Red Hat Process Automation Manager 7.0

Packaging and deploying a project

Last Updated: 2018-07-20

Red Hat Process Automation Manager 7.0 Packaging and deploying a project

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

Legal Notice

Abstract

This document describes how to package and deploy a project in Red Hat Process Automation Manager 7.0

Table of Contents

PREFACE

As a business analyst or business rules developer, you must package and deploy a developed project onto a Process Server in order to begin using the business assets you have created in Red Hat Process Automation Manager. You can package and deploy a project using Business Central or using an external Maven repository.

Prerequisite

The project to be deployed has been developed and tested. Consider using test scenarios in Business Central to test the assets in your project. For example, see *Testing a decision service using test scenarios*.

CHAPTER 1. PROJECTS

Projects contain the business assets that you develop in Red Hat Process Automation Manager and are assigned to a space (for example, **`MyProject`** within **`MySpace`**). Projects also contain configuration files such as a Maven project object model file (**`pom.xml`**), which contains build, environment, and other information about the project, and a KIE module descriptor file (**`kmodule.xml`**), which contains the KIE Base and KIE Session configuration for the assets in the project.

CHAPTER 2. PACKAGING A PROJECT

In Red Hat Process Automation Manager, each project is packaged as a Knowledge JAR (KJAR) file and deployed to a Process Server that runs the decision services, process applications, and other deployable assets (collectively referred to as *services*) from that KJAR file. These services are consumed at run time through an instantiated KIE container, or *deployment unit*. Project KJAR files are stored in a Maven repository and identified by three values: **`GroupId`**, **`ArtifactId`**, and **`Version`** (GAV). The **`Version`** value must be unique for every new version that might need to be deployed. To identify an artifact (including a KJAR file), you need all three GAV values.

Although projects are packaged automatically in Red Hat Process Automation Manager as KJAR files with assigned GAV values, ensure that you validate the GAV values and set new values as needed in order to properly package your project for deployment.

2.1. SETTING THE GROUP ID, ARTIFACT ID, AND VERSION (GAV) VALUES IN A PROJECT

The GAV values identify the project in a Maven repository. When Business Central and Process Server are on the same file system and use the same Maven repository, the project is updated in the repository each time you build a new version of your project. However, if Business Central and Process Server are on separate file systems and use separate local Maven repositories, then you must update a project GAV value, usually the version, for any new versions of the project to ensure that the project is seen as a different artifact alongside the old version. For development purposes, you can also add the **`SNAPSHOT`** suffix in the version to instruct Maven to get a new snapshot update according to the Maven policy.

You can set the GAV values in the project **Settings** screen.

Procedure

1. In Business Central, go to **Menu** → **Design** → **Projects** and click the project name. Alternatively, click **Add Project** to create a new project.
2. Click the project **Settings** tab.
3. In **General Settings**, modify the **Group ID**, **Artifact ID**, or **Version** fields as necessary. If you have deployed the project and are developing a new version, usually you need to increase the version number. For development purposes, you can also add the **`SNAPSHOT`** suffix in the version to instruct Maven to get a new snapshot update according to the Maven policy.
4. Click **Save** to finish.

2.2. DUPLICATE GAV DETECTION

All Maven repositories are checked for any duplicated **`GroupId`**, **`ArtifactId`**, and **`Version`** (GAV) attributes. If a GAV duplicate exists, the performed operation is canceled.

The duplicate GAV detection is executed every time you perform the following operations:

- Save a project definition for the project.
- Save the **`pom.xml`** file.
- Install, build, or deploy a project.

The following Maven repositories are checked for duplicate GAVs:

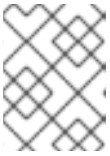
- Repositories specified in the **<repositories>** and **<distributionManagement>** elements of the **pom.xml** file.
- Repositories specified in the Maven **settings.xml** configuration file.

2.2.1. Managing duplicate GAV detection settings

Users with the **admin** role can modify the list of repositories that are checked for duplicate GAVs.

Procedure

1. In Business Central, go to **Menu** → **Design** → **Projects** and click the project name.
2. Click the project **Settings** tab and then click **Validation** to open the list of repositories.
3. Select or clear any of the listed repository options to enable or disable duplicate GAV detection. In the future, duplicate GAVs will be reported for only the repositories you have enabled for validation.



NOTE

To disable this feature, set the **org.guvnor.project.gav.check.disabled** system property to **true**.

CHAPTER 3. DEPLOYING A PROJECT

The deployment process can vary based on the requirements of your infrastructure.

In a simple deployment of Red Hat Process Automation Manager, you have one Business Central and one Process Server. You can use Business Central to develop your business assets and services and also to manage the Process Server. You can build your project in Business Central and deploy it automatically onto the Process Server. To enable automatic deployment, Business Central includes a built-in Maven repository. You can use Business Central to manage the Process Server, deploying, removing, starting, and stopping any of the services and their project versions that you have built.

