Red Hat Process Automation Manager 7.5

Installing and configuring Process Server on IBM WebSphere Application Server

Red Hat Process Automation Manager 7.5 Installing and configuring Process Server on IBM WebSphere Application Server

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

Legal Notice

Abstract

This document describes how to configure IBM WebSphere Application Server for Process Server and how to install Process Server on that IBM server instance.

Table of Contents

PREFACE

As a system administrator, you can configure your IBM WebSphere Application Server for Red Hat Process Server and install Process Server on that IBM server instance.

Prerequisites

- An IBM WebSphere Application Server instance version 9.0 or later is installed. For complete installation instructions, see the IBM WebSphere Application Server product page .

- You have access to the WebSphere Integrated Solutions Console, usually at **http://<HOST>:9060/ibm/console**.

CHAPTER 1. PROCESS SERVER

Process Server is the server where the rules and other artifacts for Red Hat Process Automation Manager are stored and run. Process Server is a standalone built-in component that can be used to instantiate and execute rules through interfaces available for REST, Java Message Service (JMS), or Java client-side applications, as well as to manage processes, jobs, and Red Hat Business Optimizer functionality through solvers.

Created as a web deployable WAR file, Process Server can be deployed on any web container. The current version of the Process Server is included with default extensions for both Red Hat Decision Manager and Red Hat Process Automation Manager.

Process Server has a low footprint with minimal memory consumption and therefore can be deployed easily on a cloud instance. Each instance of this server can open and instantiate multiple containers, which enables you to execute multiple rule services in parallel.

Process Server can be integrated with other application servers, such as Oracle WebLogic Server or IBM WebSphere Application Server, to streamline Red Hat Process Automation Manager application management.

CHAPTER 2. IBM WEBSPHERE APPLICATION SERVER

IBM WebSphere Application Server is a flexible and secure web application server that hosts Java-based web applications and provides Java EE-certified run time environments. IBM WebSphere 9.0 supports Java SE 8 and is fully compliant with Java EE 7.

CHAPTER 3. INSTALLING AND RUNNING IBM WEBSPHERE APPLICATION SERVER

IBM WebSphere Application Server must be installed and running for you to apply many of the configurations that accommodate Process Server. This section describes how to install and start IBM WebSphere.

For the most up-to-date and detailed installation instructions, see the IBM Knowledge Center.

Procedure

1. Download IBM Installation Manager version 1.8.5 or later from the IBM Installation Manager and Packaging Utility download links page. IBM Installation Manager is required for installing IBM WebSphere.

2. Extract the downloaded archive and run the following command as the root user in the new directory:

   ```
   sudo ./install
   ```

   The **IBM Installation Manager** opens.

3. Go to **File → Preferences** and click **Add Repository**.

4. In the **Add Repository** window, enter the repository URL for IBM WebSphere 9.0. You can find all the repository URLs in the Online product repositories for IBM WebSphere Application Server offerings page of the IBM Knowledge Center.

5. In your command terminal, navigate to the IBM WebSphere Application Server folder location that you specified during the installation.

6. Change to the **/bin** directory and run a command similar to the following example to create an IBM WebSphere profile, user name, and password. A profile defines the run time environment. The profile includes all the files that the server processes in the runtime environment and that you can change. The user is required for login.

   ```
   sudo ./manageprofiles.sh -create -profileName testprofile -profilePath /profiles/testprofile -adminUserName websphere -adminPassword password123
   ```

7. In your command terminal, navigate to the **bin** directory within the profile that you created (for example, **/profiles/testprofile/bin**) and run the following command to start the IBM WebSphere Application Server instance:

   ```
   sudo ./startServer.sh <SERVER_NAME>
   ```

   <SERVER_NAME> is the IBM WebSphere Application Server name defined in **Servers → Server Types → IBM WebSphere Application Servers** of the WebSphere Integrated Solutions Console.

8. Open the following URL in a web browser:

   ```
   http://<HOST>:9060/ibm/console
   ```

   <HOST> is the system name or IP address of the target server.

For example, to start the WebSphere Integrated Solutions Console for a local instance of IBM WebSphere running on your system, enter the following URL in a web browser:

```
http://localhost:9060/ibm/console
```

9. When the login page of the WebSphere Integrated Solutions Console appears, enter your administrative credentials.

CHAPTER 4. CONFIGURING IBM WEBSPHERE APPLICATION SERVER FOR PROCESS SERVER

Before you deploy Process Server with IBM WebSphere Application Server, you must configure system properties, security settings, JMS requirements, and other properties on IBM WebSphere. These configurations promote an optimal integration with Process Server.

