Abstract

This guide provides the steps necessary for administrators to install Red Hat JBoss BRMS Platform.
# Table of Contents

## CHAPTER 1. INTRODUCTION  ................................................................. 4.
  1.1. ABOUT RED HAT JBOSS BRMS  .................................................. 4
  1.2. USE CASE: BUSINESS DECISION MANAGEMENT IN THE INSURANCE INDUSTRY WITH JBOSS BRMS  .................................................. 4

## CHAPTER 2. INSTALLATION  ................................................................. 6.
  2.1. INSTALLATION OPTIONS  ................................................................. 6
  2.2. DOWNLOADING RED HAT JBOSS BRMS INSTALLER  .................................................. 6
  2.3. DOWNLOADING RED HAT JBOSS BRMS FOR JBOSS EAP  .................................................. 6
  2.4. DOWNLOADING RED HAT JBOSS BRMS GENERIC DEPLOYABLE PACKAGE  .................................................. 7
  2.5. INSTALLING RED HAT JBOSS BRMS USING THE INSTALLER  .................................................. 7
  2.6. INSTALLING RED HAT JBOSS BRMS USING THE INSTALLER IN CLI MODE  .................................................. 10
  2.7. INSTALLING RED HAT JBOSS BRMS FOR RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM  .................................................. 14
  2.8. INSTALLING RED HAT JBOSS BRMS ON RED HAT JBOSS WEB SERVER  .................................................. 15
  2.9. INSTALLING BRMS IN THE DOMAIN MODE  .................................................. 17
  2.10. DEFINING ROLES  ................................................................. 20
  2.11. CREATING USERS  ................................................................. 20

## CHAPTER 3. GIT  ................................................................................. 22
  3.1. CLONING AN EXISTING REPOSITORY  .................................................. 23
  3.2. MIGRATING A REPOSITORY FROM BRMS 5.3  .................................................. 24

## CHAPTER 4. AUTHENTICATION  .............................................................. 26.

## CHAPTER 5. TESTING THE INSTALLATION  ............................................. 27.
  5.1. STARTING THE SERVER  ................................................................. 27
  5.2. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT  .................................................. 28
  5.3. LOGGING ON TO BUSINESS CENTRAL  .................................................. 28

## CHAPTER 6. CLUSTERING  ................................................................. 29.
  GIT REPOSITORY CLUSTERING MECHANISM  .................................................. 29
  CLUSTERING MAVEN REPOSITORIES  .................................................. 29
  6.1. SETTING UP A CLUSTER  ................................................................. 29
  6.2. CONFIGURING CLUSTERING ON RED HAT JBOSS EAP  .................................................. 31

## CHAPTER 7. MAVEN REPOSITORIES  ...................................................... 35.
  7.1. ABOUT MAVEN  ................................................................. 35
  7.2. ABOUT THE PROVIDED MAVEN REPOSITORIES  .................................................. 35
  7.3. CONFIGURING MAVEN TO USE THE FILE SYSTEM REPOSITORIES  .................................................. 35
  7.4. CONFIGURING MAVEN TO USE THE ONLINE REPOSITORIES  .................................................. 40
  7.5. DEPENDENCY MANAGEMENT  ...................................................... 44

## CHAPTER 8. JBOSS DEVELOPER STUDIO  .............................................. 45.
  8.1. RED HAT JBOSS DEVELOPER STUDIO  .................................................. 45
  8.2. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS  .................................................. 45
  8.3. SETTING THE DROOLS RUNTIME  ...................................................... 45
  8.4. CONFIGURING THE JBOSS BRMS SERVER  .................................................. 46
  8.5. IMPORTING PROJECTS FROM A GIT REPOSITORY INTO JBOSS DEVELOPER STUDIO  .................................................. 47

## CHAPTER 9. BUSINESS RESOURCE PLANNER  ........................................... 51.
  9.1. INSTALLING BUSINESS RESOURCE PLANNER  .................................................. 51
  9.2. RUNNING THE BUSINESS RESOURCE PLANNER EXAMPLES  .................................................. 51
CHAPTER 1. INTRODUCTION

1.1. ABOUT RED HAT JBOSS BRMS

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

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

Business Resource Planner is included as a technical preview with this release.

