Red Hat Decision Manager 7.4

Installing and configuring Red Hat Decision Manager on Red Hat JBoss Web Server

Red Hat Decision Manager 7.4 Installing and configuring Red Hat Decision Manager on Red Hat JBoss Web Server

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

Legal Notice

Abstract

This document describes how to install Red Hat Decision Manager 7.4 on Red Hat JBoss Web Server 5.1.0.

Table of Contents

PREFACE

This document describes how to install Red Hat Decision Manager 7.4 on JBoss Web Server.

Prerequisites

- You have reviewed the information in *Planning a Red Hat Decision Manager installation* .

- You have installed Red Hat JBoss Web Server 5.1.0. For information about installing Red Hat JBoss Web Server, see the *Red Hat JBoss Web Server Installation Guide* .

CHAPTER 1. ABOUT RED HAT DECISION MANAGER

Red Hat Decision Manager is an open source decision management platform that combines business rules management, complex event processing, Decision Model & Notation (DMN) execution, and Business Optimizer for solving planning problems. It automates business decisions and makes that logic available to the entire business.

Business assets such as rules, decision tables, and DMN models are organized in projects and stored in the Business Central repository. This ensures consistency, transparency, and the ability to audit across the business. Business users can modify business logic without requiring assistance from IT personnel.

Red Hat JBoss Web Server is an enterprise ready web server designed for medium and large applications, based on Tomcat. Red Hat JBoss Web Server provides organizations with a single deployment platform for Java Server Pages (JSP) and Java Servlet technologies, PHP, and CGI.

On a Red Hat JBoss Web Server installation, you can install Decision Server and the headless Decision Manager controller. Alternatively, you can run the standalone Business Central JAR file.

The instructions in this document explain how to install Red Hat Decision Manager in a Red Hat JBoss Web Server instance.

For instruction on how to install Red Hat Decision Manager in other environments, see the following documents:

- *Installing and configuring Red Hat Decision Manager on Red Hat JBoss EAP 7.2*

- *Installing and configuring Decision Server on IBM WebSphere Application Server*

- *Installing and configuring Decision Server on Oracle WebLogic Server*

- *Deploying a Red Hat Decision Manager immutable server environment on Red Hat OpenShift Container Platform*

- *Deploying a Red Hat Decision Manager authoring or managed server environment on Red Hat OpenShift Container Platform*

- *Deploying a Red Hat Decision Manager environment on Red Hat OpenShift Container Platform using Operators*

For information about supported components, see the following documents:

- What is the mapping between Red Hat Decision Manager and the Maven library version?

- Red Hat Decision Manager 7 Supported Configurations

1.1. RED HAT DECISION MANAGER COMPONENTS

Red Hat Decision Manager is made up of Business Central and Decision Server.

- Business Central is the graphical user interface where you create and manage business rules. You can install Business Central in a Red Hat JBoss EAP instance or on the Red Hat OpenShift Container Platform (OpenShift).
  Business Central is also available as a standalone JAR file. You can use the Business Central standalone JAR file to run Business Central without needing to deploy it to an application server.

- Decision Server is the server where rules and other artifacts are executed. It is used to instantiate and execute rules and solve planning problems. You can install Decision Server in a Red Hat JBoss EAP instance, on OpenShift, in an Oracle WebLogic server instance, in an IBM WebSphere Application Server instance, or as a part of Spring Boot application.
You can configure Decision Server to run in managed or unmanaged mode. If Decision Server is unmanaged, you must manually create and maintain KIE containers (deployment units). A KIE container is a specific version of a project. If Decision Server is managed, the Decision Manager controller manages the Decision Server configuration and you interact with the Decision Manager controller to create and maintain KIE containers.

  On a Red Hat JBoss Web Server installation, you can install Decision Server and the headless Decision Manager controller. Alternatively, you can run the standalone Business Central JAR file.

1.2. RED HAT DECISION MANAGER ROLES AND USERS

To access Business Central or Decision Server, you must create users and assign them appropriate roles before the servers are started. This section describes available Red Hat Decision Manager user roles.

> **NOTE**
>
> The **admin**, **analyst**, and **rest-all** roles are reserved for Business Central. The **kie-server** role is reserved for Decision Server. For this reason, the available roles can differ depending on whether Business Central, Decision Server, or both are installed.

- **admin**: Users with the **admin** role are the Business Central administrators. They can manage users and create, clone, and manage the repositories. They have full access to make required changes in the application. Users with the **admin** role have access to all areas within Red Hat Decision Manager.

- **analyst**: Users with the **analyst** role have access to all high-level features. They can model projects. However, these users cannot add contributors to spaces or delete spaces in the **Design → Projects** view. Access to the **Deploy → Execution Servers** view, which is intended for administrators, is not available to users with the **analyst** role. However, the **Deploy** button is available to these users when they access the Library perspective.

- **rest-all**: Users with the **rest-all** role can access Business Central REST capabilities.

- **kie-server**: Users with the **kie-server** role can access Decision Server (KIE Server) REST capabilities. This role is mandatory for users to have access to Manage and Track views in Business Central.

CHAPTER 2. DOWNLOADING THE RED HAT DECISION MANAGER INSTALLATION FILES

Depending on your environment and installation requirements, download a Red Hat Decision Manager distribution.

Procedure

1. Navigate to the Software Downloads page in the Red Hat Customer Portal (login required), and select the product and version from the drop-down options:

   - **Product:** Decision Manager

   - **Version:** 7.4

2. Download one of the following product distributions, depending on your preferred installation method:

   > **NOTE**
   >
   > You only need to download one of these distributions.

   - If you want to use the installer to install Red Hat Decision Manager on Red Hat JBoss Web Server, download **Red Hat Decision Manager 7.4.0 Installer**(**rhdm-installer-7.4.0.jar**). The installer graphical user interface guides you through the installation process.

   - To install Decision Server on Red Hat JBoss Web Server using the deployable ZIP file, download the **Red Hat Decision Manager 7.4.0 Add Ons**(**rhdm-7.4.0-add-ons.zip**) file. The ZIP file does not require a graphical user interface so you can install Red Hat Decision Manager using SSH.

