-VERSION-deployable-generic.zip** and **jboss-brms-VERSION-engine.zip**.

- **btm-VERSION.jar**
- **btm-tomcat55-lifecycle-VERSION.jar**
- **h2-VERSION.jar**
- **jta-VERSION.jar**
- **slf4j-api-VERSION.jar**
- **slf4j-jdk14-VERSION.jar**

Additionally, download the following library and copy it into the **\$TOMCAT_DIR/lib/** folder: [javax.security.jacc-api.jar](#).

Add **Valve** configuration into **TOMCAT_HOME/conf/server.xml** inside the **<host>** element as last **Valve** definition:

```
<Valve className="org.kie.integration.tomcat.JACCValve" />
```

- Create the transaction manager configuration files in **\$TOMCAT_DIR/conf/**:

- **btm-config.properties**

```
bitronix.tm.serverId=tomcat-btm-node0
bitronix.tm.journal.disk.logPart1Filename=${btm.root}/work/btm1.t
log
bitronix.tm.journal.disk.logPart2Filename=${btm.root}/work/btm2.t
log
bitronix.tm.resource.configuration=${btm.root}/conf/resources.pro
perties
```

- **resources.properties** (the **resource.ds1.uniqueName** defines the data source name used in Tomcat resource definition later — make a note of this value).

Example 2.1. H2 Data Source Definition

```
resource.ds1.className=bitronix.tm.resource.jdbc.lrc.LrcXADataS
ource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=10
resource.ds1.maxPoolSize=20
resource.ds1.driverProperties.driverClassName=org.h2.Driver
resource.ds1.driverProperties.url=jdbc:h2:file:~/jbpm
resource.ds1.driverProperties.user=sa
resource.ds1.driverProperties.password=
resource.ds1.allowLocalTransactions=true
```

Example 2.2. MySQL 5.5 Data Source Definition

```
resource.ds1.className=com.mysql.jdbc.jdbc2.optional.MysqlXADat
aSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:mysql://localhost:3306/s
ampled
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.3. DB2 Type 4 Data Source Definition

```
resource.ds1.className=com.ibm.db2.jcc.DB2Driver
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:db2://localhost:50000/sa
mpledb
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.4. Oracle Data Source Definition


```

resource.ds1.className=oracle.jdbc.xa.client.OracleXADataSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:oracle:thin:@//localhost
:1521/bpms
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true

```

Example 2.5. Microsoft SQL Server Data Source Definition

```

resource.ds1.className=com.microsoft.sqlserver.jdbc.SQLServerDr
iver
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:sqlserver://localhost:14
33;databaseName=bpms;
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true

```

12. If you are using a data source other than the default provided by the underlying H2 database, you will need to set up persistence. If you are using the default H2 database, then you can ignore the rest of the steps in this procedure. In this procedure, you configure a data source with the JNDI name **jdbc/myDatasource** as defined in **uniqueName=jdbc/jbpm** in the Bitronix **resources.properties** file earlier (for the MySQL option).

- a. In **business-central/META-INF/context.xml**, replace the data source JNDI name in the **<Resource>** element. The **uniqueName** attribute refers to the **resource.ds1.uniqueName** property set in **resources.properties**:

```

<Resource name="jdbc/myDatasource" uniqueName="jdbc/jbpm"
auth="Container" removeAbandoned="true"
factory="bitronix.tm.resource.ResourceObjectFactory"
type="javax.sql.DataSource"/>

```

- b. In **business-central/WEB-INF/web.xml**, replace the data source JNDI name in the **<res-ref-name>** element with your data source name:

```

<resource-ref>
  <description>Console DS</description>
  <res-ref-name>jdbc/myDatasource</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>

```

- c. Change **business-central/WEB-INF/classes/META-INF/persistence.xml**.

In this file, change the name of the Hibernate dialect used for your database, if using a different database other than H2. The code below demonstrates the original database information for **persistence.xml**:

```
<property name="hibernate.dialect"
value="org.hibernate.dialect.H2Dialect"/>
```

This information can be updated in the following manner (as demonstrated with MySQL database below):

```
<property name="hibernate.dialect"
value="org.hibernate.dialect.MySQLDialect"/>
```



NOTE

The dialect for DB2 is **org.hibernate.dialect.DB2Dialect**, for DB2 on AS/400 is **org.hibernate.dialect.DB2400Dialect**, for Oracle is **org.hibernate.dialect.Oracle10gDialect**, and for Microsoft SQL Server is **org.hibernate.dialect.SQLServerDialect**.

- d. Change **business-central/WEB-INF/classes/META-INF/persistence.xml** file so that Red Hat JBoss BPM Suite process engine can use the new database. The code below demonstrates the original data source information for **persistence.xml**:

```
<jta-data-source>java:comp/env/jdbc/jbpm</jta-data-source>
```

Change this value to the data source defined earlier:

```
<jta-data-source>java:comp/env/jdbc/myDatasource</jta-data-source>
```

13. Set up the transaction manager listener in **\$TOMCAT_DIR/conf/server.xml** to start and stop Bitronix on container startup and shutdown. Add the following element as the last **<Listener>** element into the **<Server>** element:

```
<Listener
className="bitronix.tm.integration.tomcat55.BTMLifecycleListener" />
```

14. Define the **btm.root** system property and location where Bitronix configuration file is placed in:
In the **\$TOMCAT_DIR/bin/**, create a readable **setenv.sh** file with the following content:

```
CATALINA_OPTS="-Xmx512M -XX:MaxPermSize=512m -
Djava.security.auth.login.config=$CATALINA_HOME/webapps/business-
central/WEB-INF/classes/login.config -Dbtm.root=$CATALINA_HOME -
Dbitronix.tm.configuration=$CATALINA_HOME/conf/btm-config.properties
```

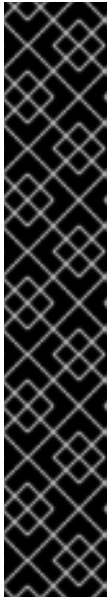
```
-Dorg.jbpm.designer.perspective=RuleFlow -
Djbpm.tsr.jndi.lookup=java:comp/env/TransactionSynchronizationRegistry
-Dorg.jboss.logging.provider=jdk"
```

The property **org.jbpm.designer.perspective** is set to **RuleFlow** to allow the default perspective for the designer to be **RuleFlow** rather than **Full**. Grant the file execute permissions if applicable.



NOTE

The **-XX:MaxPermSize=512m** JVM property is valid only for JDK 6 and 7. It is *not* valid for JDK 8+.



IMPORTANT

On Microsoft Windows systems, replace the **\$CATALINA_HOME** value in the content of the file with the equivalent environment variable name, or use the absolute path and add the values in **setenv.bat** file as shown here in the following example:

```
set "CATALINA_OPTS=-Xmx512m -XX:MaxPermSize=512m -
Djava.security.auth.login.config=C:\apache-
tomcat\webapps\business-central\WEB-
INF\classes\login.config -Dbtm.root=C:\apache-tomcat -
Dbitronix.tm.configuration=C:\apache-tomcat\conf\btm-
config.properties -
Dorg.jbpm.designer.perspective=RuleFlow -
Djbpm.tsr.jndi.lookup=java:comp/env/TransactionSynchroniz
ationRegistry"
```

15. Install the driver to your database — copy the JAR file with the relevant database driver to **\$TOMCAT_DIR/lib/**.



DRIVER TO EMBEDDED H2 DATABASE

If using the embedded H2 database, the driver is available in **business-central/WEB-INF/lib/**.

16. Start JBoss Web Server by running **startup.sh** in the **tomcat7/bin** directory.

```
./startup.sh
```

Wait a few minutes and check the server log (**\$TOMCAT_DIR/tomcat7/logs**) for any errors. If there are no errors, proceed to the next step.

17. Navigate to <http://localhost:8080/business-central> in a web browser.
18. Login with the user name/password defined in the **tomcat-users.xml** file.

After setting up Business Central, it is necessary to configure data sources for the Realtime Decision Server (**kie-server**) as well. Otherwise, Realtime Decision Server tries to find a data source under the JNDI **jboss/datasources/ExampleDS**, which is by default registered only in Red Hat JBoss EAP and *not* in Apache Tomcat.

**NOTE**

Realtime Decision Server requires a data source *only* if the jBPM extension is enabled. This extension is enabled by default.

Realtime Decision Server needs a dedicated database, which is why it is not possible to reuse the data source registered for Business Central. To add a dedicated data source for **kie-server**, do the following:

1. Copy the JAR file with the relevant database driver to **\$TOMCAT_DIR/lib/**. If you are using an H2 database, this step has already been done during the Business Central installation.
2. Add the data source into the **resources.properties** file. Note that the actual values may differ, based on the underlying database.

```
resource.ds2.className=bitronix.tm.resource.jdbc.lrc.LrcXADatasource
resource.ds2.uniqueName=jdbc/kieserver
resource.ds2.minPoolSize=10
resource.ds2.maxPoolSize=20
resource.ds2.driverProperties.driverClassName=org.h2.Driver
resource.ds2.driverProperties.url=jdbc:h2:file:~/bpm630tomcat7kieserver
resource.ds2.driverProperties.user=sa
resource.ds2.driverProperties.password=
resource.ds2.allowLocalTransactions=true
```

3. Register a new resource in the **kie-server/META-INF/context.xml**:

```
<Resource name="jdbc/kieserver"
  uniqueName="jdbc/kieserver"
  auth="Container"
  removeAbandoned="true"
  factory="bitronix.tm.resource.ResourceObjectFactory"
  type="javax.sql.DataSource" />
```

4. Update the **\$TOMCAT_DIR/bin/setenv.sh** file. Add the following data source-related properties for **kie-server**:

```
-Dorg.kie.server.persistence.ds=java:comp/env/jdbc/kieserver
-
Dorg.kie.server.persistence.tm=org.hibernate.service.jta.platform.int
ernal.BitronixJtaPlatform
```

2.8. DEFINING ROLES

Before starting the server and logging in to Business Central, you will need to create some user accounts. This section describes the different user roles that are used in Red Hat JBoss BRMS:

- **admin**: The users with **admin** role are the administrators of the application. Administrators can manage users, manage the repositories (create and clone), and have full access to make the required changes in the application. Administrators have access to all areas within the system.

- **analyst**: An **analyst** role has access to all high-level features to model projects. However, **Authoring** → **Administration** access is unavailable to users with the **analyst** role. Certain lower-level features targeted towards developers, like the **Deployment** → **Artifact Repository** view are not accessible for this role. However, the **Build & Deploy** button is available for the **analyst** role while using the Project Editor.



NOTE

Enter the above mentioned roles during the user creation process.

2.9. CREATING USERS

To start adding new users, you will need to run the **add-user.sh** script on a Unix system or the **add-user.bat** file on a Windows system from the Red Hat JBoss EAP **bin** directory.

Procedure: Creating New Users

1. Go to the Red Hat JBoss EAP **bin** directory.
2. On a Unix system, run the following command:

```
./add-user.sh
```

On a Windows system, run:

```
./add-user.bat
```

3. Enter **b** to select the application user and press **Enter**.
4. Accept the default realm (ApplicationRealm) by pressing **Enter**.
5. At the user name prompt, enter the user name and confirm. For example **helloworlduser**.



IMPORTANT

Make sure that the selected user name does *not* conflict with any known title of a role or a group.

For example, if there is a role called **admin**, you should *not* create a user with the user name **admin**.

6. Create the user password at the password prompt and reenter the password. For example **Helloworld@123**.



NOTE

The password should be at least 8 characters in length and should contain upper and lower case alphabetic characters (A-Z, a-z), at least one numerical character (0-9) and at least one special character (for example ~ ! @ # \$ % ^ * () - _ + =).

7. Enter a comma-separated list of roles the user will need at the roles prompt (see [Section 2.8, “Defining Roles”](#)).
Note that Business Central users need to have the **analyst** or the **admin** role assigned.
8. Confirm that you want to add the user.
9. Enter **yes** at the next prompt to enable clustering in the future.

CHAPTER 3. PERSISTENCE SETUPS

3.1. CONFIGURING PERSISTENCE FOR BUSINESS CENTRAL

Red Hat JBoss BRMS is configured to use an example data source with Java Naming and Directory Interface (JNDI) name `java:jboss/datasources/ExampleDS`. The example data source is not suitable for production.

To change the data source:

1. Prepare your database:
 - a. Go to the [Product Downloads](#) on the Customer Portal and select Red Hat JBoss BRMS.
 - b. Download **Red Hat JBoss BRMS 6.3.0 Supplementary Tools**.
 - c. Unzip `jboss-brms-bpmsuite-6.3-supplementary-tools/ddl-scripts`, for example into `/tmp/ddl`.
 - d. Import the DDL script for your database into the database you want to use, for example:

```
psql jbpn < /tmp/ddl/postgresql/postgresql-jbpn-schema.sql
```

2. Install the Java Database Connectivity (JDBC) driver onto your application platform. For example, to deploy a PostgreSQL JDBC driver:
 - a. Copy your JDBC jar file into `EAP_HOME/modules/org/postgres/main/`.
 - b. Create `EAP_HOME/modules/org/postgres/main/module.xml` with the following content:

```
<?xml version="1.0" ?>
<module xmlns="urn:jboss:module:1.1" name="org.postgres">
  <resources>
    <!-- Change the name of the JDBC driver to match yours. -
  ->
    <resource-root path="postgresql-9.4-1202.jdbc42.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
```

For further information about deploying JDBC drivers, see [Install a JDBC Driver as a Core Module](#) of the *Red Hat JBoss Enterprise Application Platform Administration and Configuration Guide*.

