Red Hat Process Automation Manager 7.3

Installing and configuring Red Hat Process Automation Manager on Red Hat JBoss Web Server

Last Updated: 2020-05-04

Red Hat Process Automation Manager 7.3 Installing and configuring Red Hat Process Automation 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 Process Automation Manager 7.3 on Red Hat JBoss Web Server 5.0.1 or higher.

Table of Contents

PREFACE

This document describes how to install Red Hat Process Automation Manager 7.3 on JBoss Web Server.

CHAPTER 1. ABOUT RED HAT PROCESS AUTOMATION MANAGER

Red Hat Process Automation Manager is the Red Hat middleware platform for creating business automation applications and microservices. It enables enterprise business and IT users to document, simulate, manage, automate, and monitor business processes and policies. It is designed to empower business and IT users to collaborate more effectively, so business applications can be changed easily and quickly.

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 Process Server and the headless Process Automation Manager controller. Alternatively, you can run the standalone Business Central JAR file.

The instructions in this document explain how to install Red Hat Process Automation Manager in a Red Hat JBoss Web Server instance. For instruction on how to install Red Hat Process Automation Manager in other environments, see the following documents:

- *Installing and configuring Red Hat Process Automation Manager on Red Hat JBoss EAP 7.2*
- *Installing and configuring Process Server on IBM WebSphere Application Server*
- *Installing and configuring Process Server on Oracle WebLogic Server*
- *Deploying a Red Hat Process Automation Manager immutable server environment on Red Hat OpenShift Container Platform*
- *Deploying a Red Hat Process Automation Manager authoring environment on Red Hat OpenShift Container Platform*
- *Deploying a Red Hat Process Automation Manager managed server environment on Red Hat OpenShift Container Platform*

For information on supported components, see the following documents:

- What is the mapping between Red Hat Process Automation Manager and the Maven library version?
- Red Hat Process Automation Manager 7 Supported Configurations

1.1. RED HAT PROCESS AUTOMATION MANAGER COMPONENTS

Red Hat Process Automation Manager is made up of Business Central and Process 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.

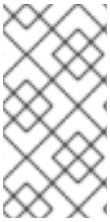
- Process Server is the server where rules and other artifacts are executed. It is used to instantiate and execute rules and solve planning problems. You can install Process Server in a Red Hat JBoss EAP instance, on OpenShift, in an Oracle WebLogic server instance, in an IBM WebSphere

Application Server instance, or as a part of Spring Boot application.
You can configure Process Server to run in managed or unmanaged mode. If Process Server is unmanaged, you must manually create and maintain KIE containers (deployment units). A KIE container is a specific version of a project. If Process Server is managed, the Process Automation Manager controller manages the Process Server configuration and you interact with the Process Automation Manager controller to create and maintain KIE containers.

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

1.2. ROLES AND USERS

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



NOTE

The **admin**, **analyst**, **developer**, **manager**, **process-admin**, **user**, and **rest-all** roles are reserved for Business Central. The **kie-server** role is reserved for Process Server. For this reason, the available roles can differ depending on whether Business Central, Process 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 Process Automation Manager.
- **analyst**: Users with the **analyst** role have access to all high-level features. They can model and execute their 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.
- **developer**: Users with the **developer** role have access to almost all features and can manage rules, models, process flows, forms, and dashboards. They can manage the asset repository, they can create, build, and deploy projects, and they can use Red Hat JBoss Developer Studio to view processes. Only certain administrative functions such as creating and cloning a new repository are hidden from users with the **developer** role.
- **manager**: Users with the **manager** role can view reports. These users are usually interested in statistics about the business processes and their performance, business indicators, and other business-related reporting. A user with this role has access only to process and task reports.
- **process-admin**: Users with the **process-admin** role are business process administrators. They have full access to business processes, business tasks, and execution errors. These users can also view business reports and have access to the Task Inbox list.
- **user**: Users with the **user** role can work on the Task Inbox list, which contains business tasks that are part of currently running processes. Users with this role can view process and task reports and manage processes.
- **rest-all**: Users with the **rest-all** role can access Business Central REST capabilities.