   - To run Business Central without needing to deploy it to an application server, download **Red Hat Decision Manager 7.4.0 Business Central Standalone** (**rhdm-7.4.0-decision-central-standalone.jar**).

CHAPTER 3. USING THE RED HAT DECISION MANAGER INSTALLER

This section describes how to install Decision Server and the headless Decision Manager controller using the installer JAR file. The JAR file is an executable file that installs Red Hat Decision Manager in an existing Red Hat JBoss Web Server 5.1.0 server installation. You can run the installer in interactive or command line interface (CLI) mode.

Next steps:

Follow the instructions in one of the following sections:

- Section 3.1, "Using the installer in interactive mode"

- Section 3.2, "Using the installer in CLI mode"

3.1. USING THE INSTALLER IN INTERACTIVE MODE

The installer for Red Hat Decision Manager is an executable JAR file. You can use it to install Red Hat Decision Manager in an existing Red Hat JBoss Web Server 5.1.0 server installation.
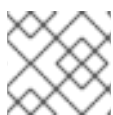
> **NOTE**
>
> For security reasons, you should run the installer as a non-root user.

Prerequisites

- A backed-up Red Hat JBoss Web Server 5.1.0 server installation is available.

- Sufficient user permissions to complete the installation are granted.

  > **NOTE**
  >
  > Ensure that you are logged in with a user that has write permission for Tomcat.

- The JAR binary is included in **$PATH** environment variable. On Red Hat Enterprise Linux, it is included in the **java-$_JAVA_VERSION_-openjdk-devel** package.

  > **NOTE**
  >
  > Red Hat Decision Manager is designed to work with UTF-8 encoding. If a different encoding system is used by the underlying JVM, unexpected errors might occur. To ensure UTF-8 is used by the JVM, use the **"-Dfile.encoding=UTF-8"** system property.

Procedure

1. In a terminal window, navigate to the directory where you downloaded the installer JAR file and enter the following command:

   ```
   java -jar rhdm-installer-7.4.0.jar
   ```

NOTE

When running the installer on Windows, you may be prompted to provide administrator credentials during the installation. To prevent this requirement, add the **izpack.mode=privileged** option to the installation command:

```
java -Dizpack.mode=privileged -jar
rhdm-installer-7.4.0.jar
```

Furthermore, when running the installer on a 32–bit Java virtual machine, you might encounter memory limitations. To prevent this issue, run this command:

```
java -XX:MaxHeapSize=4g -jar
rhdm-installer-7.4.0.jar
```

The graphical installer displays a splash screen and a license agreement page.

2. Click **I accept the terms of this license agreement** and click **Next**.

3. Specify the Red Hat JBoss Web Server 5.1.0 server home where you want to install Red Hat Decision Manager and click **Next**.

4. Select the components that you want to install and click **Next**.
   You cannot install Business Central on Red Hat JBoss Web Server. You can only install it on Red Hat JBoss EAP. However, you can install Decision Server and the headless Decision Manager controller on Red Hat JBoss Web Server. The headless Decision Manager controller is used to manage Decision Server. Install the headless Decision Manager controller if you plan to manage multiple Decision Server instances.

5. Create a user and click **Next**. By default, the new user is given the **admin**, **kie-server**, and **rest-all** roles. The **kie-server** role is required to acces Decision Server REST capabilities.

NOTE

Make sure that the specified user name is not the same as an existing user, role, or group. For example, do not create a user with the user name **admin**.

The password must have at least eight characters and must contain at least one number and one non–alphanumeric character, but not & (ampersand).

Make a note of the user name and password. You will need them to access Business Central and Decision Server.

6. On the **Installation Overview** page, click **Next** to start the installation. The Installation Overview page lists the components that you will install.

7. When the installation has completed, click **Next**.

8. When **Processing finished** appears at the top of the screen, click **Next** to complete the installation.

9. If desired, click **Generate Installation Script and Properties File** to save the installation data in an XML file, and then click **Done**. You can use this file to automatically install Red Hat Decision Manager on the same type of server. Note that you must change the **installpath** parameter in

the XML file to specify the path of the new server that you want to install Red Hat Decision Manager on. Enter the following command to perform an installation with the XML file:

```
java -jar rhdm-installer-7.4.0.jar <path-to-file>
```

You have successfully installed Red Hat Decision Manager using the installer.

3.2. USING THE INSTALLER IN CLI MODE

You can run the Red Hat Decision Manager installer through the command-line interface (CLI).

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

Prerequisites

- A backed-up Red Hat JBoss Web Server 5.1.0 server installation is available.

- Sufficient user permissions to complete the installation are granted.
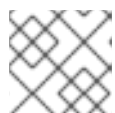
> **NOTE**
>
> Ensure that you are logged in with a user that has write permission for Tomcat.

- The JAR binary is included in the **$PATH** environment variable. On Red Hat Enterprise Linux, it is included in the **java-$*JAVA_VERSION*-openjdk-devel** package.

> **NOTE**
>
> Red Hat Decision Manager is designed to work with UTF-8 encoding. If a different encoding system is used by the underlying JVM, unexpected errors might occur. To ensure UTF-8 is used by the JVM, use the **"-Dfile.encoding=UTF-8"** system property.

Procedure

1. In a terminal window, navigate to the directory where you downloaded the installer file and enter the following command:

   ```
   java -jar rhdm-installer-7.4.0.jar -console
   ```

   The command-line interactive process will start and display the End-User License Agreement.

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

2. Read the license agreement, enter **1**, and press Enter to continue:

   ```
   Specify the home directory of one of the following servers:  Red Hat JBoss EAP 7.2 or Red Hat JBoss Web Server 5.1.0
   ```

3. Enter the parent directory of an existing Red Hat JBoss Web Server 5.1.0 installation.

The installer will verify the location of the installation at the location provided. Enter **1** to confirm and continue.

4. Follow the instructions in the installer to complete the installation.

> **NOTE**
>
> When you create the user name and password, make sure that the specified 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**.
>
> The password must have at least eight characters and must contain at least one number and one non-alphanumeric character (*not* including the character **&**).
>
> Make a note of the user name and password. You will need them to access Business Central and Decision Server.

