



Red Hat JBoss BRMS 6.1

Installation Guide

For Red Hat JBoss Administrators

Red Hat JBoss BRMS 6.1 Installation Guide

For Red Hat JBoss Administrators

Kanchan Desai
kadesai@redhat.com

Doug Hoffman

Eva Kopalova

B Long
Red Hat Engineering Content Services
belong@redhat.com

Petr Penicka

Red Hat Content Services

Gemma Sheldon
Red Hat Engineering Content Services
gsheldon@redhat.com

Legal Notice

Copyright © 2015 Red Hat, Inc.

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

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

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, 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. INSTALLATION	4
1.1. INSTALLATION OPTIONS	4
1.2. DOWNLOADING RED HAT JBOSS BRMS FOR JBOSS EAP	4
1.3. DOWNLOADING RED HAT JBOSS BRMS FOR JBOSS EAP	4
1.4. DOWNLOADING RED HAT JBOSS BRMS FOR JBOSS EAP	5
1.5. INSTALLING RED HAT JBOSS BRMS USING THE INSTALLER	5
1.6. INSTALLING RED HAT JBOSS BRMS USING THE INSTALLER IN CLI MODE	13
1.7. INSTALLING RED HAT JBOSS BRMS FOR RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM	17
1.8. INSTALLING RED HAT JBOSS BRMS ON RED HAT JBOSS WEB SERVER	18
1.9. INSTALLING RED HAT JBOSS BRMS IN THE DOMAIN MODE	22
1.10. DEFINING ROLES	25
1.11. CREATING USERS	25
CHAPTER 2. GIT	27
2.1. CLONING AN EXISTING REPOSITORY	28
2.2. MIGRATING A REPOSITORY FROM RED HAT JBOSS BRMS 5.3	29
CHAPTER 3. AUTHENTICATION	31
CHAPTER 4. TESTING THE INSTALLATION	32
4.1. STARTING THE SERVER	32
4.2. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT	33
4.3. LOGGING ON TO BUSINESS CENTRAL	34
CHAPTER 5. CLUSTERING	35
GIT REPOSITORY CLUSTERING MECHANISM	35
CLUSTERING MAVEN REPOSITORIES	35
5.1. SETTING UP A CLUSTER	35
5.2. CONFIGURING CLUSTERING ON RED HAT JBOSS EAP	38
CHAPTER 6. MAVEN REPOSITORIES	43
6.1. ABOUT MAVEN	43
6.2. ABOUT THE PROVIDED MAVEN REPOSITORIES	43
6.3. CONFIGURING MAVEN TO USE THE FILE SYSTEM REPOSITORIES	43
6.4. CONFIGURING MAVEN TO USE THE ONLINE REPOSITORIES	47
6.5. DEPENDENCY MANAGEMENT	51
CHAPTER 7. RED HAT JBOSS DEVELOPER STUDIO	52
7.1. RED HAT JBOSS DEVELOPER STUDIO	52
7.2. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS	52
7.3. SETTING THE DROOLS RUNTIME	52
7.4. CONFIGURING THE JBOSS BRMS SERVER	53
7.5. IMPORTING PROJECTS FROM A GIT REPOSITORY INTO JBOSS DEVELOPER STUDIO	53
CHAPTER 8. BUSINESS RESOURCE PLANNER	57
8.1. INSTALLING BUSINESS RESOURCE PLANNER	57
8.2. RUNNING THE BUSINESS RESOURCE PLANNER EXAMPLES	57
CHAPTER 9. PATCHING AND UPGRADING RED HAT JBOSS BPM SUITE	58
9.1. ABOUT PATCHES AND UPGRADES	58
9.2. APPLYING PATCHES IN RED HAT JBOSS BRMS 6.1	58
9.3. PATCHING OTHER PLATFORMS AND APPLICATIONS	60

APPENDIX A. REVISION HISTORY 61

CHAPTER 1. INSTALLATION

1.1. INSTALLATION OPTIONS

Red Hat JBoss BRMS comes in two versions:

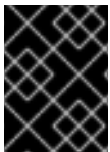
- Executable jar installer for installation on Red Hat JBoss Enterprise Application Platform (EAP) 6.4.
- Zip file install which itself comes in two versions:
 - **jboss-brms-6.MINOR_VERSION-deployable-eap6.x.zip**: version adapted for deployment on Red Hat JBoss Enterprise Application Platform (EAP 6).
 - **jboss-brms-6.MINOR_VERSION-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 JBoss BRMS 6.1 onwards, you must have JBoss EAP 6.4 or better already installed before attempting to install JBoss BRMS.

1.2. DOWNLOADING RED HAT JBOSS BRMS FOR JBOSS EAP

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **Downloads** → **Products Downloads**.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BRMS**.
4. From the **Version** drop-down menu, select version 6.1.
5. Select **Red Hat JBoss BRMS 6.1 Deployable for EAP 6.4** and then click **Download**.

1.3. DOWNLOADING RED HAT JBOSS BRMS FOR JBOSS EAP

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **Downloads** → **Products Downloads**.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BRMS**.
4. From the **Version** drop-down menu, select version 6.1.

5. Select **Red Hat JBoss BRMS 6.1 Deployable for EAP 6.4** and then click **Download**.

1.4. DOWNLOADING RED HAT JBOSS BRMS FOR JBOSS EAP

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **Downloads** → **Products Downloads**.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BRMS**.
4. From the **Version** drop-down menu, select version 6.1.
5. Select **Red Hat JBoss BRMS 6.1 Deployable for EAP 6.4** and then click **Download**.

1.5. INSTALLING RED HAT JBOSS BRMS USING THE INSTALLER

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



NOTE

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

Prerequisite

Before attempting to install JBoss BRMS, ensure you have already installed Red Hat JBoss EAP 6.4 or better.

1. Setup 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-6.1.0.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-6.1.0.GA-installer.jar**

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

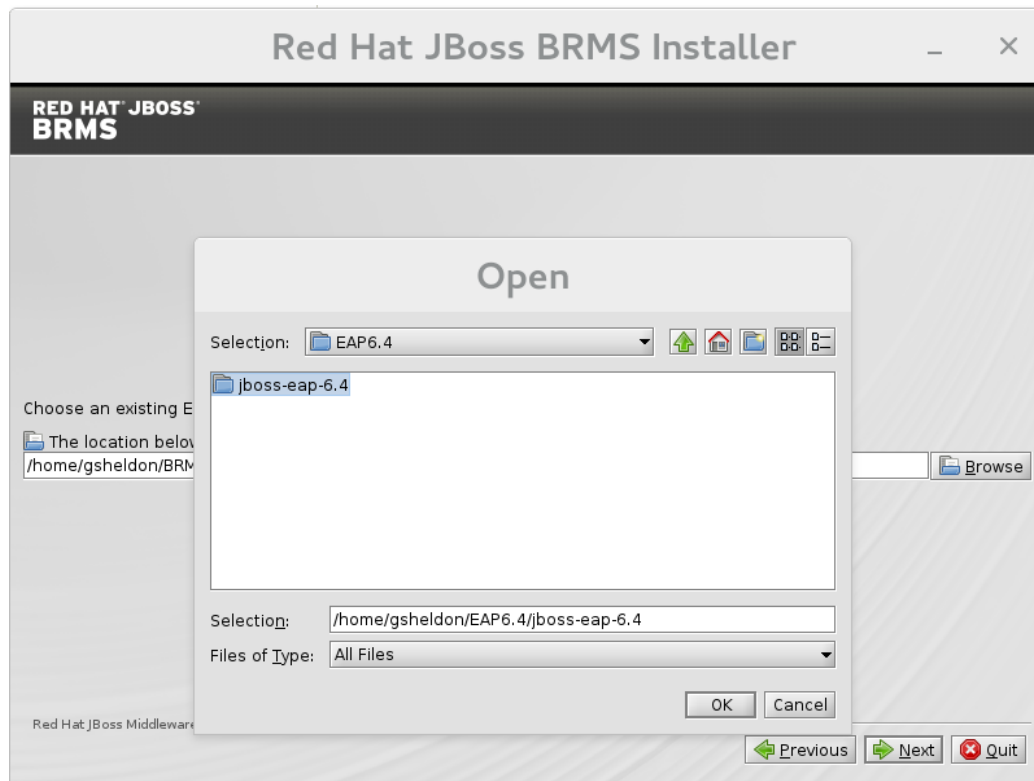


Figure 1.1. Red Hat JBoss BRMS Installer for EAP Directory Path

4. In the next two screens, create two users: the first one for the management console of the EAP (ManagementRealm) and the second one for managing JBoss BRMS itself (ApplicationRealm).

