



Red Hat JBoss BRMS 6.2

IBM WebSphere Installation and Configuration Guide

For Red Hat JBoss BRMS

Red Hat JBoss BRMS 6.2 IBM WebSphere Installation and Configuration Guide

For Red Hat JBoss BRMS

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](#). 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

A guide to installing and configuring Red Hat JBoss BRMS on IBM WebSphere

Table of Contents

CHAPTER 1. INTRODUCTION	3
1.1. ABOUT RED HAT JBOSS BRMS	3
1.2. SUPPORTED PLATFORMS	3
1.3. USE CASE: PROCESS-BASED SOLUTIONS IN THE LOAN INDUSTRY	3
1.4. ABOUT RED HAT JBOSS BRMS FOR IBM WEBSPHERE 8	4
CHAPTER 2. DOWNLOAD AND EXTRACT	6
2.1. DOWNLOAD RED HAT JBOSS BRMS FOR IBM WEBSPHERE 8	6
2.2. EXTRACT RED HAT JBOSS BRMS FOR IBM WEBSPHERE 8	6
CHAPTER 3. CONFIGURE	7
3.1. INCREASE JVM HEAP SIZE	7
3.2. MODIFY SECURITY SETTINGS	7
3.3. SETUP DATASOURCE	8
3.4. ADD CUSTOM JVM PROPERTIES	10
CHAPTER 4. INSTALL RED HAT JBOSS BRMS ON IBM WEBSPHERE 8	12
4.1. INSTALL BUSINESS CENTRAL	12
4.2. INSTALL THE REALTIME DECISION SERVER	13
APPENDIX A. REVISION HISTORY	15

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

Business Resource Planner is included with this release.

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

1.2. SUPPORTED PLATFORMS

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

- Red Hat JBoss Enterprise Application Platform 6.4
- Red Hat JBoss Web Server 2.1 (Tomcat 7) on JDK 1.7
- IBM WebSphere Application Server 8.5.5.0
- Oracle WebLogic Server 12.1.3 (12c)

1.3. USE CASE: PROCESS-BASED SOLUTIONS IN THE LOAN INDUSTRY

This section describes a use case of deploying JBoss BPM Suite to automate business processes (such as loan approval process) at a retail bank. This use case is a typical process-based specific deployment that might be the first step in a wider adoption of JBoss BPM Suite throughout an enterprise. It leverages features of both business rules and processes of JBoss BPM Suite.

A retail bank offers several types of loan products each with varying terms and eligibility requirements. Customers requiring a loan must file a loan application with the bank. The bank then processes the application in several steps, such as verifying eligibility, determining terms, checking for fraudulent activity, and determining the most appropriate loan product. Once approved, the bank creates and funds a loan account for the applicant, who can then access funds. The bank must be sure to comply with all relevant banking regulations at each step of the process, and has to manage its loan portfolio to maximize profitability. Policies are in place to aid in decision making at each step, and those policies are actively managed to optimize outcomes for the bank.

Business analysts at the bank model the loan application processes using the BPMN2 authoring tools (Process Designer) in JBoss BPM Suite. Here is the process flow:

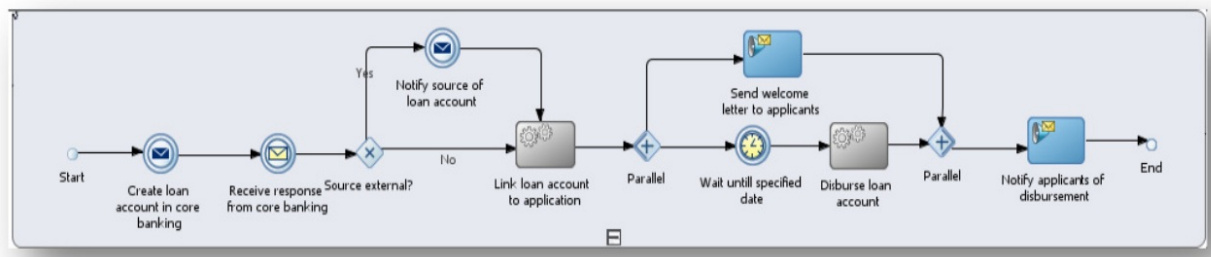


Figure 1.1. High-level loan application process flow

Business rules are developed with the rule authoring tools in JBoss BPM Suite to enforce policies and make decisions. Rules are linked with the process models to enforce the correct policies at each process step.