5. When the installation has completed, you will see this message:

> Would you like to generate an automatic installation script and properties file?

6. Enter **y** to create an XML file that contains the installation data, or   **n** to complete the installation. If you enter **y**, you are prompted to specify a path for the XML file.

7. Enter a path or press the Enter key to accept the suggested path.

8. If you installed only Decision Server, repeat these steps to install the headless Decision Manager controller on a separate server.

CHAPTER 4. DECISION SERVER ZIP FILE INSTALLATION AND CONFIGURATION

You can install Decision Server using the **rhdm-7.4-kie-server-jws.zip** file available from the Red Hat Decision Manager 7.4.0 Add Ons (**rhdm-7.4.0-add-ons.zip**) file on the Customer Portal .

4.1. INSTALLING DECISION SERVER FROM ZIP FILES

Decision Server provides the runtime environment for business assets and accesses the data stored in the assets repository (knowledge store). You can use ZIP files to install Decision Server on an existing Red Hat JBoss Web Server 5.1.0 server instance.

> **NOTE**
>
> To use the installer JAR file to install Decision Server, see Chapter 3, *Using the Red Hat Decision Manager installer*.

- The **Red Hat Decision Manager 7.4.0 Add Ons**(**rhdm-7.4.0-add-ons.zip**) file has been downloaded, as described in Chapter 2, *Downloading the Red Hat Decision Manager installation files*.

- A backed-up Red Hat JBoss Web Server 5.1.0 server installation is available. The base directory of the Red Hat JBoss Web Server installation is referred to as *JWS_HOME*.

- Sufficient user permissions to complete the installation are granted.

Procedure

1. Unzip the **rhdm-7.4.0-add-ons.zip** file.

2. From the unzipped **rhdm-7.4.0-add-ons.zip** file, extract the following files:

   - **rhdm-7.4-kie-server-jws.zip**

   - **rhdm-7.4-decision-engine.zip**

   In the following instructions, the directory that contains the extracted **rhdm-7.4-kie-server-jws.zip** file is called **JWS_TEMP_DIR** and the directory that contains the extracted **rhdm-7.4-decision-engine.zip** file is called **ENGINE_TEMP_DIR**.

3. Copy the **JWS_TEMP_DIR/rhdm-7.4-kie-server-jws/kie-server.war** directory to the *JWS_HOME*/**tomcat**/**webapps** directory.

   > **NOTE**
   >
   > Ensure the names of the Red Hat Decision Manager deployments you copy do not conflict with your existing deployments in the Red Hat JBoss Web Server instance.

4. Remove the **.war** extensions from the **kie-server.war** folder.

5. Move the **kie-tomcat-integration-7.23.0.Final-redhat-00002.jar** file from the **ENGINE_TEMP_DIR** directory to the *JWS_HOME*/**tomcat**/**lib** directory.

6. Move the **jboss-jacc-api-<VERSION>.jar**, **slf4j-api-<VERSION>.jar**, and **slf4j-jdk14-<VERSION>.jar** files from the **ENGINE_TEMP_DIR/lib** directory to the **JWS_HOME/tomcat/lib** directory, where **<VERSION>** is the version artifact file name, in the **lib** directory.

7. Add the following line to the **<host>** element in the **JWS_HOME/conf/server.xml** file after the last Valve definition:

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

8. Open the **JWS_HOME/tomcat/conf/tomcat-users.xml** file in a text editor.

9. Add users and roles to the **JWS_HOME/tomcat/conf/tomcat-users.xml** file. In the following example, **<ROLE_NAME>** is a role supported by Red Hat Decision Manager. **<USER_NAME>** and **<USER_PWD>** are the user name and password of your choice:

```
<role rolename="<ROLE_NAME>"/>
<user username="<USER_NAME>" password="<USER_PWD>" roles="<ROLE_NAME>"/>
```

If a user has more than one role, as shown in the following example, separate the roles with a comma:

```
<role rolename="admin"/>
<role rolename="kie-server"/>
<user username="rhdmUser" password="user1234" roles="admin,kie-server"/>
```

10. Complete one of the following steps in the **JWS_HOME/tomcat/bin** directory:

   - On Linux or UNIX, create the **setenv.sh** file with the following content:

```
CATALINA_OPTS="-Xmx1024m
 -Dorg.jboss.logging.provider=jdk
 -Dorg.jbpm.server.ext.disabled=true
 -Dorg.jbpm.ui.server.ext.disabled=true
 -Dorg.jbpm.case.server.ext.disabled=true"
```

   - On Windows, add the following content to the **setenv.bat** file:

```
set CATALINA_OPTS="-Xmx1024m -Dorg.jboss.logging.provider=jdk -
Dorg.jbpm.server.ext.disabled=true -Dorg.jbpm.ui.server.ext.disabled=true -
Dorg.jbpm.case.server.ext.disabled=true
```

CHAPTER 5. SECURING PASSWORDS WITH A KEYSTORE

You can use a keystore to encrypt passwords that are used for communication between Business Central and Decision Server. You should encrypt both controller and Decision Server passwords. If Business Central and Decision Server are deployed to different application servers, then both application servers should use the keystore.

Use Java Cryptography Extension KeyStore (JCEKS) for your keystore because it supports symmetric keys. Use KeyTool, which is part of the JDK installation, to create a new JCEKS.

> **NOTE**
>
> If Decision Server is not configured with JCEKS, Decision Server passwords are stored in system properties in plain text form.

Prerequisites

- Decision Server is installed in Red Hat JBoss Web Server.

- Java 8 or higher is installed.

Procedure

1. Open the *JWS_HOME***/tomcat/conf/tomcat-users.xml** file in a text editor.

2. Add a Decision Server user with the **kie-server** role to the *JWS_HOME***/tomcat/conf/tomcat-users.xml** file. In the following example, replace **<USER_NAME>** and **<PASSWORD>** with the user name and password of your choice.

