



Red Hat JBoss BRMS 6.2

Getting Started Guide

The Getting Started Guide for Red Hat JBoss BRMS

Red Hat JBoss BRMS 6.2 Getting Started Guide

The Getting Started Guide for Red Hat JBoss BRMS

Kanchan Desai
kadesai@redhat.com

Doug Hoffman

Eva Kopalova

Red Hat Content Services

Legal Notice

Copyright © 2015 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). 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 is to help users install and set up Red Hat JBoss BRMS for the first time.

Table of Contents

CHAPTER 1. QUICK START WITH RED HAT JBOSS BRMS	3
CHAPTER 2. INSTALLATION	4
2.1. DOWNLOADING RED HAT JBOSS BRMS FOR JBOSS EAP	4
2.2. INSTALLING RED HAT JBOSS BRMS FOR RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM	4
2.3. INSTALLING RED HAT JBOSS BRMS ON RED HAT JBOSS WEB SERVER	5
2.4. DEFINING ROLES	9
2.5. CREATING USERS	9
2.6. STARTING THE SERVER	10
2.7. JAVA SECURITY MANAGER AND PERFORMANCE MANAGEMENT	11
CHAPTER 3. LOGGING ON TO BUSINESS CENTRAL	13
CHAPTER 4. HELLO WORLD RULE EXAMPLE	14
4.1. CREATE YOUR FIRST RULE USING BUSINESS CENTRAL	14
4.2. SETTING UP A DECISION SERVER	17
4.3. FIRING RULES USING KIE SERVER JAVA CLIENT API	19
CHAPTER 5. RED HAT JBOSS DEVELOPER STUDIO	23
5.1. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS	23
5.2. SETTING THE DROOLS RUNTIME	23
5.3. CONFIGURING THE JBOSS BRMS SERVER	24
5.4. IMPORTING PROJECTS FROM A GIT REPOSITORY INTO JBOSS DEVELOPER STUDIO	24
5.5. CREATING A DROOLS PROJECT	27
CHAPTER 6. BUSINESS RESOURCE PLANNER	29
6.1. INSTALLING BUSINESS RESOURCE PLANNER	29
6.2. RUNNING THE BUSINESS RESOURCE PLANNER EXAMPLES	29
APPENDIX A. REVISION HISTORY	30

CHAPTER 1. QUICK START WITH RED HAT JBOSS BRMS

In this section we download, install and run Red Hat JBoss BRMS with the minimum number of instructions. For more detailed steps and alternate methods of installation, skip this section and continue reading the rest of this guide.

These instructions assume that you have the minimum supported Java version installed on your system. An existing Red Hat JBoss EAP server (version 6.4 or better) is also required as the installer does not install one.

Procedure 1.1. Red Hat JBoss BRMS Quick Start

1. Download JBoss BRMS installer from access.redhat.com.

2. Run installer by executing the following command:

```
java -jar jboss-brms-6.2.0.GA-installer.jar
```

3. Step through the GUI installer steps making note of the username/password for accessing the JBoss BRMS application after installation.

4. After successful installation, in a command prompt, navigate to the EAP home folder where you have installed JBoss BRMS.

5. Start the JBoss BRMS server by issuing the following command.

```
bin/standalone.sh
```

6. Open a web browser and login to Business Central by navigating to **<http://localhost:8080/business-central/>**. Login using the username/password combination for accessing the JBoss BRMS application that you created at installation time.

You have successfully installed and run Red Hat JBoss BRMS. You can now do the following:

Learn more about roles: [Section 2.4, “Defining Roles”](#)

Create additional users: [Section 2.5, “Creating users”](#)

Create a Hello World Rule: [Chapter 4, Hello World Rule Example](#)

Look up detailed installation instructions for using the installer: *Installation Guide*

CHAPTER 2. INSTALLATION

2.1. 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 6.2.0.
5. Select **Red Hat JBoss BRMS 6.2.0 Deployable for EAP 6** and then click **Download**.

2.2. 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.



IMPORTANT

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

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.

