



# **Red Hat Decision Manager 7.0**

## **Migrating from Red Hat JBoss BRMS 6.4 to Red Hat Decision Manager 7.0**



# Red Hat Decision Manager 7.0 Migrating from Red Hat JBoss BRMS 6.4 to Red Hat Decision Manager 7.0

---

Red Hat Customer Content Services  
brms-docs@redhat.com

## Legal Notice

Copyright © 2018 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

This document describes how to migrate user data and APIs from Red Hat BRMS 6.4 to Red Hat Decision Manager 7.0.

## Table of Contents

|  |           |
|--|-----------|
| <b>PREFACE</b> .....   | <b>3</b>  |
| <b>CHAPTER 1. OVERVIEW</b> .....   | <b>4</b>  |
| <b>CHAPTER 2. MIGRATING PROJECTS IN DECISION CENTRAL</b> .....                             | <b>5</b>  |
| <b>CHAPTER 3. MIGRATING JAVA PROJECTS</b> .....  | <b>7</b>  |
| 3.1. TROUBLESHOOTING BUILD AND EXECUTION ERRORS IN MIGRATED JAVA PROJECTS                  | 8         |
| 3.1.1. API changes in Drools modules   | 8         |
| 3.1.2. Logic changes in Drools modules   | 9         |
| <b>CHAPTER 4. RED HAT BUSINESS OPTIMIZER CHANGES IN RED HAT DECISION MANAGER 7.0</b> ..... | <b>11</b> |
| 4.1. UPDATING RED HAT BUSINESS OPTIMIZER ASSET CONFIGURATIONS IN DECISION CENTRAL          | 11        |
| <b>CHAPTER 5. UPGRADING A DECISION SERVER</b> .....  | <b>13</b> |
| <b>APPENDIX A. VERSIONING INFORMATION</b> .....  | <b>15</b> |



# PREFACE

If you have Red Hat BRMS 6.4 projects that you want to use in Red Hat Decision Manager 7.0, you must first migrate the projects.

## **Prerequisite**

You have a Red Hat BRMS 6.4 installation that contains artifacts that you want to migrate to Red Hat Decision Manager 7.0.

## CHAPTER 1. OVERVIEW

If you use BRMS version 6.4 and install Red Hat Decision Manager version 7.0, you need to migrate your existing projects to the new product.

You can move the applications (now called *decision services*) that you developed using the Business Central workbench to the new equivalent, Decision Central. You do not need to change the underlying code manually to complete this migration.

Decision services that you developed in Java code (for example, in Eclipse) require modification for Red Hat Decision Manager 7.0. You must update the dependencies in the **pom.xml** file for each project and rebuild the project. Some changes in the new version might cause known build errors; in this case, you must modify the code.

To migrate your Java client applications, you also must update the dependencies in the **pom.xml** file for each project. If your application uses embedded BRMS engines (Drools, OptaPlanner), this change also updates the engines. If the application calls the Decision Server, the API client library is updated.

If your client application is not developed with the Java client library and uses the REST API to interact with the KIE server (Decision Server), you might need to modify it to adapt to a small incompatible API change.

You can replace your KIE servers with new Decision Servers simply by stopping the old servers and starting the new ones on the same hosts. You can use Decision Server 7.0 to run KJAR files created in BRMS 6.4. However, for optimal performance, migrate your projects to Red Hat Decision Manager 7.0 and rebuild them.



### IMPORTANT

If you are using a version of BRMS older than 6.4, migrate your projects to version 6.4 before migrating to 7.0. See the [Red Hat JBoss BPM Suite 6.4 Migration Guide](#).




## CHAPTER 2. MIGRATING PROJECTS IN DECISION CENTRAL

If you have created projects in the Business Central workbench in BRMS 6.4, you can migrate them to Red Hat Decision Manager 7.0 by importing the asset repository into Decision Central.

### Prerequisites

1. In BRMS 6.4 Business Central, look up the names of the repositories that contain your project. The default repository name is **repository1**.
2. Stop BRMS 6.4 Business Central.
3. Locate the **.niogit** directory. In the default installation, its location is **\$EAP\_home/bin/.niogit**. This location can be changed with the **org.uberfire.nio.git.dir** system property in the **\$EAP\_home/standalone/configuration/standalone.xml** file. The system property can also be set with a **-D** option in the command that starts Business Central.
4. Install the Red Hat Decision Manager, including Decision Central, on the same host as BRMS 6.4 Business Central, using a separate JBoss EAP installation. For instructions, see [Installing Red Hat Decision Manager on premise](#).