   ```
   <role rolename="kie-server"/>
   <user username="<USER_NAME>" password="<PASSWORD>" roles="kie-server"/>
   ```

3. To use KeyTool to create a JCEKS, enter the following command in the Java 8 home directory:

   ```
   $<JAVA_HOME>/bin/keytool -importpassword -keystore <KEYSTORE_PATH> -keypass
   <ALIAS_KEY_PASSWORD> -alias <PASSWORD_ALIAS> -storepass
   <KEYSTORE_PASSWORD> -storetype JCEKS
   ```

   In this example, replace the following variables:

   - **<KEYSTORE_PATH>**: The path where the keystore will be stored

   - **<KEYSTORE_PASSWORD>**: The keystore password

   - **<ALIAS_KEY_PASSWORD>**: The password used to access values stored with the alias

   - **<PASSWORD_ALIAS>**: The alias of the entry to the process

4. When prompted, enter the password for the Decision Server user that you created.

5. To set the system properties, complete one of these steps in the *JWS_HOME***/tomcat/bin** directory and replace the variables as described in the following table:

   - On Linux or UNIX, create the **setenv.sh** file with the following content:

```
set CATALINA_OPTS="
-Dkie.keystore.keyStoreURL=<KEYSTORE_URL>
-Dkie.keystore.keyStorePwd=<KEYSTORE_PWD>
-Dkie.keystore.key.server.alias=<KEY_SERVER_ALIAS>
-Dkie.keystore.key.server.pwd=<KEY_SERVER_PWD>
-Dkie.keystore.key.ctrl.alias=<KEY_CONTROL_ALIAS>
-Dkie.keystore.key.ctrl.key.ctrl.pwd=<KEY_CONTROL_PWD>
```

- On Windows, add the following content to the **setenv.bat** file:

```
set CATALINA_OPTS="
-Dkie.keystore.keyStoreURL=<KEYSTORE_URL>
-Dkie.keystore.keyStorePwd=<KEYSTORE_PWD>
-Dkie.keystore.key.server.alias=<KEY_SERVER_ALIAS>
-Dkie.keystore.key.server.pwd=<KEY_SERVER_PWD>
-Dkie.keystore.key.ctrl.alias=<KEY_CONTROL_ALIAS>
-Dkie.keystore.key.ctrl.key.ctrl.pwd=<KEY_CONTROL_PWD>
```

Table 5.1. System properties used to load a Decision Server JCEKS

System property	Placeholder	Description
kie.keystore.keyStoreURL	**<KEYSTORE_URL>**	URL for the JCEKS that you want to use, for example **file:///home/kie/keystores/key store.jceks**
kie.keystore.keyStorePwd	**<KEYSTORE_PWD>**	Password for the JCEKS
kie.keystore.key.server.alias	**<KEY_SERVER_ALIAS>**	Alias of the key for REST services where the password is stored
kie.keystore.key.server.pwd	**<KEY_SERVER_PWD>**	Password of the alias for REST services with the stored password
kie.keystore.key.ctrl.alias	**<KEY_CONTROL_ALIAS>**	Alias of the key for default REST Process Automation Controller where the password is stored
kie.keystore.key.ctrl.key.ctrl. pwd	**<KEY_CONTROL_PWD>**	Password of the alias for default REST Process Automation Controller with the stored password

6. Start Decision Server to verify the configuration.

CHAPTER 6. VERIFYING THE DECISION SERVER INSTALLATION

Verify that Decision Server is installed correctly.

Prerequisites

- Decision Server is installed and configured.

Procedure

1. To start Decision Server, enter one of the following commands in the ***JWS_HOME*/tomcat/bin** directory:

   - On Linux or UNIX-based systems:

     ```
     $ ./startup.sh
     ```

   - On Windows:

     ```
     startup.bat
     ```

2. After a few minutes, review the files in the ***JWS_HOME*/tomcat/logs** directory and correct any errors.

3. To verify that Decision Server is working correctly, enter **http://localhost:8080/kie-server/services/rest/server** in a web browser.

4. Enter the user name and password stored in the **tomcat-users.xml** file.

CHAPTER 7. DOWNLOADING AND INSTALLING THE HEADLESS DECISION MANAGER CONTROLLER

You can configure Decision Server to run in managed or unmanaged mode. If Decision Server is unmanaged, you must manually create and maintain KIE containers (deployment units). If Decision Server is managed, the Decision Manager controller manages the Decision Server configuration and you interact with the Decision Manager controller to create and maintain KIE containers.

The Decision Manager controller is integrated with Business Central. If you install Business Central, use the **Execution Server** page to create and maintain KIE containers. However, if you do not install Business Central, you can install the headless Decision Manager controller and use the REST API or the Decision Server Java Client API to interact with it.

Prerequisites

- The **Red Hat Decision Manager 7.4.0 Add Ons**(**rhdm-7.4.0-add-ons.zip**) file has been downloaded, as described in Chapter 2, *Downloading the Red Hat Decision Manager installation files*.

- A Red Hat JBoss Web Server 5.1.0 server installation is available. The base directory of the Red Hat JBoss Web Server installation is referred to as *JWS_HOME*.

- Sufficient user permissions to complete the installation are granted.

Procedure

1. Unzip the **rhdm-7.4.0-add-ons.zip** file. The **rhdm-7.4-controller-jws.zip** file is in the unzipped directory.

2. Extract the **rhdm-7.4-controller-jws.zip** archive to a temporary directory. In the following examples this directory is called *TEMP_DIR*.

3. Copy the *TEMP_DIR*/**rhdm-7.4-controller-jws.zip/controller.war** directory to the *JWS_HOME*/**tomcat/webapps** directory.

   > **NOTE**
   >
   > Ensure the names of the Red Hat Decision Manager deployments you copy do not conflict with your existing deployments in the Red Hat JBoss Web Server instance.