2.3. 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 6.2 to an existing application server. The following procedure provides instructions for installation on an existing Red Hat JBoss Web Server 2.1.0 instance.

**NOTE**

In case you have already installed Red Hat JBoss BRMS on Red Hat JBoss Enterprise Application Platform, skip to [Section 2.4, “Defining Roles”](#).

Procedure 2.1. 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 6.2.0.****5. Select Red Hat JBoss BRMS 6.2.0 Deployable for All Supported Containers package and then click Download.**

Also select and download the Red Hat JBoss BRMS Core 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 2.4, “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 2.1. 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 2.2. MySQL 5.5 datasource definition

```
resource.ds1.className=com.mysql.jdbc.jdbc2.optional.MysqlXAData
Source
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:mysql://localhost:3306/s
ampled
```

```
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.3. DB2 Type 4 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
mpled
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Example 2.4. 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 2.5. 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. 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/**.

13. 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.

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

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

2.4. 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.

2.5. 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. ~ ! @ # \$ % ^ * () - _ + =).

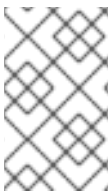
6. Enter a comma separate list of roles the user will need at the roles prompt (refer to [Section 2.4, “Defining Roles”](#)).

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).

2.6. 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.

**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:

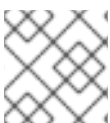
1. On the command line, move into the **\$SERVER_HOME/bin/** directory.
2. In a Unix environment run:

```
./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 2.7, “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:

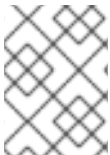
```
./standalone-secure.sh
```

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.

2.7. 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.

CHAPTER 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 4. HELLO WORLD RULE EXAMPLE

This example demonstrates basic features of Red Hat JBoss BRMS by setting up a **Hello World** project with a simple business rule.

In this project, you will:

1. Create a repository in the Artifact repository.
2. Create a project.
3. Create a rule.
4. Create a knowledgebase.
5. Build and deploy the project.
6. Create a Maven project to fire the rule.

4.1. CREATE YOUR FIRST RULE USING BUSINESS CENTRAL

Prerequisite

Ensure that you have successfully installed JBoss BRMS and Maven before you run this simple rule example using Business Central interface.

Procedure 4.1. Create and Execute your First Rule using Business Central

1. Log into Business Central

- a. On the command line, change into the **\$SERVER_HOME/bin/** directory and execute the following command:

- for Unix environment:

```
./standalone.sh
```

- for Windows environment:

```
./standalone.bat
```

- b. Once your server is up and running, open the following address in a web browser:

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

This opens the Business Central login page.

- c. Log into the Business Central with the user credentials created during installation.

2. Create a repository structure and create a project under it

- a. On the main menu of Business Central, go to **Authoring** → **Administration**.
- b. Click **Organizational Units** → **Manage Organizational Units**, then click **Add**.

- c. In the displayed **Add New Organizational Unit** dialog box, define the unit properties. For example:

- Name: EmployeeWage
- Owner: Employee

Click **OK**.

- d. On the perspective menu, click **Repositories** → **New repository**.

- e. In the displayed **Create Repository** dialog box, define the repository properties. For example:

- Repository Name: EmployeeRepo
- Organizational Unit: EmployeeWage

Click **Finish**.

- f. Go to **Authoring** → **Project Authoring**.

- g. In the Project Explorer, under the organizational unit drop-down box, select **EmployeeWage**, and in the repository drop-down box select **EmployeeRepo**.

- h. On the perspective menu, go to **New Item** → **Project**.

- i. In the displayed **Create new Project** dialog box, provide a name (for example, **MyProject**) for your project properties and click **Ok**.

- j. In the **New Project** dialog box, define the maven properties of the project. For example:

- Group ID: org.brms
- Artifact ID: MyProject
- Version ID: 1.0.0

Click **Finish**.

3. Create a fact model

- a. On the perspective menu, go to **New Item** → **Data Object**.

- b. In the displayed **Create new Data Object** dialog box, provide the values for object name and package. For example:

- Data Object: Person
- Package: org.brms.myproject

Click **Ok**.