### Procedure

1. Log in to Red Hat Decision Manager 7.0 Decision Central.
2. In the main menu, select **Projects**.
3. Click the  icon and select **Import Project**.
4. In the **Repository URL** field, type the URL:

```
file://<path>/niogit/<repository>.git
```

For example, if the location of the **.niogit** directory is **/opt/eap7.0/bin/.niogit** and you want to import projects from the **repository1** repository:

```
file:///opt/eap7.0/bin/.niogit/repository1.git
```

5. Click **Import**.
6. Select the projects to import and click **Import**.
7. A **NoRemoteRepositoryException** exception might be displayed. In this case complete the following steps:
  - a. In a shell terminal, change to the repository subdirectory (under the **.niogit**) directory you are importing
  - b. List the remotes: **git remote -v**
  - c. Remove all listed remotes: **git remote remove <remote-name>**  
For example:

–

```
$ /opt/eap7.0/bin/.niogit/repository1.git
$ git remote -v
origin  https://github.com/guvnorngtestuser1/guvnorng-playground
(fetch)
origin  https://github.com/guvnorngtestuser1/guvnorng-playground
(push)
$ git remote remove origin
```

Then import the projects again. (You can add the remotes in the new repository later if necessary).

8. Wait for the importing and indexing to complete.
9. Repeat the process for any other repositories that you wish to import.

## Result

The projects are imported into Decision Central. You can build and deploy them as normal, using the **Build & Deploy** button.



### IMPORTANT

If you have developed the project in Java and then imported it into the Business Central workbench, you might need to migrate the project as a Java project. For instructions, see [Chapter 3, \*Migrating Java projects\*](#).

## CHAPTER 3. MIGRATING JAVA PROJECTS

For Java projects that are not developed in Business Central workbench, you must modify the dependencies in the **pom.xml** file and resolve any build issues. This process is the same for decision services and client applications written in Java.

### Prerequisite

You must download the Maven repository for Red Hat Decision Manager 7.0 and make it available to the local Maven installation (as a local or remote repository).

### Procedure

1. Open the **pom.xml** file of the project.
2. Remove the **<version>** tag from any dependencies under the following groups:
  - **org.kie**
  - **org.drools**
  - **org.jbpm**
  - **org.optaplanner**
3. Add the following dependency under the **<dependencies>** tag within the **<dependencyManagement>** section:

```
<dependency>
  <groupId>org.jboss.bom.rhdm</groupId>
  <artifactId>rhdm-platform-bom</artifactId>
  <version>${rhdm.version}</version>
  <scope>import</scope>
  <type>pom</type>
</dependency>
```

Where **\${rhdm.version}** is the version of the **org.jboss.bom.rhdm:rhdm-platform-bom** artifact in the Maven repository for Red Hat Decision Manager 7.0. You can view the version of the artifact by entering the repository and navigating to **maven-repository/org/jboss/bom/rhdm/rhdm-platform-bom** directory; the name of the only subdirectory in this directory is the version. For example, **7.0.0.GA-redhat-1**.

4. If your code uses any Drools CDI annotations (**@KReleaseId**, **@KContainer**, **@KBase**, **@KSession**), also add the following dependency:

```
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-cdi</artifactId>
</dependency>
```

This dependency is necessary because the CDI extension for processing the annotations is now defined using a separate module. Without this dependency, the annotations do not work.

5. Replace the legacy **org.guvnor** dependency (if applicable) with the new **org.uberfire** dependency.

Legacy **org.guvnor** dependency:

```
<dependency>
  <groupId>org.guvnor</groupId>
  <artifactId>guvnor-rest-client</artifactId>
</dependency>
```

New **org.uberfire** dependency:

```
<dependency>
  <groupId>org.uberfire</groupId>
  <artifactId>uberfire-rest-client</artifactId>
</dependency>
```

6. Update the version of the project artifact and save the **pom.xml** file.
7. Rebuild the project, using a regular Maven build.
8. If build errors or subsequent execution errors happen, resolve them using the information in [Section 3.1, “Troubleshooting build and execution errors in migrated Java projects”](#).

## 3.1. TROUBLESHOOTING BUILD AND EXECUTION ERRORS IN MIGRATED JAVA PROJECTS

Use the following information to resolve any build or execution errors after rebuilding migrated projects for Red Hat Decision Manager 7.0.