You can also connect several Process Servers to the same Business Central and group them into different server configurations (in **Menu** → **Deploy** → **Execution Servers**). Servers belonging to the same server configuration run the same services, but you can deploy different projects or different versions of projects on different configurations. For example, you could have test servers in the **`Test`** configuration and production servers in a **`Production`** configuration. As you develop business assets and services in a project, deploy the project on the **`Test`** server configuration and then, when a version of the project is sufficiently tested, you can deploy it on the **`Production`** server configuration.

In this case, to keep developing the project, change the version in the project settings. Then the new version and the old version are seen as different artifacts in the built-in Maven repository. You can deploy the new version on the **`Test`** server configuration and keep running the old version on the **`Production`** server configuration. This deployment process is simple but has significant limitations. Notably, there is not enough access control: a developer can deploy a project directly into production.

If you require a proper integration process, you can use an external Maven repository (for example, Nexus). You can configure Business Central to push project files into the external repository instead of the built-in repository. You can still use Business Central to deploy projects on a test Process Server. However, in this process, other Process Servers (for example, staging and production) are not connected to Business Central. Instead, they retrieve the project KJAR files and any necessary dependencies from your Maven repository. You can progress the KJAR versions through your repository as necessary, in line with the integration process that you want to implement.

When you set up a Process Server, you can configure access to a Maven repository and then you can use the REST API of the server to load and start a KIE container (deployment unit) with the services from the project. If you deploy a Process Server on OpenShift, you can configure it to load and start a service from a Maven repository automatically. A Maven project and a Java client library for automating the API calls are available.

3.1. USING BUSINESS CENTRAL TO DEPLOY AND MANAGE A PROJECT

You can use Business Central to build and deploy a project. If you connect multiple Process Servers to Business Central, you can use the web UI to deploy and manage project assets on all the servers.

You can group Process Servers into different server configurations. Servers belonging to the same server configuration run the same services, but you can deploy different projects or different versions of projects on different server configurations.

IMPORTANT

You cannot move a Process Server into a different server configuration using Business Central. You must change the configuration file of the server to change the server configuration name for it.

3.1.1. Building and deploying a project in Business Central

You can build a project in Business Central and automatically deploy it on a Process Server.

Procedure

1. In Business Central, go to **Menu** → **Design** → **Projects** and click the project name.
2. In the upper-right corner, click **Build** and then **Deploy**.
   If only one Process Server is connected to Business Central, or if all connected Process Servers are in the same server configuration, the project is automatically built and deployed.

   If multiple server configurations are available, Business Central displays a dialog.
3. If the dialog appears, verify or set the following values:
   - **Deployment Unit Id** / **Deployment Unit Alias:** Verify the name and alias of the deployment unit (KIE container) running the service within the Process Server. You normally do not need to change these settings.
   - **Server Configuration:** Select the server configuration for deploying this project. You can later deploy it to other configured servers without rebuilding the project.
   - **Start Deployment Unit?:** Verify that this box is selected to start the deployment unit (KIE container). If you clear this box, the service is deployed onto the server but not started.

3.1.2. Managing project deployment in Business Central

You can use Business Central to manage project deployment on one or more Process Servers. Each Process Server must be configured to connect to Business Central. Services in a project run in a KIE container, or *deployment unit*, on a Process Server. You can create deployment units and start them on Process Servers. When you build and deploy a project, the deployment unit is created automatically in the configured server. You can start, stop, and remove deployment units as needed.

3.1.2.1. Creating a deployment unit

One or more deployment units should already exist as part of your Red Hat Process Automation Manager configuration, but if not, you can create a deployment unit from a project that was previously built in Business Central.

Procedure

1. In Business Central, go to **Menu** → **Deploy** → **Execution servers**.
2. Under **Server Configurations**, select an existing configuration or click **New Server Configuration** to create a new configuration.
3. Under **Deployment Units**, click **Add Deployment Unit**.
4. In the table within the window, select a GAV and click **Select** next to the GAV to populate the deployment unit data fields.
5. Select the **Start Deployment Unit?** box to start the service immediately, or clear the box to start it later.
6. Click **Finish**.

The new deployment unit for the service is created and placed on the Process Servers that are configured for this server configuration. If you have selected **Start Deployment Unit?**, the service is started.

3.1.2.2. Managing deployment units

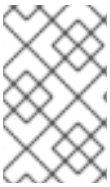
When a deployment unit is started, the service in the deployment unit is available for use. You can start, stop, or remove deployment units with services as necessary.

Procedure

1. In Business Central, go to **Menu** → **Deploy** → **Execution servers**.
2. Under **Server Configurations**, select a configuration.
3. Under **Deployment Units**, select a deployment unit.
4. Click **Start**, **Stop**, or **Remove** in the upper-right corner. To remove a running deployment unit, stop it and then remove it.

3.1.3. Configuring a Process Server to connect to Business Central

You can configure a Process Server to connect to Business Central. You can also configure the name of the server configuration. The server becomes a part of that server configuration and any projects deployed on the server configuration are deployed on the server.