Prerequisites

- IBM WebSphere Application Server is installed and running.

- You are logged in to the WebSphere Integrated Solutions Console.

4.1. ENABLING ADMINISTRATIVE SECURITY

You must enable administrative security in the WebSphere Integrated Solutions Console so that you have the required permissions to create users and groups.

Procedure

1. In the WebSphere Integrated Solutions Console, click **Security → Global Security** and ensure that the option **Enable Application Security** is selected. This may already be selected and overridden at the server level.

2. Click **Security Configuration Wizard** and click **Next**.

3. Select the repository that contains the user information. For example, select **Federated repositories** for local configurations.

4. Click **Next**

5. Enter the **Primary administrative user name** and **Password**.

6. Click **Next** and then click **Finish**.

7. Click **Save** in the **Messages** window to save your changes to the master configuration.

   Figure 4.1. Save security changes
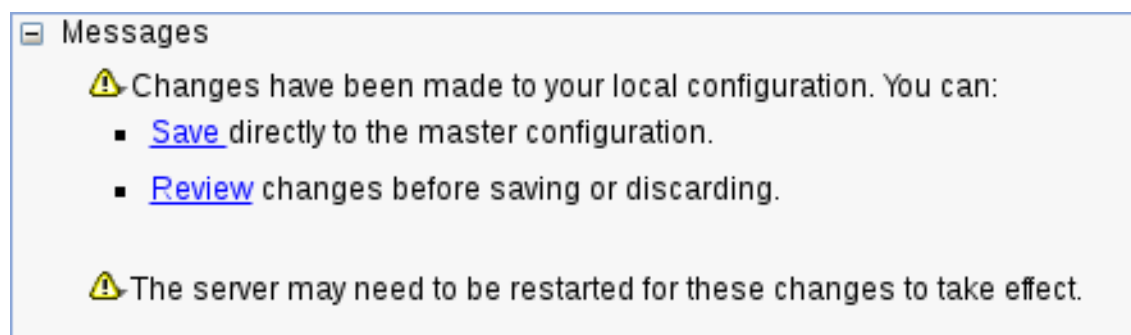
8. In your command terminal, navigate to the IBM WebSphere Application Server **/bin** directory location that you specified during installation, and run the following commands to stop and restart IBM WebSphere to apply the security changes:

   ```
   sudo ./stopServer.sh <SERVER_NAME>
   ```

```
sudo ./startServer.sh <SERVER_NAME>
```

<SERVER_NAME> is the IBM WebSphere Application Server name defined in **Servers →
Server Types → IBM WebSphere Application Servers** of the WebSphere Integrated Solutions
Console.

4.2. CONFIGURING JDBC DATA SOURCES IN IBM WEBSPHERE APPLICATION SERVER

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 IBM WebSphere Application
Server to ensure proper data exchange between the servers and the designated database.

Prerequisites

- The JDBC providers that you want to use to create database connections are configured on all
  servers on which you want to deploy the data source. For more information about JDBC
  providers, see Configuring a JDBC provider in the IBM Knowledge Center.

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:** 7.5

2. Download **Red Hat Process Automation Manager 7.5.x Add-Ons** where **x** is the current patch
   version.

3. Complete the following steps to prepare your database:

   a. Extract **rhpam-7.5.1-add-ons.zip** in a temporary directory, for example **TEMP_DIR**.

   b. Extract **TEMP_DIR/rhpam-7.5-migration-tool.zip**.

   c. Change your current directory to the **TEMP_DIR/rhpam-7.5-migration-tool/ddl-scripts**
      directory. This directory contains DDL scripts for several database types.

   d. Import the DDL script for your database type into the database that you want to use, for
      example:

      ```
      psql jbpm < /ddl-scripts/postgresql/postgresql-jbpm-schema.sql
      ```

4. In the WebSphere Integrated Solutions Console, navigate to **Resources → JDBC → Data
   sources**.

5. Select the scope at which applications can use the data source. You can choose a cell, node,
   cluster, or server.

6. Click **New** to open the **Create a data source** wizard.

7. Enter a unique name in the **Data source name** field and a Java Naming and Directory Interface (JNDI) name in the **JNDI name** field. The application server uses the JNDI name to bind resource references for an application to this data source.
   Do not assign duplicate JNDI names across different resource types, such as data sources versus J2C connection factories or JMS connection factories.

   Do not assign duplicate JNDI names for multiple resources of the same type in the same scope.