### 3.1.1. API changes in Drools modules

The following Drools modules include some incompatible changes from BRMS 6.4 to Red Hat Decision Manager 7.0:

- **kie-api**: the main API for all projects from KIE Group. The majority of changes are in the following classes:
  - **org.kie.api.task.\***
  - **org.kie.api.executor.\***
  - **org.kie.api.concurrent.\***
  - **org.kie.api.builder.\***
  - **org.kie.api.command.\***
  - **org.kie.api.runtime.\***
- **drools-core**: the rules engine. The changed classes are:
  - **org.drools.core.command.\***
  - **org.drools.core.common.\***

- **kie-server-api**: the general API, including commands,model and so on, for KIE Server (Decision Server). The changed classes are:
  - **org.kie.server.api.commands.\***
  - **org.kie.server.api.marshalling.\***
  - **org.kie.server.api.model.\***
  - **org.kie.server.api.rest.RestURI** (constants have slightly changed, omitting leading /)
- **kie-server-controller-api**: the API for the KIE Server Controller. The changed classes are:
  - **org.kie.server.controller.api.service.\***
- **kie-server-controller-rest**: the REST API for KIE Server Controller. The changed classes are:
  - **org.kie.server.controller.rest.RestSpecManagementServiceImpl**
  - **org.kie.server.controller.rest.RestKieServerControllerImpl**
- **kie-server-client**: the KIE Server Client. The changed classes are:
  - **org.kie.server.client.SolverServicesClient**
  - **org.kie.server.client.UIServicesClient**
  - **org.kie.server.client.admin.ProcessAdminServicesClient**
  - **org.kie.server.client.ProcessServicesClient**
  - **org.kie.server.client.QueryServicesClient**
  - **org.kie.server.client.JobServicesClient**
  - **org.kie.server.client.UserTaskServicesClient**
  - **org.kie.server.client.KieServicesClient**
  - **org.kie.server.client.KieServicesConfiguration**

If you encounter build errors with any of the listed classes, review the detailed reports that are available at <https://access.redhat.com/articles/3352151> . The reports indicate the API changes. You can use this information to adapt your code to the changes.

### 3.1.2. Logic changes in Drools modules

The following logic changes were made in Drools modules:

- In Red Hat BRMS 6.4, when a rule executes the sum function in an accumulate pattern, the result always returns as a double data type regardless of the data type of the inputs. Red Hat Decision Manager 7.0 preserves the data type of the inputs on which the sum is executed. This

enhancement provides a more accurate result from the sum function. In the following example, the result type of the accumulate function is **Long** instead of **double**:

```
Long(...) from accumulate(..., sum($p.getLongWeight()))
```

If the rules in your Red Hat BRMS 6.4 project include the sum function in an accumulate pattern, search for these functions and review them. Keep in mind that these functions return a double data type in Red Hat BRMS 6.4, but will return the data type of the input values in Red Hat Decision Manager 7.0.

- When no fact matches the accumulate pattern, min and max accumulate functions in Red Hat Decision Manager 7.0 do not return **+/-Integer.MAX\_VALUE**, but return **null**, therefore, unlike BRMS 6.4, the accumulate in the rule is not matched and the rule does not fire.

## CHAPTER 4. RED HAT BUSINESS OPTIMIZER CHANGES IN RED HAT DECISION MANAGER 7.0

Red Hat Business Optimizer is an embeddable planning engine in Red Hat Decision Manager that optimizes planning problems. Red Hat Business Optimizer is based on the community OptaPlanner project that is regularly updated and in some cases requires code changes for the latest Red Hat Business Optimizer features. For an overview of the latest OptaPlanner changes and migration requirements, see the [OptaPlanner upgrade recipe archive](#). OptaPlanner upgrade information for versions 7.0 through 7.6, inclusive, is relevant for upgrading from Red Hat JBoss BRMS 6.4 to Red Hat Decision Manager 7.0.

In Red Hat Decision Manager 7.0, certain Red Hat Business Optimizer configurations in Decision Central must be updated to accommodate recent OptaPlanner changes.

### 4.1. UPDATING RED HAT BUSINESS OPTIMIZER ASSET CONFIGURATIONS IN DECISION CENTRAL

If you have any solver configuration assets (`.solver.xml` files) or solution-related data objects in Decision Central, you must make certain updates to these assets in Red Hat Decision Manager 7.0 to accommodate recent Red Hat Business Optimizer changes.