Creation of the first user for the management console of JBoss EAP is optional and you may skip it if it is not required.

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



NOTE

The username that you create should *not* be the same as any of the pre-defined roles (See [Section 1.10, “Defining Roles”](#)).

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. Setup Security Environment

Next, you will setup the security environment of your new 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 setup pure IPv6 configuration on the server that the installation is taking place. This will allow you to setup runtime IPv6 specific configurations later.

7. Configure Runtime Environment

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

o Default Configuration

Choose 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.

o 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 via the check boxes.

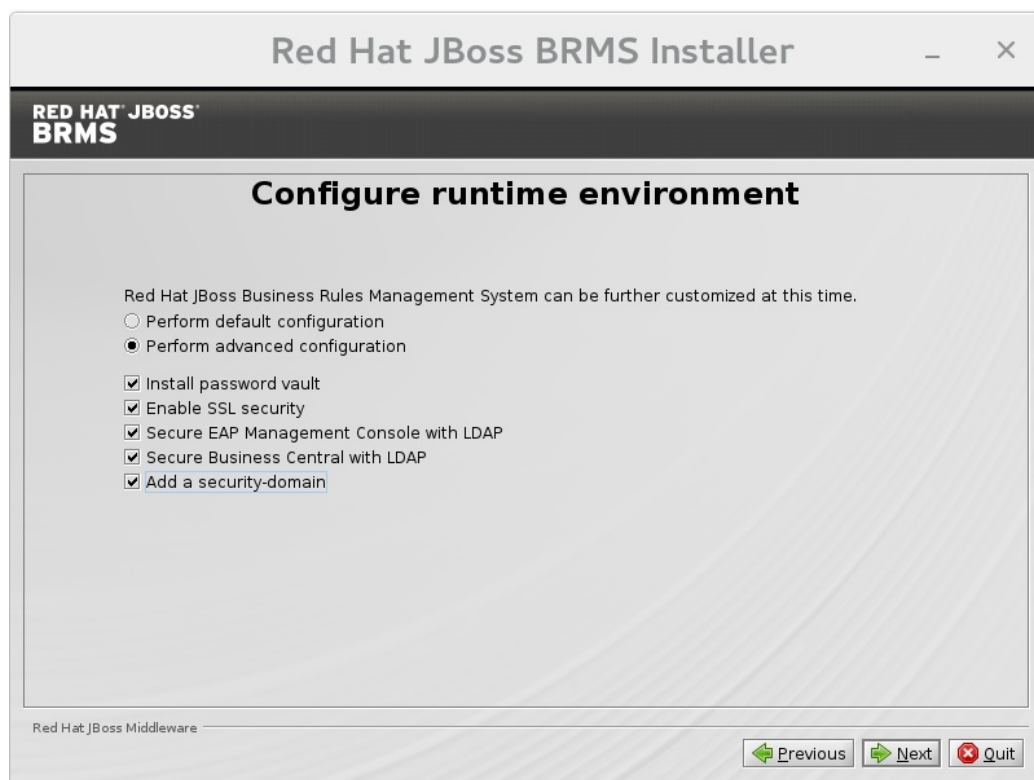
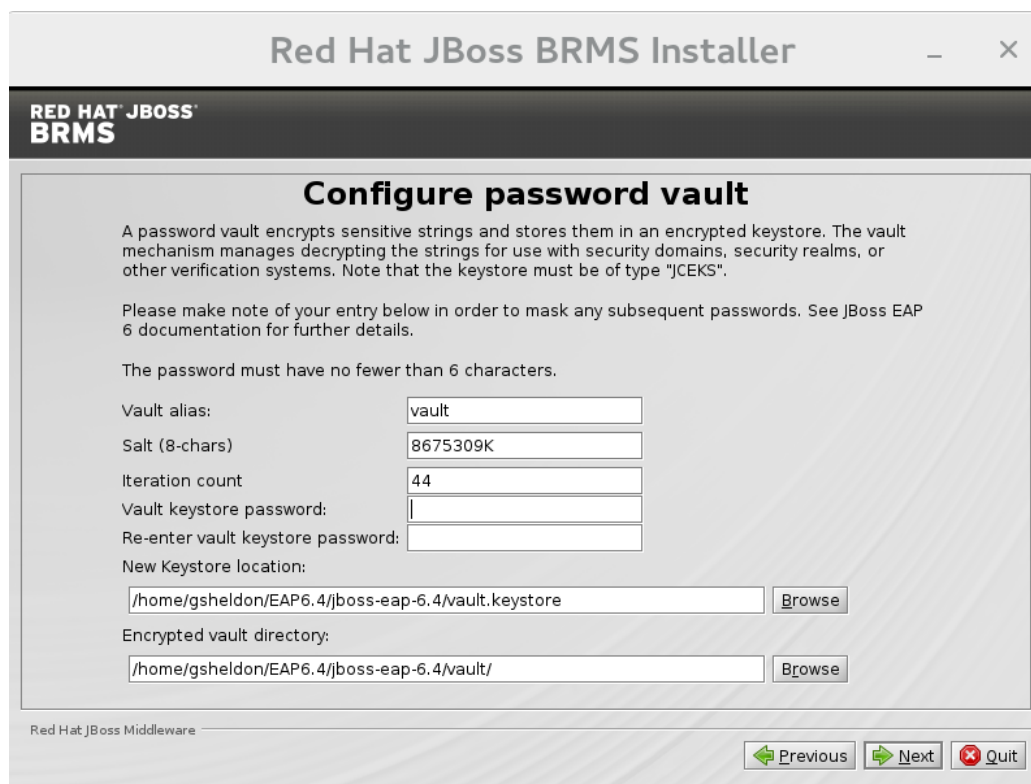


Figure 1.2. Advanced Configuration Options

■ Configure Vault Password

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*.



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

Figure 1.3. Configure vault password

■ 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 username / password.

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



Figure 1.4. SSL Security Configuration

■ LDAP Security

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.

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



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:

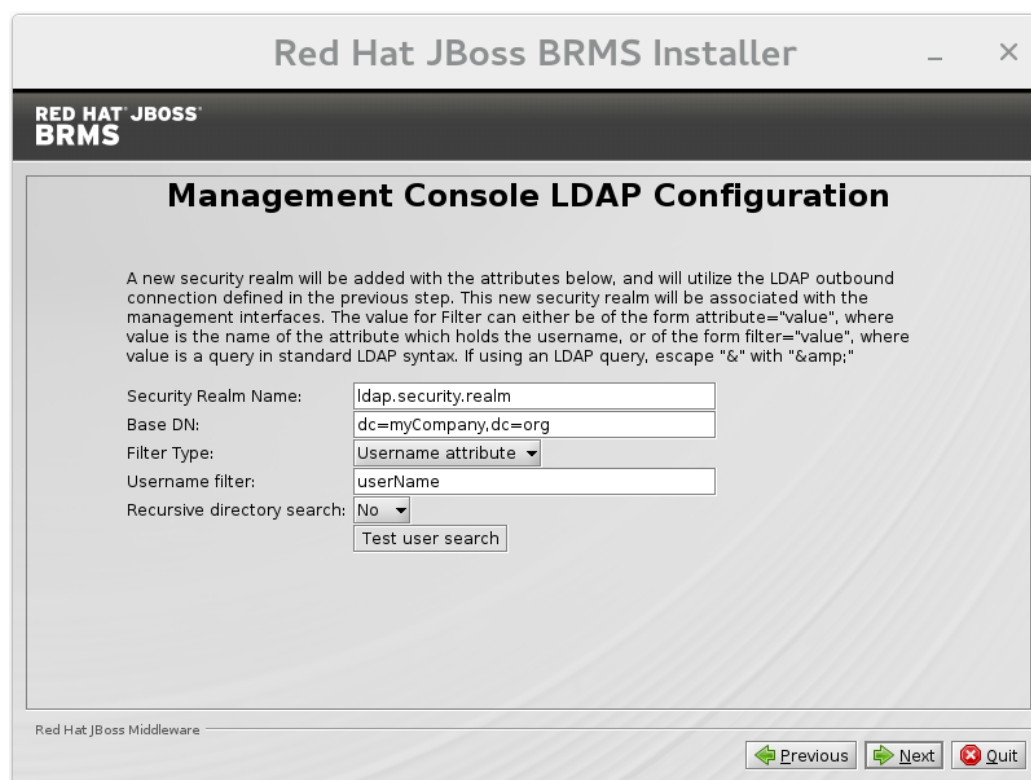
Confirm DN Password:

Red Hat JBoss Middleware

Figure 1.5. LDAP Connection Configuration

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.