8. Click **Next** and choose **Select an existing JDBC provider** if the provider has been created, or choose **Create new JDBC provider** to define details for a new provider. (Creating the JDBC provider before this point is recommended, as a prerequisite to adding data sources.)

9. On the **Enter database specific properties for the data source** panel, click **Use this data source in container managed persistence (CMP)** if container managed persistence (CMP) enterprise beans must access this data source. Then fill in any other database-specific properties.

10. Optionally, configure the security aliases for the data source. You can select **None** for any of the authentication methods, or choose one of the following types:

    - **Component-managed authentication alias:** Specifies an authentication alias to use when the component resource reference res-auth value is **Application**. To define a new alias, navigate to **Related Items → J2EE Connector Architecture (J2C) authentication data entries**. A component-managed alias represents a combination of ID and password that is specified in an application for data source authentication. Therefore, the alias that you set on the data source must be identical to the alias in the application code.

    - **Mapping-configuration alias:** Used only in the absence of a login configuration on the component resource reference. The specification of a login configuration and the associated properties on the component resource reference is the preferred way to define the authentication strategy when the res-auth value is set to **Container**. If you specify the **DefaultPrincipalMapping** login configuration, the associated property is a JAAS - J2C authentication data entry alias.

    - **Container-managed authentication alias:** Used only in the absence of a login configuration on the component resource reference. The specification of a login configuration and the associated properties on the component resource reference determines the container-managed authentication strategy when the res-auth value is set to **Container**.

11. Click **Next**, review the information for the data source, and click **Finish** to save the configuration and exit the wizard.
    The **Data sources** panel displays your new configuration in a table along with any other data sources that are configured for the same scope.

For more information about IBM WebSphere Application Server data sources, see Configuring a JDBC provider and data source in the IBM Knowledge Center.

4.3. CONFIGURING JAVA MESSAGE SERVICE (JMS)

The Java Message Service (JMS) is a Java API that Process Server uses to exchange messages with other application servers such as Oracle WebLogic Server and IBM WebSphere Application Server. You must configure your application server to send and receive JMS messages through Process Server to ensure proper collaboration between the two servers.

4.3.1. Create a service bus and add IBM WebSphere

You muse create a service bus and add the IBM WebSphere Application Server as a member of it in order to use JMS.

Procedure

1. In the WebSphere Integrated Solutions Console, navigate to **Service Integration → Buses → New**.

2. Enter a new bus name and clear the **Bus Security** option.

3. Click **Next** and then **Finish** to create the service bus.

4. Select the service bus that you have created.

5. Under **Topology**, click **Bus Members → Add**.

6. In the **Add a New Bus Member** wizard, choose the IBM WebSphere Application Server and the type of message store for persistence. You can also specify the properties of the message store.

7. Click **Finish** to add the new bus member.

4.3.2. Create JMS connection factories

To enable messaging with Process Server, you must create certain JMS connection factories for sending and receiving messages.

Prerequisites

- You have created a service bus for IBM WebSphere Application Server.

Procedure

1. In the WebSphere Integrated Solutions Console, navigate to **Resources → JMS → Connection Factories**.

2. Select the correct scope and click **New**.

3. Select the **Default Messaging Provider** option and click **OK**.

4. For each of the following required connection factories, enter the name of the connection factory (for example, **KIE.SERVER.REQUEST**) and the JNDI name (for example, **jms/cf/KIE.SERVER.REQUEST**), and then select the service bus from the **Bus Name** drop-down list. Leave the default values for the remaining options.

5. Click **Apply** and **Save** to save the changes to the master configuration, and repeat for each required factory.

4.3.2.1. JMS connection factories for Process Server

The following are the required Java Message Service (JMS) connection factories that enable JMS messaging with Process Server:

Table 4.1. Required JMS connection factories for Process Server

Name	Default value	Used for
KIE.SERVER.REQUE ST	**jms/cf/KIE.SERVER.REQUEST**	Sending all requests to Process Server
KIE.SERVER.RESPO NSE	**jms/cf/KIE.SERVER.RESPONS E**	Receiving all responses produced by Process Server
KIE.SERVER.EXECU TOR	**jms/cf/KIE.SERVER.EXECUTO R**	Process Server executor services

4.3.3. Create JMS queues

JMS queues are the destination end points for point-to-point messaging. You must create certain JMS queues to enable JMS messaging with Process Server.

Prerequisites

- You have created a service bus for IBM WebSphere Application Server.