Report a bug

1.2. USE CASE: BUSINESS DECISION MANAGEMENT IN THE INSURANCE INDUSTRY WITH JBOSS BRMS

BRMS comprises a high-performance rule engine from the Drools project, a rule repository and easy to use rule authoring tools from the Drools Guvnor project, and Complex Event Processing rule engine extensions from the Drools Fusion project. It also includes Business Resource Planner, a solver for complex planning problems, as a technology preview.

The consumer insurance market is extremely competitive, and it is imperative that customers receive efficient, competitive, and comprehensive services when visiting an online insurance quotation solution. An insurance provider increased revenue from their online quotation solution by upselling to the visitors of the solution relevant, additional products during the quotation process.

JBoss BRMS was integrated with the insurance provider's infrastructure so that when a request for insurance was processed, BRMS was consulted and appropriate additional products were presented with the insurance quotation:
Figure 1.1. BRMS Use Case: Insurance Industry Decision Making

BRMS provided the decision management functionality, i.e. the automatic determination of the products to present to the applicant based on the rules defined by business analysts. The rules were implemented as decision tables, so they could be easily understood and modified without requiring additional support from IT.

Report a bug
CHAPTER 2. INSTALLATION

2.1. INSTALLATION OPTIONS

Red Hat JBoss BRMS comes in two versions:

- Executable jar installer for installation on Red Hat JBoss Enterprise Application Platform (EAP) 6.1.1.

- Zip file install which itself comes in two versions:
  
  
  - jboss-brms-6.MINOR_VERSION-redhat-x-deployable-generic.zip: the deployable version with additional libraries adapted for deployment on Red Hat JBoss Web Server (WS) 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".

Report a bug

2.2. DOWNLOADING RED HAT JBOSS BRMS INSTALLER

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 6.0.3.

5. Select Red Hat JBoss BRMS 6.0.3 Installer for EAP 6.1.1 and then click Download.

Report a bug

2.3. 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 6.0.3.
5. Select Red Hat JBoss BRMS 6.0.3 Deployable for EAP 6.1.1 and then click Download.

Report a bug

2.4. DOWNLOADING RED HAT JBOSS BRMS GENERIC DEPLOYABLE PACKAGE

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 6.0.3.
5. Select Red Hat JBoss BRMS 6.0.3 Deployable for all supported containers and then click Download.

Report a bug

2.5. 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 BRMS on an existing EAP 6.1.1 installation or a brand new one. The installer gives you the option to install EAP 6.1.1 if you do not already have it.

NOTE

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

1. Setup Location and Users
   Navigate to the folder where you downloaded the installer file in a command prompt and execute the following command (replace the VERSION number and x with the actual file name).

   java -jar jboss-brms-installer-VERSION.GA-redhat-x.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-installer-VERSION.GA-redhat-x.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 EAP where BRMS needs to be installed. If you instead prefer to install BRMS in a brand new EAP, then point this location to a parent directory where there is no existing EAP. BRMS will then be installed on top of a fresh EAP server bundled with the installer. The screenshot below depicts an example 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 BRMS itself (ApplicationRealm). Make a note of these usernames and passwords as you will need them to access the EAP server and the BRMS application respectively.

**NOTE**

The passwords that you create must have at least 8 characters and must contain at least one number and one alphanumeric character.

Depicted below is a screenshot of the username and password page:
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 BRMS install. Decide to enable or disable the Java Security Manager in this step by clicking on the checkbox. The Java Security Manager makes your system more secure but may downgrade performance. You need to make a decision based on your environment.

6. Final Configuration Steps

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. 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.

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

Report a bug
2.6. 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 BRMS.

1. Navigate to the folder where you downloaded the installer file in a command prompt and execute the following command (replace the VERSION number and x with the actual file name).