4. Remove the **.war** extensions from the **controller.war** folder.

5. Copy the contents of the *TEMP_DIR*/**rhdm-7.4-controller-jws/SecurityPolicy/** directory to *JWS_HOME*/**bin**

6. When asked to overwrite files, select **Yes**.

7. Add the **kie-server** role and user to the *JWS_HOME*/**tomcat/conf/tomcat-users.xml** file. In the following example, **<USER_NAME>** and **<PASSWORD>** are the user name and password of your choice:

```
<role rolename="kie-server"/>
<user username="<USER_NAME>" password="<PASSWORD>" roles="kie-server"/>
```

8. Complete one of the following tasks in the *JWS_HOME*/**tomcat**/**bin** directory of the instance running Decision Server:

   - On Linux or UNIX, create the **setenv.sh** file with the following content:

     ```
     CATALINA_OPTS="-Xmx1024m
      -Dorg.jbpm.server.ext.disabled=true
      -Dorg.jbpm.ui.server.ext.disabled=true
      -Dorg.jbpm.case.server.ext.disabled=true
      -Dorg.kie.server.controller.user=<CONTROLLER_USER>
      -Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
      -Dorg.kie.server.id=<KIE_SERVER_ID>
      -Dorg.kie.server.location=http://<HOST>:<PORT>/kie-server/services/rest/server
      -Dorg.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller"
     ```

   - On Windows, add the following content to the **setenv.bat** file:

     ```
     set CATALINA_OPTS= -Xmx1024m
      -Dorg.jbpm.server.ext.disabled=true
      -Dorg.jbpm.ui.server.ext.disabled=true
      -Dorg.jbpm.case.server.ext.disabled=true
      -Dorg.kie.server.controller.user=<CONTROLLER_USER>
      -Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
      -Dorg.kie.server.id=<KIE_SERVER_ID>
      -Dorg.kie.server.location=http://<HOST>:<PORT>/kie-server/services/rest/server
      -Dorg.kie.server.controller=http://<CONTROLLER_HOST>:
     <CONTROLLER_PORT>/controller/rest/controller
     ```

   In the preceding examples, replace the following variables:

   - Replace **<CONTROLLER_USER>** and **<CONTROLLER_PWD>** with the user name and password for the **kie-server** role that you defined earlier in this procedure.

   - Replace **<KIE_SERVER_ID>** with a unique identifier.

   - Replace **<CONTROLLER_HOST>:<CONTROLLER_PORT>** with the IP address (host and port) of the controller. If you use the same server for Decision Server and the controller, **<CONTROLLER_HOST>:<CONTROLLER_PORT>** is **localhost:8080**.

9. In the *JWS_HOME*/**tomcat**/**bin** directory of the instance running the headless Decision Manager controller, create a readable **setenv.sh** file with the following content, where **<USERNAME>** is the Decision Server user and **<USER_PWD>** is the password for that user: **CATALINA_OPTS="-Dorg.kie.server.user=<USERNAME> -Dorg.kie.server.pwd= <USER_PWD>"**

10. To start the headless Decision Manager controller, enter one of the following commands in the *JWS_HOME*/**tomcat**/**bin** directory:

    - On Linux or UNIX–based systems:

      ```
      $ ./startup.sh
      ```

    - On Windows:

      ```
      startup.bat
      ```

11. After a few minutes, review the *JWS_HOME*/**tomcat**/**logs** directory and correct any errors.

12. To verify that the headless Decision Manager controller is working correctly, enter **http://<CONTROLLER_HOST>:<CONTROLLER_PORT>/controller/rest/controller/management/servers** in a web browser. If you use the same server for Decision Server and the controller, **<CONTROLLER_HOST>:<CONTROLLER_PORT>** is **localhost:8080**.

13. Enter the user name and password stored in the **tomcat-users.xml** file.

CHAPTER 8. RUNNING STANDALONE BUSINESS CENTRAL

You can use the Business Central standalone JAR file to run Business Central without needing to deploy it to an application server such as Red Hat JBoss EAP.

> **NOTE**
>
> Red Hat supports this installation type only when it is installed on Red Hat Enterprise Linux.

Prerequisites

- The **Red Hat Decision Manager 7.4.0 Business Central Standalone**(**rhdm-7.4.0-decision-central-standalone.jar**) file has been downloaded, as described in Chapter 2, *Downloading the Red Hat Decision Manager installation files*.

Procedure

1. Create a directory and move the **rhdm-7.4.0-decision-central-standalone.jar** file to this directory.

2. In a terminal window, navigate to the directory that contains the standalone JAR file.

3. Create the **application-users.properties** file. Include an administrative user and if this Business Central instance will be a Decision Manager controller for Decision Server, include a Decision Manager controller user, for example:

   ```
   rhdmAdmin=password1
   controllerUser=controllerUser1234
   ```

4. Create the **application-roles.properties** file to assign roles to the users that you included in the **application-users.properties** file, for example:

   ```
   rhdmAdmin=admin
   controllerUser=kie-server
   ```

5. Create the **application-config.yaml** configuration file with the following contents, where **<APPLICATION_USERS>** is the path to the **application-users.properties** file and **<APPLICATION_ROLES>** is the path to the **application-roles.properties** file:

   ```
   thorntail:
     management:
       security-realms:
         ApplicationRealm:
           local-authentication:
             default-user: local
             allowed-users: local
             skip-group-loading: true
           properties-authentication:
             path: <APPLICATION_USERS>
             plain-text: true
           properties-authorization:
             path: <APPLICATION_ROLES>
   datasource:
   ```

```
management:
  wildfly:
    admin: admin
```

6. Enter the following command:

```
java -jar rhdm-7.4.0-decision-central-standalone.jar -s application-config.yaml
```

In addition, you can set any properties supported by Business Central by including the **-D<property>=<value>** parameter in this command, for example:

```
java -jar rhdm-7.4.0-decision-central-standalone.jar -s application-config.yaml -D<property>=
<value> -D<property>=<value>
```

See Section 8.1, "Supported properties" for more information.

8.1. SUPPORTED PROPERTIES

The Business Central system properties listed in this section are passed to **standalone*.xml** files or when you install standalone Business Central, you can use the properties listed in this section in the following command:

```
java -jar rhdm-7.4.0-decision-central-standalone.jar -s application-config.yaml -D<property>=<value>
-D<property>=<value>
```

In this command, **<property>** is a property from the following list and **<value>** is a value that you assign to that property:

- **org.uberfire.nio.git.dir**: Location of the Decision Server Git directory.