Procedure

1. In the WebSphere Integrated Solutions Console, navigate to **Resources → JMS → Queues**.

2. Select the correct scope and click **New**.

3. Select the **Default Messaging Provider** option and click **OK**.

4. For each of the following required queues, enter the name of the queue (for example, **KIE.SERVER.REQUEST**) and the JNDI name (for example, **jms/KIE.SERVER.REQUEST**), and then select the service bus from the **Bus Name** drop-down list.

5. From the **Queue Name** drop-down list, select the **Create Service Integration Bus Destination**, enter a unique identifier, and select the bus member that you created previously.

6. Click **Apply** and **Save** to save the changes to the master configuration, and repeat for each required queue.

4.3.3.1. JMS queues for Process Server

The following are the required Java Message Service (JMS) queues that enable JMS messaging with Process Server:

Table 4.2. Required JMS queues for Process Server

Name	Default value	Used for
KIE.SERVER.REQUE ST	**jms/KIE.SERVER.REQUEST**	Sending all requests to Process Server
KIE.SERVER.RESPO NSE	**jms/KIE.SERVER.RESPONSE**	Receiving all responses produced by Process Server

Name	Default value	Used for
KIE.SERVER.EXECUTOR	**jms/KIE.SERVER.EXECUTOR**	Process Server executor services

4.3.4. Create JMS activation specifications

A JMS activation specification is required in order to bridge the queue and the message-driven bean that enables JMS.

Prerequisites

- You have created a service bus for IBM WebSphere Application Server.

- You have created JMS queues.

Procedure

1. In the WebSphere Integrated Solutions Console, navigate to **Resources → JMS → Activation Specifications**.

2. Select the correct scope and click **New**.

3. Select the **Default Messaging Provider** option and click **OK**.

4. For each of the following required activation specifications, enter the name of the activation specification (for example, **KIE.SERVER.REQUEST**) and the JNDI name (for example, **jms/activation/KIE.SERVER.REQUEST**), and then select the service bus from the **Bus Name** drop-down list.

5. From the **Destination Type** drop-down list, select **Queue** and enter the name of the corresponding queue as a **Destination lookup** (for example, **jms/KIE.SERVER.REQUEST**).

6. Click **Apply** and **Save** to save the changes to the master configuration, and repeat for each required activation specification.

4.3.4.1. JMS activation specifications for Process Server

The following are the required Java Message Service (JMS) activation specifications that enable JMS messaging with Process Server:

Table 4.3. Required JMS activation specifications for Process Server

Name	Default value	Used for
KIE.SERVER.REQUEST	**jms/activation/KIE.SERVER.REQUEST**	Sending all requests to Process Server

Name	Default value	Used for
KIE.SERVER.RESPONSE	**jms/activation/KIE.SERVER.RESPONSE**	Receiving all responses produced by Process Server
KIE.SERVER.EXECUTOR	**jms/activation/KIE.SERVER.EXECUTOR**	Process Server executor services

4.4. SETTING SYSTEM PROPERTIES IN IBM WEBSPHERE APPLICATION SERVER

Set the system properties listed in this section on your IBM WebSphere Application Server before you deploy Process Server.

Procedure

1. In the WebSphere Integrated Solutions Console, navigate to **Servers** → **Server Types** → **IBM WebSphere Application Servers**.

2. In the list of application servers, choose the server on which you are going to deploy Process Server.

3. Under the **Server Infrastructure**, click **Java and Process Management** → **Process Definition**.

Figure 4.2. WebSphere configuration page

4. Under **Additional Properties**, click **Java Virtual Machine**

Figure 4.3. Process definition configuration page

This opens the configuration properties for the JVM that is used to start IBM WebSphere.

5. Set both the **Initial heap size** and **Maximum heap size** to **2048** and click **Apply** to increase the Java Virtual Machine (JVM) memory size. Process Server has been tested with these values. If you do not increase the JVM memory size, IBM WebSphere Application Server freezes or causes deployment errors when deploying Process Server.

6. Under **Additional Properties**, click **Custom Properties**.

7. Click **New → Custom JVM Properties** and add the following properties to IBM WebSphere:

Table 4.4. System properties for Process Server

Name	Value	Description
kie.server.jms.queues.response	**jms/KIE.SERVER.RESPONSE**	The JNDI name of JMS queue for responses used by Process Server.
org.kie.server.domain	**WSLogin**	JAAS **LoginContext** domain used to authenticate users when using JMS.