- c. In the displayed **Person** window of the newly created **Person** data object, click **add field** to open the **New field** dialogue. Add a variable name in the **Id** field, select data type for the variable in the **Type** field, and click **Create and continue** until you have defined all the necessary variables. For example:

- Id: firstName

Type: String

- Id: lastName

Type: String

- Id: hourlyRate

Type: Integer

- Id: wage

Type: Integer

Click **Create** for the last variable and then **Save**.

4. Create a rule

- On the perspective menu, click **New Item** → **DRL File**.
- In the Create new dialog box, provide the name and package name of your rule file. For example:

- DRL file name: **MyRule**

- Package: org.brms.myproject

Click **Ok**.

- In the displayed DRL editor with the **MyRule.drl** file, write your rule as shown below:

```
package org.brms.myproject;
rule "MyRule"
when

    Person(hourlyRate*wage > 100)
    Person(name : firstName, surname : lastName)

then
    System.out.println( "Hello" + " " + name + " " + surname + "!"
);
    System.out.println( "You are rich!" );
end
```

- Click **Save**.

Rules are simple when-then statements stored in DRL files. This example has two conditions:

- `Person(hourlyRate*wage > 100)`
- `Person(name : firstName, surname : lastName)`

The first line evaluates a logical condition. The second line searches for an instance of the data object `Person` and saves its attributes, that is, the `firstName` and `lastName` into variables used in the second part of the rule.

If the conditions are met, the *then* part of the rule executes. In this example, two lines will be printed out in the JBoss BRMS command line.

5. Create a Knowledge Base

Following steps demonstrate how to create your own knowledge base. For more information about `KieSession`, see [Chapter 16.2.3](#) from the Red Hat JBoss BPM Suite Development Guide.

- a. Click **Open Project Editor**.
- b. Navigate to the drop-down menu and click **Project Settings: Project General Settings → Knowledge Base Settings: Knowledge bases and sessions**.
- c. Click **Add** and enter the name of your knowledge base. For this example, enter: **myBase**.
- d. Click **Make Default**
- e. Click **Add** under **Packages**.
- f. In this step, you are specifying which packages should your knowledge base include. In this example, we want all the packages. Therefore, enter `*` (an asterisk).
- g. Click **Add** under **Knowledge Sessions**.
- h. Enter the name of your session. For example, **mySession**. Check **Default** and select the **stateless** state.
- i. Click **Save** in the top right corner.

6. Build and deploy your rule

- a. Click **Open Project Editor** and then click **Build & Deploy**.

A green notification appears in the upper part of the screen informing you that the project has been built and deployed successfully to the Execution Server.



NOTE

In case a red notification appears, informing you that the build has failed, you will be presented with information about the build failure in the JBoss BRMS console. This is crucial information in case of troubleshooting your application. Make sure you have saved all assets before attempting to build your project.

4.2. SETTING UP A DECISION SERVER

To fire all the rules through the REST API, you must register a decision server. This section illustrates:

1. How to register a decision server
2. How to add a new container to the decision server

See the [How to set up Decision Server in 6.2 Knowledgebase](#) article for more information about decision servers.

Procedure 4.2. Registering a Decision Server

1. Creating a User

You will need a new user. Due to a bug, this user needs to be named **kieserver**.

- a. In the command line, move into the **\$SERVER_HOME/bin/** directory and execute the following command:

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

2. Creating a Decision Server


- a. On the command line, change into the **\$SERVER_HOME/standalone/configuration** directory and edit **standalone.xml** so that it contains the following properties in addition to the properties located in the **system-properties** tag:

```
<property name="org.kie.server.user"
value="helloworlduser"></property>
<property name="org.kie.server.pwd"
value="Helloworld@123"></property>
<property name="org.kie.server.location"
value="http://localhost:8080/kie-
server/services/rest/server"></property>
<property name="org.kie.server.controller"
value="http://localhost:8080/business-
central/rest/controller"></property>
<property name="org.kie.server.controller.user"
value="kieserver"></property>
<property name="org.kie.server.controller.pwd"
value="kieserver1!"></property>
<property name="org.kie.server.id" value="local-server-
123"></property>
```