   ```
   java -jar jboss-brms-installer-VERSION.GA-redhat-x.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. If you specify a directory where EAP does not exist, the installer will install a new version of EAP for you.

   ```
   If you wish to install on an existing EAP installation, the parent directory of that EAP installation should be specified. [/home/user]
   ```

   If you decide to create a new folder for the installation to take place, you will be prompted to select the following:

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

4. Create a user for the management console of EAP (Management Realm):

   ```
   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.
   
   Admin username: [admin]
   ```

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

   ```
   The password must be at least eight characters long, with one alphabetic character, one digit, and one non-alphanumeric character.
   
   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 BRMS user:

   Create a Business Rules Management System User
   Create a BRMS 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'.

   BRMS username: [brmsAdmin]

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

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

   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 by either pressing 1 to select it or 0 to deselect it.

    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.

    [x] 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:

[ ] Enable LDAP authentication
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/Documents/BRMS_Installer
   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/Documents/BRMS_Installer/<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-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]$ 
   ```


    ```
    ./standalone.sh
    ```


22. Login with the correct username/password as given to the BRMS user in the "Create and confirm a password for the BRMS user" step.
2.7. INSTALLING RED HAT JBOSS BRMS FOR RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

Red Hat JBoss BRMS can be installed on JBoss Enterprise Application Platform 6.1.1.

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-6.1 directory).

2. Unzip the downloaded zip archive; make sure it is merged into the EAP_HOME directory (jboss-eap-6.1).

   **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 BRMS instance. Make sure that there is no existing 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-6.1 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-6.1):

   - jboss-eap-6.1/domain/configuration/*
3. Ensure the target EAP does not include a deployment with a colliding name. Copy the folder `jboss-eap-6.1/standalone/deployments` into the `EAP_HOME` directory from the BRMS distribution.

4. Make sure no EAP module layer is already called BRMS and copy the folder `jboss-eap-6.1/modules/system/layers/brms` into the EAP 6.1.1 folder.

---

### 2.8. 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 to an existing application server. The following procedure provides instructions for installation on an existing Red Hat JBoss Web Server 2.0.1 instance.

**Procedure 2.1. Installing the Generic Deployable Package**

1. To download the generic deployable package zip file from the Red Hat Customer Support Portal, go to [https://access.redhat.com](https://access.redhat.com) 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 6.0.3

5. Select **Red Hat JBoss BRMS 6.0.3 Deployable for All Supported Containers** package and then click **Download**.

6. Extracting the downloaded archive creates the following zip archives:
   - `jboss-brms-engine.zip`
   - `jboss-brms-manager.zip`

7. Extract **business-central.war** from the `jboss-brms-manager.zip` archive and copy to `tomcat7/webapps/`.
8. Remove the .war extension from the business-central.war so that directory is now called business-central.

9. Define the users and the roles in tomcat7/conf/tomcat-users.xml as shown below:

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

10. Edit the web.xml file in the tomcat7/webapps/business-central/WEB-INF directory by uncommenting all entries that are marked with TOMCAT-JEE-SECURITY. For example:

```
<!-- TOMCAT-JEE-SECURITY -->
<!--
<init-param>
 <param-name>org.uberfire.auth.scheme</param-name>
 <param-value>org.uberfire.security.server.auth.JACCAuthenticationScheme</param-value>
 </init-param>
 -->
```

Should be uncommented like the following:

```
<!-- TOMCAT-JEE-SECURITY -->

<init-param>
 <param-name>org.uberfire.auth.scheme</param-name>
 <param-value>org.uberfire.security.server.auth.JACCAuthenticationScheme</param-value>
 </init-param>
```


13. Download the file below and place it in the tomcat7/lib directory.

   - javax.security.jacc-api.jar

14. Add the below Tomcat Valve declaration into tomcat7/conf/server.xml (inside the relevant <host> element):

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

org.uberfire.security.auth.AuthenticationSource inside the business-
central/WEB-INF/classes/META-INF/services


   ./startup.sh


18. Login with the correct username/password as given in the tomcat-users.xml file where you
defined user roles.

Report a bug

2.9. INSTALLING BRMS IN THE DOMAIN MODE

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

1. Extract the zip package deployable for EAP you downloaded from the Red Hat Customer Portal
   and copy the following directories into the installation of EAP 6.1:

   - bin
   - domain
   - modules

   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 2.11, “Creating users” and Section 2.10, “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 2.11, “Creating users” section.

**Installing Multiple 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 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`
When multiple 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:

```xml
<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:

```xml
<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="8002" boot-time="false"/>
</system-properties>
```
2.10. 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.