Name	Value	Description
org.kie.server.persistence.ds	**jdbc/jbpm**	Data source JNDI name for Process Server.
org.kie.server.persistence.tm	**org.hibernate.engine.transaction.jta.platform.internal.WebSphereJtaPlatform**	Transaction manager platform for setting Hibernate properties.
org.kie.server.persistence.dialect	Example: **org.hibernate.dialect.H2Dialect**	Specifies the Hibernate dialect to be used. Set according to data source.
org.kie.executor.jms.queue	**jms/KIE.SERVER.EXECUTOR**	Job executor JMS queue for Process Server.
org.kie.executor.jms.cf	**jms/cf/KIE.SERVER.EXECUTOR**	Job executor JMS connection factory for Process Server.
org.kie.server.router	Example: **http://localhost:9000**	(Optional) Specifies one or more URLs for one or more Process Server routers (Smart Routers) that the application server is part of in a clustered Process Server environment.
org.jboss.logging.provider	**jdk**	This property is only required where a **CA SiteMinder TAI (SMTAI)** is installed in the environment. Using this property forces Hibernate to use **JDK** instead of **log4j** for logging within Dashbuilder. **CA SiteMinder TAI (SMTAI)** contains an old version of **log4j**, which causes conflicts.

8. Click **Save** to save the changes to the master configuration.

4.5. STOPPING AND RESTARTING IBM WEBSPHERE APPLICATION SERVER

After you have configured all required system properties in IBM WebSphere Application Server, stop and restart the IBM server to ensure that the configurations are applied.

Procedure

In your command terminal, navigate to the IBM WebSphere Application Server /**bin** directory location that you specified during installation, and run the following commands to stop and restart IBM WebSphere to apply the configuration changes:

```
sudo ./stopServer.sh <SERVER_NAME>
```

```
sudo ./startServer.sh <SERVER_NAME>
```

<SERVER_NAME> is the IBM WebSphere Application Server name defined in **Servers → Server Types → IBM WebSphere Application Servers** of the WebSphere Integrated Solutions Console.

CHAPTER 5. INSTALLING AND RUNNING PROCESS SERVER WITH IBM WEBSPHERE APPLICATION SERVER

After you have configured all required system properties in IBM WebSphere Application Server, you can install Process Server with IBM WebSphere to streamline Red Hat Process Automation Manager application management.

Prerequisites

- An IBM WebSphere Application Server instance is configured as described in Chapter 4, *Configuring IBM WebSphere Application Server for Process Server* .

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:** 7.5

2. Download **Red Hat Process Automation Manager 7.5.1 Process Server for All Supported EE7 Containers**.

3. Extract the **rhpam-7.5.1-kie-server-ee7.zip** archive to a temporary directory. In the following examples this directory is called *TEMP_DIR*.

4. Repackage the **kie-server.war** directory:

   a. Navigate to the *TEMP_DIR*/**rhpam-7.5.1-kie-server-ee7/kie-server.war** directory.

   b. Select the contents of the *TEMP_DIR*/**rhpam-7.5.1-kie-server-ee7/kie-server.war** directory and create the **kie-server.zip** file.

   c. Rename **kie-server.zip** to **kie-server.war**. This is the file that you will use to deploy Process Server.

   d. Optionally, copy the new **kie-server.war** file to a location that is more convenient to deploy from.

5. In the WebSphere Integrated Solutions Console, navigate to **Applications → Application Types → WebSphere Enterprise Applications**.

6. Click **Install**.

7. Navigate to the **kie-server.war** file that you repackaged and select it to upload.

8. Select **Fast Path** and click **Next**.
   The **Install New Application** wizard opens.

9. Change the **Application Name** to **kie-server** and click **Next**.

10. Map the Process Server modules to servers according to your specific requirements and click **Next**.

11. For **Bind Listeners for Message-Driven Beans**, select **Activation Specification** for both

beans, enter **jms/activation/KIE.SERVER.REQUEST** in the Target Resource JNDI Name field, and enter the **jms/cf/KIE.SERVER.REQUEST** JNDI name for the **KIE.SERVER.REQUEST** connection factory.

12. In the **Map Virtual Hosts for Web Modules**section, keep the default values and click Next.

13. Set the context root to **kie-server**.

14. In the **Metadata for Modules** section, keep the default values and click Next.

15. Click **Finish** to install Process Server and click Save to save the changes to the master configuration.

5.1. CREATING THE PROCESS SERVER GROUP AND ROLE

After Process Server is installed, you must create the **kie-server** group and a user.

Prerequisites

- Process Server is installed on the IBM WebSphere Application Server instance.