3. Setting Up Containers

- a. On the command line, change into the **\$SERVER_HOME/bin/** directory and execute the following command:

```
./standalone.sh
```

- b. In the Business Central, click **Deploy** → **Rule Deployment**
- c. In the displayed **Server Management Browser** window, you see **local-server-123**. Click **plus** on the right.
- d. The **Create Container...** dialog opens. Enter the following:
 - Name: **myContainer**
 - Click **Search** and next to **MyProject-1.0.0.jar**, click **select**.
 - Click **Ok** to create the container.
- e. Click  to select **myContainer** and click **Start** to start the container

4.3. FIRING RULES USING KIE SERVER JAVA CLIENT API

To fire the rules, you need to send a request to the decision server. In this guide, you will use the Kie Server Java Client API to send requests using REST API. You can use any client to send XML requests to the server. For more information about sending requests to the decision server, see the [How to generate and send requests to Decision Server?](#) Knowledgebase article.



NOTE

Make sure you have configured your Maven installation to use JBoss BRMS online repositories. For more information about Maven configuration, see [Chapter 7.4](#) from the Red Hat JBoss BPM Suite Installation Guide.

1. Create a Basic Maven Archetype

- a. Navigate to a directory of your choice in your system and execute the following command:

```
mvn archetype:generate -DgroupId=com.MyProject.app -
DartifactId=my-app -DarchetypeArtifactId=maven-archetype-
quickstart -DinteractiveMode=false
```

- b. This creates a directory called **my-app** with the following structure:

```
my-app
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |   |   |-- com
    |   |   |   |-- MyProject
    |   |   |   |   |-- app
    |   |   |   |   |   App.java
    |-- test
    |   |-- java
    |   |   |-- com
    |   |   |   |-- MyProject
    |   |   |   |   |-- app
    |   |   |   |   |   AppTest.java
```

2. Import Maven Dependencies

- a. You must declare the libraries your Maven project will use. Edit the **my-app/pom.xml** file to set the JBoss BRMS dependencies. Import following dependencies:

```
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-compiler</artifactId>
  <version>6.3.0.Final-redhat-5</version>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-core</artifactId>
  <version>6.3.0.Final-redhat-5</version>
</dependency>
```

```

<dependency>
  <groupId>org.kie.server</groupId>
  <artifactId>kie-server-api</artifactId>
  <version>6.3.0.Final-redhat-5</version>
</dependency>

<dependency>
  <groupId>org.kie.server</groupId>
  <artifactId>kie-server-client</artifactId>
  <version>6.3.0.Final-redhat-5</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-simple</artifactId>
  <version>1.7.13</version>
</dependency>
<dependency>
  <groupId>org.brms</groupId>
  <artifactId>MyProject</artifactId>
  <version>1.0.0</version>
</dependency>

```

NOTE

The last dependency is the project you have built in JBoss BRMS. The model class (**org.brms.myproject.Person** in this example) in the client code, that is your Maven application, needs to match your server side, that is the JBoss BRMS. Otherwise, the rules will not match when sending a request. The easiest way to achieve this is to share the model class between both sides. You achieve this by adding it as a Maven dependency.

Therefore, make sure that the following attributes match what you had entered while creating a new project in the Business Central:

- groupId
- artifactId
- version

3. Firing the Rule from TestApp

- a. Locate the **testApp** method in the **my-app/src/test/java/com/MyProject/app/AppTest.java** file (which is created by default by Maven).
- b. Edit the **AppTest.java**:
 - i. Add the following imports:

```

import org.kie.api.command.BatchExecutionCommand;
import org.kie.server.api.model.ServiceResponse;
import org.kie.server.client.KieServicesClient;
import org.kie.server.client.KieServicesConfiguration;

```



```

import org.kie.server.client.KieServicesFactory;
import org.kie.server.client.RuleServicesClient;
import org.kie.api.KieServices;
import org.kie.api.command.Command;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import org.brms.myproject.Person;
import org.drools.core.command.impl.GenericCommand;