3. Create a new data source:
 - a. Open `EAP_HOME/standalone/configuration/standalone.xml` and locate `<datasources>`.
 - b. Add your data source and the driver:

■

```

<!-- Default data source not suitable for production, now
disabled. -->
<datasources>
    <datasource jndi-name="java:jboss/datasources/ExampleDS"
pool-name="ExampleDS" enabled="false" use-java-context="true">
        <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-
1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
        <driver>h2</driver>
        <security>
            <user-name>sa</user-name>
            <password>sa</password>
        </security>
    </datasource>

    <!-- The new data source configuration. -->
    <datasource jndi-name="java:jboss/datasources/PostgresqlDS"
pool-name="PostgresqlDS" enabled="true">
        <connection-
url>jdbc:postgresql://localhost:5432/jbpm</connection-url>
        <driver>postgresql-9.4-1202.jdbc42.jar</driver>
        <security>
            <user-name>jbpm</user-name>
            <password>jbpm</password>
        </security>
    </datasource>

    <!-- The drivers section. -->
    <drivers>
        <driver name="h2" module="com.h2database.h2">
            <xa-datasource-
class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
        </driver>
        <!-- Register your new driver. -->
        <driver name="postgres" module="org.postgres">
            <driver-class>org.postgresql.Driver</driver-class>
        </driver>
    </drivers>
</datasources>

```

For more information, see [Example PostgreSQL Datasource](#) of the *Red Hat JBoss Enterprise Application Platform Administration and Configuration Guide*.

4. Register the data source in Business Central:

- a. Open **EAP_HOME/standalone/deployments/business-central.war/WEB-INF/classes/META-INF/persistence.xml**.
- b. Locate the **<jta-data-source>** tag and change it to the JNDI name of your data source, for example:

```

<jta-data-source>java:jboss/datasources/PostgresqlDS</jta-data-
source>

```

- c. Locate the **<properties>** tag and change the **hibernate.dialect** property, for example:


```
<property name="hibernate.dialect"
value="org.hibernate.dialect.PostgreSQLDialect" />
```

3.2. CONFIGURING PERSISTENCE FOR THE REALTIME DECISION SERVER

When changing the database for the Realtime Decision Server, the best practice is to use a separate database from the one for Business Central and Dashbuilder applications. To change the database for the Realtime Decision Server:

1. Open **EAP_HOME/standalone/configuration/standalone.xml** and locate the **<system-properties>** tag.

2. Add the following properties:

- **org.kie.server.persistence.ds**: The JNDI name of your data source.
- **org.kie.server.persistence.dialect**: The hibernate dialect for your database.

For example:

```
<system-properties>
  <property name="org.kie.server.repo"
value="${jboss.server.data.dir}"/>
  <property name="org.kie.example" value="true"/>
  <property name="org.jbpm.designer.perspective" value="full"/>
  <property name="designerdataobjects" value="false"/>
  <property name="org.kie.server.user" value="bpmsUser"/>
  <property name="org.kie.server.pwd" value="bpms123!"/>
  <property name="org.kie.server.location"
value="http://localhost:8080/kie-server/services/rest/server"/>
  <property name="org.kie.server.controller"
value="http://localhost:8080/business-central/rest/controller"/>
  <property name="org.kie.server.controller.user"
value="kieserver"/>
  <property name="org.kie.server.controller.pwd"
value="kieserver1!"/>
  <property name="org.kie.server.id" value="local-server-123"/>

  <!-- Data source properties. -->
  <property name="org.kie.server.persistence.ds"
value="java:jboss/datasources/PostgresqlDS"/>
  <property name="org.kie.server.persistence.dialect"
value="org.hibernate.dialect.PostgreSQLDialect"/> -->
</system-properties>
```

3.3. TROUBLESHOOTING

IBM DB2 database has problems with Dashbuilder

If you want to use an IBM DB2 database as the underlying data source for Business Central, increase the page size for the database. The default page size of 4 kB is not sufficient for the Dashbuilder table columns size.

When creating the database, force the page size to 16384 as in the example below:

Example 3.1. Adjusting Page Size

```
CREATE DATABASE dashb PAGESIZE 16384
```

This increase in page size for the underlying database must be performed before the Red Hat JBoss BPM Suite has been run for the first time.

Non-English characters are not displayed in Dashbuilder

If you want to use UTF 8 to display non-English characters, set the encoding at the level of database for Dashbuilder to work correctly. For example, in MySQL, add the following to the server configuration file:

```
[mysqld]
character-set-server=utf8
collation-server=utf8_general_ci
```

Oracle 11 produces multiple warnings without any cause

When you use Oracle 11 as the data source, multiple warning (**WARN**) messages are produced in the logs without any corresponding Business Central activity being performed. This is expected behavior. To turn off these messages, set the level of the **org.hibernate.loader** category of the logger to **ERROR** in the **standalone.xml** file:

```
<logger category="org.hibernate.loader">
  <level name="ERROR"/>
</logger>
```

CHAPTER 4. GIT

Git is a distributed version control system and it implements revisions as commit objects. Every time when you commit your changes into a repository this creates a new commit object in the Git repository. Similarly, the user can also copy an existing repository. This copying process is typically called cloning and the resulting repository can be referred to as clone. Every clone contains the full history of the collection of files and a cloned repository has the same functionality as the original repository.

The local repository consists of three "trees" maintained by Git as shown in the following figure:

- *Working Directory* which holds the actual files.
- *Index* which acts as a staging area.
- *Head* which points to the last commit the user has made.

The following table provides with a summary of important Git terminology.

Table 4.1. Git Terminology

Term	Definition
Branches	A branch is a named pointer to a commit. Selecting a branch in Git terminology is called to checkout a branch. If you are working in a certain branch, the creation of a new commit advances this pointer to the newly created commit. Each commit knows their parents (predecessors). Successors are retrieved by traversing the commit graph starting from branches or other refs, symbolic reference (for example HEAD) or explicit commit objects. This way a branch defines its own line of descendants in the overall version graph formed by all commits in the repository. You can create a new branch from an existing one and change the code independently from other branches. One of the branches is the default (typically named master). The default branch is the one for which a local branch is automatically created when cloning the repository.
Commit	When you commit your changes into a repository this creates a new commit object in the Git repository. This commit object uniquely identifies a new revision of the content of the repository. This revision can be retrieved later, for example, if you want to see the source code of an older version. Each commit object contains the author and the committer, thus making it possible to identify who did the change. The author and committer might be different people. The author did the change and the committer applied the change to the Git repository.
HEAD	HEAD is a symbolic reference most often pointing to the currently checked out branch. Sometimes the HEAD points directly to a commit object, this is called detached HEAD mode. In that state creation of a commit will not move any branch. The first predecessor of HEAD can be addressed through HEAD~1 , HEAD~2 and so on. If you switch branches, the HEAD pointer moves to the last commit in the branch. If you checkout a specific commit, the HEAD points to this commit.

Term	Definition
Index	Index is an alternative term for the staging area.
Repository	A repository contains the history, the different versions over time and all different branches and tags. In Git each copy of the repository is a complete repository. If the repository is not a bare repository, it allows you to checkout revisions into your working tree and to capture changes by creating new commits. Bare repositories are only changed by transporting changes from other repositories. This tutorial uses the term repository to talk about a non bare repository. If it talks about a bare repository, this is explicitly mentioned.
Revision	Represents a version of the source code. Git implements revisions as commit objects (or short commits). These are identified by an SHA-1 secure hash. SHA-1 IDs are 160 bits long and are represented in hexadecimal notation.
Staging area	The staging area is the place to store changes in the working tree before the commit. The staging area contains the set of the snapshots of changes in the working tree (change or new files) relevant to create the next commit and stores their mode (file type, executable bit).
Tags	A tag points to a commit which uniquely identifies a version of the Git repository. With a tag, you can have a named point to which you can always revert to. You can revert to any point in a Git repository, but tags make it easier. The benefit of tags is to mark the repository for a specific reason, for example with a release. Branches and tags are named pointers, the difference is that branches move when a new commit is created while tags always point to the same commit. Technically, a tag reference can also point to an annotated tag object.
URL	A URL in Git determines the location of the repository. Git distinguishes between fetchurl for getting new data from other repositories and pushurl for pushing data to another repository.
Working tree	The working tree contains the set of working files for the repository. You can modify the content and commit the changes as new commits to the repository.

Import projects from an existing Git repository in Red Hat JBoss Developer Studio (see [Section 8.5, “Importing Projects from Git Repository into Red Hat JBoss Developer Studio”](#)).

4.1. CLONING EXISTING REPOSITORY

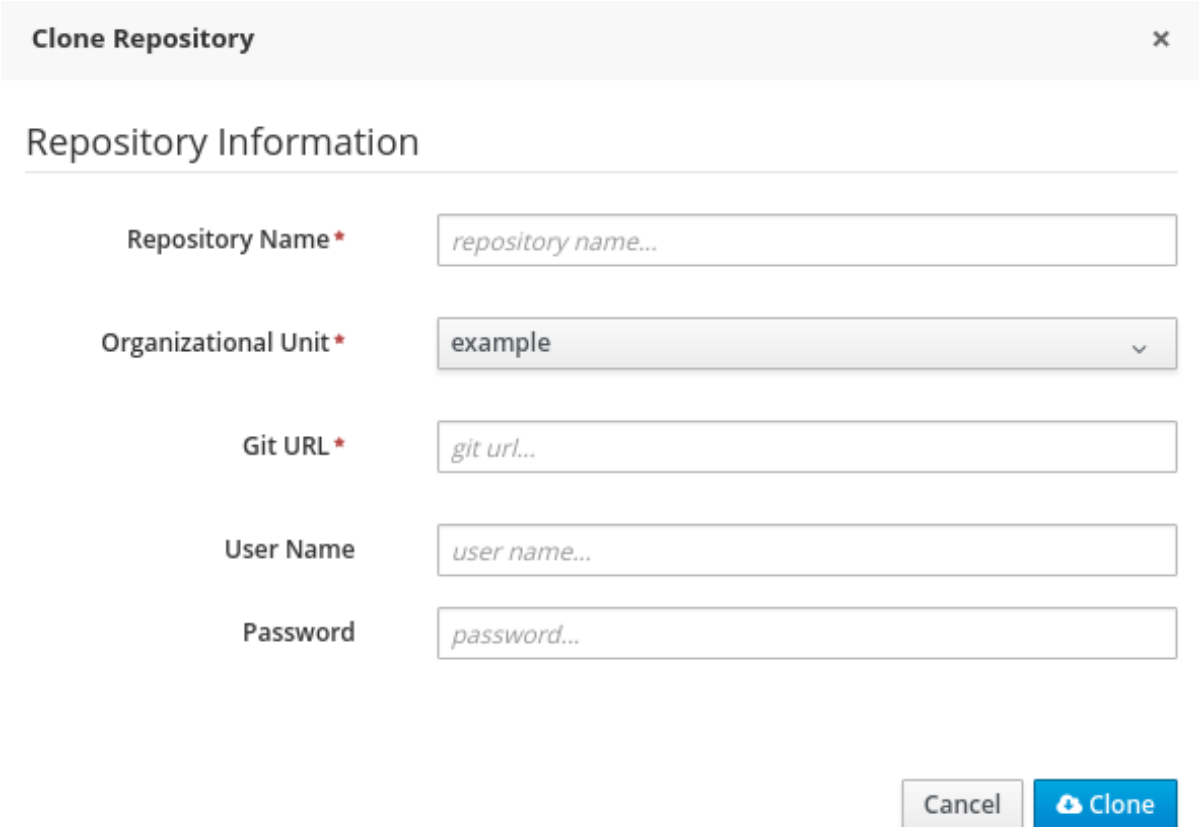
An existing Git repository can be cloned and used in Red Hat JBoss BRMS.

Procedure: Cloning Repository

1. Open the **Administration** perspective: on the main menu, click **Authoring** → **Administration**.

2. On the perspective menu, click **Repository** → **Clone repository**.
The **Clone Repository** pop-up window is displayed.

Figure 4.1. Clone Repository Pop-up



Clone Repository x

Repository Information

Repository Name *

Organizational Unit *

Git URL *

User Name

Password

3. Enter the mandatory details:
 - A repository name.
 - Select an organizational unit in which the repository is to be created from the **Organizational Unit** drop-down option.
 - Enter a Git URL.
 - Enter a user name and a password.
4. Click **Clone**.

4.2. MIGRATING REPOSITORY FROM RED HAT JBOSS BRMS 5.3

To migrate data from Red Hat JBoss BRMS 5, do the following:

1. Download the migration tool from [Red Hat Customer Portal](#) and unzip the downloaded ZIP archive.
2. For [production databases](#), copy the JDBC driver for the database that is used by the JCR repository into the **libs** directory of the migration tool.
3. On the command line, move into the **bin/** directory of the exploded ZIP archive.
In a Unix environment, run:

```
./runMigration.sh -i SOURCE_PATH -o DESTINATION_PATH -r  
REPOSITORY_NAME
```

In a Windows environment, run:

```
./runMigration.bat -i SOURCE_PATH -o DESTINATION_PATH -r  
REPOSITORY_NAME
```

Where:

- **SOURCE_PATH** is the path to the source JCR repository.
- **DESTINATION_PATH** is the path to the destination Git VFS.
- **REPOSITORY_NAME** is an arbitrary name for the new repository.

The repository is then migrated to the specified location.

CHAPTER 5. AUTHENTICATION

Authentication and user management is handled by the application server that Red Hat JBoss BRMS has been installed to. Users should see the application server documentation for more information.

CHAPTER 6. TESTING INSTALLATION

6.1. STARTING SERVER



NOTE

If you installed Red Hat JBoss BRMS using the generic deployable version on Red Hat Java Web Server, [Section 2.6, “Installing Red Hat JBoss BRMS on Red Hat JBoss Enterprise Application Platform”](#) contains the instructions for starting the server. You can ignore the following discussion.