Procedure

1. In the WebSphere Integrated Solutions Console, click **Users and Groups** → **Manage Groups**.

2. In the **Manage Groups** screen, click **Create**.

3. In the **Create a Group** screen, enter **kie-server** in the Group name box, then click Create.

4. To create a user to add to the **kie-server** group, click **Users and Groups** → **Manage Users**.

5. In the **Create a User** screen, complete the required information.

6. Click **Group Membership**.

7. In the **Group Membership** screen, click **kie-server**, move it to Mapped To, and click Close.

8. On the **Create a User** screen click Create.

5.2. MAPPING THE PROCESS SERVER GROUP AND ROLE

After Process Server is installed, you must map the **kie-server** role to the **kie-server** group in the WebSphere Integrated Solutions Console to run Process Server.

Prerequisites

- Process Server is installed on the IBM WebSphere Application Server instance.

- IBM WebSphere Application Server has the **kie-server** group with at least one user.

Procedure

1. In the WebSphere Integrated Solutions Console, navigate to **Applications** → **Application Types** → **WebSphere Enterprise Applications** and select the newly installed **kie-server** application.

2. Under **Detail Properties**, click **Security Role to User/Group Mapping**

3. Select the **kie-server** role and click **Map Groups** to search for the **kie-server** group.

4. Move the **kie-server** group from the **Available** list to the **Selected** list and click **OK**. This mapping gives users in the IBM WebSphere Application Server **kie-server** group access to Process Server.

5. Click **Save** to complete the mapping.

5.3. CONFIGURING CLASS LOADING FOR PROCESS SERVER

After Process Server is installed, you must configure class loading to set parent classes to load last.

Procedure

1. Navigate to **Applications → Application Types → WebSphere Enterprise Applications** and click **kie-server**.

2. Click **Class Loading and Update Detection** under the **Detail Properties** heading on the left.

3. In the properties, change *Class Loader Order* to **Classes loaded with local class loader first (parent last)** and *WAR Class Loader Policy* to **Single class loader for application**

4. Save the changes to the master configuration.

5.4. VERIFYING THE INSTALLATION

After you install Process Server and define the Process Server group mapping, verify that the server is running.

Prerequisites

- Process Server is installed on the IBM WebSphere Application Server instance.

- You have set all required system properties for the headless Process Automation Manager controller.

- You have defined the Process Server group mapping in IBM WebSphere Application Server.

Procedure

Navigate to the Process Server URL **http://<HOST>:<PORT>/kie-server** to verify that the server is running, or send a **GET** request to **http://<HOST>:<PORT>/kie-server/services/rest/server** to check whether the Process Server REST API responds.

<HOST> is the ID or name of the Process Server host, for example, **localhost** or **192.7.8.9**.

<PORT> is the port of the Process Server host, for example, **9060**.

If Process Server is not running, stop and restart the IBM WebSphere Application Server instance and try again to access the Process Server URL or API.

CHAPTER 6. INSTALLING AND RUNNING THE HEADLESS PROCESS AUTOMATION MANAGER CONTROLLER WITH IBM WEBSPHERE APPLICATION SERVER

To use the Process Server REST API or Java Client API to interact with Process Server, install the headless Process Automation Manager controller with IBM WebSphere Application Server. The headless Process Automation Manager controller manages Process Server configuration in a centralized way so that you can use the headless Process Automation Manager controller to create and maintain containers and perform other server-level tasks.

> **NOTE**
>
> For optimal results, install Process Server and the headless Process Automation Manager controller on different servers in production environments. In development environments, you can install Process Server and the headless Process Automation Manager controller on the same server.

Prerequisites

- The IBM WebSphere Application Server instance is configured as described in Chapter 4, *Configuring IBM WebSphere Application Server for Process Server* .

- Process Server is installed on the IBM WebSphere Application Server instance.

- You have sufficient user permissions to complete the installation.

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:** 7.5

2. Download **Red Hat Process Automation Manager 7.5.1 Add-Ons**

3. Extract the downloaded **rhpam-7.5.1-add-ons.zip** file to a temporary directory.

4. Extract the **rhpam-7.5.1-add-ons.zip** file to a temporary directory. In the following examples this directory is called *TEMP_DIR*.