```

- ii. Locate the **testApp()** method, load the knowledge base and fire your rule by adding the following code:

```

Person p1 = new Person();
p1.setFirstName("Anton");
p1.setLastName("RedHat");
p1.setHourlyRate(11);
p1.setWage(20);

String url = "http://localhost:8080/kie-
server/services/rest/server";
String username = "kieserver";
String password = "kieserver1!";
String container = "myContainer";
String session = "mySession";

KieServicesConfiguration config =
KieServicesFactory.newRestConfiguration(url, username,
password);
Set<Class<?>> allClasses = new HashSet<Class<?>>();
allClasses.add(Person.class);
config.addJaxbClasses(allClasses);

KieServicesClient client =
KieServicesFactory.newKieServicesClient(config);
RuleServicesClient ruleClient =
client.getServicesClient(RuleServicesClient.class);
List<GenericCommand<?>> commands = new
ArrayList<GenericCommand<?>>();
commands.add((GenericCommand<?>)
KieServices.Factory.get().getCommands().newInsert(p1,"Person
Insert ID"));
commands.add((GenericCommand<?>)
KieServices.Factory.get().getCommands().newFireAllRules("fire-
identifier"));
BatchExecutionCommand batchCommand =
KieServices.Factory.get().getCommands().newBatchExecution(comm
ands,session);
ServiceResponse<String> response =
ruleClient.executeCommands(container, batchCommand);
System.out.println(response.getResult());

```

Note that the default marshaller on the server side is JAXB, unless set differently. That means you must set its context with any custom classes you plan to use. In this example, that is **Person.class**. Since the example uses a **stateless** session, the marshaller is disposed of once a single command is executed. That is why you need to wrap multiple commands in the **BatchCommand** object.

To change the marshaller type, enter one of the following commands and add the **MarshallingFormat** import to your project:

- `config.setMarshallingFormat(MarshallingFormat.JSON);`
- `config.setMarshallingFormat(MarshallingFormat.XSTREAM);`
- `import org.kie.server.api.marshalling.MarshallingFormat;`

- c. Navigate to the **my-app** directory and execute the following command from the command line:

```
mvn clean install
```

- d. Note that there is the Maven output and the JBoss BRMS output. The expected output in the JBoss BRMS console is:

```
16:26:56,119 INFO [stdout] (http-/127.0.0.1:8080-5) Hello Anton RedHat!
16:26:56,119 INFO [stdout] (http-/127.0.0.1:8080-5) You are rich!
```

CHAPTER 5. 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"`

5.1. INSTALLING THE JBOSS DEVELOPER STUDIO PLUG-INS

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

Procedure 5.1. 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.

5.2. 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 5.2. 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.

5.3. CONFIGURING THE JBOSS BRMS SERVER

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

Procedure 5.3. 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**.

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



NOTE

This is an additional feature not required for working with 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 5.4. 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 5.1. 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 5.5. 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**.