2.11. 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`.

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 2.10, “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).

Report a bug
CHAPTER 3. 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 3.1. Git Terminology**

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branches</td>
<td>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.</td>
</tr>
<tr>
<td>Commit</td>
<td>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.</td>
</tr>
<tr>
<td>Head</td>
<td>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<del>1, HEAD</del>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.</td>
</tr>
<tr>
<td>Index</td>
<td>Index is an alternative term for the staging area.</td>
</tr>
</tbody>
</table>
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 8.5, “Importing Projects from a Git Repository into JBoss Developer Studio”).

Report a bug

3.1. CLONING AN EXISTING REPOSITORY

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

Procedure 3.1. 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.

![Clone Repository Pop-up](image)

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

Report a bug

### 3.2. MIGRATING A REPOSITORY FROM 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:
   
   ```bash
   ./runMigration.sh -i <source-path> -o <destination-path> -r <repository-name>
   ```

   In a Windows environment, run:
   
   ```bat
   .\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.

Report a bug
CHAPTER 4. 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.

Report a bug
CHAPTER 5. TESTING THE INSTALLATION

5.1. STARTING THE SERVER

If you have installed BRMS using either the Installer or via the EAP6 bundle install, you can now start your server in one of two modes.

**NOTE**

If you installed 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:
   ```bash
   ./standalone.sh
   ```
   In a Windows environment run:
   ```bash
   ./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 5.2, "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:
5.2. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT

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

5.3. 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 6. 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).
- kie-commons provides VFS implementation and clustering
- uber fire framework provides 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.

Report a bug

6.1. 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-bpms-brms-VERSION-redhat-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/tmp/
   # the port at which the clients connects
   clientPort=2181
   ```

   Make sure the dataDir location exists and is accessible.

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.

4. Set up the cluster with the Zookeeper server as the master of the configuration:
   
   a. Create the cluster:

   ```
   $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_NAME:UNIQUE_ID
   ```

   **Example 6.1. Adding two cluster nodes**

   ```
   ./helix-admin.sh --zkSvr localhost:2181 --addNode brms-cluster
   nodeOne:12345
   ./helix-admin.sh --zkSvr localhost:2181 --addNode brms-cluster
   nodeTwo:12346
   ```

5. Add resources to the cluster.

   **Example 6.2. Adding vfs-repo as resource**

   ```
   ./helix-admin.sh --zkSvr localhost:2181 --addResource brms-cluster
   vfs-repo 1 LeaderStandby AUTO_REBALANCE
   ```
6. Rebalance the cluster.

Example 6.3. Rebalancing the brms-cluster

```
./helix-admin.sh --zkSvr localhost:2181 --rebalance brms-cluster
```

7. Start the Helix controller.

Example 6.4. Starting the Helix controller