- **org.uberfire.nio.git.dirname**: Name of the Decision Server Git directory. Default value: **.niogit**.

- **org.uberfire.nio.git.proxy.ssh.over.http**: Specifies whether SSH should use an HTTP proxy. Default: **false**

- **http.proxyHost**: Defines the host name of the HTTP proxy. Default: **null**

- **http.proxyPort**: Defines the host port (integer value) of the HTTP proxy. Default: **null**

- **org.uberfire.nio.git.proxy.ssh.over.https**: Specifies whether SSH should use an HTTPS proxy. Default: **false**

- **https.proxyHost**: Defines the host name of the HTTPS proxy. Default: **null**

- **https.proxyPort**: Defines the host port (integer value) of the HTTPS proxy. Default: **null**

- **org.uberfire.nio.git.http.enabled**: Enables or disables the HTTP daemon. Default: **true**

- **org.uberfire.nio.git.http.host**: If the HTTP daemon is enabled, it uses this property as the host identifier. This is an informative property that is used to display how to access the Git repository over HTTP. The HTTP still relies on the servlet container. Default: **localhost**

- **org.uberfire.nio.git.http.hostname**: If the HTTP daemon is enabled, it uses this property as the host name identifier. This is an informative property that is used to display how to access the Git repository over HTTP. The HTTP still relies on the servlet container. Default: **localhost**

- **org.uberfire.nio.git.http.port**: If the HTTP daemon is enabled, it uses this property as the port number. This is an informative property that is used to display how to access the Git repository over HTTP. The HTTP still relies on the servlet container. Default: **8080**

- **org.uberfire.nio.git.https.enabled**: Enables or disables the HTTPS daemon. Default: **false**

- **org.uberfire.nio.git.https.host**: If the HTTPS daemon is enabled, it uses this property as the host identifier. This is an informative property that is used to display how to access the Git repository over HTTPS. The HTTPS still relies on the servlet container. Default: **localhost**

- **org.uberfire.nio.git.https.hostname**: If the HTTPS daemon is enabled, it uses this property as the host name identifier. This is an informative property that is used to display how to access the Git repository over HTTPS. The HTTPS still relies on the servlet container. Default: **localhost**

- **org.uberfire.nio.git.https.port**: If the HTTPS daemon is enabled, it uses this property as the port number. This is an informative property that is used to display how to access the Git repository over HTTPS. The HTTPS still relies on the servlet container. Default: **8080**

- **org.uberfire.nio.git.daemon.enabled**: Enables or disables the Git daemon. Default value: **true**.

- **org.uberfire.nio.git.daemon.host**: If the Git daemon is enabled, it uses this property as the local host identifier. Default value: **localhost**.

- **org.uberfire.nio.git.daemon.hostname**: If the Git daemon is enabled, it uses this property as the local host name identifier. Default: **localhost**

- **org.uberfire.nio.git.daemon.port**: If the Git daemon is enabled, it uses this property as the port number. Default value: **9418**.

- **org.uberfire.nio.git.http.sslVerify**: Enables or disables SSL certificate checking for Git repositories. Default: **true**
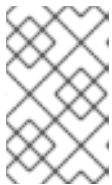
> **NOTE**
>
> If the default or assigned port is already in use, a new port is automatically selected. Ensure that the ports are available and check the log for more information.

- **org.uberfire.nio.git.ssh.enabled**: Enables or disables the SSH daemon. Default value: **true**.

- **org.uberfire.nio.git.ssh.host**: If the SSH daemon enabled, it uses this property as the local host identifier. Default value: **localhost**.

- **org.uberfire.nio.git.ssh.hostname**: If the SSH daemon is enabled, it uses this property as local host name identifier. Default: **localhost**

- **org.uberfire.nio.git.ssh.port**: If the SSH daemon is enabled, it uses this property as the port number. Default value: **8001**.
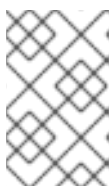
> **NOTE**
>
> If the default or assigned port is already in use, a new port is automatically selected. Ensure that the ports are available and check the log for more information.

- **org.uberfire.nio.git.ssh.cert.dir**: Location of the **.security** directory where local certificates are stored. Default: the working directory.

- **org.uberfire.nio.git.ssh.passphrase**: Pass phrase used to access the public key store of your operating system when cloning git repositories with SCP style URLs. Example: **git@github.com:user/repository.git**.

- **org.uberfire.nio.git.ssh.algorithm**: Algorithm used by SSH. Default value: **RSA**.

- **org.uberfire.nio.git.ssh.ciphers**: A comma-separated string of ciphers. The available ciphers are **aes128-ctr**, **aes192-ctr**, **aes256-ctr**, **arcfour128**, **arcfour256**, **aes192-cbc**, **aes256-cbc**. If the property is not used, all available ciphers are loaded.

- **org.uberfire.nio.git.ssh.macs**: A comma-separated string of message authentication codes (MACs). The available MACs are **hmac-md5**, **hmac-md5-96**, **hmac-sha1**, **hmac-sha1-96**, **hmac-sha2-256**, **hmac-sha2-512**. If the property is not used, all available MACs are loaded.
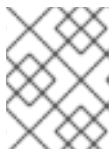
> **NOTE**
>
> If you plan to use RSA or any algorithm other than DSA, make sure you set up your application server to use the Bouncy Castle JCE library.

- **org.uberfire.metadata.index.dir**: Place where the Lucene **.index** directory is stored. Default: the working directory

- **org.uberfire.ldap.regex.role_mapper**: Regex pattern used to map LDAP principal names to the application role name. Note that the variable role must be part of the pattern because it is substituted by the application role name when matching a principal value to a role name. Default: Not used.