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

Figure 1.6. Management Console LDAP Configuration

- **Base DN:** Will typically define a 'base search' or 'root context' to begin searching for users.
- **Filter type:** Tells JBoss EAP how to find the LDAP attribute that defines a user; it is can be a simple attribute, but can also be a complex "LDAP Filter".
- **Username attribute:** The LDAP attribute which holds the username values. A username 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, 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 Username 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*.

Red Hat JBoss BRMS Installer

RED HAT JBOSS BRMS

Business Central LDAP Configuration

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.

User context:

User filter:

User roles context:

User roles filter:

Role context:

Role filter:

Role attribute id:

Role name id:

☐ Role attribute is distinguished name

Red Hat JBoss Middleware

Figure 1.7. Business Central LDAP Configuration

■ Security Domain and JSSE

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*.



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 1.8. Security Domain



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

Figure 1.9. JSSE Configuration

- The installer will go through the steps to install JBoss BRMS and will perform post installation configuration steps when you click **next**. The installer will also start the 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.

1.6. 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 JBoss BRMS.

Prerequisite

Before attempting to install JBoss BRMS, ensure you have already installed Red Hat JBoss EAP 6.4 or better.

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

```
java -jar jboss-brms-6.1.0.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 EAP installation.

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

```
[/home/user/BRMS-6.1.0/jboss-eap-6.4]
```

The installer will verify the location of the 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 then create a 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]
```

8. Create and confirm a password for the 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:

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

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

- `[] Secure EAP Management Console with LDAP`
Input 1 to select, 0 to deselect:
- `[] Secure Business Central with LDAP`
Input 1 to select, 0 to deselect:
- `[] Add a security-domain`
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-6.1.0/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-
6.1.0/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-6.1.0.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 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. Login with the correct username/password as given to the JBoss BRMS user in the "Create and confirm a password for the JBoss BRMS user" step.

1.7. INSTALLING RED HAT JBOSS BRMS FOR RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

Red Hat JBoss BRMS can be installed on JBoss Enterprise Application Platform 6.4.

Installation on a fresh EAP instance

To install the deployable package for a Red Hat JBoss Enterprise Application Platform that has yet to be configured, do the following:

1. Move the downloaded zip archive to the parent directory of the Red Hat JBoss Enterprise Application Platform home directory (*EAP_HOME*; the **jboss-eap-6.4** directory).
2. Unzip the downloaded zip archive: make sure it is merged into the *EAP_HOME* directory (**jboss-eap-6.4**).



WARNING

This step must be performed by the same user account that was used to install EAP. This account must not be a superuser account.

3. It is necessary to overwrite the files that already exist in the *EAP_HOME* directory with their versions from the downloaded zip archive. When prompted to do so, accept overwriting the original files.

Installation on an existing EAP configuration

**WARNING**

These instructions are for installing, and *NOT* for updating an existing JBoss BRMS instance. Make sure that there is no existing JBoss BRMS install in the target EAP.

To install the deployable package for a previously configured Red Hat JBoss Enterprise Application Platform, do the following:

1. Download the zip archive and prepare to manually merge files into the Red Hat JBoss Enterprise Application Platform home directory (*EAP_HOME*; the **jboss-eap-6.4** directory).
2. Unzip the downloaded zip archive; however, do not overwrite all of the files. Manually merge the following files into the *EAP_HOME* directory (**jboss-eap-6.4**):
 - **jboss-eap-6.4/domain/configuration/***
 - **jboss-eap-6.4/standalone/configuration/***
 - **jboss-eap-6.4/modules/layers.conf**
 - **jboss-eap-6.4/bin/product.conf**

**WARNING**

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

3. Ensure the target EAP does not include a deployment with a colliding name. Copy the folder **jboss-eap-6.4/standalone/deployments** into the *EAP_HOME* directory from the JBoss BRMS distribution.
4. Make sure no EAP module layer is already called JBoss BRMS and copy the folder **jboss-eap-6.4/modules/system/layers/brms** into the EAP 6.4 folder.

1.8. 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 6.1 to an existing application server. The following procedure provides instructions for installation on an existing Red Hat JBoss Web Server 2.1.0 instance.

Procedure 1.1. Installing the Generic Deployable Package

1. Download and Extract

To download the generic deployable package zip file from the Red Hat Customer Support Portal, go to <https://access.redhat.com> and log in.

2. Click **Downloads**.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BRMS**.
4. From the **Version** drop-down menu, select version 6.1
5. Select **Red Hat JBoss BRMS 6.1 Deployable for All Supported Containers** package and then click **Download**.

Also select and download the Red Hat JBoss BRMS Engine files.

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** folders.
8. Move the **kie-tomcat-integration-VERSION.jar** file from **business-central/WEB-INF/lib** in JBoss BRMS distribution to **tomcat7/lib**.

9. Setup Users

Define the users and the roles in **tomcat7/conf/tomcat-users.xml** as shown below. Make sure that username and roles do not conflict. For example, you should not create a user with the username of **admin** as that is a defined role. See [Section 1.10, “Defining Roles”](#) for a list of defined roles.

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

10. 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:

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

- `slf4j-jdk14-VERSION.jar`

In addition, download the following library and copy it into the **`$TOMCAT_DIR/lib/`** folder as well:

- [javax.security.jacc-api.jar](#)

11. 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 datasource name used in tomcat resource definition later - make a note of this value).

Example 1.1. H2 datasource 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 1.2. MySQL 5.5 datasource 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 1.3. DB2 Type 4 datasource 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 1.4. Oracle datasource 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 1.5. Microsoft SQL Server datasource 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. 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" />

```

13. Define the btm.root system property and location where bitronix config file is placed in:

\$TOMCAT_DIR/bin/, create a readable **setenv.sh** file with the following content:

```

CATALINA_OPTS="-Xmx512M -XX:MaxPermSize=512m -
Dbtm.root=$CATALINA_HOME -
Dbitronix.tm.configuration=$CATALINA_HOME/conf/btm-config.properties

```

```
-  
Djbpm.tsr.jndi.lookup=java:comp/env/TransactionSynchronizationRegist  
ry"
```

14. Install the Driver to Your Database

Copy the jar file with the relevant database driver to **\$TOMCAT_DIR/lib/**.



NOTE

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

15. Start Server and Test

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.

16. Navigate to <http://localhost:8080/business-central> in a web browser.

17. Login with the username/password defined in the **tomcat-users.xml** file.

1.9. INSTALLING RED HAT JBOSS BRMS IN THE DOMAIN MODE

These are the steps to install Red Hat JBoss BRMS 6 for EAP 6 deployable in the domain mode.

1. Download and extract the Red Hat JBoss BRMS 6.1.0 Deployable for EAP 6 ZIP file from [Red Hat Customer Portal](#) and copy the following directories into the installation of EAP 6.4:

- o **bin**
- o **domain**

Skip the **standalone** directory.

2. On the command line, move to the **/bin** directory and start the domain as follows:

In a Unix environment, run:

```
./domain.sh
```

In a Windows environment, run:

```
./domain.bat
```

3. Deploy the archive either via **\${jboss-eap-home}/bin/jboss-cli.sh / \${jboss-eap-home}/bin/jboss-cli.bat**, or via management web UI (localhost:9990/):



NOTE

The web application **business-central.war** supplied in the EAP deployable binaries in the **/standalone/deployments** directory is a directory, but for deployment into the domain, you have to use a WAR archive. To create it, zip the content of the **business-central.war** directory.

- a. To deploy the archive via **`${jboss-eap-home}/bin/jboss-cli.sh`** or **`${jboss-eap-home}/bin/jboss-cli.bat`**, move into the **`${jboss-eap-home}/bin`** directory and deploy the WAR file:

In a Unix environment, run:

```
./jboss-cli.sh
```

In a Windows environment, run:

```
./jboss-cli.bat
```

Then run:

```
deploy location_of_business-central.war_file
```

- b. To deploy the archive via management web UI (localhost:9990/):

- to create an EAP management account, add a Management User from the **/bin** directory as follows.

On a Unix system, run:

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

In a Windows environment, run:

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

(refer to [Section 1.11, “Creating users”](#) and [Section 1.10, “Defining Roles”](#)).

- log in using your EAP management account
- select **Domain -> Manage Deployments -> Content Repository -> Add**
- select the web archive from the file system, upload the web archive
- select the deployment, click the **Assign** button
- select the server group



NOTE

In order to log in to 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 the [Section 1.11, “Creating users”](#) section.

Installing Multiple JBoss BRMS Server Instances

In many situations, users may want to group together a set of EAP 6 nodes on the same machine and give them a meaningful name for easy maintenance. Unique values need to be incorporated for the system properties for each server instance. Listed below are the common properties that can be specified with a single JBoss BRMS node to change the default configuration; however, they should be specified for multiple nodes running on a single machine so every node can point to a different directory:

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

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

- **org.uberfire.nio.git.daemon.host** - can be left on default to bind to localhost.
- **org.uberfire.nio.git.daemon.port**
- **org.uberfire.nio.git.ssh.host** - can be left on default to bind to 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.

Incorporate the previous properties in the **\$EAP_HOME/domain/configuration/host.xml** file as illustrated in the two nodes below:

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 **.niogit** files. These files, which should be used by a respective node, would then be grouped in a particular domain.

1.10. DEFINING ROLES

Before starting the server and logging onto 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. Admins 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.



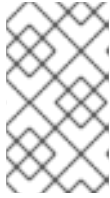
NOTE

Enter the above mentioned roles during the user creation process.

1.11. 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 EAP bin directory.

1. Run `./add-user.sh` on a Unix system or `add-user.bat` on a Windows system from the bin directory.
2. Enter `b` to select an Application User at the type of user prompt and press Enter.
3. Accept the default Realm (ApplicationRealm): by pressing Enter.
4. At the username prompt, enter a user name and confirm. For example: **helloworlduser**.

**NOTE**

Make sure that the usernames don't conflict with any known groups. For example, if there is a group called **admin**, you should not create a user with the username **admin**.

5. Create the user's 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 (e.g. A-Z, a-z), at least one numerical character (e.g. 0-9) and at least one special character (e.g. ~ ! @ # \$ % ^ * () - _ + =).

6. Enter a comma separate list of roles the user will need at the roles prompt (refer to [Section 1.10, "Defining Roles"](#)).

Business Central users need to have the **analyst** or the **admin** role.

7. Confirm you want to add the user.
8. Enter yes at the next prompt (this is to enable clustering in the future if required).

CHAPTER 2. 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 2.1. 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 (e.g. 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 via 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.
Index	Index is an alternative term for the staging area.

Term	Definition
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 e.g. 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 JBoss Developer Studio (refer to [Section 7.5, “Importing Projects from a Git Repository into JBoss Developer Studio”](#)).

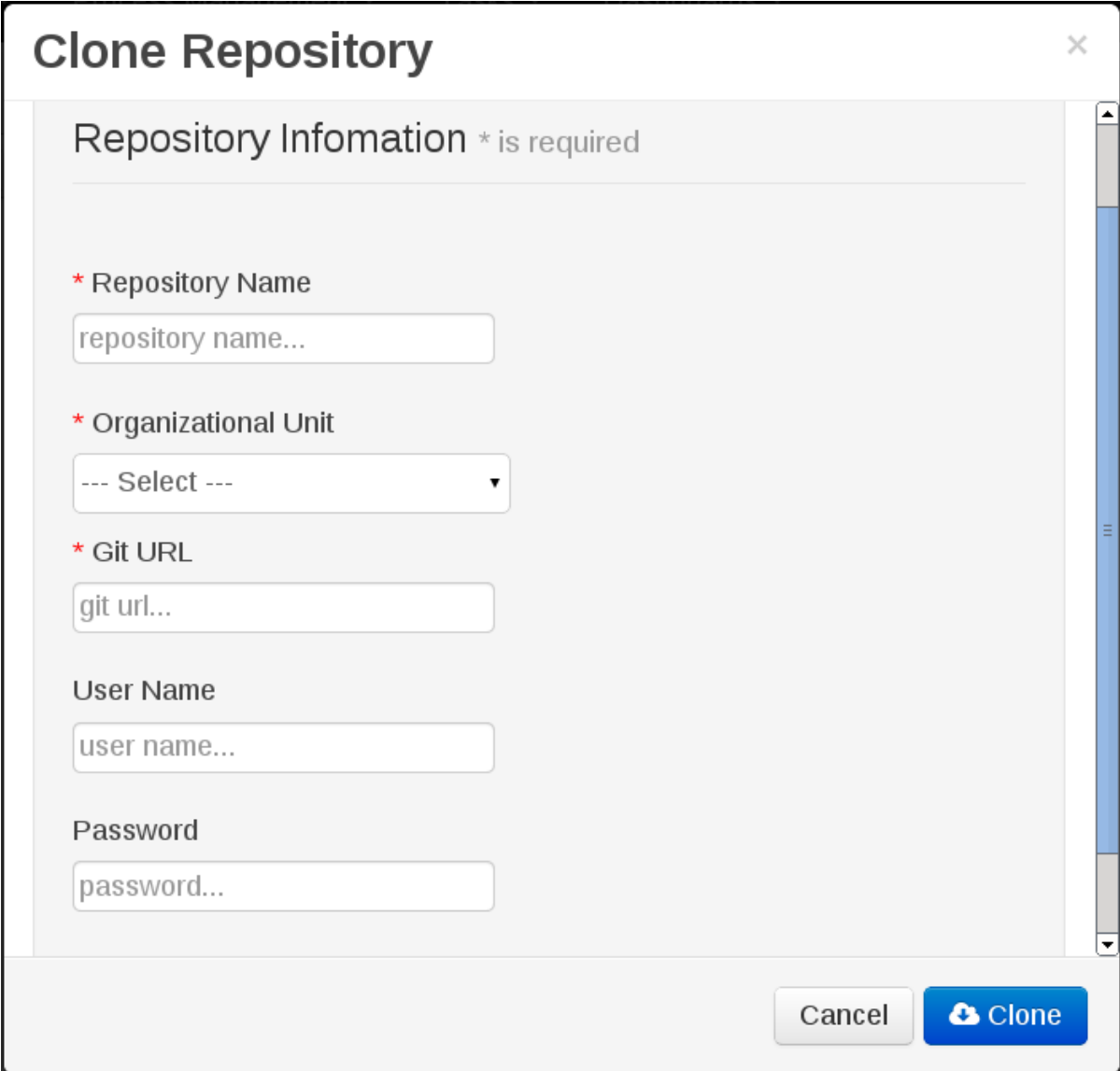
2.1. CLONING AN EXISTING REPOSITORY

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

Procedure 2.1. Cloning a repository

1. Open the **Administration** perspective: on the main menu, click **Authoring** → **Administration**.
2. On the perspective menu, click **Repository** → **Clone repository**.

3. The **Clone Repository** pop-up window is displayed.



The image shows a 'Clone Repository' pop-up window. At the top, the title 'Clone Repository' is displayed with a close button (X) on the right. Below the title, a section titled 'Repository Information * is required' contains several input fields. The first three fields are marked with a red asterisk, indicating they are mandatory: 'Repository Name' (text input), 'Organizational Unit' (dropdown menu), and 'Git URL' (text input). Below these are two more fields: 'User Name' (text input) and 'Password' (text input). At the bottom right of the window, there are two buttons: a 'Cancel' button and a 'Clone' button with a cloud icon.

Figure 2.1. Clone Repository Pop-up

4. Enter the mandatory details:
 - Repository name.
 - Select an organizational unit in which the repository is to be created from the **Organizational Unit** drop-down option.
 - Enter the GIT URL.
 - Enter Username and Password.
5. Click **Clone**

2.2. MIGRATING A REPOSITORY FROM RED HAT JBOSS BRMS 5.3

To migrate data from 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 3. AUTHENTICATION

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

CHAPTER 4. TESTING THE INSTALLATION

4.1. STARTING THE SERVER

If you have installed Red Hat JBoss BRMS using the JBoss EAP 6 bundle install, you can now start your server in one of two modes.



WARNING

In addition to the previous install steps, we strongly recommend to set the system property **org.kie.tx.lock.enabled** to **false** in order to prevent unresponsiveness or deadlock issues. You can either start your server with the option **-Dorg.kie.tx.lock.enabled=false** or edit the **standalone.xml** file:

```
<system-properties>
  <property name="org.kie.tx.lock.enabled" value="false"/>
  ...
</system-properties>
```

For further details, please refer to this article:
<https://access.redhat.com/solutions/1610723>.



NOTE

If you installed JBoss BRMS using the generic deployable version on Red Hat Java Web Server, the instructions for download and install also contain the instructions for starting the server. You can ignore the following discussion.

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 **\$SERVER_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 4.2, “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 **\$SERVER_HOME/bin/** directory.
2. In a Unix environment run:

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

In a Windows environment run:

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

**NOTE**

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

**NOTE**

If you are starting the server in the domain mode, the corresponding scripts are **domain.sh** and **domain-secure.sh** respectively.

4.2. 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 JBoss BRMS application. Use the following guidelines to deploy secure and high performance 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.

4.3. LOGGING ON TO 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**.

CHAPTER 5. CLUSTERING

When clustering Red Hat JBoss BRMS, consider which components need to be clustered. You can cluster the following:

- GIT repository: virtual-file-system (VFS) repository that holds the business assets so that all cluster nodes use the same repository
- Web applications: The web applications need to be clustered so that nodes share the same runtime data.

For instructions on clustering the application, refer to the container clustering documentation.

GIT REPOSITORY CLUSTERING MECHANISM

To cluster the GIT repository the following is used:

- Apache Zookeeper brings all parts together.
- Apache Helix is the cluster management component that registers all cluster details (the cluster itself, nodes, resources).
- uberfire framework which provides the backbone of the web applications

A typical clustering setup involves the following:

- Setting up the cluster itself using Zookeeper and Helix
- Configuring clustering on your container (this documentation provides only clustering instructions for Red Hat JBoss EAP 6)

CLUSTERING MAVEN REPOSITORIES

Various operations within the Business Central publish JARs to the Business Central's internal Maven Repository.

This repository exists on the application server's 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 alternate 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 this case clustering of the Maven repository folder is not needed.

5.1. SETTING UP A CLUSTER

To cluster your GIT (VFS) repository in Business Central, do the following (If you don't use Business Central, you may skip this section):

1. Download the **jboss-bpmsuite-brms-*VERSION*-supplementary-tools.zip**, which contains Apache Zookeeper, Apache Helix, and quartz DDL scripts. After downloading, unzip the archive: the **Zookeeper** directory (**\$ZOOKEEPER_HOME**) and the **Helix** directory (**\$HELIX_HOME**) are created.
2. Now Configure ZooKeeper:
 - a. In the ZooKeeper directory, go to **conf** directory and do the following:

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

- b. Open **zoo.cfg** for editing and adjust the settings including the following:

```
# the directory where the snapshot is stored.
dataDir=$ZOOKEEPER_HOME/data/
# the port at which the clients connects
clientPort=2181
server.1=server1:2888:3888
server.2=server2:2888:3888
server.3=server3:2888:3888
```

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" respectively for node 1, node 2 and node 3 respectively. ZooKeeper should have an odd number of instances, at least 3 in order to recover from failure.

The node ID is specified in a field called **myid** under the data directory of ZooKeeper on each node. For example, on node 1, run: **\$ echo "1" > /zookeeper/data/myid**

3. Set up ZooKeeper, so you can use it when creating the cluster with Helix:

- a. Go to the **\$ZOOKEEPER_HOME/bin/** directory 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.

4. Once the ZooKeeper ensemble is started, the next step is to configure and start Helix. Helix only needs to be configured once and from a single node. The configuration is then stored by the ZooKeeper ensemble and shared as appropriate.

Set up 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 CLUSTER_NAME
```

- b. Add your nodes to the cluster:

```
$HELIX_HOME/bin/helix-admin.sh --zkSvr
ZOOKEEPER_HOST:ZOOKEEPER_PORT --addNode CLUSTER_NAME
NODE_NAMEUNIQUE_ID
```

Example 5.1. Adding two cluster nodes

```
./helix-admin.sh --zkSvr server1:2181,server2:2181,server3:2181
--addNode brms-cluster nodeOne:12345
./helix-admin.sh --zkSvr server1:2181,server2:2181,server3:2181
```



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

5. Add resources to the cluster.

Example 5.2. Adding vfs-repo as resource

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

6. Rebalance the cluster with the three nodes.

Example 5.3. Rebalancing the brms-cluster

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

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

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

Example 5.4. Starting the Helix controller

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



NOTE

Zookeeper should an odd number of instances, at least 3 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.

Stopping Helix and Zookeeper

To stop Helix processes and the Zookeeper server, use the following procedure.

Procedure 5.1. Stopping Helix and Zookeeper

1. Stop JBoss EAP server processes.
2. Stop the Helix process that has been created by **run-helix-controller.sh**, for example, **kill -15 <pid of HelixControllerMain>**.
3. Stop ZooKeeper server using the **zkServer.sh stop** command.

5.2. CONFIGURING CLUSTERING ON RED HAT JBOSS EAP

The information provided in this section is a simple clustering recipe. For additional information on clustering refer to *Red Hat JBoss EAP documentation*.

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

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

1. Configure individual server nodes in the **main-server-group** element in the **`$EAP_HOME/domain/configuration/host.xml`** file with properties defined in [Table 5.1](#), “Cluster node properties”:

Note that a when configuring a JBoss EAP cluster with Zookeeper, a different number of JBoss EAP nodes than Zookeeper nodes is possible (keeping in mind that Zookeeper should to have an odd number of nodes). However, having the same node count for both Zookeeper and JBoss EAP is considered best practice.

Table 5.1. Cluster node properties

Property name	Value	Description
<code>org.uberfire.nio.git.dir</code>	<code>/home/jbrm/node[N]/repo</code>	GIT (VFS) repository location on node[N]
<code>jboss.node.name</code>	<code>nodeOne</code>	node name unique within the cluster
<code>org.uberfire.cluster.id</code>	<code>brms-cluster</code>	Helix cluster name
<code>org.uberfire.cluster.zk</code>	<code>server1:2181</code>	Zookeeper location
<code>org.uberfire.cluster.local.id</code>	<code>nodeOne_12345</code>	unique ID of the Helix cluster node Note that <code>:</code> is replaced with <code>_</code> .
<code>org.uberfire.cluster.vfs.lock</code>	<code>vfs-repo</code>	name of the resource defined on the Helix cluster
<code>org.uberfire.nio.git.daemon.port</code>	<code>9418</code>	port used by the VFS repo to accept client connections The port must be unique for each cluster member.
<code>org.uberfire.metadata.index.dir</code>	<code>/home/jbrm/node[N]/index</code>	location where the index for search is to be created (maintained by Apache Lucene)

Property name	Value	Description
org.uberfire.nio.git.ssh.port	8003	the unique port number for ssh access to the GIT repo for a cluster running on physical machines.
org.uberfire.nio.git.daemon.host	nodeOne	the name of the daemon host machine in a physical cluster.
org.uberfire.nio.git.ssh.host	nodeOne	the name of the SSH host machine in a physical cluster.
org.uberfire.nio.git.ssh.hostport and org.uberfire.nio.git.daemon.hostport	8003 and 9418	In a virtualized environment, the outside port to be used.