```
./run-helix-controller.sh --zkSvr localhost:2181 --cluster brms-cluster 2>&1 > /tmp/controller.log &
```

6.2. 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*.

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 `SEAP_HOME/domain/configuration/host.xml` file with properties defined in Table 6.1, “Cluster node properties”:

   **Table 6.1. Cluster node properties**

<table>
<thead>
<tr>
<th>Property name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>org.uberfire.nio.git.dir</code></td>
<td><code>/home/jbrm/node[N]/repo</code></td>
<td>GIT (VFS) repository location on node[N]</td>
</tr>
<tr>
<td><code>jboss.node.name</code></td>
<td><code>nodeOne</code></td>
<td>node name unique within the cluster</td>
</tr>
<tr>
<td><code>org.uberfire.cluster.id</code></td>
<td><code>brms-cluster</code></td>
<td>Helix cluster name</td>
</tr>
<tr>
<td><code>org.uberfire.cluster.zk</code></td>
<td><code>localhost:2181</code></td>
<td>Zookeeper location</td>
</tr>
<tr>
<td><code>org.uberfire.cluster.local.id</code></td>
<td><code>nodeOne_12345</code></td>
<td>unique ID of the Helix cluster node</td>
</tr>
</tbody>
</table>
   | Note that : is replaced with _ .
<p>| <code>org.uberfire.cluster.vfs.lock</code> | <code>vfs-repo</code> | name of the resource defined on the Helix cluster |</p>
<table>
<thead>
<tr>
<th>Property name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>org.uberfire.nio.git.daemon.port</td>
<td>9418</td>
<td>port used by the VFS repo to accept client connections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The port must be unique for each cluster member.</td>
</tr>
<tr>
<td>org.uberfire.metadata.index.dir</td>
<td>/home/jbrm/node[N]/index</td>
<td>location where the index for search is to be created (maintained by Apache Lucene)</td>
</tr>
<tr>
<td>org.uberfire.nio.git.ssh.port</td>
<td>8003</td>
<td>the unique port number for ssh access to the GIT repo for a cluster running on physical machines.</td>
</tr>
<tr>
<td>org.uberfire.nio.git.daemon.host</td>
<td>nodeOne</td>
<td>the name of the daemon host machine in a physical cluster.</td>
</tr>
<tr>
<td>org.uberfire.nio.git.ssh.host</td>
<td>nodeOne</td>
<td>the name of the SSH host machine in a physical cluster.</td>
</tr>
</tbody>
</table>

Example 6.5. Cluster nodeOne configuration

```xml
<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="localhost:2181" boot-time="false"/>
    <property name="org.uberfire.cluster.local.id" value="nodeOne_12345" boot-time="false"/>
    <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo" boot-time="false"/>

    <property name="org.uberfire.nio.git.daemon.port" value="9418" boot-time="false"/>
    <property name="org.uberfire.metadata.index.dir" value="/tmp/jbrm/nodeone" boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.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" />
</system-properties>
```
Example 6.6. Cluster nodeTwo configuration

```xml
<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="localhost: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.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.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"/>
</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 *BRMS Administration and Configuration Guide*.

3. Start the application server:

   ```bash
   $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`).
Report a bug
CHAPTER 7. MAVEN REPOSITORIES

7.1. ABOUT MAVEN

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

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

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

For more information about Maven, see 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.

Report a bug

7.2. 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. You can configure Maven to use the repositories following the procedure in Section 7.4, “Configuring Maven to Use the Online Repositories” or Section 7.3, “Configuring Maven to Use the File System 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.

Report a bug

7.3. 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 7.1.

1. Download the following ZIP archives containing the required repositories:

- [http://maven.repository.redhat.com/techpreview/brms6/6.0.3/jboss-bpms-brms-6.0.3.GA-redhat-1-maven-repository.zip](http://maven.repository.redhat.com/techpreview/brms6/6.0.3/jboss-bpms-brms-6.0.3.GA-redhat-1-maven-repository.zip)


**NOTE**

Alternatively, the ZIP archives can be downloaded from [http://access.redhat.com/jbossnetwork/](http://access.redhat.com/jbossnetwork/).

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
<?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 BRMS/BPMS -->
        <profile>
            <id>brms-bpms-local-profile</id>
            <repositories>
                <repository>
                    <id>jboss-bpms-brms-6.0.3.GA-redhat-1-maven-repository</id>
                    <name>BRMS/BPMS 6.0.3 GA Repository</name>
                    <url>file:///jboss-bpms-brms-6.0.3.GA-redhat-1-maven-repository</url>
                    <layout>default</layout>
                    <releases>
                        <enabled>true</enabled>
                        <updatePolicy>never</updatePolicy>
                    </releases>
                    <snapshots>
                        <enabled>false</enabled>
                        <updatePolicy>never</updatePolicy>
                    </snapshots>
                </repository>
                <repository>
                    <id>jboss-eap-6.1.1.GA-maven-repository</id>
                    <name>EAP 6.1.1 GA Repository</name>
                </repository>
            </repositories>
        </profile>
    </profiles>
</settings>
```
<url>file://</url> <!-- path to the repository -->/jboss-eap-6.1.1.GA-maven-repository</url>
<layout>default</layout>
<releases>
  <enabled>true</enabled>
  <updatePolicy>never</updatePolicy>
</releases>
<snapshots>
  <enabled>false</enabled>
  <updatePolicy>never</updatePolicy>
</snapshots>
</repository>
<repository id="jboss-public-repository">
<url>http://repository.jboss.org/nexus/content/repositories/public/</url>
<releases>
  <enabled>true</enabled>
  <updatePolicy>never</updatePolicy>
</releases>
<snapshots>
  <enabled>false</enabled>
  <updatePolicy>never</updatePolicy>
</snapshots>
</repository>
<repositories>
<pluginRepositories>
<pluginRepository id="jboss-bpms-brms-6.0.3.GA-redhat-1-maven-repository">
  <name>BRMS/BPMS 6.0.3 GA Repository</name>
<url>file://</url> <!-- path to the repository -->/jboss-bpms-brms-6.0.3.GA-redhat-1-maven-repository</url>
<layout>default</layout>
<releases>
  <enabled>true</enabled>
  <updatePolicy>never</updatePolicy>
</releases>
<snapshots>
  <enabled>false</enabled>
  <updatePolicy>never</updatePolicy>
</snapshots>
</pluginRepository>
<pluginRepository id="jboss-eap-6.1.1.GA-maven-repository">
  <name>EAP 6.1.1 GA Repository</name>
<url>file://</url> <!-- path to the repository -->/jboss-eap-6.1.1.GA-maven-repository</url>
<layout>default</layout>
<releases>
  <enabled>true</enabled>
  <updatePolicy>never</updatePolicy>
</releases>
<snapshots>
  <enabled>false</enabled>
  <updatePolicy>never</updatePolicy>
</snapshots>
</pluginRepository>
</pluginRepositories>
</repositories>

