



Red Hat JBoss BRMS 6.4

Oracle Weblogic Installation and Configuration Guide

For Red Hat JBoss BRMS

Red Hat JBoss BRMS 6.4 Oracle Weblogic Installation and Configuration Guide

For Red Hat JBoss BRMS

Red Customer Content Services

brms-docs@redhat.com

Emily Murphy

Gemma Sheldon

Michele Haglund

Mikhail Ramendik

Stetson Robinson

Vidya Iyengar

Legal Notice

Copyright © 2019 Red Hat, Inc.

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

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

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

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

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

A guide to installing and configuring Red Hat JBoss BRMS on Oracle Weblogic.

Table of Contents

CHAPTER 1. INTRODUCTION	3
1.1. ABOUT RED HAT JBOSS BRMS FOR ORACLE WEBLOGIC SERVER	3
CHAPTER 2. DOWNLOAD AND EXTRACT	4
2.1. DOWNLOAD RED HAT JBOSS BRMS FOR ORACLE WEBLOGIC SERVER	4
2.2. EXTRACT RED HAT JBOSS BRMS FOR ORACLE WEBLOGIC SERVER	4
CHAPTER 3. CONFIGURE	5
3.1. SETTING ENVIRONMENT VARIABLES	5
JVM Memory Size	5
JVM Custom Properties	5
3.2. CONFIGURING SECURITY SETTINGS	6
3.3. CREATING DATA SOURCE	6
3.4. CONFIGURING JAVA MESSAGE SERVICE (JMS)	8
Create JMS Module	8
Create JMS Connection Factories	8
Create JMS Queues	9
CHAPTER 4. INSTALL	10
4.1. INSTALLING BUSINESS CENTRAL	10
4.2. INSTALLING REALTIME DECISION SERVER	10
CHAPTER 5. NEXT STEPS	12
APPENDIX A. ADDITIONAL NOTES	13
APPENDIX B. VERSIONING INFORMATION	14

CHAPTER 1. INTRODUCTION

1.1. ABOUT RED HAT JBOSS BRMS FOR ORACLE WEBLOGIC SERVER

Red Hat JBoss BRMS for Oracle WebLogic Server is provided as two deployable web application archives:

- **business-central.war** – The main business rules management application.
- **kie-server.war** – Application for executing rules through REST, JMS or a Java client side application.

Installation of Red Hat JBoss BRMS on Oracle WebLogic Server is supported from the 6.1 version of Red Hat JBoss BRMS. In this guide, you will explore how it can be installed on a full profile version of Oracle WebLogic Server.

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

Before you proceed, make sure you have access to the server on Oracle WebLogic Server and that you are able to successfully access Oracle WebLogic Server's administrative console through a web browser (usually at **http://TARGET_SERVER:7001/console**).

As noted earlier, Red Hat JBoss BRMS for Oracle WebLogic Server is distributed as a WAR file. It is then deployed as an exploded archive and then configured as any other web application.

CHAPTER 2. DOWNLOAD AND EXTRACT

2.1. DOWNLOAD RED HAT JBOSS BRMS FOR ORACLE WEBLOGIC SERVER

You can download the deployable Red Hat JBoss BRMS package file for Oracle WebLogic Server from Red Hat Customer Portal:

1. Go to the [Red Hat Customer Portal](#) and log in.
2. Click **DOWNLOADS** at the top of the page.
3. In the **Product Downloads** page that opens, click **Red Hat JBoss BRMS**.
4. From the **Version** drop-down menu, select **6.4**.
5. On the **Releases** tab, navigate to **Red Hat JBoss BRMS 6.4.0 Deployable for Oracle Weblogic 12c** and click **Download**.
6. On the **Patches** tab, download the latest patch (if applicable).

2.2. EXTRACT RED HAT JBOSS BRMS FOR ORACLE WEBLOGIC SERVER

The installation ZIP file for Red Hat JBoss BRMS that you have downloaded contains all necessary WAR deployable archives mentioned in [Section 1.1, "About Red Hat JBoss BRMS for Oracle WebLogic Server"](#).

Copy the installation ZIP file to your WebLogic Server host and extract it to a location the server can access.

```
unzip ~/jboss-brms-VERSION-deployable-wls12c.zip
```

See the [Patching and Upgrading Red Hat JBoss BRMS](#) chapter from *Red Hat JBoss BRMS Installation Guide* for instructions on applying the patch updates.

CHAPTER 3. CONFIGURE

3.1. SETTING ENVIRONMENT VARIABLES

Certain environment variables on your Oracle WebLogic Server require configuration before deploying the application.