Once the Red Hat JBoss BRMS server is installed on Red Hat JBoss EAP 6, you can run it either in the *standalone* or the *domain* mode.

6.1.1. Standalone Mode

If you used the JAR installer for JBoss EAP 6, you can run the server in one of the standalone modes: *standalone* or *standalone-secure*.



NOTE

If you chose the Deployable ZIP for JBoss EAP, the configuration steps are described in [Section 2.2, “Downloading Red Hat JBoss BRMS for Red Hat JBoss EAP”](#).

The default startup script, **standalone.sh** that Red Hat JBoss BRMS ships with, is optimized for performance. To run your server in the performance mode, do the following:

1. On the command line, move into the **`$EAP_HOME/bin/`** directory.
2. In a Unix environment, run:

```
./standalone.sh
```

In a Windows environment, run:

```
./standalone.bat
```

Red Hat JBoss BRMS also ships with a separate script, **standalone-secure.sh**, that is optimized for security. This script applies a security policy by default that protects against a known security vulnerability.



NOTE

It is recommended that production environments use **standalone-secure.sh** script.

**WARNING**

The use of a security manager imposes a significant performance penalty that you should be aware of. The tradeoff between security and performance must be made by taking into consideration individual circumstances. See [Section 6.2, “Java Security Manager and Performance Management”](#).

To run your server in the secure mode with this script, do the following:

1. On the command line, move into the `$EAP_HOME/bin/` directory.
2. In a Unix environment, run:

```
./standalone-secure.sh
```

In a Windows environment, run:

```
./standalone-secure.bat
```

**NOTE**

If you installed Red Hat JBoss BRMS using the installer, an option to apply the security policy is given to you at the time of install. The installer does not provide a separate **standalone-secure.sh** script.

6.1.2. Domain Mode

If you used the JAR installer, Red Hat JBoss BRMS is already configured for running in the domain mode.

**NOTE**

If you chose the Deployable ZIP for Red Hat JBoss EAP, the configuration steps for domain mode are described in [Section 2.2, “Downloading Red Hat JBoss BRMS for Red Hat JBoss EAP”](#).

To start Red Hat JBoss BRMS in the domain mode, perform the following steps:

1. On the command line, move into the `$EAP_HOME/bin/` directory.
2. In a Unix environment, run:

```
./domain.sh
```

In a Windows environment, run:

```
./domain.bat
```

6.2. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT

As noted earlier, enabling the Java Security Manager (JSM) to sandbox the evaluation of MVEL scripts in Red Hat JBoss BRMS introduces a performance hit in high load environments. Environments and performance markers must be kept in mind when deploying a Red Hat JBoss BRMS application. Use the following guidelines to deploy secure and high performance Red Hat JBoss BRMS applications.

- In high load environments where performance is critical it is recommended to only deploy applications that have been developed on other systems and properly reviewed. It is also recommended not to create any users with **analyst** role on such systems. If these safeguards are followed, it is safe to leave JSM disabled on these systems so it does not introduce any performance degradation.
- In testing and development environments without high loads, or in environments where rule and process authoring is exposed to external networks, it is recommended to have JSM enabled in order to achieve security benefits of properly sandboxed evaluation of MVEL.

Allowing users with **analyst** role to log in to the Business Central console with JSM disabled is not secure and not recommended.

6.3. LOGGING INTO BUSINESS CENTRAL

Log into Business Central after the server has successfully started.

1. Navigate to <http://localhost:8080/business-central> in a web browser. If the user interface has been configured to run from a domain name, substitute **localhost** for the domain name. For example <http://www.example.com:8080/business-central>.
2. Log in with the user credentials that were created during installation. For example, user: **helloworlduser** and password: **HelloWorld@123**.

Troubleshooting

Loading... screen does not disappear

When you log into Business Central, it is possible that the *Loading...* screen does not disappear. This can be caused by your firewall interfering with Server Sent Events (SSE) used by Business Central.

To work around the problem, disable SSE usage by the Business Central:

1. Create an **ErraiService.properties** file, which contains:
errai.bus.enable_sse_support=false.
2. Copy the file to **INSTALL_PATH/standalone/deployments/business-central.war/WEB-INF/classes/**.
3. Redeploy **business-central.war**.

You can create two types of Red Hat JBoss BRMS clusters:

Design-Time Clustering

Enables you to share assets in the Git repository, such as processes, rules, data objects,

and others, with all the Red Hat JBoss BRMS nodes in your cluster. It is suitable in case of concerns about a single point of failure and high availability during the development process. Design-time clustering makes use of *Apache Helix* and *Apache ZooKeeper*. Design-time clustering is not required for runtime execution.

Runtime Clustering

Enables you to use the clustering capabilities of your container, such as Red Hat JBoss EAP. Runtime clustering does not require you to manage any Apache Helix or Apache ZooKeeper nodes. *Quartz Enterprise Job Scheduler* is supported if you use timers in your application.



NOTE

If you use the Websphere Application Server, Quartz setup is not necessary. Instead, use clustered EJB Timers. For more information, see the [How to setup BPM Suite Timers to work in Websphere Application Server clustering](#) support article.

You can cluster the following components of Red Hat JBoss BRMS:

- Design-time cluster
 - *Git repository*: virtual-file-system (VFS) repository that holds the business assets.
- Runtime cluster
 - *RealTime Decision Server, or Web applications* the web application nodes must share runtime data.
For instructions on clustering the RealTime Decision Server, see [Section 6.5.5, “Clustering the Realtime Decision Server”](#), or the clustering documentation of your container.

6.4. GIT REPOSITORY CLUSTERING MECHANISM

To cluster the Git repository, Red Hat JBoss BRMS uses:

Apache Helix

Provides cluster management functionality that enables you to synchronize and replicate data among the nodes in your cluster. Apache Helix cluster is managed by Apache ZooKeeper. With Apache Helix, you can define a cluster, add nodes to the cluster, remove nodes from the cluster, and perform other cluster-management tasks. Additional information:

- Apache Helix needs to be configured on a single node only. The configuration is then stored and distributed by ZooKeeper.
- Apache Helix cluster is administered by the **helix-admin.sh** script. See Apache Helix [documentation](#) for the list of commands as well as alternative ways of managing Apache Helix cluster.
- Apache Helix cluster needs exactly one controller, which must be aware of all the nodes. See Apache Helix [controller](#) documentation and Apache Helix [architecture](#) documentation.

Apache ZooKeeper

Enables you to synchronize and replicate data from the Apache Helix cluster. An Apache ZooKeeper cluster is known as an *ensemble* and requires a majority of the servers to be functional for the service to be available.

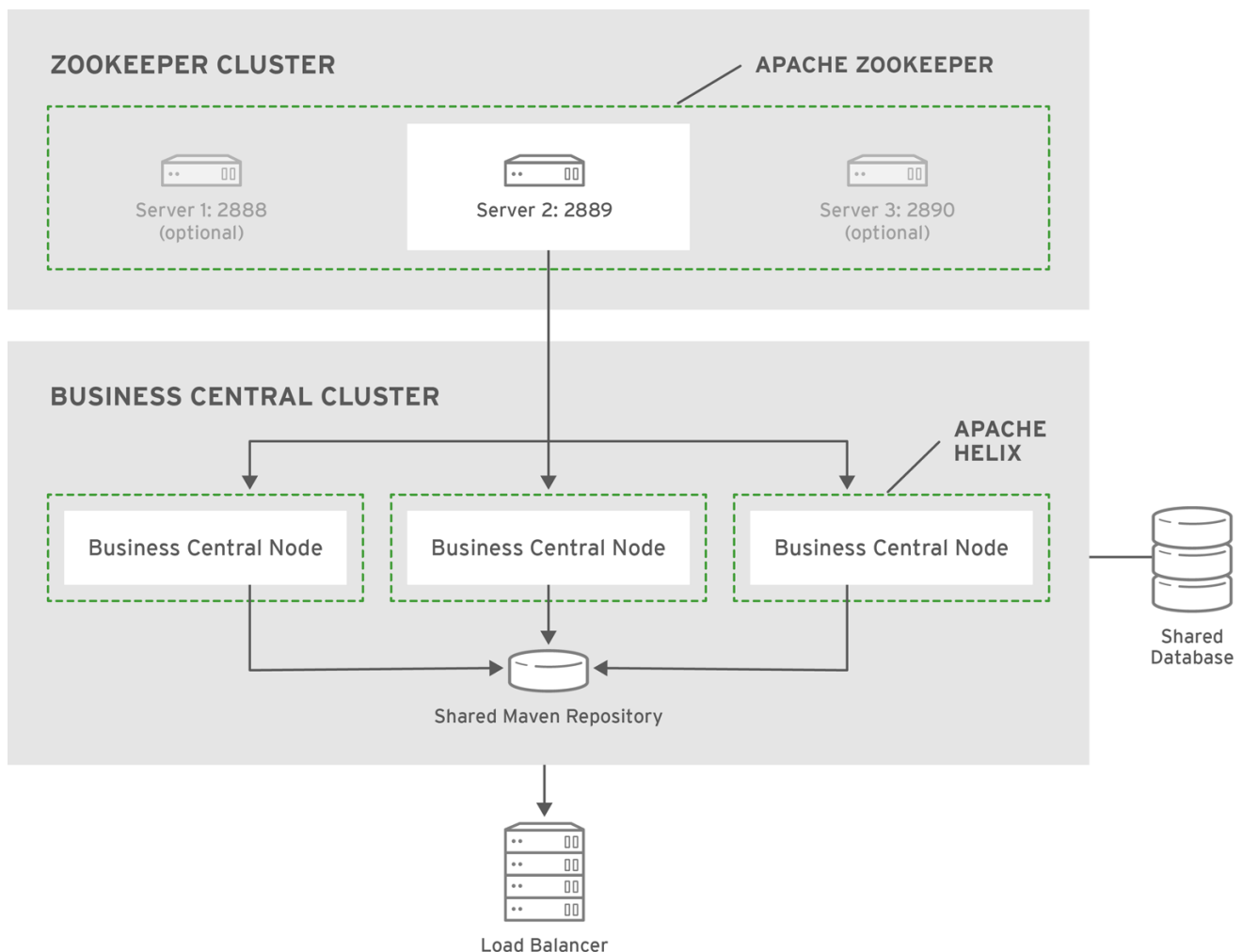
However, an ensemble is not required for any type of clustering. Only a single instance of ZooKeeper is required to allow Red Hat JBoss BRMS to replicate its data; the ZooKeeper ensemble serves to provide redundancy and protect against the failure of ZooKeeper itself.

Additional information:

- For more information about failure recovery, see Apache ZooKeeper [Data File Management](#).
- For a list of commands, see Apache ZooKeeper [ZooKeeper Commands: The Four Letter Words](#).

The relationship between Apache Helix and Apache ZooKeeper:

Figure 6.1. Schema of Red Hat JBoss BRMS Cluster



A typical clustering setup involves the following:

1. Configuring the cluster using Apache ZooKeeper and Apache Helix. This is required only for design-time clustering.
2. Configuring clustering on your container. *Red Hat JBoss BRMS Installation Guide*

provides only clustering instructions for Red Hat JBoss EAP 6.

Clustering Maven Repositories

Various Business Central operations publish JAR files to the Business Central's internal Maven Repository.

This repository exists on the application server file-system as regular files and is not cluster aware. This folder is not synchronized across the various nodes in the cluster and must be synchronized using external tools like **rsync**.

An alternative to the use of an external synchronization tool is to set the system property **org.guvnor.m2repo.dir** on each cluster node to point to a SAN or NAS. In such case, clustering of the Maven repository folder is not needed.

6.5. CLUSTERING ON RED HAT JBOSS EAP

To install Red Hat JBoss BRMS in the clustered mode, the JAR installer provides a sample setup. You can configure clustering with the deployable ZIP for EAP as well.

6.5.1. Clustering Using the JAR Installer



NOTE

The JAR installer provides sample setup only. Adjusting the configuration is necessary for it to suit your project's needs.

Using the JAR installer, you can set up a basic clustering configuration of Red Hat JBoss BRMS.

The automatic configuration creates:

- ZooKeeper ensemble with three ZooKeeper nodes
- A Helix cluster
- Two Quartz datastores (one managed, one unmanaged)

This Red Hat JBoss BRMS setup consists of two EAP nodes that share a Maven repository, use Quartz for coordinating timed tasks, and have **business-central.war**, **dashbuilder.war**, and **kie-server.war** deployed. To customize the setup to fit your scenario, or to use clustering with the deployable ZIP, see [Section 6.5.4, “Custom Configuration \(Deployable ZIP\)”](#) and the clustering documentation of your container.

Follow the installation process described in [Section 2.3, “Installing Red Hat JBoss BRMS Using Installer”](#).

1. In **Configure runtime environment**, select **Install clustered configuration** and click **Next**.
2. Select the JDBC vendor for your database.
3. Provide the corresponding driver JAR(s):
 - Select one or more files on the filesystem.

- Provide one or more URLs. The installer downloads the files automatically.

The installer copies the JAR(s) into **EAP_HOME/modules** and creates corresponding **module.xml** file.

4. Enter the url, username, and password for accessing the database used by Quartz. The installer creates:

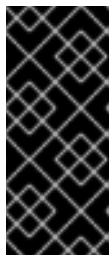
- The Quartz definition file in **EAP_HOME/domain/configuration/quartz-definition.properties**
- Two Quartz data sources in **EAP_HOME/domain/domain.xml**
Edit the **domain.xml** file to customize the setup.



NOTE

During the installation, Quartz DDL scripts will be run on the database selected in this step. The scripts make changes needed for Quartz to operate, such as adding tables. You can view the scripts in **EAP_HOME/jboss-brms-bpmsuite-6.3-supplementary-tools/ddl-scripts**. No modifications should be necessary.