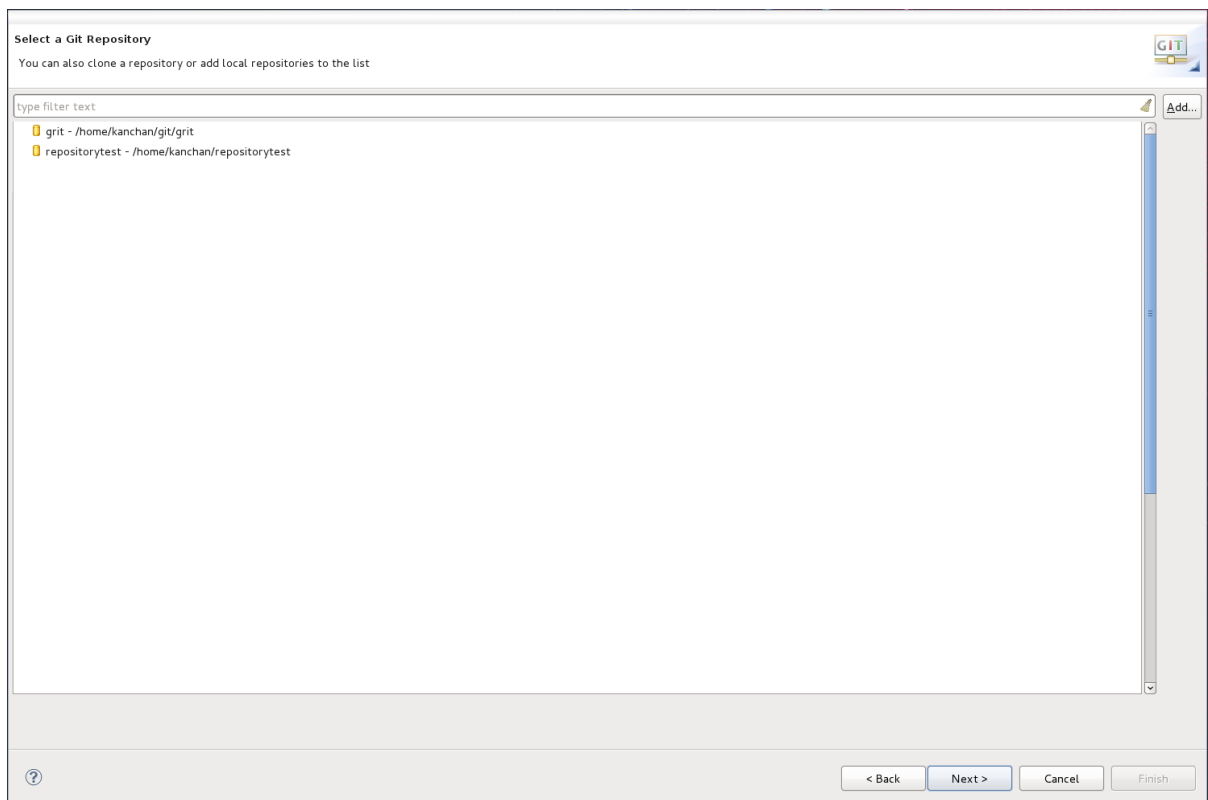


Figure 5.2. Git Repository Details

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**.