JVM Memory Size

With the default JVM memory size, the WebLogic Server freezes and/or causes deployment errors when deploying Business Central. Increase the memory size by setting the following environment variable:

```
USER_MEM_ARGS=-Xms512m -Xmx1024m -XX:MaxPermSize=512m
```

JVM Custom Properties

The following custom properties require configuration on the WebLogic Server:

Properties Required for Business Central and Realtime Decision Server

- **org.kie.executor.jms.cf**: JNDI name of the connection factory for sending messages with job executor requests.
- **org.kie.executor.jms.queue**: JNDI name of the destination (queue) for sending messages with job executor requests.

Properties Required for Business Central

- **org.jbpm.designer.perspective=ruleflow**: Changes the default perspective in the designer to **RuleFlow** instead of **Full**.
- **org.uberfire.start.method**: Defines startable beans for Uberfire. Set to **ejb** (Enterprise Java Beans).
- **org.uberfire.domain**: Sets the domain for Uberfire to use. Set to **OracleDefaultLoginConfiguration**.
- **com.sun.jersey.server.impl.cdi.lookupExtensionInBeanManager**: Uses **BeanManager** to look up extensions. This avoids conflicts across multiple instances. Set to **true**.

Properties Required for Realtime Decision Server

- **kie.server.jms.queues.response**: JNDI name of the response queue for the Realtime Decision Server. Set to **jms/KIE.SERVER.RESPONSE**.
- **org.kie.server.domain**: JAAS LoginContext domain used to authenticate users when using JMS. Set to **OracleDefaultLoginConfiguration**.
- **org.jbpm.server.ext.disabled**: Set to **true** to disable BPM support (for example, processes support).
- **org.jbpm.ui.server.ext.disabled**: Set to **true** to disable the Intelligent Process Server UI extension..

For a full list of available Realtime Decision Server system properties, see chapter [Realtime Decision Server](#) of the Red Hat JBoss BRMS *User Guide*.

Set the following custom properties in a single environment variable:

```
JAVA_OPTIONS="-Dkie.server.jms.queues.response=jms/KIE.SERVER.RESPONSE
-Dorg.jbpm.designer.perspective=ruleflow
-Dorg.uberfire.start.method=ejb
-Dorg.uberfire.domain=OracleDefaultLoginConfiguration
-Dorg.kie.executor.jms.cf=jms/cf/KIE.EXECUTOR
-Dorg.kie.executor.jms.queue=jms/KIE.EXECUTOR
-Dcom.sun.jersey.server.impl.cdi.lookupExtensionInBeanManager=true
-Dorg.jbpm.server.ext.disabled=true
-Dorg.jbpm.ui.server.ext.disabled=true"
```

3.2. CONFIGURING SECURITY SETTINGS

Several security settings on Oracle WebLogic Server must be set for the Business Central application to work. The following settings enable the container managed authentication mechanisms provided by the WebLogic server:

1. In the WebLogic administrative console, click **Security Realms**.
2. Choose your desired security realm or click **New** to create a new security realm.
3. Navigate to **Users and Groups** → **Groups** to get to the groups list for your security realm.
4. Click **New** to create a new group. Create the following three new groups: **admin**, **analyst**, and **kie-server**. Also create the REST API groups if you will use API. For further information about API roles, see section [REST API](#) of *Red Hat JBoss BPM Suite Development Guide* .
5. Click on the **Users** tab and click **New** to create a new user. Provide this new user with a name (for example **business-central-admin**) and a password. Click **OK** to save.



IMPORTANT

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

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

6. Click on the newly created user, then click the **Groups** tab. Use the selection tool to move the **admin** group from the **Available** field to the **Chosen** field. Click **Save** to save.



NOTE

You may assign this user to any of the groups previously created and in 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.

3.3. CREATING DATA SOURCE

The Business Central application requires a data source 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 data source connects to. Whatever your underlying database, make sure you have the data source

ready.

1. Navigate to **Services** → **Data Sources**, which takes you to your JDBC Data Sources list.
2. Click **New** → **Generic Data Source** to start creating a new data source.
3. Provide your data source with the following details:
 - **Name:** enter a name for your data source.
 - **JNDI Name:** set to **jdbc/jbpm**.
 - **Database Type:** set to **MySQL**.



NOTE

You can use alternative database types by editing **WEB-INF/classes/META-INF/persistence.xml** in the Business Central WAR archive. Edit the **hibernate.dialect** property to your preferred database.

For example, to change to Oracle 12c, edit the **hibernate.dialect** to the following:

```
<property name="hibernate.dialect"
value="org.hibernate.dialect.Oracle10gDialect" />
```

Click **Next** to advance to the next configuration screen.