#### Prerequisite

Decision Central data has been migrated from Red Hat JBoss BRMS to Red Hat Decision Manager 7.0. For migration instructions, see [Chapter 2, Migrating projects in Decision Central](#).

#### Procedure

1. Log in to Decision Central for Red Hat Decision Manager 7.0.
2. Go to **Menu** → **Design** → **Projects** and select the project name.
3. Open any solver configuration asset (`.solver.xml` file), if present.
4. In the solver configuration designer, click **Save** without making any changes. This step is required to generate the new code for solver configuration assets in Red Hat Decision Manager 7.0. Do this step with any other solver configuration assets.
5. Under **Data Objects** in the **Project Explorer** (left menu), open any data object (`.java` file) configured as a **Planning Solution**, if applicable.  
To verify if this setting is selected for this data object, click the OptaPlanner icon on the right side of the data objects designer. If **Planning Solution** is not selected, then this procedure does not apply.
6. Under **general properties** in the data objects designer, set the **Superclass** drop-down option to **Nothing selected**. This setting is no longer required by Red Hat Business Optimizer in Red Hat Decision Manager 7.0.
7. On the right side of the data objects designer, click the OptaPlanner icon to reveal the **Planner Settings**. The **Planning Solution** option should be selected (if not, then this procedure does not apply).
8. Select **No selected**, then re-select **Planning Solution** and specify the **Solution Score Type**. This step is required to generate the new code for planning solutions in Red Hat Decision Manager 7.0.

9. Click **Save** in the data objects designer and make the same changes to any other data objects configured as a **Planning Solution**.



## CHAPTER 5. UPGRADING A DECISION SERVER

You can upgrade a BRMS 6.4 KIE Server to Red Hat Decision Manager 7.0 Decision Server by installing the new server on the same host. You can use existing KJAR files. However, for optimal performance, upgrade and rebuild the projects.

### Procedure

1. Install Red Hat Decision Manager 7.0 Decision Server on the same host as a BRMS 6.4 KIE Server. Use a separate JBoss EAP installation. For instructions, see [Installing Red Hat Decision Manager on premise](#).
2. Configure the Red Hat Decision Manager 7.0 Decision Server to bind to the same port as the BRMS 6.4 KIE Server. (The default settings are the same).
3. Configure the Decision Server to use a Maven repository that contains your decision services (KJAR files). Use one of the following approaches:
  - Use the same KJAR files as of the BRMS 6.4 KIE Server. To copy the Maven configuration, copy the value of the **kie.maven.settings.custom** system property from the old version **\$EAP\_home/standalone/configuration/standalone.xml** file to the same file in the new version.
  - Upgrade your decision services to the new version. For instructions, see [Chapter 2, Migrating projects in Decision Central](#) and [Chapter 3, Migrating Java projects](#). Upload the new KJAR files to a Maven repository. Then configure the Decision Server to use the repository:
    - a. Create a **settings.xml** file for the Maven repository
    - b. In the **\$EAP\_home/standalone/configuration/standalone.xml** file for Red Hat Decision Manager 7.0 Decision Server, under the **<system-properties>** tag, set the **kie.maven.settings.custom** property to the full pathname of the **settings.xml** file, for example:
 

```
<property name="kie.maven.settings.custom" value="/opt/custom-config/settings.xml"/>
```

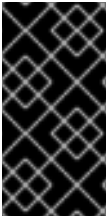
In this case, you need to change the version of the KJAR files. Change the version of the services that you start accordingly. You can still use the same container name to ensure maximum compatibility with existing clients.
4. Stop the BRMS 6.4 KIE Server.
5. Start the Red Hat Decision Manager 7.0 Decision Server.
6. Use the API to start the necessary decision services. For instructions, see [Packaging and deploying a decision service](#).

### Result

The Red Hat Decision Manager 7.0 Decision Server operates on the same URLs and provides the same services as the BRMS 6.4 KIE Server.

You might need to change client applications that use the REST API in order to adapt to the following API change:

- The ServiceResponse wrapper was removed from the Planner service responses.



### IMPORTANT

You can also configure the Decision Server to connect to a Decision Central. The Decision Central can manage several Decision Servers and deploy the same or different decision services on them. For instructions, see [Packaging and deploying a decision service](#).

## APPENDIX A. VERSIONING INFORMATION

Documentation last updated on: Monday, October 1, 2018.