The bank's IT organization deploys the JBoss BPM Suite so that the entire loan application process can be automated.

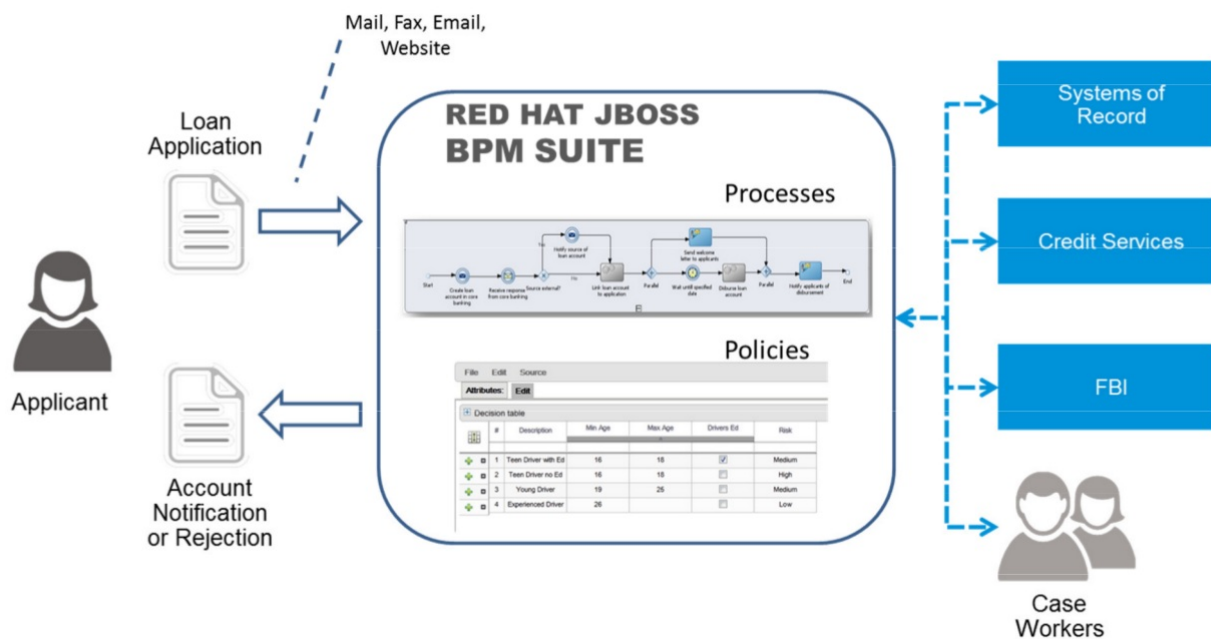


Figure 1.2. Loan Application Process Automation

The entire loan process and rules can be modified at any time by the bank's business analysts. The bank is able to maintain constant compliance with changing regulations, and is able to quickly introduce new loan products and improve loan policies in order to compete effectively and drive profitability.

1.4. ABOUT RED HAT JBOSS BRMS FOR IBM WEBSHERE 8

Red Hat JBoss BRMS for IBM WebSphere 8 is provided as a single deployable web application archive: business-centar.war.

Installation of JBoss BRMS on IBM WebSphere is supported from the 6.0.2 version of JBoss BRMS. In this guide, you will explore how it can be installed on a full profile version of IBM WebSphere 8.

Before installation, several configuration steps have to be performed to enable a successful setup. This guide will outline these steps.

Before you proceed, ensure you have root access to the server on which IBM WebSphere 8 is installed and that you are able to successfully access IBM WebSphere's administrative console via a web browser (Usually at <http://target-server:9060/ibm/console>).