5. Click **Next** to initiate the installation.



IMPORTANT

When using the JAR installer, the **war** archives are automatically created from the applications in **EAP_HOME/standalone/deployments/**. That means additional space is necessary as the applications exist both in uncompressed and compressed state in the storage during the installation.

Three ZooKeeper instances are created in **EAP_HOME/jboss-brms-bpmsuite-6.3-supplementary-tools/** in directories **zookeeper-one**, **zookeeper-two**, and **zookeeper-three**.

After the installation finishes, do not start the server from the installer. To make Apache Helix aware of the cluster nodes, Apache ZooKeeper instances, and start the cluster:

1. Change into **EAP_HOME/jboss-brms-bpmsuite-6.3-supplementary-tools/helix-core**.
2. Execute the launch script:
On UNIX systems:

```
./startCluster.sh
```

On Windows:

```
./startCluster.bat
```

3. Change into **EAP_HOME/bin**.
4. Execute the following script to start Red Hat JBoss EAP:

On UNIX systems:

```
./domain.sh
```

On Windows:

```
./domain.bat
```

6.5.2. Starting a Cluster

The **startCluster.sh** script in **EAP_HOME/jboss-brms-bpmsuite-6.3-supplementary-tools/helix-core** initializes and starts the cluster. Once initialized, further usage of **startCluster.sh** results in errors. If you installed Red Hat JBoss BRMS cluster with the installer:

- **ZOOKEEPER_HOME** is located in **EAP_HOME/jboss-brms-bpmsuite-6.3-supplementary-tools/zookeeper-NUMBER**
- **HELIX_HOME** is located in **EAP_HOME/jboss-brms-bpmsuite-6.3-supplementary-tools/helix-core**

To start a cluster:

1. Start all your ZooKeeper servers, for example:

On UNIX systems:

```
./ZOOKEEPER_HOME_ONE/bin/zkServer.sh start &
./ZOOKEEPER_HOME_TWO/bin/zkServer.sh start &
./ZOOKEEPER_HOME_THREE/bin/zkServer.sh start &
```

On Windows:

```
ZOOKEEPER_HOME_ONE/bin/zkServer.cmd start
ZOOKEEPER_HOME_TWO/bin/zkServer.cmd start
ZOOKEEPER_HOME_THREE/bin/zkServer.cmd start
```

2. Make the Helix Controller aware of the ZooKeeper instance(s). For example:

```
./HELIX_HOME/bin/run-helix-controller.sh --zkSvr
localhost:2181,localhost:2182,localhost:2183 --cluster bpms-cluster
2>&1 > /tmp/controller.log &
```

3. Change into **EAP_HOME/bin** and start Red Hat JBoss EAP:

On UNIX systems:

```
./domain.sh
```

On Windows:

```
./domain.bat
```

4. You can access your Red Hat JBoss BRMS nodes. For example, if you created Red Hat JBoss BRMS cluster by using the installer, you can access your nodes at:

```
localhost:8080/business-central  
localhost:8230/business-central
```

6.5.3. Stopping a Cluster

To stop your cluster, stop the components in the reversed order from starting it:

1. Stop the instance of Red Hat JBoss EAP, or the container you are using.
2. Stop the Helix Controller process.
On UNIX systems, find the PID of the process:

```
ps aux|grep HelixControllerMain
```

Once you have the PID, terminate the process:

```
kill -15 <pid of HelixControllerMain>
```

On Windows, use the Task Manager to stop the process.

3. Stop the ZooKeeper server(s). For each server instance, execute:
On UNIX systems:

```
./ZOOKEEPER_HOME_ONE/bin/zkServer.sh stop  
./ZOOKEEPER_HOME_TWO/bin/zkServer.sh stop  
./ZOOKEEPER_HOME_THREE/bin/zkServer.sh stop
```

On Windows:

```
ZOOKEEPER_HOME_ONE/bin/zkServer.cmd stop  
ZOOKEEPER_HOME_TWO/bin/zkServer.cmd stop  
ZOOKEEPER_HOME_THREE/bin/zkServer.cmd stop
```

6.5.4. Custom Configuration (Deployable ZIP)

When using Red Hat JBoss EAP clustering, a single Red Hat JBoss EAP domain controller exists with other Red Hat JBoss EAP slaves connecting to it as management users. You can deploy Business Central and dashbuilder as a management user on a domain controller, and the WAR deployments will be distributed to other members of the Red Hat JBoss EAP cluster.

To configure clustering on Red Hat JBoss EAP 6, do the following:

1. Configure ZooKeeper and Helix according to [Section 6.6.1, “Setting a Cluster”](#).
2. Configure individual server nodes in the **main-server-group** element in the **EAP_HOME/domain/configuration/host.xml** file with properties defined in [_cluster_properties_BRMS](#).

When configuring a Red Hat JBoss EAP cluster with Apache ZooKeeper, a different number of Red Hat JBoss EAP nodes than Apache ZooKeeper nodes is possible. However, having the same node count for both ZooKeeper and Red Hat JBoss EAP is considered best practice.

Cluster Node Properties

jboss.node.name

A node name unique in a Red Hat JBoss BRMS cluster.

Values	Default
String	N/A

org.uberfire.cluster.id

The name of the Helix cluster, for example: **kie-cluster**. You must set this property to the same value as defined in the Helix Controller.

Values	Default
String	N/A

org.uberfire.cluster.local.id

The unique ID of the Helix cluster node. Note that ':' is replaced with '_', for example **node1_12345**.

Values	Default
String	N/A

org.uberfire.cluster.vfs.lock

The name of the resource defined on the Helix cluster, for example: **kie-vfs**.

Values	Default
String	N/A

org.uberfire.cluster.zk

The location of the Zookeeper servers.

Values	Default
String of the form host1:port1,host2:port2,host3:port3,...	N/A

org.uberfire.metadata.index.dir

The location of the **.index** directory, which Apache Lucene uses when indexing and searching.

Values	Default
Path	Current working directory

org.uberfire.nio.git.daemon.host

If the Git daemon is enabled, it uses this property as the localhost identifier.

Values	Default
URL	localhost

org.uberfire.nio.git.daemon.hostport

When running in a virtualized environment, the host's outside port number for the Git daemon.

Values	Default
Port number	9418

org.uberfire.nio.git.daemon.port

If the Git daemon is enabled, it uses this property as the port number.

Values	Default
Port number	9418

org.uberfire.nio.git.dir

The location of the directory **.niogit**. Change the value for example for backup purposes.

Values	Default
Path	Current working directory

org.uberfire.nio.git.ssh.host

If the SSH daemon is enabled, it uses this property as the localhost identifier.

Values	Default
URL	localhost

org.uberfire.nio.git.ssh.hostport

When running in a virtualized environment, the host's outside port number for the SSH daemon.

Values	Default
Port number	8003

org.uberfire.nio.git.ssh.port

If the SSH daemon is enabled, it uses this property as the port number.

Values	Default
Port number	8001

Example 6.1. Cluster nodeOne Configuration

```
<system-properties>
  <property name="org.uberfire.nio.git.dir"
value="/tmp/brms/nodeone"
      boot-time="false"/>
  <property name="jboss.node.name" value="nodeOne" boot-
time="false"/>
  <property name="org.uberfire.cluster.id" value="brms-cluster"
boot-time="false"/>
  <property name="org.uberfire.cluster.zk"
value="server1:2181,server2:2181,server3:2181" boot-
time="false"/>
  <property name="org.uberfire.cluster.local.id"
value="nodeOne_12345"
      boot-time="false"/>
  <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.port" value="9418"
boot-time="false"/>
  <property name="org.uberfire.metadata.index.dir"
value="/tmp/jbrm/nodeone"
      boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/tmp/jbpm/nodeone"
      boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.port" value="8003"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.host"
value="nodeOne" />
  <property name="org.uberfire.nio.git.ssh.host" value="nodeOne"
/>
  <property name="org.uberfire.nio.git.ssh.hostport" value="8003"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.hostport"
value="9418"
      boot-time="false"/>
</system-properties>
```

Example 6.2. Cluster nodeTwo Configuration

```

<system-properties>
  <property name="org.uberfire.nio.git.dir"
value="/tmp/brms/nodetwo"
    boot-time="false"/>
  <property name="jboss.node.name" value="nodeTwo" boot-
time="false"/>
  <property name="org.uberfire.cluster.id" value="brms-cluster"
boot-time="false"/>
  <property name="org.uberfire.cluster.zk"
    value="server1:2181,server2:2182,server3:2183" boot-
time="false"/>
  <property name="org.uberfire.cluster.local.id"
value="nodeTwo_12346"
    boot-time="false"/>
  <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.port" value="9418"
boot-time="false"/>
  <property name="org.uberfire.metadata.index.dir"
value="/tmp/jbrm/nodetwo"
    boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/tmp/jbpm/nodetwo"
    boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.port" value="8003"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.host"
value="nodeTwo" />
  <property name="org.uberfire.nio.git.ssh.host" value="nodeTwo"
/>
  <property name="org.uberfire.nio.git.ssh.hostport" value="8003"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.hostport"
value="9418"
    boot-time="false"/>
</system-properties>

```

Example 6.3. Cluster nodeThree Configuration

```

<system-properties>
  <property name="org.uberfire.nio.git.dir"
value="/tmp/brms/nodethree"
    boot-time="false"/>
  <property name="jboss.node.name" value="nodeThree" boot-
time="false"/>
  <property name="org.uberfire.cluster.id" value="brms-cluster"
boot-time="false"/>
  <property name="org.uberfire.cluster.zk"
    value="server1:2181,server2:2182,server3:2183" boot-
time="false"/>
  <property name="org.uberfire.cluster.local.id"
value="nodeThree_12347"

```

```

        boot-time="false"/>
    <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo"
boot-time="false"/>
    <property name="org.uberfire.nio.git.daemon.port" value="9418"
boot-time="false"/>
    <property name="org.uberfire.metadata.index.dir"
value="/tmp/jbrm/nodethree"
        boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/tmp/jbpm/nodethree"
        boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.port" value="8003"
boot-time="false"/>
    <property name="org.uberfire.nio.git.daemon.host"
value="nodeThree" />
    <property name="org.uberfire.nio.git.ssh.host"
value="nodeThree" />
    <property name="org.uberfire.nio.git.ssh.hostport" value="8003"
boot-time="false"/>
    <property name="org.uberfire.nio.git.daemon.hostport"
value="9418"
        boot-time="false"/>
</system-properties>

```

3. Add management users as instructed in the *Administration and Configuration Guide* for Red Hat JBoss EAP and application users as instructed in *Red Hat JBoss BRMS Administration and Configuration Guide*.
4. Change to **EAP_HOME/bin** and start the application server in domain mode:
On UNIX systems:

```
./domain.sh
```

On Windows:

```
./domain.bat
```

5. Check that the nodes are available.

Deploy the Business Central application to your servers:

1. Log in as the management user to the server **Administration** console of your domain and add the new deployments using the **Runtime view** of the console. Once the deployment is added to the domain, assign it to the correct server group (**main-server-group**).

**NOTE**

It is important users explicitly check deployment unit readiness with every cluster member.

When a deployment unit is created on a cluster node, it takes some time before it is distributed among all cluster members. Deployment status can be checked using the UI and REST, however, if the query goes to the node where the deployment was originally issued, the answer is **deployed**. Any request targeting this deployment unit sent to a different cluster member fails with **DeploymentNotFoundException**.

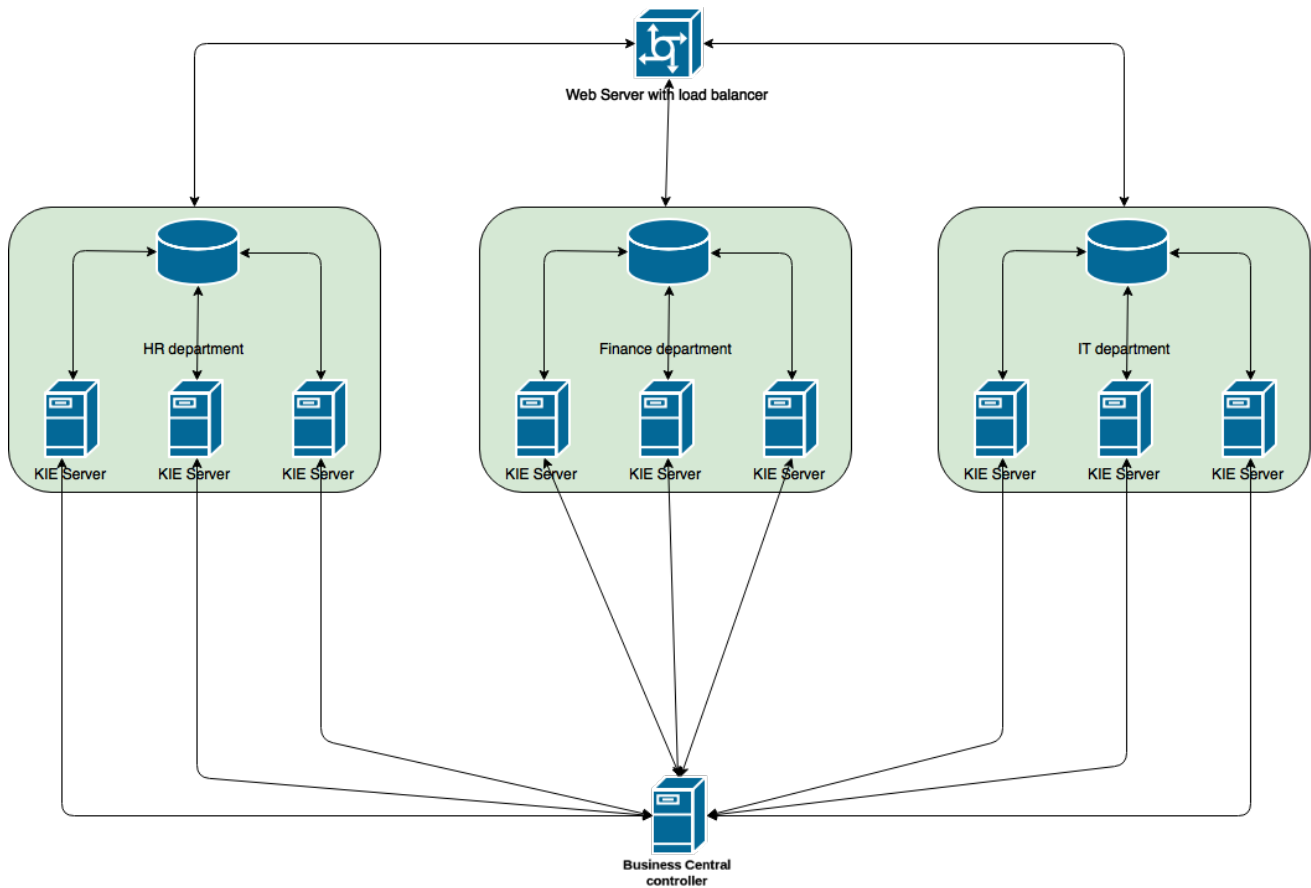
6.5.5. Clustering the Realtime Decision Server

The Realtime Decision Server is a lightweight and scalable component. Clustering it provides many benefits. For example:

- You can partition your resources based on deployed containers.
- You can scale individual instances independently from each other.
- You can distribute the cluster across network and manage it by a single controller.
 - The controller can be clustered into a ZooKeeper ensemble.
- No further components are required.