Example 5.5. 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.cluster.autostart" value="true"
boot-time="true"/>

  <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 5.6. 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,server2:2181,server3:2181" 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.cluster.autostart" value="true"
boot-time="true"/>

  <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 5.7. 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,server2:2181,server3:2181" 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.cluster.autostart" value="true"
boot-time="true"/>

    <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>

```

2. 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*.
3. Start the application server:

```
]$ $JBOSS_HOME/bin/domain.sh
```

4. Check that the nodes are available.

Deploy the Business Central application to your servers:

- Log on 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 via 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**.

CHAPTER 6. MAVEN REPOSITORIES

6.1. 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.

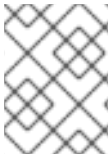
Maven uses repositories to store Java libraries, plug-ins, and other build artifacts. Repositories can be either local or remote. A local repository is a download of artifacts from a remote repository cached on a local machine. A remote repository is any other repository accessed using common protocols, 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 [Welcome to Apache Maven](#).

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

For more information about Maven POM files, see the [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.

6.2. ABOUT THE 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. If you provided the location of Maven's **settings.xml** file during installation, Maven is already configured to use the online 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 <https://access.redhat.com/site/maven-repository>.

6.3. CONFIGURING MAVEN TO USE THE FILE SYSTEM REPOSITORIES

Overview

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

Procedure 6.1.

1. Download the following ZIP archives containing the required repositories:
 - o <https://access.redhat.com/jbossnetwork/restricted/listSoftware.html?product=brms&version=6.1.0>
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 JBoss
    BRMS/JBoss BPM Suite -->
    <profile>
      <id>brms-bpms-local-profile</id>
      <repositories>
        <repository>
          <id>jboss-brms-bpmsuite-repository</id>
          <name>BRMS/BPMS 6.1.0 GA Repository</name>
          <url>file://<!-- path to the repository -->/jboss-brms-
bpmsuite-6.1.0.GA-redhat-2-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 6.1.0 GA Repository</name>
          <url>file://<!-- path to the repository -->/jboss-brms-
bpmsuite-6.1.0.GA-redhat-2-maven-repository/maven-repository</url>
          <layout>default</layout>
          <releases>
            <enabled>true</enabled>
            <updatePolicy>never</updatePolicy>
          </releases>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
```



```

        <snapshots>
          <enabled>>false</enabled>
          <updatePolicy>never</updatePolicy>
        </snapshots>
      </pluginRepository>
    </pluginRepositories>
  </profile>
</profiles>

  <!-- 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>
  -->

  <activeProfiles>
    <!-- Activation of the JBoss BRMS/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.

Troubleshooting

Q: Why do I still get errors when building or deploying my applications?

A: Issue

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
```

Cause

Your cached local Maven repository might contain outdated artifacts.

Resolution

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.

Q: Why is JBoss Developer Studio using my old Maven configuration?

A: Issue

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

Cause

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

Resolution

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 JBoss Developer Studio.

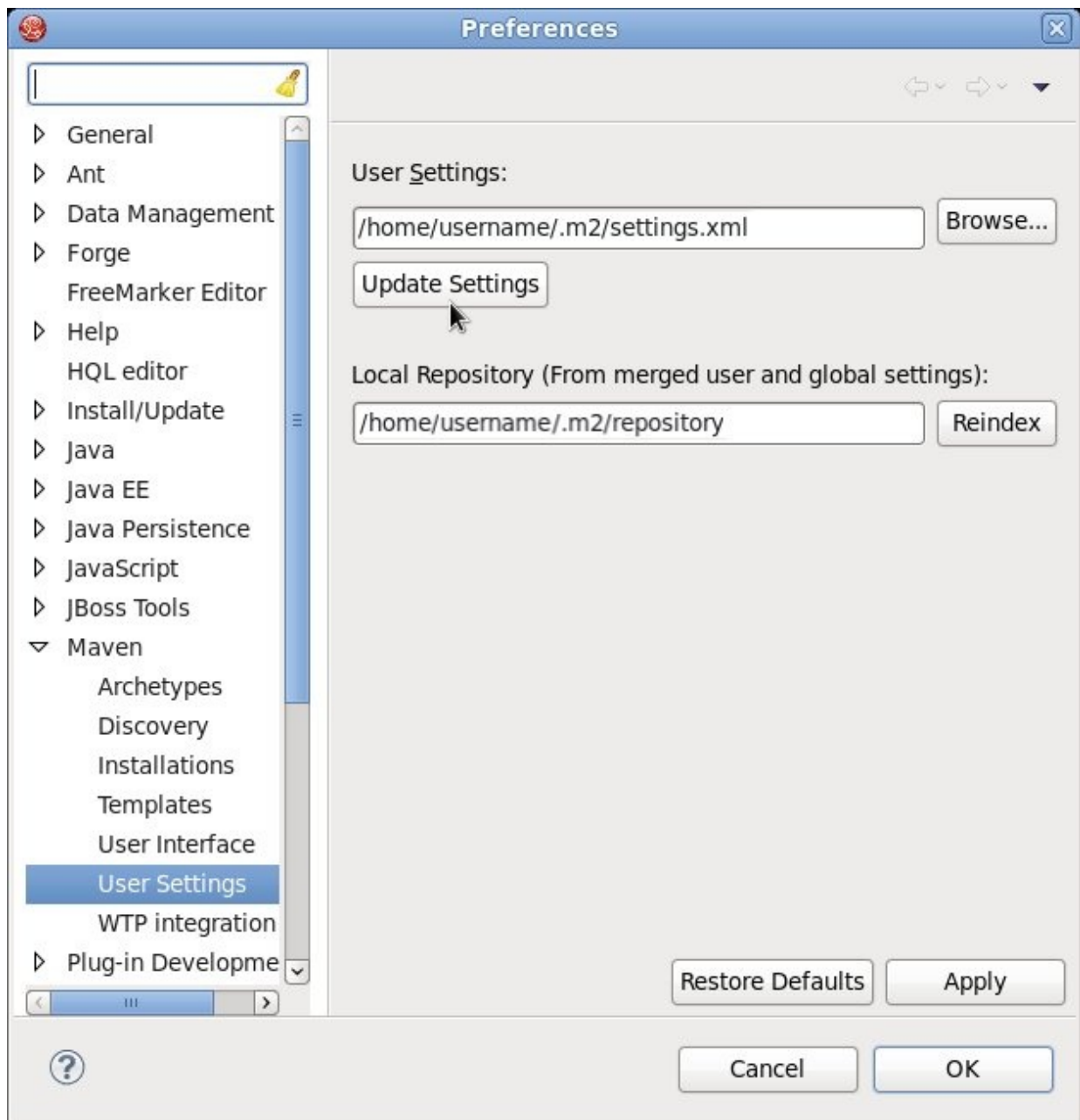


Figure 6.1. Update Maven User Settings

6.4. CONFIGURING MAVEN TO USE THE ONLINE REPOSITORIES

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

If you did not configure the Maven repository during installation, you can configure it using the following procedure. (It is also possible to do this using the project's POM file, but this is not recommended.)