As noted earlier, JBoss BRMS for IBM WebSphere is distributed as a WAR file. It is then deployed and configured as any other web application.

CHAPTER 2. DOWNLOAD AND EXTRACT

2.1. DOWNLOAD RED HAT JBOSS BRMS FOR IBM WEBSHERE 8

You can download the deployable Red Hat JBoss BRMS package file for WebSphere 8 from Red Hat Customer Portal:

1. Go to the [Red Hat Customer Portal](#) 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 6.2.
5. In the table, navigate to **Red Hat JBoss BRMS 6.2 Deployable for IBM WebSphere 8.5** row and then click **Download**.

2.2. EXTRACT RED HAT JBOSS BRMS FOR IBM WEBSHERE 8

The installation zip file for JBoss BRMS that you have just downloaded contains the Business Central WAR deployable (**business-central.war**) in un-exploded format.

Extract the zip file so that you have access to the deployable WAR.

CHAPTER 3. CONFIGURE

Before JBoss BRMS can be deployed as a web archive in IBM WebSphere 8, WebSphere server itself must be configured to accept the deployable. In this section, we will go through the different configuration steps that must be taken before deploying.

Login to your WebSphere 8 console using an administrative login before performing any of these steps. The usual login URL is: `http://target-server:9060/ibm/console`.

3.1. INCREASE JVM HEAP SIZE

With the default JVM heap size, the WebSphere 8 server freezes and/or causes deployment errors when deploying Business Central. Increase the heap size by following these steps:

1. Click on **Servers --> Server Types --> WebSphere Application Servers** in your WebSphere administrative console.
2. In the list of application servers, click on the server on which you are going to deploy Business Central. This will open up the configuration page for that server.
3. Under **Server Infrastructure**, click on **Java and Process Management** and then **Process Definition**. This will open the JVM process definition page.
4. Click on **Java Virtual Machine** under the Additional Properties heading. This will open up the configuration properties for the JVM that is used to start WebSphere.
5. Enter a value of 1536 for the maximum heap size changing it from the default of 1024. Click Apply at the bottom.
6. You can choose to save these configuration settings to the master WebSphere configuration at this stage.
7. Restart WebSphere at this point or wait till other configuration changes have been made.



NOTE

At this stage, you may want to increase the heap size in the Deployment Manager, if installed and being used. To do so, click on **System administration → Deployment manager → Java and process management → Process definition → Java virtual machine** in your WebSphere administrative console. Enter the maximum heap size and restart the Deployment Manager.

3.2. MODIFY SECURITY SETTINGS

Several security settings on IBM WebSphere 8 need to be changed and/or confirmed to be in place for the Business Central application to work on WebSphere 8. The following settings enable the container managed authentication mechanisms provided by the WebSphere server.

1. In WebSphere 8 administrative console, click on **Security --> Global Security**. Ensure that the option **Enable Application security** is checked. This may already be checked and overridden at the server level.
2. Click on **Custom properties** and then **New** to enter a new custom property with the following details: **Name: `com.ibm.ws.security.web.logoutOnHTTPSessionExpire`** and **Value: `true`**. This property instructs the server to invalidate LTPA tokens on session

invalidation, thereby making the logout process consistent across multiple users using the same browser.

Click **Apply** and then **OK**.

3. Next, click on **Users and Groups** --> **Manage Groups** and create 2 new groups: **admin**, and **analyst**.

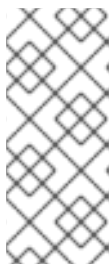


NOTE

Add the **kie-server** group as well if you will install the Realtime Decision Server. Also add the REST API groups if you will use API. For further information about API roles, refer to [Chapter 17. Remote API](#).

4. Click on **Users and Groups** --> **Manage Users** and create a user (for example: **business-central-admin**) and assign it to the **admin** user group that was created in the previous step.

Note that you should not create users with usernames that are identical to group names. For example, do not create a user with the username of **admin**.