The basic runtime cluster consists of:

- Multiple Red Hat JBoss EAP instances with Realtime Decision Server
- A controller instance with Business Central



This section describes how to start Realtime Decision Server cluster on Red Hat JBoss EAP 6.4.

Creating a Realtime Decision Server Cluster

1. Change into **CONTROLLER_HOME/bin**.
2. Add a user with the **kie-server** role:

```
$ ./add-user.sh -a --user kieserver --password kieserver1! --role
kie-server
```

3. Start your controller:

```
$ ./standalone.sh
```

4. Change into **SERVER_1_HOME**.
5. Deploy **kie-server.war**. Clustered servers do not need **business-central.war** or other applications.
6. See the **<servers>** part of the following **host.xml** as an example of required properties:

```
<server name="server-one" group="main-server-group">
  <system-properties>
    <property name="org.kie.server.location"
value="http://localhost:8180/kie-
server/services/rest/server"></property> 1
```

```

    <property name="org.kie.server.controller"
value="http://localhost:8080/business-
central/rest/controller"></property> ❷
    <property name="org.kie.server.controller.user"
value="kieserver"></property> ❸
    <property name="org.kie.server.controller.pwd"
value="kieserver1!"></property> ❹
    <property name="org.kie.server.id" value="HR"></property> ❺
  </system-properties>
  <socket-bindings port-offset="100"/>
</server>

<server name="server-two" group="main-server-group" auto-
start="true">
  <system-properties>
    <property name="org.kie.server.location"
value="http://localhost:8230/kie-
server/services/rest/server"></property>
    <property name="org.kie.server.controller"
value="http://localhost:8080/business-
central/rest/controller"></property>
    <property name="org.kie.server.controller.user"
value="kieserver"></property>
    <property name="org.kie.server.controller.pwd"
value="kieserver1!"></property>
    <property name="org.kie.server.id" value="HR"></property>
  </system-properties>
  <socket-bindings port-offset="150"/>
</server>

```

- ❶ **org.kie.server.location**: URL of the server instance.
- ❷ **org.kie.server.controller**: Comma-separated list of the controller URL(s).
- ❸ **org.kie.server.controller.user**: Username you created for controller authentication. Uses **kieserver** by default.
- ❹ **org.kie.server.controller.pwd**: Password for controller authentication. Uses **kieserver1!** by default.
- ❺ **org.kie.server.id**: Server identifier that corresponds to template ID defined by the controller instance. Give the same ID to multiple server instances that represent one template.

The example above is defined for Red Hat JBoss EAP domain mode. For further list of bootstrap switches, see [Realtime Decision Server Setup](#) of *Red Hat JBoss BRMS User Guide*.

7. Repeat the previous step for as many servers as you need. To start Red Hat JBoss EAP in the domain mode, execute:

```
$ ./SERVER_HOME/bin/domain.sh
```

After connecting the servers to your controller, check the controller log:

■


```

13:54:40,315 INFO
[org.kie.server.controller.impl.KieServerControllerImpl] (http-
localhost/127.0.0.1:8080-1) Server http://localhost:8180/kie-
server/services/rest/server connected to controller
13:54:40,331 INFO
[org.kie.server.controller.impl.KieServerControllerImpl] (http-
localhost/127.0.0.1:8080-2) Server http://localhost:8230/kie-
server/services/rest/server connected to controller
13:54:40,348 INFO
[org.kie.server.controller.rest.RestKieServerControllerImpl] (http-
localhost/127.0.0.1:8080-1) Server with id 'HR' connected
13:54:40,348 INFO
[org.kie.server.controller.rest.RestKieServerControllerImpl] (http-
localhost/127.0.0.1:8080-2) Server with id 'HR' connected

```

Alternatively, to verify in controller Business Central:

1. Log into the controller Business Central.
2. Click **Deploy → Rule Deployments**.
3. View the remote servers connected to each template.

6.6. GENERIC BUNDLE CLUSTERING

6.6.1. Setting a Cluster



NOTE

If you do not use Business Central, skip this section.

To cluster your Git (VFS) repository in Business Central:

1. Download the **jboss-bpmsuite-brms-*VERSION*-supplementary-tools.zip**, which contains Apache ZooKeeper, Apache Helix, and Quartz DDL scripts.
2. Unzip the archive: the **ZooKeeper** directory (**ZOOKEEPER_HOME**) and the **Helix** directory (**HELIX_HOME**) are created.
3. Configure Apache ZooKeeper:
 - a. In the ZooKeeper directory, change to **conf** and execute:

```
cp zoo_sample.cfg zoo.cfg
```

- b. Edit **zoo.cfg**:

```

# The directory where the snapshot is stored.
dataDir=$ZOOKEEPER_HOME/data/

# The port at which the clients connects.
clientPort=2181

# Defining ZooKeeper ensemble.

```

```
# server.{ZooKeeperNodeID}={server}:{port:range}
server.1=localhost:2888:3888
server.2=localhost:2889:3889
server.3=localhost:2890:3890
```

**NOTE**

Multiple ZooKeeper nodes are not required for clustering.

Make sure the **dataDir** location exists and is accessible.

- c. Assign a node ID to each member that will run ZooKeeper. For example, use **1**, **2**, and **3** for node 1, node 2 and node 3 respectively.

The ZooKeeper node ID is specified in a field called **myid** under the data directory of ZooKeeper on each node. For example, on node 1, execute:

```
echo "1" > /zookeeper/data/myid
```

4. Provide further ZooKeeper configuration if necessary.
5. Change to **ZOOKEEPER_HOME/bin/** and start ZooKeeper:

```
./zkServer.sh start
```

You can check the ZooKeeper log in the **ZOOKEEPER_HOME/bin/zookeeper.out** file. Check this log to ensure that the ensemble (cluster) is formed successfully. One of the nodes should be elected as leader with the other two nodes following it.

6. Once the ZooKeeper ensemble is started, configure and start Helix. Helix needs to be configured from a single node only. The configuration is then stored by the ZooKeeper ensemble and shared as appropriate. Configure the cluster with the ZooKeeper server as the master of the configuration:

- a. Create the cluster by providing the ZooKeeper Host and port as a comma-separated list:

```
$HELIX_HOME/bin/helix-admin.sh --zkSvr
ZOOKEEPER_HOST:ZOOKEEPER_PORT --addCluster <clustername>
```

- b. Add your nodes to the cluster:

```
HELIX_HOME/bin/helix-admin.sh --zkSvr
ZOOKEEPER_HOST:ZOOKEEPER_PORT --addNode <clustername>:
<name_uniqueID>
```

Example 6.4. Adding Two Cluster Nodes

```
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183
--addNode brms-cluster nodeOne:12345
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183
--addNode brms-cluster nodeTwo:12346
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183
--addNode brms-cluster nodeThree:12347
```

7. Add resources to the cluster.

```
helix-admin.sh --zkSvr ZOOKEEPER_HOST:ZOOKEEPER_PORT --addResource
<clustername> <resourceName> <numPartitions> <stateModelName>
```

Learn more about state machine configuration at [Helix Tutorial: State Machine Configuration](#).

Example 6.5. Adding vfs-repo as Resource

```
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183 --
addResource brms-cluster vfs-repo 1 LeaderStandby AUTO_REBALANCE
```

8. Rebalance the cluster with the three nodes.

```
helix-admin.sh --zkSvr ZOOKEEPER_HOST:ZOOKEEPER_PORT --rebalance
<clustername> <resourcename> <replicas>
```

Learn more about rebalancing at [Helix Tutorial: Rebalancing Algorithms](#).

Example 6.6. Rebalancing brms-cluster

```
./helix-admin.sh --zkSvr server1:2181,server2:2182,server3:2183 --
rebalance brms-cluster vfs-repo 3
```

In the above command, **3** stands for three ZooKeeper nodes.

9. Start the Helix controller in all the nodes in the cluster.

Example 6.7. Starting Helix Controller

```
./run-helix-controller.sh --zkSvr
server1:2181,server2:2182,server3:2183 --cluster brms-cluster 2>&1
> ./controller.log &
```

NOTE

In case you decide to cluster ZooKeeper, add an odd number of instances in order to recover from failure. After a failure, the remaining number of nodes still need to be able to form a majority. For example a cluster of five ZooKeeper nodes can withstand loss of two nodes in order to fully recover. One ZooKeeper instance is still possible, replication will work, however no recover possibilities are available if it fails.

6.6.2. Starting and Stopping a Cluster

To start your cluster, see [Section 6.5.2, “Starting a Cluster”](#). To stop your cluster, see [Section 6.5.3, “Stopping a Cluster”](#).

CHAPTER 7. MAVEN REPOSITORIES

7.1. ABOUT MAVEN

Apache Maven is a distributed build automation tool used in Java application development to build and manage software projects. Maven uses configuration XML files called POM (Project Object Model) to define project properties and manage the build process. POM files describe the project's module and component dependencies, build order, and targets for the resulting project packaging and output. This ensures that projects are built in a correct and uniform manner.

Maven uses repositories to store Java libraries, plug-ins, and other build artifacts. Repositories can be either local or remote. A local repository is a download of artifacts from a remote repository cached on a local machine. A remote repository is any other repository accessed using common protocols, such as **http://** when located on an HTTP server, or **file://** when located on a file server. The default repository is the public remote [Maven 2 Central Repository](#).

Configuration of Maven is performed by modifying the **settings.xml** file. You can either configure global Maven settings in the **M2_HOME/conf/settings.xml** file, or user-level settings in the **USER_HOME/.m2/settings.xml** file.

For more information about Maven, see the [Welcome to Apache Maven](#) page.

For more information about Maven repositories, see the [Apache Maven Project — Introduction to Repositories](#) article.

For more information about Maven POM files, see [Apache Maven Project — POM Reference](#).



NOTE

Your Red Hat JBoss product has been built with Maven 3.0.x. Therefore, this is the recommended Maven version for building your own SwitchYard applications.

7.2. ABOUT PROVIDED MAVEN REPOSITORIES

A set of repositories containing artifacts required to build applications based on Red Hat JBoss BRMS is provided with this release. Maven must be configured to use these repositories and the Maven Central Repository in order to provide correct build functionality.

Two interchangeable sets of repositories ensuring the same functionality are provided. The first set is available for download and storage in a local file system, the second set is hosted online for use as remote repositories.



IMPORTANT

The set of online remote repositories is a technology preview source of components. As such, it is not in scope of patching and is supported only for use in development environment. Using the set of online repositories in production environment is a potential source of security vulnerabilities and is therefore not a supported use case. For more information, see the [JBoss Enterprise Maven Repository](#).

7.3. CONFIGURING MAVEN TO USE FILE SYSTEM REPOSITORIES

Overview

In situations where you cannot use the online repositories, you will have to download and configure the required repositories locally.