CHAPTER 7. MAVEN REPOSITORIES
<pluginRepository>
    <id>jboss-public-plugin-repository</id>
</pluginRepository>

<url>http://repository.jboss.org/nexus/content/repositories/public/</url>

    <releases>
        <enabled>true</enabled>
    </releases>
    <snapshots>
        <enabled>false</enabled>
    </snapshots>
</pluginRepository>
</pluginRepositories>
</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-local-profile</activeProfile>
</activeProfiles>
</settings>
IMPORTANT

The **bpm-service** quickstart application will not build without two additional repositories. If you want to build this application, download the following repositories and add them to your **setting.xml** file in the same way as that shown above.

- [http://maven.repository.redhat.com/techpreview/eap6/6.0.1/jboss-eap-6.0.1-maven-repository.zip](http://maven.repository.redhat.com/techpreview/eap6/6.0.1/jboss-eap-6.0.1-maven-repository.zip)
- [http://maven.repository.redhat.com/techpreview/wfk2/2.5.0/jboss-wfk-2.5.0-maven-repository.zip](http://maven.repository.redhat.com/techpreview/wfk2/2.5.0/jboss-wfk-2.5.0-maven-repository.zip)

**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.
7.4. 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/ and
http://repository.jboss.org/nexus/content/repositories/public/.

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 7.2. 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:

```xml
<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>
        <repository>
          <id>jboss-public-repository</id>
          <url>http://repository.jboss.org/nexus/content/repositories/public/</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>false</enabled>
          </snapshots>
        </repository>
      </repositories>
    </profile>
    <profile>
      <id>jboss-ga-plugin-repository</id>
    </profile>
    <profile>
      <id>jboss-public-plugin-repository</id>
    </profile>
  </profiles>
</settings>
```
Installation Guide

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

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.
7.5. DEPENDENCY MANAGEMENT

In order to use correct Maven dependencies in your Red Hat JBoss BRMS project, relevant Bill Of Materials (BOM) and parent POM files must be added to the project's \texttt{pom.xml} file. Adding the BOM and parent will ensure that correct versions of plugins and transitive dependencies from the provided Maven repositories are included in the project.