4. Select your database driver in the drop-down menu. Click **Next** to advance to the next configuration screen.
5. Leave the **Transaction Options** as the defaults and click **Next** to advance to the next configuration screen.
6. Provide the following **Connection Properties** for your data source:
 - **Database Name:** the name of the database to use on your data source.
 - **Host Name:** the hostname or IP address of the system containing the database.
 - **Port:** the port used to connect to the database. Unless you have configured the database to use a different port, use the default port provided in this field.
 - **Database User Name:** the database user that interacts with the database. Make sure the chosen user has the required permissions to access and write to the chosen database.
 - **Password:** the password for the chosen database user.

Click **Next** to advance to the next configuration screen.

7. The **Test Database Connection** page provides a means to test and confirm your database connection. Click **Test Configuration** and the page will refresh with a valid connection message. Click **Finish** to complete the data source configuration.
8. Click on the name of the new data source and navigate to **Targets** tab. Click the checkbox for the server chosen to host the Business Central deployment. Click **Save** to save your selection.

3.4. CONFIGURING JAVA MESSAGE SERVICE (JMS)

Oracle WebLogic Server must be configured to send and receive JMS messages through Red Hat JBoss BRMS Realtime Decision Server. This requires a JMS server. Follow the steps below to create a JMS server:

1. Navigate to **Services** → **Messaging** → **JMS Servers**.
2. Click **New** to start creating a new JMS Server.
3. Provide your JMS Server with a name. Click **Next** to advance to the next configuration screen.
4. Select the **Target** server chosen for the Business Central deployment. Click **Finish** to complete the JMS Server creation.

Create JMS Module

A JMS Module stores your JMS resources, such as connection factories and queues. Use the following steps to create a new JMS Module:

1. Navigate to **Services** → **Messaging** → **JMS Modules**.
2. Click **New** to start creating a new module.
3. Provide your module with a name and click **Next** to advance to the next configuration screen.
4. Select the **Target** server chosen for the Realtime Decision Server deployment. Click **Finish** to complete the JMS Module creation.
5. Click on the newly created module's name, then click on **Subdeployments**.
6. Click **New** to create a subdeployment for your module.
7. Provide your subdeployment with a name and click **Next** to advance to the next configuration screen.
8. Choose the previously created JMS Server by marking the checkbox. Click **Finish** to complete the subdeployment configuration.

Create JMS Connection Factories

To send and receive messages from Red Hat JBoss BRMS Realtime Decision Server, you will need to create the JMS connection factories – one for receiving messages and one for sending them. The following connection factories are required:

- **KIE.SERVER.REQUEST**: for all requests to the Realtime Decision Server.
Default value: **jms/cf/KIE.SERVER.REQUEST**.
- **KIE.SERVER.RESPONSE**: receiving all responses produced by the Realtime Decision Server.
Default value: **jms/cf/KIE.SERVER.RESPONSE**.
- **KIE.EXECUTOR**: sending executor requests for jobs running in Business Central.
Default value: **jms/cf/KIE.EXECUTOR**.
- **KIE.SERVER.EXECUTOR**: sending executor requests for jobs running on the Realtime Decision Server.
Default value: **jms/cf/KIE.SERVER.EXECUTOR**.

Use the following procedure to create each connection factory:

1. Navigate to **Services** → **Messaging** → **JMS Modules** to see your list of JMS Modules.
2. Click on your previously created module, then click **New** to start creating a new JMS resource.
3. Select **Connection Factory** and click **Next**.
4. Enter the name of the connection factory (for example **KIE.SERVER.REQUEST**) and the JNDI name (for example **jms/cf/KIE.SERVER.REQUEST**).
Click **Next** to advance to the next configuration screen.
5. The connection factory automatically selects the servers assigned to the JMS Module as the default. Click **Finish** to complete the connection factory creation.

Repeat the above procedure for each connection factory.

Create JMS Queues

You now need to create the JMS Queues. These queues are the destination end points for point-to-point messaging. You will create:

- **KIE.SERVER.REQUEST**: for all requests to the Realtime Decision Server.
Default value: **jms/KIE.SERVER.REQUEST**.
- **KIE.SERVER.RESPONSE**: for the Realtime Decision Server responses.
Default value: **jms/KIE.SERVER.RESPONSE**.
- **KIE.EXECUTOR**: sending executor requests for jobs running in Business Central.
Default value: **jms/KIE.EXECUTOR**.
- **KIE.SERVER.EXECUTOR**: sending executor requests for jobs running on the Realtime Decision Server.
Default value: **jms/KIE.SERVER.EXECUTOR**.

Use the following procedure to create each queue:

1. If you are not there already, navigate to **Services** → **Messaging** → **JMS Modules** to see your list of JMS Modules.
2. Click on your previously created module, then click **New** to start creating a new JMS resource.
3. Select **Queue** and click **Next**.
4. Enter the name of the queue (for example **KIE.SERVER.REQUEST**) and the JNDI name (for example **jms/KIE.SERVER.REQUEST**).
Click **Next** to advance to the next configuration screen.
5. Choose the JMS Module subdeployment that connects to the JMS Server. Click **Finish** to complete the queue creation.