1. Download the following ZIP archives containing the required repositories:
 - <https://access.redhat.com/jbossnetwork/restricted/softwareDetail.html?softwareId=43621&product=brms&version=6.3&downloadType=distributions>
2. Unzip the downloaded ZIP files into an arbitrary location in a local file system.
3. Add entries for the unzipped repositories to Maven's **settings.xml** file. The following code sample contains a profile with the repositories, configuration of authentication for access to the repositories, and an activation entry for the profile:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/xsd/settings-
1.0.0.xsd">
  <localRepository/>
  <profiles>
    <!-- Profile with local repositories required by
         Red Hat JBoss BRMS/Red Hat JBoss BPM Suite -->
    <profile>
      <id>brms-bpms-local-profile</id>
      <repositories>
        <repository>
          <id>jboss-brms-bpmsuite-repository</id>
          <name>BRMS/BPMS 6.3.0 GA Repository</name>
          <url>
            file://<!-- path to the repository -->
              /jboss-brms-bpmsuite-6.3.0.GA-maven-
repository/maven-repository
          </url>
          <layout>default</layout>
          <releases>
            <enabled>true</enabled>
            <updatePolicy>never</updatePolicy>
          </releases>
          <snapshots>
            <enabled>false</enabled>
            <updatePolicy>never</updatePolicy>
          </snapshots>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>jboss-brms-bpmsuite-repository</id>
          <name>BRMS/BPMS 6.3.0 GA Repository</name>
          <url>
            file://<!-- path to the repository -->
```

```

        /jboss-brms-bpmsuite-6.3.0.GA-maven-
repository/maven-repository
    </url>
    <layout>default</layout>
    <releases>
        <enabled>true</enabled>
        <updatePolicy>never</updatePolicy>
    </releases>
    <snapshots>
        <enabled>false</enabled>
        <updatePolicy>never</updatePolicy>
    </snapshots>
</pluginRepository>
</pluginRepositories>
</profile>
</profiles>

<servers>
    <!-- Configuring pre-emptive authentication for the repository
server -->
    <server>
        <id>brms-bpms-m2-repo</id>
        <username>admin</username>
        <password>admin</password>
        <configuration>
            <wagonProvider>httpClient</wagonProvider>
            <httpConfiguration>
                <all>
                    <usePreemptive>true</usePreemptive>
                </all>
            </httpConfiguration>
        </configuration>
    </server>

    <!-- Alternative to enabling pre-emptive authentication -
configuring
        the Authorization HTTP header with Base64-encoded
credentials
    <server>
        <id>brms-bpms-m2-repo</id>
        <configuration>
            <httpHeaders>
                <property>
                    <name>Authorization</name>
                    <value>Basic YWRtaW46YWRtaW4= </value>
                </property>
            </httpHeaders>
        </configuration>
    </server>
    -->
</servers>

<activeProfiles>
    <!-- Activation of the Red Hat JBoss BRMS/Red Hat JBoss BPM
Suite profile -->

```

```

    <activeProfile>brms-bpms-local-profile</activeProfile>
  </activeProfiles>
</settings>

```

Result

The Maven repositories are downloaded, unzipped in a local file system, registered in Maven's **settings.xml** file, and ready to be used when performing Maven builds.

7.3.1. Troubleshooting

7.3.1.1. Why do I still get errors when building or deploying my applications?

When you build or deploy a project, it fails with one or both of the following errors:

- **[ERROR] Failed to execute goal on project *PROJECT_NAME***
- **Could not find artifact *ARTIFACT_NAME***

Your cached local Maven repository might contain outdated artifacts.

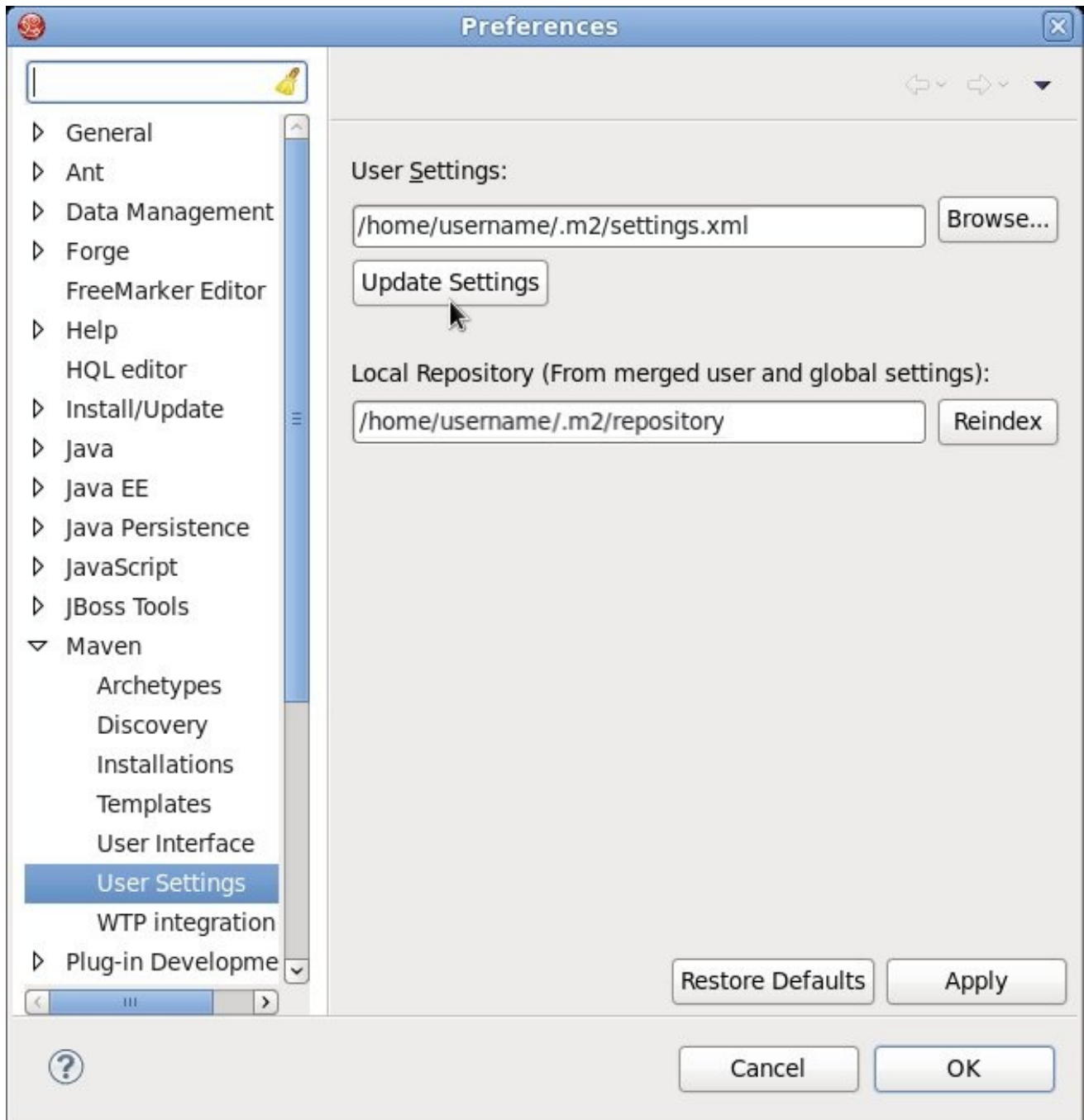
To resolve the issue, delete the cached local repository — the `~/.m2/repository/` directory on Linux or the `%SystemDrive%\Users\USERNAME\.m2\repository\` directory on Windows — and run **`mvn clean install -U`**. This will force Maven to download correct versions of required artifacts when performing the next build.

7.3.1.2. Why is Red Hat JBoss Developer Studio using my old Maven configuration?

You have updated your Maven configuration, but this configuration is not reflected in Red Hat JBoss Developer Studio.

If Red Hat JBoss Developer Studio is running when you modify your Maven **settings.xml** file, this configuration will not be reflected in Red Hat JBoss Developer Studio.

Refresh the Maven settings in the IDE. From the menu, choose **Window → Preferences**. In the **Preferences** window, expand **Maven** and choose **User Settings**. Click the **Update Settings** button to refresh the Maven user settings in Red Hat JBoss Developer Studio.



7.4. CONFIGURING MAVEN TO USE ONLINE REPOSITORIES

The online repositories required for Red Hat JBoss BRMS applications are located at <https://maven.repository.redhat.com/ga/>.

It is possible to configure Maven to use online repositories using the project's POM file, but this is not recommended.

Procedure: Configuring Maven to Use Online Repositories

1. Add entries for the online repositories and configuration of authentication for accessing them to Maven's **settings.xml** file as in the code sample below:

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```



```

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">
<profiles>
  <!-- Profile with online repositories required by BRMS/BPMS -->
  <profile>
    <id>brms-bpms-online-profile</id>
    <repositories>
      <repository>
        <id>jboss-ga-repository</id>
        <url>https://maven.repository.redhat.com/ga/</url>
        <releases>
          <enabled>true</enabled>
        </releases>
        <snapshots>
          <enabled>>false</enabled>
        </snapshots>
      </repository>
    </repositories>
    <pluginRepositories>
      <pluginRepository>
        <id>jboss-ga-plugin-repository</id>
        <url>https://maven.repository.redhat.com/ga/</url>
        <releases>
          <enabled>true</enabled>
        </releases>
        <snapshots>
          <enabled>>false</enabled>
        </snapshots>
      </pluginRepository>
    </pluginRepositories>
  </profile>
</profiles>

<servers>
  <!-- Configuring pre-emptive authentication for the repository
server -->
  <server>
    <id>brms-bpms-m2-repo</id>
    <username>admin</username>
    <password>admin</password>
    <configuration>
      <wagonProvider>httpclient</wagonProvider>
      <httpConfiguration>
        <all>
          <usePreemptive>true</usePreemptive>
        </all>
      </httpConfiguration>
    </configuration>
  </server>

  <!-- Alternative to enabling pre-emptive authentication -
configuring
      the Authorization HTTP header with Base64-encoded
credentials
  <server>

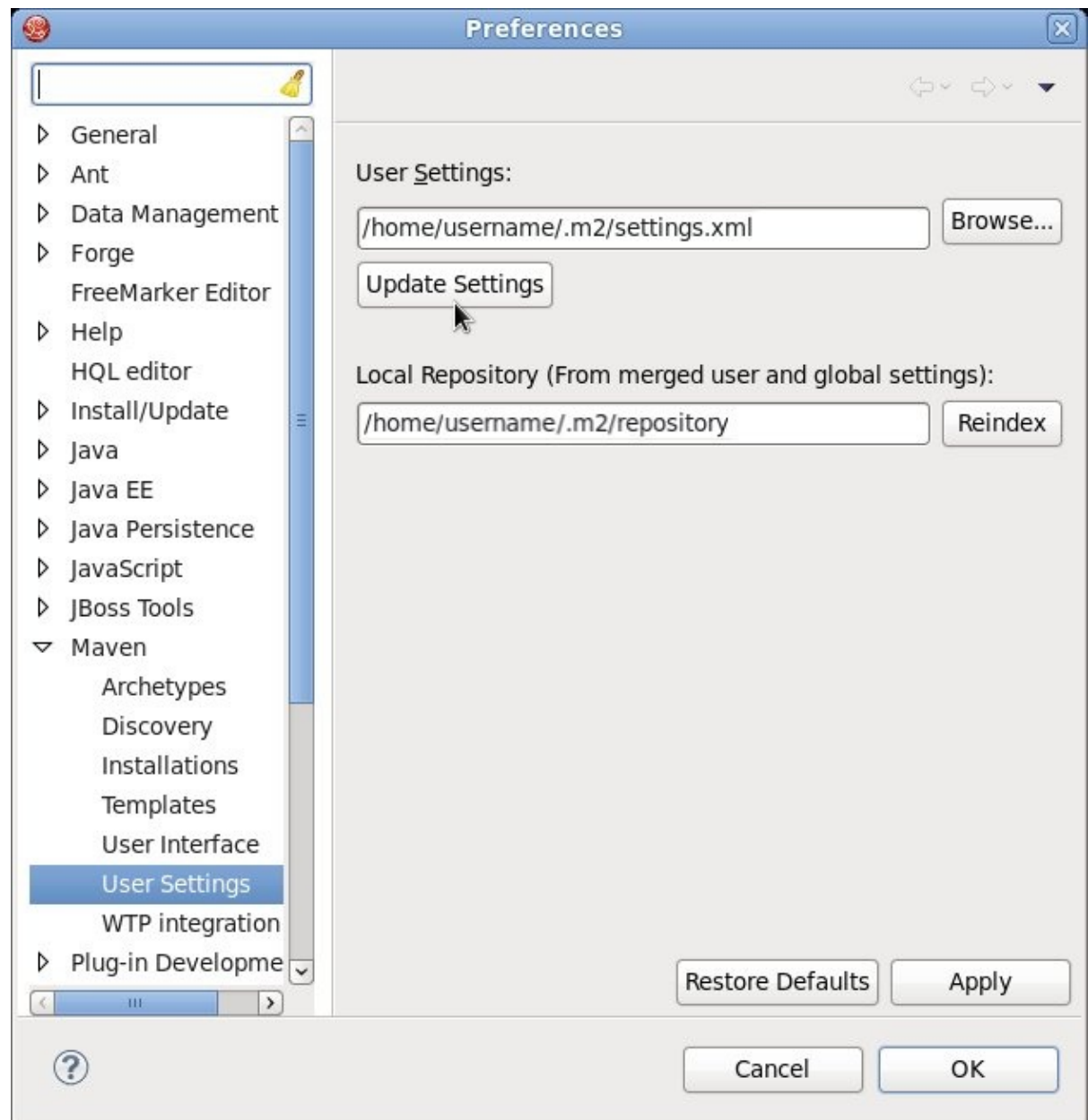
```

```
<id>brms-bpms-m2-repo</id>
<configuration>
  <httpHeaders>
    <property>
      <name>Authorization</name>
      <value>Basic YWRtaW46YWRtaW4= </value>
    </property>
  </httpHeaders>
</configuration>
</server>
-->
</servers>

<activeProfiles>
  <!-- Activation of the BRMS/BPMS profile -->
  <activeProfile>brms-bpms-online-profile</activeProfile>
</activeProfiles>