NOTE

You may assign this user to any of the groups you have just created. In the actual production systems, you are likely to create separate users for separate groups that align with business roles. The **admin** group is all encompassing and is therefore useful for the purposes of this setup. Also note that users who need access to the REST API need to be also assigned to appropriate REST API roles.

5. Setup session management custom settings

Next, click on **Servers** --> **Server Types** --> **WebSphere Application Servers** in your WebSphere administrative console and then select the server on which you are deploying Business Central.

6. Click on **Session management** --> **Custom properties**. Click **NEW**.
7. In the settings page, enter ***InvalidateOnUnauthorizedSessionRequestException*** and set its value to **true**.
8. Click **Apply** and then **OK**.

3.3. SETUP DATASOURCE

The Business Central application requires a datasource which must be created prior to the deployment of the actual WAR file. This also means that you must have access to an underlying database that the datasource connects to. Whatever your underlying database, make sure you have the datasource ready. The steps below will help you set this up.

1. Open up the JDBC providers panel by clicking on **Resources** --> **JDBC** --> **JDBC Providers**.
2. Select the scope of this JDBC provider to include your server and node. It cannot be **All scopes**. Then click the **New...** button. This will bring up the **Create new JDBC provider** form page.

3. Fill out the form based on the database driver that you have available. If your database is not listed, select **Others** Database type selection box and provide the implementation class name. For example, for H2, Postgres or MySQL, the implementation class name will be *org.h2.jdbcx.JdbcDataSource*, *org.postgresql.xa.PGXADatasource* and *com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource* respectively.
4. Make sure to give the JDBC Provider a name that is descriptive and click the **Next** button. You will be asked to provide the CLASSPATH for the JDBC driver class files that you have just defined. Enter this value and click the **Next** button to be taken to the summary screen. Click the **Finish** button to accept and add this new JDBC Provider.

Using this new JDBC provider, you will now need to setup the actual datasource for Business Central. Before you create the datasource, however, open up the `persistence.xml` file in the Business Central war file that you have downloaded. You will need to know the name of the datasource defined within this file. It is normally *jdbc/jbpm*.

You will also need to change the `hibernate.dialect` to suit your actual database. For example, if your underlying database is DB2, you would change the this property from `org.hibernate.dialect.H2Dialect` to `org.hibernate.dialect.DB2Dialect`.

Setting up a Datasource

1. Open up the datasources panel by clicking on **Resources --> JDBC --> Data sources** in the WebSphere administrative panel. Click the **New...** button making sure the appropriate scope has been selected from the Scope drop-down.
2. Step 1 of the "Create a Data source" form is displayed. Enter a unique Data source name that you will use to refer to this datasource by. The JDBC provider is the one you created earlier in this section, and the JNDI name is the one you looked up from the `persistence.xml` file. Click **Next** and select the JDBC provider created earlier from the drop-down menu. Click **Next** 2 more times and accept the default values.
3. Click the **Finish** button to save the changes and then click to save the changes in the master configuration.
4. You will now need to provide the basic meta properties for this datasource. Properties like *servername*, *dbname*, *username* and *password* must now be defined and vary for different databases. These properties can be defined by editing the datasource just created and then clicking on **Custom properties** link under the **Additional properties** menu. Some example database properties are shown below.

For H2 databases, you will need to define the following custom properties:

- URL
- user
- password

For MySQL databases, you will need to define the following custom properties:

- serverName
- dbName
- port

- user
- password

Finally for Postgres databases, you will need to define the following custom properties:

- serverName
- databaseName
- portNumber
- user
- password

5. Once all the connection properties have been defined, click the **Test Connection** to ensure the validity of the datasource.

3.4. ADD CUSTOM JVM PROPERTIES

At this point, you need to add several custom properties to the JVM that is used to start IBM WebSphere 8. These custom properties take into consideration the configuration changes you have made so far.