Repeat the above procedure for each queue.

CHAPTER 4. INSTALL

Now that the basic configuration is done, the Oracle WebLogic Server is set to deploy Red Hat JBoss BRMS.

As noted earlier, the Red Hat JBoss BRMS ZIP file for Oracle WebLogic Server contains the deployable WAR file for Business Central and Realtime Decision Server.

4.1. INSTALLING BUSINESS CENTRAL

Business Central is uploaded as a web archive and then accessed by a familiar through a web browser. Start this deployment by installing the Business Central exploded WAR as a WebLogic Application.

1. In your Oracle WebLogic Server administrative console, click **Deployments**. This will show you all the existing applications in the system and allow you to install a new one.
2. Click **Install** to start the process.
3. Navigate to the exploded archive location for **business-central.war** and select it. Click **Next** to continue.
4. Select **Install this deployment as an application** as the targeting style and click **Next**.
5. Set the application name to **business-central** and set the security model to **DD Only**. Leave the remaining options as default and click **Next** to continue.
6. In the **Additional Configuration** section, choose **No, I will review the configuration later** and click **Finish**.
7. To ensure correct functionality of process diagram visualization, copy **xercesImpl.jar** to the server libraries, which is in **/DOMAIN_HOME/lib/** by default.

You have now successfully installed Business Central on Oracle WebLogic Server. To access the application, navigate to **http://TARGET_SERVER:7001/business-central**.

4.2. INSTALLING REALTIME DECISION SERVER

The Realtime Decision Server is distributed as a deployable WAR. Follow the steps below to install the server:

1. Navigate to **Deployments**. This will show you all the existing applications in the system and allow you to install a new one.
2. Click **Install** to start the process.
3. Navigate to the exploded archive location for **kie-server.war** and select it. Click **Next** to continue.
4. Select **Install this deployment as an application** as the targeting style and click **Next**.
5. Set the application name to **kie-server** and set the security model to **DD Only**. Leave the remaining options as default and click **Next** to continue.
6. In the **Additional Configuration** section, choose **No, I will review the configuration later** and click **Finish**.

7. To ensure correct functionality of UI extension, copy **xercesImpl.jar** to the server libraries, which is in **/DOMAIN_HOME/lib/** by default.

You can now access the Realtime Decision Server at **http://TARGET_SERVER:7001/kie-server/services/rest/server**.

CHAPTER 5. NEXT STEPS

Now that you have completed the installation, use the following guides to start using Red Hat JBoss BRMS:

- *Red Hat JBoss BRMS Getting Started Guide* : provides an introductory tutorial on the core features of Red Hat JBoss BRMS.
- *Red Hat JBoss BRMS User Guide* : provides steps on how to start creating your business rules.
- *Red Hat JBoss BRMS Administration and Configuration Guide* : provides steps on how to configure aspects of your Red Hat JBoss BRMS deployment, including migration, data management, imports and exports, integration, and monitoring.

APPENDIX A. ADDITIONAL NOTES

- The Oracle WebLogic Server class-loading mechanism does not provide access to the application JARs located in your application's **WEB-INF/lib** directory, which can cause problems when looking to inject KIE-CDI classes. As a workaround, Red Hat JBoss BRMS includes a CDI extension that temporarily swaps class loaders to load the application: **org.kie.workbench.backend.weblogic.SwapClassLoaderExtension**.

Enabling CDI Extension workaround

- Get the extension's class file either by using the pre-compiled **SwapClassLoaderExtension.class** file found in Business Central's **business-central.war/WEB-INF/classes/org/kie/workbench/backend/weblogic/** subdirectory, or by downloading and compiling its [source](#). Put the file in your application's **/WEB-INF/classes/org/kie/workbench/backend/weblogic/** directory.
- Enable the extension by putting its fully qualified name in the **WEB-INF/classes/META-INF/services/javax.enterprise.inject.spi.Extension** file in your application. If the file does not exist, create it.

Contents of javax.enterprise.inject.spi.Extension

```
org.kie.workbench.backend.weblogic.SwapClassLoaderExtension
```



NOTE

When using Business Central, these steps are not necessary—the workaround has already been applied.

- If using a non-exploded archive, Weblogic packs contents of Business Central's **WEB-INF/classes** into **WEB-INF/lib/_wl_cls_gen.jar**. If a developer aims to create **KieBase** and **KieSession** from resources, KIE-Spring cannot find the classes. For this reason, the recommended deployment method is to use the exploded archives contained within the product's ZIP file.

APPENDIX B. VERSIONING INFORMATION

Documentation last updated on: Monday, May 13, 2019.