- **org.uberfire.sys.repo.monitor.disabled**: Disables the configuration monitor. Do not disable unless you are sure. Default value: **false**

- **org.uberfire.secure.key**: Password used by password encryption. Default value: **org.uberfire.admin**

- **org.uberfire.secure.alg**: Crypto algorithm used by password encryption. Default value: **PBEWithMD5AndDES**

- **org.uberfire.domain**: Security-domain name used by uberfire. Default value: **ApplicationRealm**

- **org.guvnor.m2repo.dir**: Place where the Maven repository folder is stored. Default value: **<working-directory>/repositories/kie**

- **org.guvnor.project.gav.check.disabled**: Disables group ID, artifact ID, and version (GAV) checks. Default value: **false**

- **org.kie.build.disable-project-explorer**: Disables automatic build of a selected project in Project Explorer. Default value: **false**

- **org.kie.verification.disable-dtable-realtime-verification**: Disables the real-time validation and verification of decision tables. Default value: **false**

- **org.kie.server.controller**: URL for connecting with a Decision Manager controller, for example: **ws://localhost:8080/decision-central/websocket/controller**

- **org.kie.server.user**: User name used to connect with the Decision Server nodes from the Decision Manager controller. This property is only required when using this Business Central installation as a Decision Manager controller.

- **org.kie.server.pwd**: Password used to connect with the Decision Server nodes from the Decision Manager controller. This property is only required when using this Business Central installation as a Decision Manager controller.

- **kie.maven.offline.force**: Forces Maven to behave as offline. If true, disable online dependency resolution. Default: **false**.

> **NOTE**
>
> Use this property for Business Central only. If you share a runtime environment with any other component, isolate the configuration and apply it only to Business Central.

- **org.uberfire.gzip.enable**: Enables or disables Gzip compression on GzipFilter. Default: **true**

- **designerdataobjects**: Disables the data object functionality. Set the value of this parameter to **false**.

CHAPTER 9. MAVEN SETTINGS AND REPOSITORIES FOR RED HAT DECISION MANAGER

When you create a Red Hat Decision Manager project, Business Central uses the Maven repositories that are configured for Business Central. You can use the Maven global or user settings to direct all Red Hat Decision Manager projects to retrieve dependencies from the public Red Hat Decision Manager repository by modifying the Maven project object model (POM) file (**pom.xml**). You can also configure Business Central and Decision Server to use an external Maven repository or prepare a Maven mirror for offline use.

For more information about Red Hat Decision Manager packaging and deployment options, see *Packaging and deploying a Red Hat Decision Manager project* .

9.1. CONFIGURING MAVEN USING THE PROJECT CONFIGURATION FILE (POM.XML)

To use Maven for building and managing your Red Hat Decision Manager projects, you must create and configure the POM file (**pom.xml**). This file holds configuration information for your project. For more information, see Apache Maven Project.

Procedure

1. Generate a Maven project. A **pom.xml** file is automatically generated when you create a Maven project.

2. Edit the **pom.xml** file to add more dependencies and new repositories.
   Maven downloads all of the JAR files and the dependent JAR files from the Maven repository when you compile and package your project.

Find the schema for the **pom.xml** file at http://maven.apache.org/maven-v4_0_0.xsd. For more information about POM files, see Apache Maven Project POM .

9.2. MODIFYING THE MAVEN SETTINGS FILE

Red Hat Decision Manager uses Maven **settings.xml** file to configure it's Maven execution. You must create and activate a profile in the **settings.xml** file and declare the Maven repositories used by your Red Hat Decision Manager projects.

For information about the Maven **settings.xml** file, see the Apache Maven Project Setting Reference.

Procedure

1. In the **settings.xml** file, declare the repositories that your Red Hat Decision Manager projects use. Usually, this is either the online Red Hat Decision Manager Maven repository or the Red Hat Decision Manager Maven repository that you download from the Red Hat Customer Portal and any repositories for custom artifacts that you want to use.

2. Ensure that Business Central or Decision Server is configured to use the **settings.xml** file. For example, specify the **kie.maven.settings.custom=<SETTINGS_FILE_PATH>** property where **<SETTINGS_FILE_PATH>** is the path to the **settings.xml** file.
   On Red Hat JBoss Web Server, for Decision Server add **-Dkie.maven.settings.custom= <SETTINGS_FILE_PATH>** to the **CATALINA_OPTS** section of the **setenv.sh** (Linux) or **setenv.bat** (Windows) file. For standalone Business Central, enter the following command:

```
java -jar rhpam-7.4.0-business-central-standalone.jar -s application-config.yaml -
Dkie.maven.settings.custom=<SETTINGS_FILE_PATH>
```

9.3. ADDING MAVEN DEPENDENCIES FOR RED HAT DECISION MANAGER

To use the correct Maven dependencies in your Red Hat Decision Manager project, add the Red Hat Business Automation bill of materials (BOM) files to the project's **pom.xml** file. The Red Hat Business Automation BOM applies to both Red Hat Decision Manager and Red Hat Process Automation Manager. When you add the BOM files, the correct versions of transitive dependencies from the provided Maven repositories are included in the project.

For more information about the Red Hat Business Automation BOM, see What is the mapping between Red Hat Decision Manager and the Maven library version?.

Procedure

1. Declare the Red Hat Business Automation BOM in the **pom.xml** file:

   ```xml
   <dependencyManagement>
    <dependencies>
     <dependency>
      <groupId>com.redhat.ba</groupId>
      <artifactId>ba-platform-bom</artifactId>
      <version>7.4.0.GA-redhat-00002</version>
      <type>pom</type>
      <scope>import</scope>
     </dependency>
    </dependencies>
   </dependencyManagement>
   <dependencies>
   <!-- Your dependencies -->
   </dependencies>
   ```

2. Declare dependencies required for your project in the **<dependencies>** tag. After you import the product BOM into your project, the versions of the user-facing product dependencies are defined so you do not need to specify the **<version>** sub-element of these **<dependency>** elements. However, you must use the **<dependency>** element to declare dependencies which you want to use in your project.

3. For standalone projects that are not authored in Business Central, specify all dependencies required for your projects. In projects that you author in Business Central, the basic decision engine dependencies are provided automatically by Business Central.