Procedure 6.2. Configuring Maven to Use the 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>http://maven.repository.redhat.com/techpreview/all</url>
          <releases>
            <enabled>>true</enabled>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>jboss-ga-plugin-repository</id>

<url>http://maven.repository.redhat.com/techpreview/all</url>
          <releases>
            <enabled>>true</enabled>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>

  <!-- 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>
-->

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

</settings>

```

2. If you modified the **settings.xml** file while 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 JBoss Developer Studio.

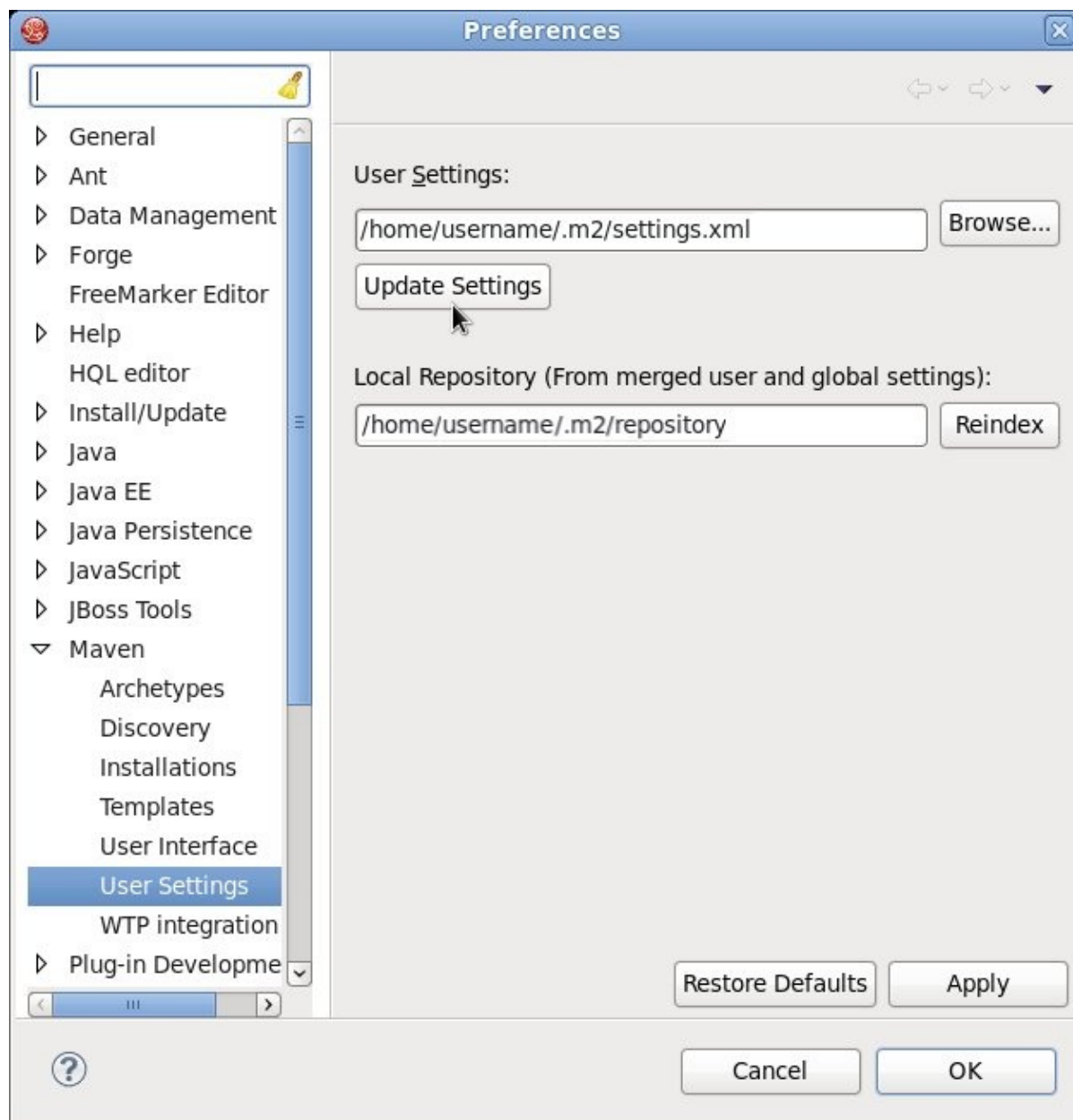


Figure 6.2. Update Maven User Settings

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. This will force Maven to download correct versions of required artifacts during the next build.

6.5. 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.

The Maven repository in 6.1.0 is designed to be used only in combination with Maven Central and no other repositories are required.

Depending on your project requirements, declare the dependencies in your POM file in the dependencies section:

- `org.jboss.bom.brms:jboss-brms-bpmsuite-bom:VERSION`: This is the basic BOM without any Java EE6 support.
- `org.jboss.bom.brms:jboss-javaee-6.0-with-brms-bpmsuite:VERSION`: This provides support for Java EE6.

CHAPTER 7. RED HAT JBOSS DEVELOPER STUDIO

7.1. RED HAT JBOSS DEVELOPER STUDIO

Red Hat JBoss Developer Studio is the JBoss Integrated Development Environment (IDE) based on Eclipse. Get the latest JBoss Developer Studio from the Red Hat customer support portal at <https://access.redhat.com>. 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 JBoss BRMS plug-in is called the Drools plug-in and the JBoss BPM Suite plug-in is called the jBPM plug-in.

Refer to the *Red Hat JBoss Developer Studio* documentation for installation and set-up 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"`

7.2. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS

The Drools plug-ins for JBoss Developer Studio are available via the update site.

Procedure 7.1. Install the Drools JBoss Developer Studio Plug-in

1. Start 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.jboss.com/updates/8.0/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. After installation of the plug-ins has completed, restart JBoss Developer Studio.

7.3. SETTING THE DROOLS RUNTIME

In order to use the Red Hat JBoss BRMS plug-in with Red Hat JBoss Developer Studio, it is necessary to set up the runtime.

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

Procedure 7.2. Configure JBoss BRMS Runtime

1. Extract the runtime jar files located in the **jboss-brms-VERSION-engine.zip** archive that you can download from [Red Hat Customer Portal](#).
2. From the JBoss Developer Studio menu, select **Window** and click **Preferences**.
3. Select **Drools** → **Installed Drools Runtimes**.
4. Click **Add . . .**; provide a name for the new runtime, and click **Browse** to navigate to the directory where you extracted the runtime files in step 1. Click **OK** to register the selected runtime in JBDS.
5. Mark the runtime you have created as the default Drools 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 JBoss Developer Studio to update the Runtime.

7.4. CONFIGURING THE JBOSS BRMS SERVER

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

Procedure 7.3. Configure the Server

1. Open the Drools view by selecting **Window** → **Open Perspective** → **Other** and 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 clicking **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 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**.

7.5. IMPORTING PROJECTS FROM A GIT REPOSITORY INTO JBOSS DEVELOPER STUDIO