1. Click on **Servers --> Server Types --> WebSphere Application Servers** in your WebSphere administrative console.
2. In the list of application servers, click on the server on which you are going to deploy Business Central. This will open up the configuration page for that server.
3. Under **Server Infrastructure**, click on **Java and Process Management** and then **Process Definition**. This will open the JVM process definition page.
4. Click on **Java Virtual Machine** under the Additional Properties heading. This will open up the configuration properties for the JVM that is used to start WebSphere.
5. Click on **Custom properties** under Additional Properties.
6. Create the following properties by pressing the **New . . .** button.
 - `org.jbpm.designer.perspective=ruleflow` - this argument on the command line forces the default perspective in the designer to **RuleFlow** instead of **Full**.
 - `org.jboss.logging.provider=jdk` - you only need to set this property if you have **CA SiteMinder TAI (SMTAI)** installed in your environment. By setting this property you are forcing Hibernate within Dashbuilder to use JDK instead of log4j for logging. **CA SiteMinder TAI (SMTAI)** contains an old version of log4j which causes conflicts.



NOTE

JBoss BRMS uses an embedded version of GIT for its artifact versioning. This version of GIT uses ports 9418 and 8003 for normal and SSH access, respectively.

Make sure that these embedded GIT ports are not already in use in your version of IBM WebSphere.

If these ports are being used and you need to change the default GIT ports, then change them by setting the `org.uberfire.nio.git.daemon.port` and `org.uberfire.nio.git.ssh.port` properties using the steps described here.

7. You can choose to save these configuration settings to the master WebSphere configuration at this stage.
8. Restart IBM WebSphere for these changes to take effect.

CHAPTER 4. INSTALL RED HAT JBOSS BRMS ON IBM WEBSHERE 8

Now that the basic configuration is done and IBM WebSphere 8 is set to deploy JBoss BRMS, you can upload the WAR deployable that was extracted earlier.

As noted earlier, the JBoss BRMS zip file for IBM WebSphere 8 contains the deployable WAR file for Business Central.

Let's install Business Central for BRMS on IBM WebSphere 8.

4.1. INSTALL BUSINESS CENTRAL

Business Central is uploaded as a Web archive and then accessed by a familiar URL: `http://target-server/business-central`. Start this deployment by installing the Business Central WAR as a WebSphere Application.

1. In your IBM WebSphere 8 administrative console, open up **Applications --> Application Types --> WebSphere enterprise applications**. This will show you all the existing applications in the system and allow you to install a new one.
2. Click the **Install** button to start the process.
3. Upload the Business Central war file (**business-central.war**) from the local file system. See [Section 2.2, “Extract Red Hat JBoss BRMS for IBM WebSphere 8”](#). This process may take some time.
4. Accept the default choices for most of the options presented including choosing the **Fast Path** option.
5. Change Application name to be **business-central** in Step 1 and click **Next**.
6. In Step 2, map the Business Central modules to servers depending upon your requirements.
7. Accept the default values for rest of the options and set the context root to: **business-central**. Click the **Finish** button to install Business Central at the Summary page. This process will take a while. Choose to save to the master configuration at the end of this process.
8. You will be returned to the **WebSphere enterprise applications** page where business-central will be a new application listed. However, this application will be stopped at this stage. Before you start it, there are a few application configuration options that need to be done.
9. **Map Groups to Roles**
Click on the business-central application and open up this applications configuration page. Click on **Security role to user/group mapping**.

Select the **admin** role, click **Map Groups . . .**, search for the **admin** group in the next screen (or just press the **Search** button) and move it from the **Available list** to the **Selected list**. Click **OK**. Mapping this will give the admin user that we created earlier (**business-central-admin**) access to the Business Central application.

Do the same for the **analyst** role and save the configuration.

If you have other groups or users that should have access to Business Central, use the same steps to map them to the **admin** or **analyst** roles.



NOTE

Give this user access to the `kie-server` role if you are also installing the Realtime Decision Server. Furthermore, map appropriate REST API roles if you will use REST API. For further information about API roles, refer to [Chapter 17. Remote API](#) from Red Hat JBoss BPM Suite Development Guide.