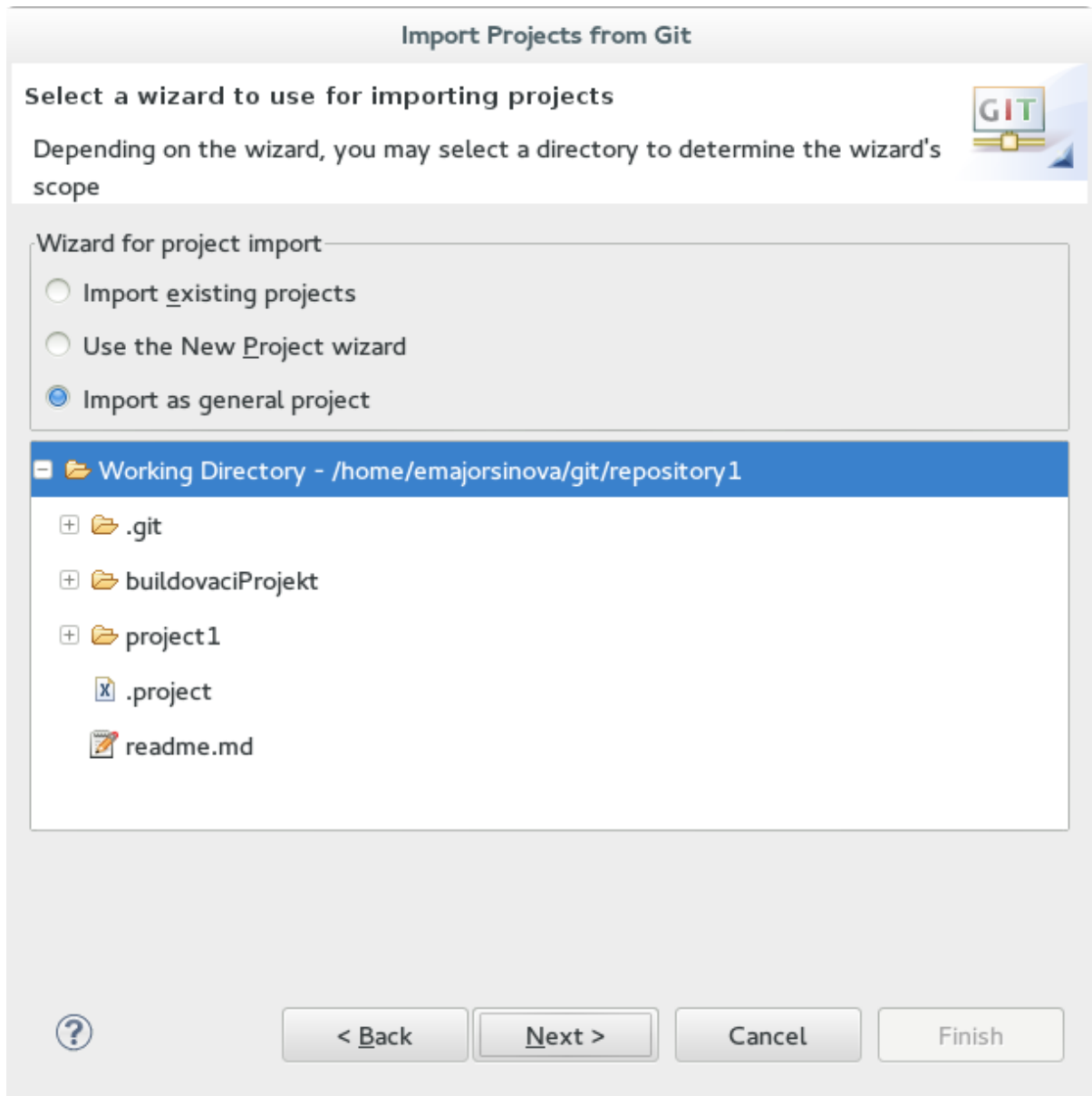


Figure 5.3. Wizard for Project Import

5.5. CREATING A DROOLS PROJECT

Procedure 5.6. Creating a New Red Hat JBoss Developer Studio Project

1. From the main menu, select **File** → **New** → **Project**.

Select **Drools** → **Drools Project** and click **Next**.

2. Enter a name for the project into the **Project name:** text box and click **Next**.



NOTE

JBoss Developer Studio provides the option to add a sample HelloWorld Rule file to the project. Accept this default by clicking **Next** to test the sample project in the following steps.

3. Select the Drools runtime (or use the default).

4. Select code compatible with **Drools 6.0.x**. Enter a **GroupID**, an **ArtifactID**, and **Version**, and click **Finish**.
5. To test the project, right click the Java file that contains the main method and select **Run** → **run as** → **Java Application**.

The output will be displayed on the console tab.

CHAPTER 6. 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.

6.1. 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 6.2.0.
5. Select **Red Hat JBoss BRMS 6.2.0 Business Resource Planner** and then click **Download**.

6.2. 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

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

Revision 6.2.0-7 Republished.	Fri Apr 21 2017	Tomas Radej
Revision 6.2.0-6 Translation fix.	Wed Apr 19 2017	Tomas Radej
Revision 6.2.0-5 Updated with latest fixes.	Thu Apr 28 2016	Tomas Radej
Revision 6.2.0-4 Build for release update 2 of JBoss BRMS.	Tue Mar 29 2016	Tomas Radej
Revision 6.2.0-3 Added note about versions in Revision History, fixed changelog dates.	Mon Nov 30 2015	Tomas Radej
Revision 6.2.0-2 Product names and minor typos corrected.	Mon Nov 30 2015	Tomas Radej
Revision 6.2.0-1 Initial build for release 6.2.0 of JBoss BRMS.	Mon Nov 30 2015	Tomas Radej