- **kie-server**: Users with the **kie-server** role can access Process 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 PROCESS AUTOMATION MANAGER INSTALLATION FILES

Depending on your environment and installation requirements, download a Red Hat Process Automation 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:** Process Automation Manager
   - **Version:** 7.3
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 Process Automation Manager on Red Hat JBoss Web Server, download **Red Hat Process Automation Manager 7.3.0 Installer** (**rhpam-installer-7.3.0.jar**). The installer graphical user interface guides you through the installation process.
- To install Process Server on Red Hat JBoss Web Server using the deployableZIP files, download the following files:
  - **Red Hat Process Automation Manager 7.3.0 Add Ons**(**rhpam-7.3.0-add-ons.zip**)
  - **Red Hat Process Automation Manager 7.3.0 Maven Repository**(**rhpam-7.3.0-maven-repository.zip**)
    The ZIP files do not require a graphical user interface so you can install Red Hat Process Automation Manager using SSH.
- To run Business Central without needing to deploy it to an application server, download **Red Hat Process Automation Manager 7.3.0 Business Central Standalone** (**rhpam-7.3.0-business-central-standalone.jar**).

CHAPTER 3. USING THE RED HAT PROCESS AUTOMATION MANAGER INSTALLER

This section describes how to install Process Server and the headless Process Automation Manager controller using the installer JAR file. The JAR file is an executable file that installs Red Hat Process Automation Manager in an existing Red Hat JBoss Web Server 5.0.1 or higher 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

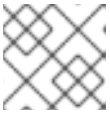
The installer for Red Hat Process Automation Manager is an executable JAR file. You can use it to install Red Hat Process Automation Manager in an existing Red Hat JBoss Web Server 5.0.1 or higher server installation.

NOTE

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

Prerequisites

- A backed-up Red Hat JBoss Web Server 5.0.1 or higher 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 Process Automation 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 rhpam-installer-7.3.0.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
rhpam-installer-7.3.0.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
rhpam-installer-7.3.0.jar
```

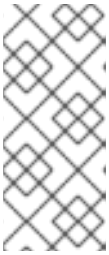
The graphical installer displays a splash screen and a license agreement page.

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

3. Specify the Red Hat JBoss Web Server 5.0.1 or higher server home where you want to install Red Hat Process Automation 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 Process Server and the headless Process Automation Manager controller on Red Hat JBoss Web Server. The headless Process Automation Manager controller is used to manage Process Server. Install the headless Process Automation Manager controller if you plan to manage multiple Process 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 Process 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 Process 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. On the **Configure Runtime Environment**page, choose whether to perform the default installation or perform an advanced configuration.
   If you choose **Perform advanced configuration**, you can choose to configure database settings or customize certain Process Server options.

9. If you selected **Customize database settings**, on the **JDBC Drive Configuration** page specify a data source JDBC driver vendor, select one or more Driver JAR files, and click **Next**.
A data source is an object that enables a Java Database Connectivity (JDBC) client, such as an application server, to establish a connection with a database. Applications look up the data source on the Java Naming and Directory Interface (JNDI) tree or in the local application context and request a database connection to retrieve data. You must configure data sources for Process Server to ensure proper data exchange between the servers and the designated database.

10. If you selected **Customize Process Server settings**, change any of the following, if desired:
    - Change the name of the Process Server property.
    - Change the URL of the headless Process Automation Manager controller.
    - Deselect any Process Server server options.

11. Click **Next** to configure the runtime environment.

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

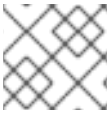
13. If desired, click **Generate Installation Script and Properties File** to save the installation data in an XML file, and then click **Done**. You can use this file to automatically install Red Hat Process Automation Manager on the same type of server. Note that you must change the **installpath** parameter in the XML file to specify the path of the new server that you want to install Red Hat Process Automation Manager on. Enter the following command to perform an installation with the XML file:

```
java -jar rhpam-installer-7.3.0.jar <path-to-file>
```

You have successfully installed Red Hat Process Automation Manager using the installer. If you installed only Business Central, repeat these steps to install Process Server on a separate server.

3.2. USING THE INSTALLER IN CLI MODE

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



NOTE

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

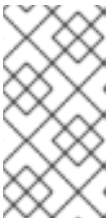
Prerequisites

- A backed-up Red Hat JBoss Web Server 5.0.1 or higher 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 Process Automation 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 rhpam-installer-7.3.0.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
```

3. Enter the parent directory of an existing Red Hat JBoss Web Server 5.0.1 or higher 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 Process 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.

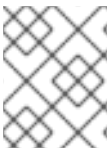
8. If you installed only Process Server, repeat these steps to install the headless Process Automation Manager controller on a separate server.

CHAPTER 4. PROCESS SERVER ZIP FILE INSTALLATION AND CONFIGURATION

You can install Process Server using the **rhpam-7.3-kie-server-jws.zip** file and then configure the Java Database Connectivity (JDBC) web server data sources on Red Hat JBoss Web Server .

4.1. INSTALLING PROCESS SERVER FROM ZIP FILES

Process 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 Process Server on an existing Red Hat JBoss Web Server 5.0.1 or higher server instance.



NOTE

To use the installer JAR file to install Process Server, see Chapter 3, *Using the Red Hat Process Automation Manager installer*.

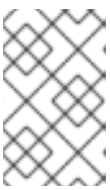
Prerequisites

- The following files have been downloaded, as described in Chapter 2, *Downloading the Red Hat Process Automation Manager installation files*:
  - **Red Hat Process Automation Manager 7.3.0 Add Ons**(**rhpam-7.3.0-add-ons.zip**)
  - **Red Hat Process Automation Manager 7.3.0 Maven Repository**(**rhpam-7.3.0-maven-repository.zip**)
- A backed-up Red Hat JBoss Web Server 5.0.1 or higher 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 **rhpam-7.3.0-add-ons.zip** file.
2. From the unzipped **rhpam-7.3.0-add-ons.zip** file, extract the following files:
   - **rhpam-7.3-kie-server-jws.zip**
   - **rhpam-7.3-process-engine.zip**

   In the following instructions, the directory that contains the extracted **rhpam-7.3-kie-server-jws.zip** file is called **JWS_TEMP_DIR** and the directory that contains the extracted **rhpam-7.3-process-engine.zip** file is called **ENGINE_TEMP_DIR**.
3. Copy the **JWS_TEMP_DIR/rhpam-7.3-kie-server-jws/kie-server.war** directory to the ***JWS_HOME*/tomcat/webapps** directory.



NOTE

Ensure the names of the Red Hat Process Automation Manager deployments you are copying 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-7.18.0.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 ***TOMCAT_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 Process Automation 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="rhpamUser" 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"
```

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

```
set CATALINA_OPTS=-Xmx1024m -Dorg.jboss.logging.provider=jdk
```

4.2. CONFIGURING JDBC WEB SERVER DATA SOURCES ON RED HAT JBOSS WEB SERVER

Java Database Connectivity (JDBC) is an API specification that is used to connect programs written in Java to the data in popular databases. A data source is an object that enables a JDBC client, such as an application server, to establish a connection with a database. Applications look up the data source on the Java Naming and Directory Interface (JNDI) tree or in the local application context and request a database connection to retrieve data. You must configure data sources for Process Server to ensure proper data exchange between the servers and the designated database.

Prerequisite

Red Hat Process Automation Manager for Red Hat JBoss Web Server is installed.

Procedure

1. Copy the following libraries from the offline Maven repository to the **JWS_HOME/tomcat/lib** folder:

```
org.jboss.spec.javax.transaction:jboss-transaction-api_1.2_spec
org.jboss.integration:narayana-tomcat
org.jboss.narayana.jta:narayana-jta
org.jboss:jboss-transaction-spi
```

2. Copy your database JDBC driver to the **JWS_HOME/tomcat/lib** folder.

3. Configure the pooling XA data source in the **JWS_HOME/tomcat/conf/context.xml** file:

NOTE

Some of the properties in the following examples might not apply to your database server. Check the documentation for your JDBC driver to determine which properties to set.

a. Configure an XA data source without pooling capabilities. This XA data source is used to create new connections to the target database. In the following example, the XA datasourse is **xads** and the variables are defined in Table 4.1, "XA data source variables":

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
<Resource
auth="Container"
databaseName="${datasource.dbName}"
description="XA Data Source"
factory="org.apache.tomcat.jdbc.naming.GenericNamingResourcesFactory"
loginTimeout="0"
name="xads"
uniqueName="xads"
portNumber="${datasource.port}"
serverName="${datasource.hostname}"
testOnBorrow="false"
type="${datasource.class}"
url="${datasource.url}"
URL="${datasource.url}"
user="${datasource.username}"
password="${datasource.password}"
driverType="4"
schema="${datasource.schema}"
/>
</Context>
```

Table 4.1. XA data source variables

Variable	Description
<datasource.dbName>	The name of the database.

Variable	Description
<datasource.port>	The port number of the database.
<datasource.hostname>	The name of the database host.
<datasource.class>	**XADataSource** class of JDBC driver.
<datasource.url>	The JDBC database connection URL. With some databases, the URL property is **url** and with other databases (for example H2 databases) this property is **URL**.
<datasource.username>	User name for the database connection.
<datasource.password>	Password for the database connection.
<datasource.schema>	The database schema.

b. Configure a pooling data source that relies on the XA data source for creating new connections. In this example, the data source is **poolingXaDs**, **<datasource.username>** is the user name for the database connection, and **<datasource.password>** is the password for the database connection:

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
<Resource
name="poolingXaDs"
uniqueName="poolingXaDs"
auth="Container"
description="Pooling XA Data Source"
factory="org.jboss.narayana.tomcat.jta.TransactionalDataSourceFactory"
testOnBorrow="true"
transactionManager="TransactionManager"
transactionSynchronizationRegistry="TransactionSynchronizationRegistry"
type="javax.sql.XADataSource"
username="${datasource.username}"
password="${datasource.password}"
xaDataSource="xads"
/>
</Context>
```

The data source is now available under the **java:comp/env/poolingXaDs** JNDI name and passes it to the Process Server through the **org.kie.server.persistence.ds** system property as described in the next steps.

NOTE

The pooling data source configuration relies on additional resource that have been previously configured in **context.xml** file in **kie-server** application, specifically **TransactionManager** and **TransactionSynchronizationRegistry**.

1. Configure Process Server to use the data source:

c. Open one of the following scripts in a text editor:

- For Linux or Unix:

```
JWS_HOME/tomcat/bin/setenv.sh
```

- For Windows:

```
JWS_HOME/tomcat/bin/setenv.bat
```

d. Add the following properties to **CATALINA_OPS** where **<hibernate.dialect>** is the Hibernate dialect for your database:

```
CATALINA_OPTS="
-Dorg.kie.server.persistence.ds=java:comp/env/poolingXaDs
-Dorg.kie.server.persistence.tm=JBossTS
-Dorg.kie.server.persistence.dialect=${<hibernate.dialect>}"
```

The following dialects are supported:

- DB2: **org.hibernate.dialect.DB2Dialect**
- MSSQL: **org.hibernate.dialect.SQLServer2012Dialect**
- MySQL: **org.hibernate.dialect.MySQL5InnoDBDialect**
- MariaDB: **org.hibernate.dialect.MySQL5InnoDBDialect**
- Oracle: **org.hibernate.dialect.Oracle10gDialect**
- PostgreSQL: **org.hibernate.dialect.PostgreSQL82Dialect**
- PostgreSQL plus: **org.hibernate.dialect.PostgresPlusDialect**
- Sybase: **org.hibernate.dialect.SybaseASE157Dialect**

NOTE

The **setenv.sh** script should already exist. However, if it does not, create it.

CHAPTER 5. VERIFYING THE PROCESS SERVER INSTALLATION

Verify that Process Server is installed correctly.

Prerequisites

- Process Server is installed and configured.

1. To start Red Hat JBoss Web 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 Process Server is working on Red Hat JBoss Web Server, enter the following command:

   ```
   curl -X GET "http://localhost:8080/kie-server/services/rest/server" -H  "accept:
   application/xml" -u '<USER_NAME>:<USER_PWD>'
   ```

   In this command, replace **<USER_NAME>** and **<USER_PWD>** with the values in the **tomcat-users.xml** file.

   The output of this command provides information about the Process Server instance.

CHAPTER 6. DOWNLOADING AND INSTALLING THE HEADLESS PROCESS AUTOMATION MANAGER CONTROLLER

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

The Process Automation 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 Process Automation Manager controller and use the REST API or the Process Server Java Client API to interact with it.

Prerequisites

- The **Red Hat Process Automation Manager 7.3.0 Add Ons**(**rhpam-7.3.0-add-ons.zip**) file has been downloaded, as described in Chapter 2, *Downloading the Red Hat Process Automation Manager installation files*.
- A Red Hat JBoss Web Server 5.0.1 or higher 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 **rhpam-7.3.0-add-ons.zip** file. The **rhpam-7.3-controller-jws.zip** file is in the unzipped directory.
2. Extract the **rhpam-7.3-controller-jws.zip** archive to a temporary directory. In the following examples this directory is called ***TEMP_DIR***.
3. Copy the ***TEMP_DIR*/rhpam-7.3-controller-jws.zip/controller.war** directory to the ***JWS_HOME*/tomcat/webapps** directory.

NOTE

Ensure the names of the Red Hat Process Automation Manager deployments you are copying 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*/rhpam-7.3-controller-jws/SecurityPolicy/** directory to ***JWS_HOME*/bin**. When asked to overwrite files, select **Yes**.
6. Add the **kie-server** role and user to the ***JWS_HOME*/tomcat/conf/tomcat-users.xml** file. In the following example, **<USERNAME>** and **<PASSWORD>** are the user name and password of your choice:

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

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

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

     ```
     CATALINA_OPTS="-Xmx1024m -Dorg.jboss.logging.provider=jdk
      -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.jboss.logging.provider=jdk
      -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
     ```

8. In the ***JWS_HOME*/tomcat/bin** directory of the instance running the headless Process Automation Manager controller, create a readable **setenv.sh** file with the following content: **CATALINA_OPTS="-Dorg.kie.server.user=<USERNAME> -Dorg.kie.server.pwd= <USER_PWD>"**

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

   - On Linux or UNIX-based systems:

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

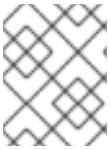
   - On Windows:

     ```
     startup.bat
     ```

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

11. To verify that the headless Process Automation Manager controller is working on Red Hat JBoss Web Server, enter the following command. In this command, replace **<CONTROLLER>** and **<CONTROLLER_PWD>** with the values in the **tomcat-users.xml** file. The output of this command provides information about the Process Server instance.

    ```
    curl -X GET "http://<HOST>:<PORT>/controller/rest/controller/management/servers" -H
    "accept: application/xml" -u '<CONTROLLER>:<CONTROLLER_PWD>'
    ```

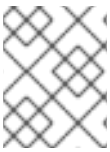


NOTE

Alternatively, you can use the Process Server Java API Client to access the headless Process Automation Manager controller.

CHAPTER 7. 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 Process Automation Manager 7.3.0 Business Central Standalone**(**rhpam-7.3.0-business-central-standalone.jar**) file has been downloaded, as described in Chapter 2, *Downloading the Red Hat Process Automation Manager installation files* .

Procedure

1. Create a directory and move the **rhpam-7.3.0-business-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 Process Automation Manager controller for Process Server, include a Process Automation Manager controller user, for example:

```
rhpamAdmin=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:

```
rhpamAdmin=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:

```
swarm:
  security:
    security-domains:
      other:
        classic-authentication:
          login-modules:
            myloginmodule:
              code: org.kie.security.jaas.KieLoginModule
              flag: optional
              module: deployment.business-central-webapp.war
  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 rhpam-7.3.0-business-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 rhpam-7.3.0-business-central-standalone.jar -s application-config.yaml -D<property>=<value> -D<property>=<value>
```

See Section 7.1, "Supported properties" for more information.

7.1. 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 rhpam-7.3.0-business-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 Process Server Git directory.
- **org.uberfire.nio.git.dirname**: Name of the Process 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.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.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.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 Process Automation Manager controller, for example: **ws://localhost:8080/business-central/websocket/controller**
- **org.kie.server.user**: User name used to connect with the Process Server nodes from the Process Automation Manager controller. This property is only required when using this Business Central installation as a Process Automation Manager controller.
- **org.kie.server.pwd**: Password used to connect with the Process Server nodes from the Process Automation Manager controller. This property is only required when using this Business Central installation as a Process Automation 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 8. MAVEN SETTINGS AND REPOSITORIES FOR RED HAT PROCESS AUTOMATION MANAGER

You can use an external Maven repository to deploy a project. When you create a 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 Process Automation Manager projects to retrieve dependencies from the public Red Hat Process Automation Manager repository by modifying the following files:

- The Maven **settings.xml** file.
- The Maven project object model (POM) file (**pom.xml**).

For more information, see *Packaging and deploying a Red Hat Process Automation Manager project* .

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

To use Maven for building and managing your Red Hat Process Automation 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 .

8.2. MODIFYING THE MAVEN SETTINGS FILE

Red Hat Process Automation 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 Process Automation Manager projects.

Procedure

1. In the **settings.xml** file, declare the repositories that your Red Hat Process Automation Manager projects use. Usually, this is either the online Red Hat Process Automation Manager Maven repository or the Red Hat Process Automation 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 Process Server is configured to use the **settings.xml** file, for example by specifying the **kie.maven.settings.custom=<path_to_settings.xml>** property.

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

8.3. ADDING MAVEN DEPENDENCIES FOR RED HAT PROCESS AUTOMATION MANAGER

To use the correct Maven dependencies in your Red Hat Process Automation 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 Process Automation Manager and the Maven library version?.

Procedure

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

```
<dependencyManagement>
 <dependencies>
  <dependency>
   <groupId>com.redhat.ba</groupId>
   <artifactId>ba-platform-bom</artifactId>
   <version>7.3.0.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 and process engine dependencies are provided automatically by Business Central.

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

     Embedded process engine dependencies

```
<!-- Public KIE API -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
  <groupId>org.jbpm</groupId>
```

```
  <artifactId>jbpm-flow</artifactId>
</dependency>

<dependency>
  <artifactId>jbpm-flow-builder</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-bpmn2</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-runtime-manager</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-persistence-jpa</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-query-jpa</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-audit</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<!-- Dependency needed for default WorkItemHandler implementations. -->
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-workitems-core</artifactId>
</dependency>

<!-- Logging dependency. You can use any logging framework compatible with slf4j. -->
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>${logback.version}</version>
</dependency>
```

- For a Red Hat Process Automation Manager project that uses CDI, you typically declare the following dependencies:

  CDI-enabled process engine dependencies

  ```
  <dependency>
  ```

```
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-services-cdi</artifactId>
</dependency>
```

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

  Embedded decision engine dependencies

```
<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 Process Server, declare the following dependencies:

  Client application Process Server dependencies

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

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

  Client dependency

```
<dependency>
  <groupId>org.uberfire</groupId>
</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 Process Automation 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>
```

CHAPTER 9. 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 Process Automation Manager projects exist in an external Git repository.
- Credentials required for read access to that external Git repository are available.

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 10. INTEGRATING LDAP AND SSL

With Red Hat Process Automation 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 Wednesday, May 8, 2019.