You can configure 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 7.4. Cloning a Remote Git Repository

1. Start the Red Hat JBoss BRMS/BPM Suite server (whichever is applicable) by selecting the server from the server 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 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**.

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 ☐

Figure 7.1. Git Repository Details

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 7.5. Importing a Local Git Repository

1. Start the Red Hat JBoss BRMS/BPM Suite server (whichever is applicable) by selecting the server from the server tab and click the start icon.
2. In 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**.

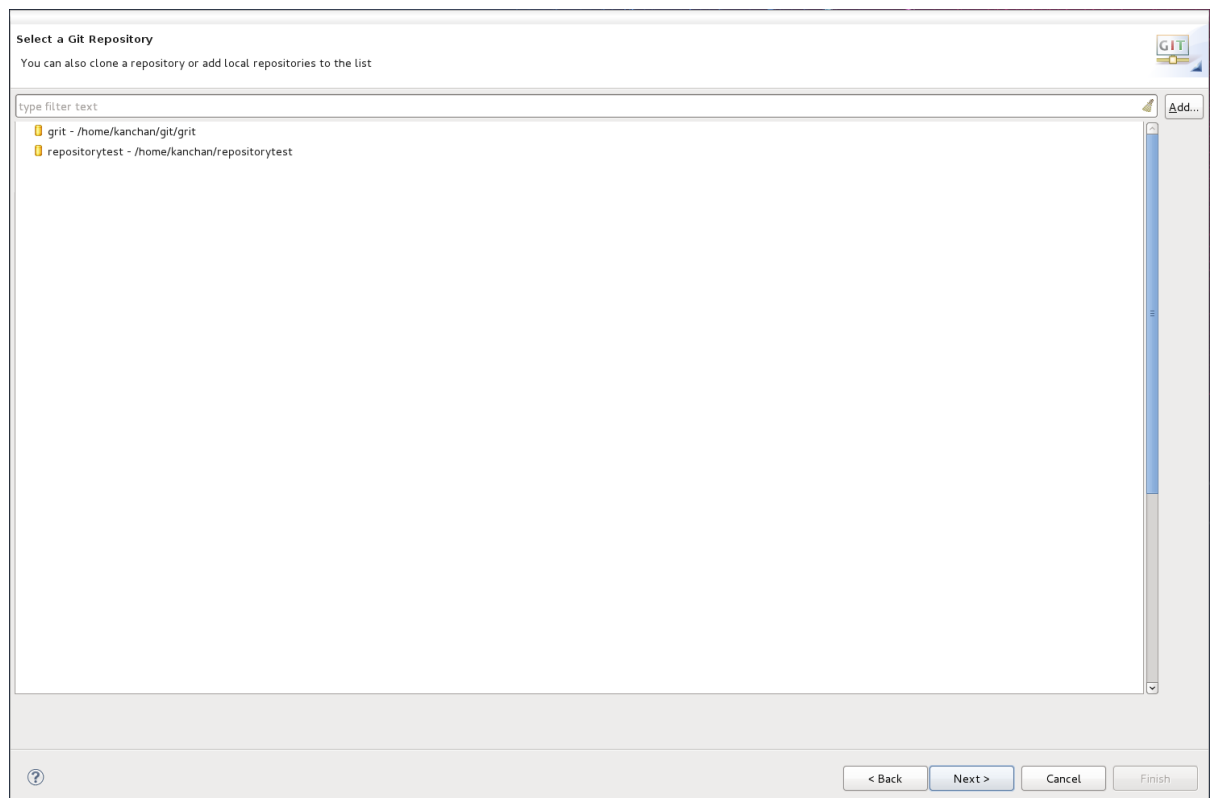


Figure 7.2. Git Repository Details

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 group** and click **Next**. Name the project and click **Finish**.

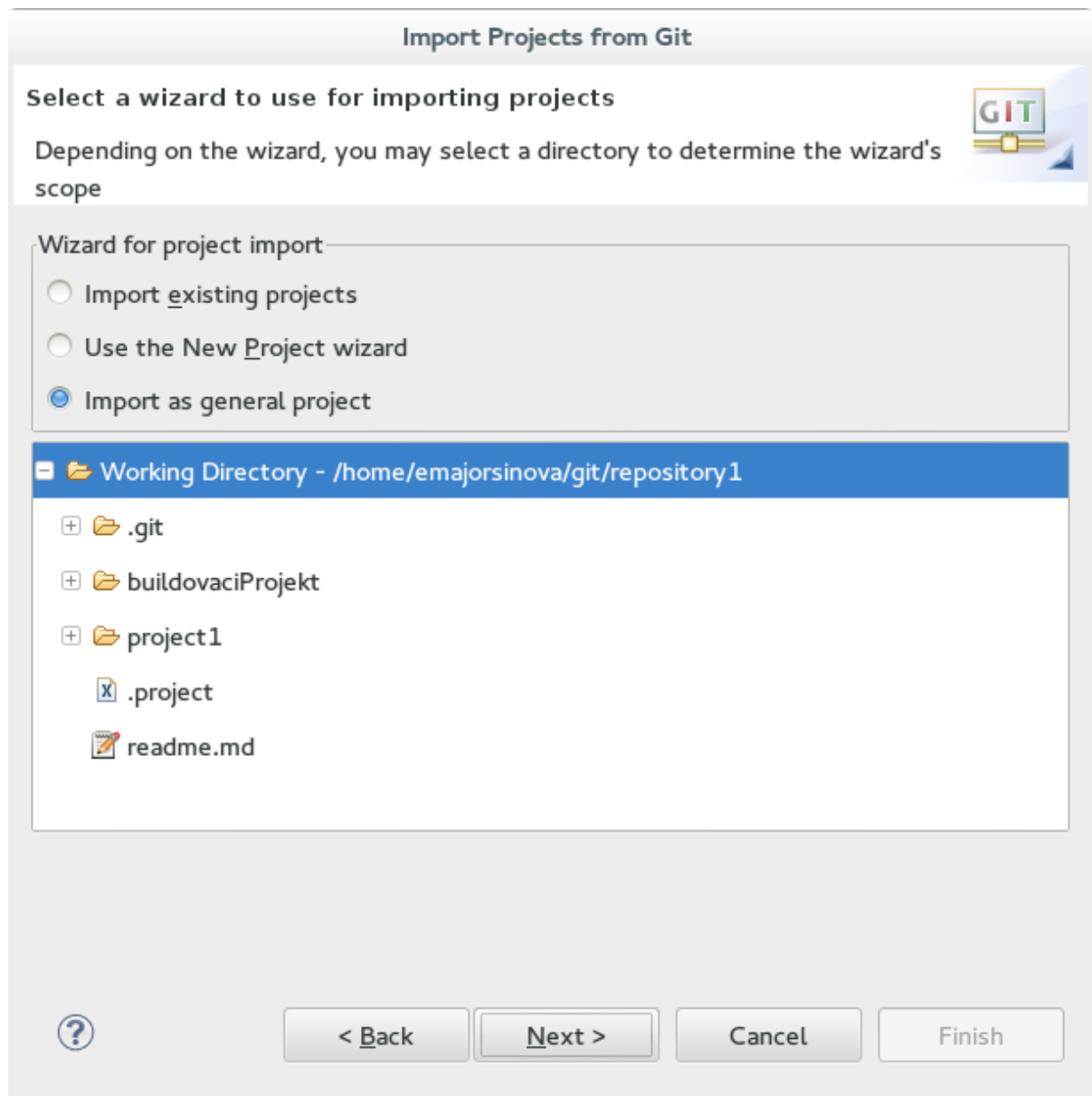


Figure 7.3. Wizard for Project Import

CHAPTER 8. BUSINESS RESOURCE PLANNER

Business Resource Planner is a lightweight, embeddable planning engine that optimizes planning problems. It helps normal Java TM 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.

8.1. INSTALLING BUSINESS RESOURCE PLANNER

1. Navigate to the [Red Hat Customer Portal](#) and log in with your user credentials.
2. Select **Downloads** → **Product Downloads**.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BRMS**.
4. From the **Version** drop-down menu, select version 6.1.
5. Select **Red Hat JBoss BRMS 6.1 Business Resource Planner** and then click **Download**.

8.2. RUNNING THE 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 9. PATCHING AND UPGRADING RED HAT JBOSS BPM SUITE

9.1. 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.

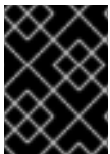
Red Hat JBoss BRMS patches can be downloaded from <https://access.redhat.com/downloads/>.

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

- JBoss BRMS customers - **jboss-brms-<version>-patch.zip**
- JBoss BPM Suite customers - **jboss-bpmsuite-<version>-patch.zip**
- Maven repo updates (Same for both JBoss BRMS and JBoss BPM Suite customers) - **jboss-brms-bpmsuite-<version>-incremental-maven-repository.zip**

9.2. APPLYING PATCHES IN RED HAT JBOSS BRMS 6.1

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 JBoss BRMS 6.1 (or better) installation.



IMPORTANT

The patching tool is for use with JBoss BRMS 6.1 or better, and should not be used for earlier versions. For more information, see <https://access.redhat.com/articles/1455733>.

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 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 BRMS/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 really useful feature that helps you to preserve your configuration files from being overwritten automatically by the update process. Of course, 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
```

It is now up to you to compare the contents of the two files and merge the changes.

What happens 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**. It is then up to you 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
```

9.3. PATCHING OTHER PLATFORMS AND APPLICATIONS

The following commands can be used for updating other supported platforms and common applications in Red Hat JBoss BRMS.

Patch EAP 6.x Business Central WAR

```
$ ./apply-updates.[sh|bat] <some-path>/jboss-eap-
6.4/standalone/deployments/business-central.war eap6.x-bc
```

Patch Generic KIE Server WAR

```
$ ./apply-updates.[sh|bat] <some-path-to-tomcat-home>/webapps/kie-
server.war generic-kie-server
```

Patch Whole WebLogic 12c Bundle

```
$ ./apply-updates.[sh|bat] <path-to-unzipped-wls12c-bundle> wls12c
```

Patch Planner Engine Bundle

```
$ ./apply-updates.[sh|bat] <path-to-unzipped-planner-bundle> planner-
engine
```


APPENDIX A. REVISION HISTORY

Revision 1.0.0-33

Thu Dec 17 2015

Vidya Iyengar

Build includes various enhancements and fixes