10. Classloading Configuration

Ensure by going to the **Class loading and update detection** setting for your business-central application that the parameters **Classes loaded with local class loader first (parent last)** and **Single class loader for application** are checked.

11. Click the **OK** button, choose to save to master configuration and restart the WebSphere server.

12. Enable Bouncy Castle Crypto API

To enable GIT SSH repository cloning and `kie-config-cli` from within Business Central, the Bouncy Castle Crypto API must be setup. Enable this API by following these steps:

1. Set the `org.apache.sshd.registerBouncyCastle` property to true and `org.uberfire.domain` property to `WSLogin`. These are custom properties so see [Section 3.4, “Add Custom JVM Properties”](#) for details on how to set them.
2. Setup the Bouncy Castle API as a shared library referenced from Business Central using the appropriate version of Bouncy Castle:
 1. Navigate to **Environment** → **Shared Libraries**, choose your scope and then click on the **New** button to create a new library. Give this library a name and set the classpath to the Bouncy Castle library (`bcprov-jdk16-1.46.jar`). Click **Ok** and save the configuration.
 2. Navigate to **Applications** → **Application Types**. Open up **WebSphere enterprise applications** and then click the Business Central application that you have just installed. Click on **Shared library references**, select the web module, click on **Reference shared libraries** and select the Bouncy Castle library created in the previous step. Click the **Ok** button and save the configuration.

You have now successfully installed Business Central on IBM WebSphere 8. Start the application by going back to **Applications** --> **WebSphere enterprise applications** page and then selecting **business-central** checkbox before clicking on the **Start** button.

To access the application, navigate to `http://target-server/business-central`.

4.2. INSTALL THE REALTIME DECISION SERVER

The Realtime Decision Server is distributed as a web application archive (WAR) file with the name of `kie-server.war` and is present in your IBM WebSphere download. To install it, follow these steps:



NOTE

It is assumed here that you have followed the steps to create the role `kie-server`, required by the Decision Server when you were installing Business Central. If you haven't done so, revisit the respective section in this guide.

1. In your IBM WebSphere 8 administrative console, open up **Applications --> Application Types --> WebSphere enterprise applications**. This will show you all the existing applications in the system and allow you to install a new one.
2. Click the **Install** button to start the process.
3. Upload the Decision Server WAR file (**kie-server.war**) from the local file system.
4. Accept the default choices for most of the options presented including choosing the **Fast Path** option.
5. Change Application name to **kie-server** in Step 1 and click **Next**.
6. In Step 2, map the KIE Server modules to servers depending upon your requirements.
7. Accept the default values for rest of the options and set the context root to: **kie-server**. Click the **Finish** button to install the Decision Server. Choose to save to the master configuration at the end of this process.
8. Navigate to **Applications → Application Types → WebSphere enterprise applications** and then click on **kie-server** (this should be in the stopped state). This will show you the specific application settings. Click on "Class loading and update detection". In the properties, change "Class loader order" to 'Classes loaded with local class loader first (parent last)' and "WAR class loader policy" to 'Single class loader for application'. Save this configuration.
9. If you have already mapped the **kie-server** role to a user/group, you may ignore this step, otherwise, go back to the main configuration page for this newly installed server application. Click on **Security role to user/group mapping**.

Select the **kie-server** role, click **Map Groups . . .**, search for the **admin** group in the next screen (or just press the **Search** button) and move it from the **Available list** to the **Selected list**. Click **OK**. Mapping this will give the admin user that we created earlier (for example, **business-central-admin**) access to the Realtime Decision Server

10. Save and start the **kie-server** application.

You can check whether the Realtime Decision Server REST API works by sending a GET request at <http://target-server/kie-server/services/rest/server>.

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 6.2.0-4 Updated with latest fixes.	Thu Apr 28 2016	Tomas Radej
Revision 6.2.0-3 Build for release update 2 of JBoss BRMS.	Tue Mar 29 2016	Tomas Radej
Revision 6.2.0-2 Added note about versions in Revision History, fixed changelog dates.	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