To ensure correct dependency usage in your project, declare the following parent in the project's \texttt{pom.xml} file:

\begin{verbatim}
<parent>
  <groupId>org.jboss.ip.component.management</groupId>
  <artifactId>ip-parent</artifactId>
  <version>1.2.1-redhat-2</version>
</parent>
\end{verbatim}

and add the following two BOM files as dependencies in the \texttt{dependencyManagement} section:

\begin{verbatim}
<dependency>
  <groupId>org.jboss.ip.component.management</groupId>
  <artifactId>ip-dependency-management-all</artifactId>
  <type>pom</type>
  <version>1.2.1-redhat-2</version>
  <scope>import</scope>
</dependency>
<dependency>
  <groupId>org.jboss.component.management</groupId>
  <artifactId>jboss-dependency-management-all</artifactId>
  <type>pom</type>
  <version>6.1.2.Final-redhat-5</version>
  <scope>import</scope>
</dependency>
\end{verbatim}

Use the entries from the code sample below for this purpose.
CHAPTER 8. JBOSS DEVELOPER STUDIO

8.1. RED HAT JBOSS DEVELOPER STUDIO

Red Hat JBoss Developer Studio is the JBoss integrated development environment (IDE) based on Eclipse and available 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 BRMS plugin is called the Drools plugin and the BPM Suite plugin is called the jBPM plugin.

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 need to ensure that the instance of JBDS is started with the file encoding set to UTF-8. You can do this by editing the $JBDS_HOME/studio/jbdevstudio.ini file and adding the following property: 

"-Dfile.encoding=UTF-8"

8.2. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS

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

Procedure 8.1. 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/7.0/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.

8.3. SETTING THE DROOLS RUNTIME
In order to use the 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 8.2. Configure BRMS Runtime**

1. Extract the runtime jar files located in the `jboss-brms-engine.zip` archive of the JBoss BRMS Generic Deployable zip archive (not the EAP6 deployable zip archive) available 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.

**Report a bug**

**8.4. CONFIGURING THE JBOSS BRMS SERVER**

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

**Procedure 8.3. 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 6.1+** and clicking **Next**.

5. Set the home directory by clicking the **Browse** button. Navigate to and select the installation directory for JBoss EAP 6.1.1 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**.

**Report a bug**
8.5. IMPORTING PROJECTS FROM A GIT REPOSITORY INTO JBOSS DEVELOPER STUDIO

JBoss Developer Studio can be configured to connect to a central Git asset repository. The repository is the space where versions of rules, models, functions and processes are stored. This Git repository must already be defined by the KIE Workbench.

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

Procedure 8.4. Cloning a Remote Git Repository

1. Start the Red Hat JBoss BRMS server (if not already running) 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 will fail and you may safely ignore this step.

   ```bash
   /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.
6. Select which branch you want to import in the following window and click Next.

7. You will be presented with the option 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 8.5. Importing a Local Git Repository**

1. Start the Red Hat JBoss BRMS server (if not already running) 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 8.2. Git Repository Details

4. Select the repository that is to be configured from the list of available repositories and click Next.

5. In the dialog that opens, select the radio button Import as general project from the Wizard for project import group and click Next. Name the project and click Finish.
Figure 8.3. Wizard for Project Import

Report a bug
CHAPTER 9. BUSINESS RESOURCE PLANNER

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

Planner helps solve various use cases like the following:

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

9.1. INSTALLING BUSINESS RESOURCE PLANNER

1. Navigate to the Red Hat Customer Portal and log in with your user credentials.
2. Select Downloads → Product Downloads.
3. In the Product Downloads page that opens, click Red Hat JBoss BRMS.
4. From the Version drop-down menu, select version 6.0.3.
5. Select Red Hat JBoss BRMS 6.0.3 Business Resource Planner and then click Download.

9.2. 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.
## APPENDIX A. REVISION HISTORY

<table>
<thead>
<tr>
<th>Revision 1.0.0-47</th>
<th>Mon Dec 01 2014</th>
<th>Vikram Goyal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built from Content Specification: 22697, Revision: 729903 by vigoyal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>