   - For a basic Red Hat Decision Manager project, declare the following dependencies, depending on the features that you want to use:

   - For a basic Red Hat Decision Manager project, declare the following dependencies:

     Embedded decision engine dependencies

     ```xml
     <dependency>
       <groupId>org.drools</groupId>
       <artifactId>drools-compiler</artifactId>
     ```

```
</dependency>

<!-- Dependency for persistence support. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-decisiontables</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-templates</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-scorecards</artifactId>
</dependency>

<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
</dependency>
```

- To use the Decision Server, declare the following dependencies:

  Client application Decision Server dependencies

  ```
  <dependency>
    <groupId>org.kie.server</groupId>
    <artifactId>kie-server-client</artifactId>
  </dependency>
  ```

- To create a remote client for Red Hat Decision Manager, declare the following dependency:

  Client dependency

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

- When creating a JAR file that includes assets, such as rules and process definitions, specify the packaging type for your Maven project as **kjar** and use **org.kie:kie-maven-plugin** to process the **kjar** packaging type located under the **<project>** element. In the following example, **${kie.version}** is the Maven library version listed in What is the mapping between Red Hat Decision Manager and the Maven library version?:

  ```
  <packaging>kjar</packaging>
  <build>
  ```

```
    <plugins>
     <plugin>
      <groupId>org.kie</groupId>
      <artifactId>kie-maven-plugin</artifactId>
      <version>${kie.version}</version>
      <extensions>true</extensions>
     </plugin>
    </plugins>
   </build>
```

9.4. PREPARING A MAVEN MIRROR REPOSITORY FOR OFFLINE USE

If your Red Hat Decision Manager deployment does not have outgoing access to the public Internet, you must prepare a Maven repository with a mirror of all the necessary artifacts and make this repository available to your environment.

> **NOTE**
>
> You do not need to complete this procedure if your Red Hat Decision Manager deployment is connected to the Internet.

Prerequisites

- A computer that has outgoing access to the public Internet is available.

Procedure

1. On the computer that has an outgoing connection to the public Internet, complete the following steps:

   a. Download the latest version of the Offliner tool.

   b. Download the **rhdm-7.4.0-offliner.txt** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.

   c. Enter the following command to use the Offliner tool to download the required artifacts:

   ```
   java -jar offliner-<version>.jar -r https://maven.repository.redhat.com/ga/ -r
   https://repo1.maven.org/maven2/ -d /home/user/temp rhdm-7.4.0-offliner.txt
   ```

   Replace **/home/user/temp** with an empty temporary directory and **<version>** with the version of the Offliner tool that you downloaded. The download can take a significant amount of time.

   d. If the tool reports failed downloads, enter the following commands to download the artifacts that failed to download the first time:

   ```
   grep Path: errors.log | sed -n -e 's/^.*Path: //p' > failed-downloads.txt
   java -jar offliner-<version>.jar -r https://maven.repository.redhat.com/ga/ -r
   https://repo1.maven.org/maven2/ -d /home/user/temp failed-downloads.txt
   ```

   If failures are reported again and are a minority of the total number downloaded the first time, you can proceed.

2. If you developed services outside Business Central and they have additional dependencies, add the dependencies to the mirror repository. If you developed the services as Maven projects, you can use the following steps to prepare these dependencies automatically. Complete the steps on the computer that has an outgoing connection to the public Internet.

   a. Create a backup of the local Maven cache directory (~/**.m2/repository**) and then clear the directory.

   b. Build the source of your projects using the **mvn clean install** command.

   c. For every project, enter the following command to ensure that Maven downloads all runtime dependencies for all the artifacts generated by the project:

      mvn -e -DskipTests dependency:go-offline -f /path/to/project/pom.xml --batch-mode -Djava.net.preferIPv4Stack=true

      Replace **/path/to/project/pom.xml** with the correct path to the **pom.xml** file of the project.

   d. Copy the contents of the local Maven cache directory (~/**.m2/repository**) to the temporary directory that you are using.

3. Copy the contents of the temporary directory to a directory on the computer on which you deployed Red Hat Decision Manager. This directory becomes the offline Maven mirror repository.

4. Create and configure a **settings.xml** file for your Red Hat Decision Manager deployment, according to instructions in Section 9.2, "Modifying the Maven settings file".

5. Make the following changes in the **settings.xml** file:

   - Under the **<profile>** tag, if a **<repositories>** or **<pluginRepositores>** tag is absent, add the tags as necessary.

   - Under **<repositories>** add the following sequence:

     ```
     <repository>
       <id>offline-repository</id>
       <url>file:///path/to/repo</url>
       <releases>
         <enabled>true</enabled>
       </releases>
       <snapshots>
         <enabled>false</enabled>
       </snapshots>
     </repository>
     ```

     Replace **/path/to/repo** with the full path to the local Maven mirror repository directory.

   - Under **<pluginRepositories>** add the following sequence:

     ```
     <repository>
       <id>offline-plugin-repository</id>
       <url>file:///path/to/repo</url>
       <releases>
         <enabled>true</enabled>
       </releases>
     ```

```
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
  </repository>
```

Replace **/path/to/repo** with the full path to the local Maven mirror repository directory.

CHAPTER 10. IMPORTING PROJECTS FROM GIT REPOSITORIES

Git is a distributed version control system. It implements revisions as commit objects. When you save your changes to a repository, a new commit object in the Git repository is created.

Business Central uses Git to store project data, including assets such as rules and processes. When you create a project in Business Central, it is added to a Git repository that is embedded in Business Central. If you have projects in other Git repositories, you can import those projects into the Business Central Git repository through Business Central spaces.

Prerequisites

- Red Hat Decision Manager projects exist in an external Git repository.

- You have the credentials required for read access to that external Git repository.

Procedure

1. In Business Central, click **Menu → Design → Projects**.

2. Select or create the space into which you want to import the projects. The default space is **mySpace**.

3. Click the three vertical dots on the right side of the screen and select **Import Project**.

4. In the **Import Project** window, enter the URL and credentials for the Git repository that contains the projects that you want to import and click **Import**. The projects are added to the Business Central Git repository and are available from the current space.

CHAPTER 11. INTEGRATING LDAP AND SSL

With Red Hat Decision Manager you can integrate LDAP and SSL through RH-SSO. For more information, see the *Red Hat Single Sign-On Server Administration Guide* .

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Friday, May 22, 2020.