5. Extract the **rhpam-7.5.1-add-ons/rhpam-7.5-controller-ee7.zip** file.

6. Repackage the **controller.war** directory:

   a. Navigate to the *TEMP_DIR*/**rhpam-7.5.1-add-ons/rhpam-7.5-controller-ee7/controller.war** directory.

   b. Select the contents of the *TEMP_DIR*/**rhpam-7.5.1-add-ons/rhpam-7.5-controller-ee7/controller.war** directory and create the **controller.zip** file.

   c. Rename **controller.zip** to **controller.war**. This is the file that you will use to deploy the headless Process Automation Manager controller.

      d. Optionally, copy the new **controller.war** file to a location that is more convenient to deploy from.

7. In the WebSphere Integrated Solutions Console, navigate to **Applications → Application Types → WebSphere Enterprise Applications**.

8. Click **Install**.

9. Navigate to the **controller.war** file that you repackaged and select it to upload.

10. Select **Fast Path** and click **Next**.
The **Install New Application** wizard opens.

11. Change the **Application Name** to **controller** and click **Next**.

12. Map the headless Process Automation Manager controller modules to servers according to your specific requirements and click **Next**.

13. For **Bind Listeners for Message-Driven Beans**, select **Activation Specification** for both beans, enter **jms/activation/KIE.SERVER.REQUEST** in the **Target Resource JNDI Name** field, and enter the **jms/cf/KIE.SERVER.REQUEST** JNDI name for the **KIE.SERVER.REQUEST** connection factory.

14. In the **Map Virtual Hosts for Web Modules** section, keep the default values and click **Next**.

15. Set the context root to **controller**.

16. In the **Metadata for Modules** section, keep the default values and click **Next**.

17. Click **Finish** to install the headless Process Automation Manager controller and click **Save** to save the changes to the master configuration.

6.1. CONFIGURING CLASS LOADING FOR THE HEADLESS PROCESS AUTOMATION MANAGER CONTROLLER

After the headless Process Automation Manager controller is installed, you must configure Process Server class loading to set parent classes to load last.

Procedure

1. Navigate to **Applications → Application Types → WebSphere Enterprise Applications** and click **kie-server**.

2. Click **Class Loading and Update Detection** under the **Detail Properties** heading on the left.

3. In the properties, change *Class Loader Order* to **Classes loaded with local class loader first (parent last)** and *WAR Class Loader Policy* to **Single class loader for application**

4. Save the changes to the master configuration.

6.2. SETTING SYSTEM PROPERTIES FOR THE HEADLESS PROCESS AUTOMATION MANAGER CONTROLLER

After you install the headless Process Automation Manager controller, set the system properties listed in this section on your application server or servers to enable proper interaction with the headless Process Automation Manager controller.

> **NOTE**
>
> For optimal results, install Process Server and the headless Process Automation Manager controller on different servers in production environments. In development environments, you can install Process Server and the headless Process Automation Manager controller on the same server. In either case, be sure to make these property changes on all application servers where the headless Process Automation Manager controller is installed.

Prerequisites

- Process Server and the headless Process Automation Manager controller are installed on the application server instance.

Procedure

1. Specify the following JVM property values on the application server instance where the headless Process Automation Manager controller is installed:

   Table 6.1. Required properties for the headless Process Automation Manager controller

Name	Requirement
org.kie.server.user	A user with the **kie-server** role
org.kie.server.pwd	The password for the user specified in the **org.kie.server.user** property

2. Specify the following JVM property values on the application server instance where Process Server is installed:

   Table 6.2. Required properties for Process Server when headless Process Automation Manager controller is installed

Name	Requirement
org.kie.server.controller.user	A user with the **kie-server** role
org.kie.server.controller.pwd	The password for the user specified for the **org.kie.server.controller.user** property
org.kie.server.id	The ID or name of the Process Server installation, such as **rhdm700-decision-server-1**
org.kie.server.location	The URL of the Process Server, **http://\<HOST\>:\<PORT\>/kie-server/services/rest/server**

Name	Requirement
org.kie.server.controller	The URL of the headless Process Automation Manager controller, **http://<HOST>: <PORT>/controller/rest/controller**

<HOST> is the ID or name of the Process Server host, for example, **localhost** or **192.7.8.9**.

<PORT> is the port of the Process Server host, for example, **7001**.

6.3. MAPPING THE HEADLESS PROCESS AUTOMATION MANAGER CONTROLLER GROUP AND ROLE

After the headless Process Automation Manager controller is installed, in the WebSphere Integrated Solutions Console, you must create the **kie-server** group and then map the **kie-server** role to the **kie-server** group.

Prerequisites

- The headless Process Automation Manager controller is installed on the IBM WebSphere Application Server instance.