</settings>
```

2. If you modified the **settings.xml** file while Red Hat JBoss Developer Studio was running, you must refresh Maven settings in the IDE. From the menu, choose **Window → Preferences**. In the **Preferences** window, expand **Maven** and choose **User Settings**. Click the **Update Settings** button to refresh the Maven user settings in Red Hat JBoss Developer Studio.

Figure 7.1. Update Maven User Settings

Result

Maven has been configured to use the online repositories provided for your Red Hat JBoss product.



IMPORTANT

If your cached local Maven repository contains outdated artifacts, you may encounter one of the following Maven errors when you build or deploy your project:

- **Missing artifact *ARTIFACT_NAME***
- **[ERROR] Failed to execute goal on project *PROJECT_NAME*;
Could not resolve dependencies for *PROJECT_NAME***

To resolve the issue, delete the cached local repository — the `~/.m2/repository/` directory on Linux or the `%SystemDrive%\Users\USERNAME\.m2\repository\` directory on Windows — and run `mvn clean install -U`. This will force Maven to download correct versions of required artifacts when performing the next build.

7.5. DEPENDENCY MANAGEMENT

In order to use the correct Maven dependencies in your Red Hat JBoss BRMS project, you must add relevant Bill Of Materials (BOM) files to the project's `pom.xml` file. Adding the BOM files ensures that the correct versions of transitive dependencies from the provided Maven repositories are included in the project.

To use the Red Hat JBoss BRMS Maven artifacts, you need to configure one of the following:

- Red Hat JBoss BRMS product online Maven repository — for instructions, see [Section 7.4, “Configuring Maven to Use Online Repositories”](#).
- Red Hat JBoss BRMS product local-filesystem-based Maven repository — for instructions, see [Section 7.3, “Configuring Maven to Use File System Repositories”](#).

The Maven Central online repository, which is also required, is configured by default in Maven. It has to be reachable for your project to function properly, even if you selected the local Maven repository.

Depending on your project requirements, declare one of the following the dependencies in your POM file in the dependencies section. For further information about the versions, see the Supported Component Versions chapter of *Red Hat JBoss BPM Suite Installation Guide*

- `org.jboss.bom.brms:jboss-brms-bpmsuite-platform-bom:$VERSION`
- `org.jboss.bom.brms:jboss-brms-bpmsuite-bom:$VERSION`

CHAPTER 8. RED HAT JBOSS DEVELOPER STUDIO

8.1. RED HAT JBOSS DEVELOPER STUDIO

Red Hat JBoss Developer Studio is the JBoss Integrated Development Environment (IDE) based on Eclipse. Get the latest Red Hat JBoss Developer Studio from the [Red Hat Customer Portal](#). Red Hat JBoss Developer Studio provides plug-ins with tools and interfaces for Red Hat JBoss BRMS and Red Hat JBoss BPM Suite. These plugins are based on the community version of these products. So, the Red Hat JBoss BRMS plug-in is called the Drools plug-in and the Red Hat JBoss BPM Suite plug-in is called the jBPM plug-in.

See the *Red Hat JBoss Developer Studio* documentation for installation and setup instructions.



WARNING

Due to an issue in the way multi-byte rule names are handled, you must ensure that the instance of JBoss Developer Studio is started with the file encoding set to UTF-8. You can do this by editing the `$JBDS_HOME/studio/jbdevstudio.ini` file and adding the following property: `"-Dfile.encoding=UTF-8"`.

8.2. INSTALLING THE RED HAT JBOSS DEVELOPER STUDIO PLUG-INS

The Drools plug-ins for Red Hat JBoss Developer Studio are available on the update site.

Procedure: Installing the Drools Red Hat JBoss Developer Studio Plug-in

1. Start Red Hat JBoss Developer Studio.
2. Select **Help** → **Install New Software**.
3. Click **Add** to enter the **Add Repository** menu.
4. Give the software site a name next to **Name** field and add the following URL in the **Location** field:
<https://devstudio.redhat.com/9.0/stable/updates/integration-stack/>.
5. Click **OK**.
6. Select the **JBoss Business Process and Rule Development** feature from the available options and click **Next** and then **Next** again.
7. Read the license and accept it by selecting the appropriate radio button, and click **Finish**.
8. Once the plug-in installation is complete, restart Red Hat JBoss Developer Studio.

8.3. SETTING THE DROOLS AND JBPM RUNTIME ENVIRONMENTS

To use the Red Hat JBoss BRMS and Red Hat JBoss BPM Suite plug-ins with Red Hat JBoss Developer Studio, you must set up the runtimes.

A runtime is a collection of JAR files that represents a specific release of the software and provides libraries needed for compilation and running of your business assets.

Procedure: Configuring Red Hat JBoss BRMS and Red Hat JBoss BPM Suite Runtimes

1. Extract the runtime JAR files located in the **jboss-brms-*VERSION*-engine.zip** or **jboss-bpmsuite-*VERSION*-engine.zip** archive that you can download from the [Red Hat Customer Portal](#).
2. From the Red Hat JBoss Developer Studio menu, select **Window** and click **Preferences**.
3. To install the Drools runtime, select **Drools** → **Installed Drools Runtimes**. To install the jBPM runtime, select **jBPM** → **Installed jBPM Runtimes**.
4. Click **Add...**, provide a name and a version of the new runtime, and click **Browse** to navigate to the directory where you extracted the runtime files in the first step. Click **OK** to register the selected runtime in Red Hat JBoss Developer Studio.
5. Mark the runtime you have created as the default runtime by clicking on the check box next to it.
6. Click **OK**. If you have existing projects, a dialog box will indicate that you have to restart Red Hat JBoss Developer Studio to update the runtime.

8.4. CONFIGURING RED HAT JBOSS BRMS SERVER

Red Hat JBoss Developer Studio can be configured to run the Red Hat JBoss BRMS Server.

Procedure: Configuring Server

1. Open the Drools view by selecting **Window** → **Open Perspective** → **Other**. Select **Drools** and click **OK**.
2. Add the server view by selecting **Window** → **Show View** → **Other...** and select **Server** → **Servers**.
3. Open the server menu by right clicking the **Servers** panel and select **New** → **Server**.
4. Define the server by selecting **JBoss Enterprise Middleware** → **JBoss Enterprise Application Platform 6.4+** and click **Next**.
5. Set the home directory by clicking the **Browse** button. Navigate to and select the installation directory for JBoss EAP 6.4 which has Red Hat JBoss BRMS installed.
6. Provide a name for the server in the **Name** field, make sure that the configuration file is set, and click **Finish**.

8.5. IMPORTING PROJECTS FROM GIT REPOSITORY INTO RED HAT JBOSS DEVELOPER STUDIO

You can configure Red Hat JBoss Developer Studio to connect to a central Git asset repository. The repository stores rules, models, functions, and processes.

You can either clone a remote Git repository or import a local Git repository.

Procedure: Cloning Remote Git Repository

1. Start the Red Hat JBoss BRMS server by selecting the server from the **Servers** tab and click the start icon.
2. Simultaneously, start the Secure Shell server, if not running already, by using the following command. The command is Linux and Mac specific only. On these platforms, if **sshd** has already been started, this command fails. In that case, you may safely ignore this step.

```
/sbin/service sshd start
```


3. In Red Hat JBoss Developer Studio, select **File → Import...** and navigate to the Git folder. Open the Git folder to select **Projects from Git** and click **Next**.
4. Select the repository source as **Clone URI** and click **Next**.
5. Enter the details of the Git repository in the next window and click **Next**.

Figure 8.1. Git Repository Details

Clone Git Repository

Source Git Repository

Enter the location of the source repository.



Location

URI:

Host:

Repository path:

Connection

Protocol: ▼


Port:

Authentication

User:

Password:

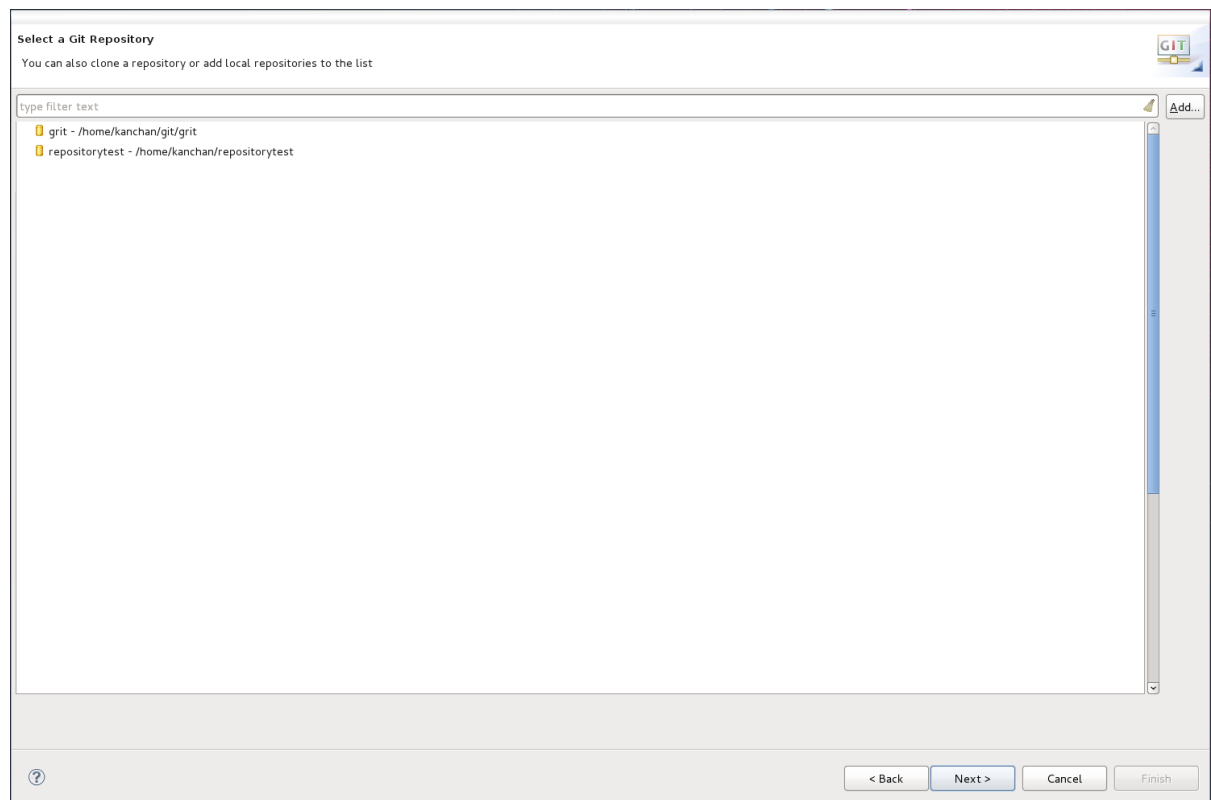
Store in Secure Store ☐



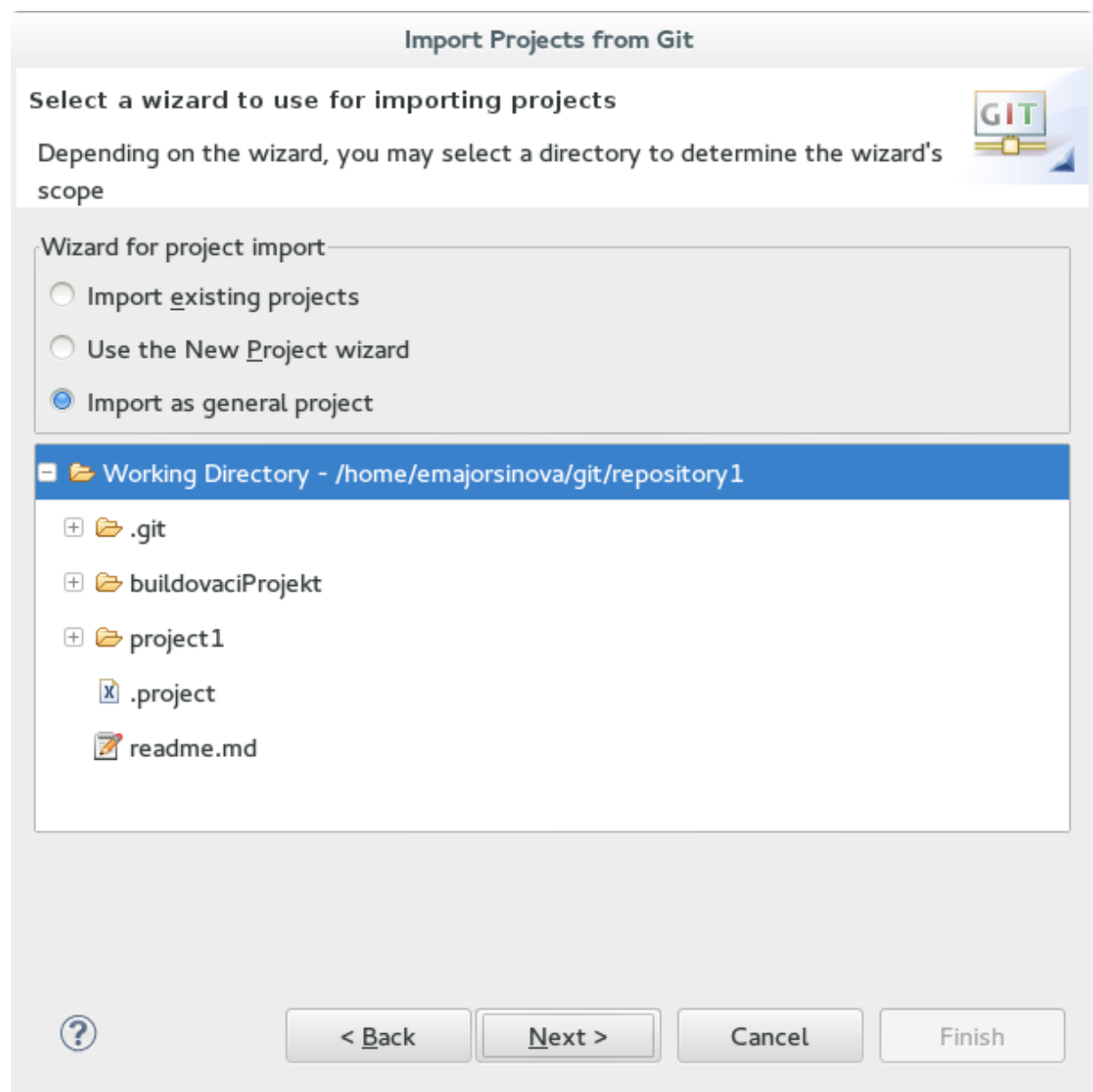
6. Select the branch you wish to import in the following window and click **Next**.
7. To define the local storage for this project, enter (or select) a non-empty directory, make any configuration changes and click **Next**.
8. Import the project as a general project in the following window and click **Next**. Name the project and click **Finish**.

Procedure: Importing Local Git Repository

1. Start the Red Hat JBoss BRMS server by selecting the server from the **Servers** tab and click the start icon.
2. In Red Hat JBoss Developer Studio, select **File → Import...** and navigate to the Git folder. Open the Git folder to select **Projects from Git** and click **Next**.
3. Select the repository source as **Existing local repository** and click **Next**.

Figure 8.2. Git Repository Details

4. Select the repository that is to be configured from the list of available repositories and click **Next**.
5. In the dialog that opens, select the radio button **Import as general project** from the **Wizard for project import** and click **Next**. Name the project and click **Finish**.

Figure 8.3. Wizard for Project Import

CHAPTER 9. BUSINESS RESOURCE PLANNER

Business Resource Planner is a lightweight, embeddable planning engine that optimizes planning problems. It helps normal Java™ programmers solve planning problems efficiently, and it combines optimization heuristics and metaheuristics with very efficient score calculations.

Planner helps solve various use cases like the following:

- *Employee/Patient Rosters.* Planner helps create timetables for nurses and keeps track of patient bed management.
- *Educational Timetables.* Planner helps schedule lessons, courses, exams, and conference presentations.
- *Shop Schedules.* Planner tracks car assembly lines, machine queue planning, and workforce task planning.
- *Cutting Stock.* Planner minimizes waste by reducing the consumption of resources such as paper and steel.

9.1. INSTALLING BUSINESS RESOURCE PLANNER

1. Navigate to the [Red Hat Customer Portal](#) and log in with your user credentials.
2. Select **DOWNLOADS** at the top of the page.
3. In the **Product Downloads** page that opens, select **Red Hat JBoss BRMS**.
4. From the **Version** drop-down menu, select version **6.3**.
5. Select **Red Hat JBoss BRMS 6.3.0 Business Resource Planner** and click **Download**.

9.2. RUNNING BUSINESS RESOURCE PLANNER EXAMPLES

1. On the command line, move into the **examples/** directory.
2. In a Unix environment, run the following command:

```
./runExamples.sh
```

In a Windows environment, run the following command:

```
./runExamples.bat
```

3. Pick an example from the Examples GUI application that opens and run it in your favorite IDE.

CHAPTER 10. PATCHING AND UPGRADING RED HAT JBOSS BRMS

10.1. ABOUT PATCHES AND UPGRADES

Red Hat JBoss BRMS patches can be either an asynchronous update, or a planned update:

- *Asynchronous updates:* Individual patches which are released outside the normal update cycle of the existing product. These may include security patches, as well as other individual patches provided by Red Hat Global Support Services (GSS) to fix specific issues.
- *Planned updates:* The cumulative patches of an existing product, which includes all previously developed updates for that version of the product.

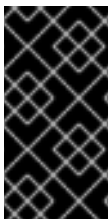
Red Hat JBoss BRMS patches can be downloaded from the [Red Hat Customer Portal](#)

The following files are included as part of a Red Hat JBoss BRMS and Red Hat JBoss BPM Suite patch release.

- Red Hat JBoss BRMS customers — **jboss-brms-*VERSION*-patch.zip**.
- Red Hat JBoss BPM Suite customers — **jboss-bpmsuite-*VERSION*-patch.zip**.
- Maven repository updates (same for both Red Hat JBoss BRMS and Red Hat JBoss BPM Suite customers) — **jboss-brms-bpmsuite-*VERSION*-incremental-maven-repository.zip**.

10.2. APPLYING PATCHES IN RED HAT JBOSS BRMS 6.3

In Red Hat JBoss BRMS, the client patching tool is distributed as a ZIP file that includes simple **.sh** and **.bat** scripts, allowing for easy and automatic application of updates to an existing Red Hat JBoss BRMS 6.1 (or better) installation.



IMPORTANT

The patching tool is for use with Red Hat JBoss BRMS 6.1 or better, and should not be used for earlier versions. For more information, see the [Maintenance Release Changes in BRMS and BPM Suite 6.1+](#) article at Red Hat Knowledgebase.

The script requires two mandatory parameters: **<path-to-distribution-root>** and **<type-of-distribution>**. For example, the following command applies the updates to the specified Red Hat JBoss EAP bundle:



NOTE

Patch updates should not be applied while you are running an instance of Red Hat JBoss BRMS. Make sure that the server is shut down before running the following command.

```
$ ./apply-updates.sh ~/EAP_HOME/jboss-eap-6.4 eap6.x
```

The following distribution types are supported:

- **eap6.x**
- **eap6.x-bc**
- **eap6.x-kie-server**
- **generic**
- **generic-bc**
- **generic-kie-server**
- **was8**
- **was8-bc**
- **was8-kie-server**
- **wls12c**
- **wls12c-bc**
- **wls12c-kie-server**
- **brms-engine**
- **planner-engine**
- **supplementary-tools**

The quickstarts and migration tool are also included in the patch and are available for download as a ZIP file.



NOTE

Only updates for Red Hat JBoss BRMS or Red Hat JBoss BPM Suite are included in the patch distribution. Patches to EAP itself must be applied using the EAP patching mechanism. See the [Red Hat JBoss EAP Installation Guide](#)

Backup Feature

Before applying any updates, the client script takes a backup of the specified distribution. It copies the distribution file or directory into the **backup/CURRENT_TIMESTAMP** subdirectory. The top-level backup directory is created at the same filesystem level as the **apply-updates** script.

Blacklist Feature

The client patching tool provides a blacklist feature that allows you to tell the script the files that must not be updated. This is a feature that helps you preserve your configuration files from being overwritten automatically by the update process. You can specify non-configuration files as well if required.

To specify the blacklisted files, open the file **blacklist.txt** present within the patch distribution. Enter the relative path to the files that must not be updated. Each file must be specified on a line by itself.

–

```
# Lines with a '#' are comment lines, like this one.
# Blank lines are ignored.

# We have made changes to the web.xml that must be preserved:
WEB-INF/web.xml

# This file has custom modifications:
styles/base.css
```

Files specified in the **blacklist.txt** file that have updated content in the patch, are not touched by the update tool. Instead, the tool copies the new, updated file in the same location and appends the new suffix to it. For example, after running the patch tool, both these files will exist in the **styles** folder, continuing with the **blacklist.txt** file in the example above.

```
$ ls styles
base.css base.css.new
```

Now, compare the contents of the two files and merge the changes.

If there are files that are no longer being distributed but you want to preserve them, put them into the **blacklist.txt** file as well. The patch update tool will not delete these files, and instead create an empty marker file with the suffix **removed**. You can then choose to either keep or delete these files manually.

Continuing with the previous example, if the **base.css** file was removed and you had this file listed in the **blacklist.txt** file, then after the patch tool has run, the contents of the styles directory would be similar to:

```
$ ls styles
base.css base.css.removed
```

10.3. PATCHING OTHER PLATFORMS AND APPLICATIONS

Use the following commands for updating other supported platforms and common applications in Red Hat JBoss BRMS.



IMPORTANT

On a Microsoft Windows system, run **./apply-updates.bat** instead of **./apply-updates.sh**.

Patch EAP 6.x Business Central WAR

```
$ ./apply-updates.sh PATH/jboss-eap-6.4/standalone/deployments/business-
central.war eap6.x-bc
```

Patch Generic KIE Server WAR

```
$ ./apply-updates.sh PATH_TO_TOMCAT_HOME/webapps/kie-server.war generic-
kie-server
```

Patch Whole WebLogic 12c Bundle

```
$ ./apply-updates.sh PATH_TO_UNZIPPED_wlsc12c_BUNDLE wls12c
```

Patch Planner Engine Bundle

```
$ ./apply-updates.sh PATH_TO_UNZIPPED_PLANNER_BUNDLE planner-engine
```

Patch IBM WebSphere Application Server Bundle

```
$ ./apply-updates.sh PATH_TO_UNZIPPED_WAS_BUNDLE was8
```



NOTE

When patching the IBM WebSphere Application Server, do *not* extract the target WAR files.

See [Section 10.2, “Applying Patches in Red Hat JBoss BRMS 6.3”](#) and [Section 10.4, “Upgrading to Latest Minor Release”](#) for more information.

10.4. UPGRADING TO LATEST MINOR RELEASE

Apart from supporting upgrade to the latest micro release, Red Hat JBoss BRMS also supports upgrading between minor releases. For example, upgrading from:

- Red Hat JBoss BRMS 6.2.2 to Red Hat JBoss BRMS 6.3.0
- Red Hat JBoss BRMS 6.1.5 to Red Hat JBoss BRMS 6.3.0

The Red Hat JBoss BRMS upgrade tool is distributed as ZIP files with naming convention that states the upgrade path. For example, **jboss-brms-6.2.2-to-6.3.0-patch.zip** is used to upgrade from 6.2.x to the 6.3.0 version. These ZIP files can be downloaded from the [Red Hat Customer Portal](#):

- Use **jboss-brms-6.2.2-to-6.3.0-patch.zip** to upgrade from Red Hat JBoss BRMS 6.2.2 to Red Hat JBoss BRMS 6.3.0.
- Use **jboss-brms-6.1.5-to-6.3.0-patch.zip** to upgrade from Red Hat JBoss BRMS 6.1.5 to Red Hat JBoss BRMS 6.3.0.

Each ZIP file contains the following scripts:

- **apply-updates.bat**
- **apply-updates.sh**

To upgrade to the next minor release using these upgrade scripts, you must specify arguments indicating the path of distribution and the type of distribution you want to upgrade in your command:

```
$ ./apply-updates.sh DISTRIBUTION_PATH DISTRIBUTION_NAME
```

For example:

```
$ ./apply-updates.sh ~/EAP_HOME/jboss-eap-6.4 eap6.x
```

The supported distribution types are:

- **eap6.x**
- **eap6.x-bc**
- **eap6.x-kie-server**
- **generic**
- **generic-bc**
- **generic-kie-server**
- **was8**
- **was8-bc**
- **was8-kie-server**
- **wls12c**
- **wls12c-bc**
- **wls12c-kie-server**
- **brms-engine**
- **planner-engine**
- **supplementary-tools**

The upgrade tool allows you to upgrade the entire distribution, or only a part of the distribution as per your requirement. For example, for the **eap6.x** distribution, you can choose to patch the entire **eap6.x** or choose to patch any of the war files (**eap6.x-bc**, **eap6.x-kie-server**) that the patch contains.

Note that the upgrade tool does not upgrade the configuration files if you have your custom updates in them. The upgrade tool checks if the configuration files have any changes. If there are no changes made to the configuration files, the tool replaces the configuration files with the latest version. However, if the tool finds custom changes made to any of the configuration files, it adds those files to blacklist, and does not replace them with the latest version. So you do not need to manually compare the configuration files and place them in the blacklist to ensure that your custom configurations are intact.



NOTE

It is recommended that you add your custom changes to the **.new** files instead of trying to update the current configuration files with changes from the new Red Hat JBoss BRMS version. For example, if you have custom changes such as data source name/location in the **persistence.xml** file, the recommended approach is to add your custom changes to the **.new** files created by the upgrade tool. Once you have updated the **.new** files with all the required changes, rename them to their original names (without the **.new** suffix). This ensures that the applications pick the updated configuration files containing your custom changes.

APPENDIX A. REVISION HISTORY

Note that revision numbers relate to the edition of this manual, not to version numbers of Red Hat JBoss BRMSBPM Suite.

Revision 6.3.0-17 Rebuilt.	Mon Mar 20 2017	Tomas Radej
Revision 6.3.0-16 Rebuilt.	Wed Feb 22 2017	Tomas Radej
Revision 6.3.0-15 Rebuilt.	Fri Dec 23 2016	Tomas Radej
Revision 6.3.0-14 Rebuilt.	Mon Nov 28 2016	Tomas Radej
Revision 6.3.0-13 Rebuilt.	Wed Oct 5 2016	Tomas Radej
Revision 6.3.0-12 Built for release 6.3.3.	Mon Oct 3 2016	Tomas Radej
Revision 6.3.0-11 Rebuilt.	Thu Sep 15 2016	Tomas Radej
Revision 6.3.0-10 Published the AsciiDoc version of the docs.	Thu Sep 15 2016	Tomas Radej
Revision 6.3.0-9 Updated Installation Guide.	Tue Aug 2 2016	Tomas Radej
Revision 6.3.0-8 Updated documentation with release 6.3.1.	Thu Jul 14 2016	Tomas Radej
Revision 6.3.0-7 Updated documentation with release 6.3.1.	Thu Jul 14 2016	Tomas Radej
Revision 6.3.0-6 Releasing the newest documentation.	Thu Jun 2 2016	Marek Czernek
Revision 6.3.0-5 Built with live links.	Thu May 5 2016	Tomas Radej
Revision 6.3.0-4 Bootstrapping links.	Thu May 5 2016	Tomas Radej
Revision 6.3.0-3 All books rebuilt.	Thu Apr 28 2016	Tomas Radej
Revision 6.3.0-2 All books rebuilt.	Thu Apr 28 2016	Tomas Radej
Revision 6.3.0-1 Initial build for release 6.3.0 of JBoss BPM SuiteJBoss BRMS.	Thu Apr 28 2016	Tomas Radej