Procedure

1. In the WebSphere Integrated Solutions Console, create the **kie-server** group and a user for that group, as described in *Creating the Process Server group and role* .

2. Navigate to **Applications → Application Types → WebSphere Enterprise Applications** and select the newly installed headless Process Automation Manager controller.

3. Under **Detail Properties**, click **Security Role to User/Group Mapping**

4. Select the **kie-server** role and click **Map Groups** to search for the **kie-server** group.

5. Move the **kie-server** group from the **Available** list to the **Selected** list and click **OK**. This mapping gives the previously created administrator user access to Process Automation Manager.

6. Click **Save** to complete the mapping.

6.4. VERIFYING THE INSTALLATION

After you install the headless Process Automation Manager controller and define the required system properties and role requirements on the application server, verify that the headless Process Automation Manager controller works correctly.

Prerequisites

- Process Server and the headless Process Automation Manager controller are installed on the application server instance.

- You have set all required system properties and role requirements for the headless Process Automation Manager controller on the application server.

Procedure

In your command terminal, enter the following command to verify that the headless Process Automation Manager controller is working:

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

<HOST> is the ID or name of the Process Server host, for example, **localhost** or **192.7.8.9**.

<PORT> is the port of the Process Server host, for example, **7001**.

<CONTROLLER> and **<CONTROLLER_PWD>** are the user credentials that you created in this section.

The command should return information about the Process Server instance.

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

If the headless Process Automation Manager controller is not running, stop and restart the application server instance and try again to access the headless Process Automation Manager controller URL or API.

CHAPTER 7. CONFIGURING AN EMBEDDED PROCESS ENGINE AND DECISION ENGINE IN IBM WEBSPHERE APPLICATION SERVER

An embedded engine is a light-weight workflow and rule engine that enables you to execute your decisions and business processes. An embedded engine can be part of a Red Hat Process Automation Manager application or it can be deployed as a service through OpenShift, Kubernetes, and Docker. You can embed an engine in a Red Hat Process Automation Manager application through the API or as a set of contexts and dependency injection (CDI) services.

If you intend to use an embedded engine with your Red Hat Process Automation Manager application, you must add Maven dependencies to your project by adding 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. 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:

```xml
<dependencyManagement>
 <dependencies>
  <dependency>
   <groupId>com.redhat.ba</groupId>
   <artifactId>ba-platform-bom</artifactId>
   <version>7.5.1.redhat-00001</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.

   - 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

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

     <!-- Core dependencies for process engine -->
     ```

27

```xml
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-flow</artifactId>
</dependency>

<dependency>
 <groupId>org.jbpm</groupId>
  <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>
```

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>
  <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 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>
```

3. If you use a process engine or decision engine with persistence support in your project, you must declare the following hibernate dependencies in the **dependencyManagement** secion of your **pom.xml** file by copying the **version.org.hibernate-4ee7** property from the Red Hat Business Automation BOM file:

Hibernate dependencies

```
<!-- hibernate dependencies -->
<dependencyManagement>
 <dependencies>
   <dependency>
   <groupId>org.hibernate</groupId>
   <artifactId>hibernate-entitymanager</artifactId>
   <version>${version.org.hibernate-4ee7}</version>
   </dependency>

   <dependency>
   <groupId>org.hibernate</groupId>
   <artifactId>hibernate-core</artifactId>
   <version>${version.org.hibernate-4ee7}</version>
```

```
        </dependency>
      </dependencies>
    </dependencyManagement>
```

CHAPTER 8. SECURING PASSWORDS WITH A KEYSTORE

You can use a keystore to encrypt passwords that are used for communication between Business Central and Process Server. You should encrypt both controller and Process Server passwords. If Business Central and Process 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.

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

Prerequisites

- Process Server is installed in IBM WebSphere Application Server.

- A Process Server user with the **kie-server** role has been created, as described in Section 5.1, "Creating the Process Server group and role".

- Java 8 or higher is installed.

Procedure

1. Create a JCEKS keystore.

2. When prompted, enter the password for the Process Server user that you created.

3. Set the system properties listed in the following table:

   Table 8.1. System properties used to load a Process 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/keystore.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

System property	Placeholder	Description
kie.keystore.key.ctrl.pwd	**<KEY_CONTROL_P WD>**	Password of the alias for default REST Process Automation Controller with the stored password

4. Start Process Server to verify the configuration.

CHAPTER 9. NEXT STEPS

- *Getting started with decision services*

- *Designing a decision service using guided decision tables*

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Thursday, October 31, 2019.