NOTE

If you are deploying Process Server on OpenShift, see *Deploying a Red Hat Process Automation Manager 7.0 managed server environment on Red Hat OpenShift Container Platform* for instructions about configuring it to connect to Business Central.

Prerequisite

Process Server is installed. For installation options, see *Planning a Red Hat Process Automation Manager installation*.

Procedure

1. Create a **`settings.xml`** file for connecting to the Maven repository that Business Central uses. If you have not changed the configuration of Business Central, it uses its own built-in Maven repository. Use the following contents for **`settings.xml`**:

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">

  <servers>
    <server>
      <id>remote-repo</id>
      <username>user</username>
      <password>pwd</password>
    </server>
  </servers>
```

```
  <profiles>
    <profile>
      <id>additional-maven-repos</id>
      <repositories>
        <repository>
          <id>remote-repo</id>
          <url>http://centralhost:centralport/business-
central/maven2/</url>
        </repository>
      </repositories>
    </profile>
  </profiles>
  <activeProfiles>
    <activeProfile>additional-maven-repos</activeProfile>
  </activeProfiles>
</settings>
```

Replace the following values:

- **user:** The user name for a user who can log in to Business Central
- **pwd:** The password for the user who can log in to Business Central
- **centralhost:** The host name for Business Central
- **centralport:** The port for Business Central

If Business Central is deployed on OpenShift, remove **`business-central`** from the URL.

2. Save the file in a known location, for example, **`/opt/maven/settings.xml`**.
3. In your Red Hat Process Automation Manager installation directory, navigate to the **`standalone-full.xml`** file. For example, if you use a Red Hat JBoss EAP installation for Red Hat Process Automation Manager, go to **`$EAP_HOME/standalone/configuration/standalone-full.xml`**.
4. Open **`standalone-full.xml`** and under the **`<system-properties>`** tag, set the following properties:
   - **org.kie.server.controller.user:** The user name of a user who can log in to the Business Central.
   - **org.kie.server.controller.pwd:** The password of the user who can log in to the Business Central.
   - **org.kie.server.controller:** The URL for connecting to the API of Business Central.

   Normally, the URL is **`http://<centralhost>:<centralport>/business-central/rest/controller`**, where **`<centralhost>`** and **`<centralport>`** are the host name and port for Business Central. If Business Central is deployed on OpenShift, remove **`business-central/`** from the URL.

   - **org.kie.server.location:** The URL for connecting to the API of Process Server. Normally, the URL is **`http://<serverhost>:<serverport>/kie-server/services/rest/server`**, where **`<serverhost>`** and **`<serverport>`** are the host name and port for Process Server.

- **org.kie.server.id:** The name of a server configuration. If this server configuration does not exist in Business Central, it is created automatically when Process Server connects to Business Central.
- **kie.maven.settings.custom:** The full path to the `settings.xml` file for connecting to the Maven repository.

For example:

```
<property name="org.kie.server.controller.user"
value="central_user"/>
<property name="org.kie.server.controller.password"
value="central_password"/>
<property name="org.kie.server.controller"
value="http://central.example.com:8080/business-
central/rest/controller"/>
<property name="org.kie.server.location"
value="http://kieserver.example.com:8080/kie-
server/services/rest/server"/>
<property name="org.kie.server.id" value="production-servers"/>
<property name="kie.maven.settings.custom"
value="/opt/maven/settings.xml"/>
```

5. Start or restart the Process Server.

3.2. USING AN EXTERNAL MAVEN REPOSITORY TO DEPLOY A PROJECT

You can use an external Maven repository (for example, Nexus) to set up an integration process. You can configure Business Central to push project files into the external repository instead of the built-in repository. You can still use Business Central to deploy projects on a test Process Server. However, in this process, other Process Servers (for example, staging and production) are not connected to Business Central. Instead, they retrieve the project KJAR files and any necessary dependencies from your Maven repository. You can progress the KJAR versions through your repository as necessary, in line with the integration process that you want to implement.

When you set up a Process Server, you can configure access to a Maven repository and then you can use the REST API of the server to load and start a KIE container (deployment unit) with the services from the project. If you deploy a Process Server on OpenShift, you can configure it to load and start a service from a Maven repository automatically. A Maven project and a Java client library for automating the API calls are available.

3.2.1. Configuring an external Maven repository for Business Central and Process Server

You can configure Business Central and Process Server to use an external Maven repository instead of the built-in repository. In this case, all built project KJAR files are pushed into this repository. You can progress these files through the repository as necessary to implement your integration process.

Prerequisite

Business Central and Process Server are installed. For installation options, see *Planning a Red Hat Process Automation Manager installation*.

Procedure

1. Create a Maven **`settings.xml`** file with connection and access details for your external repository. For details about the **`settings.xml`** file, see the Maven Settings Reference.
2. Save the file in a known location, for example, **`/opt/custom-config/settings.xml`**.
3. In your Red Hat Process Automation Manager installation directory, navigate to the **`standalone-full.xml`** file. For example, if you use a Red Hat JBoss EAP installation for Red Hat Process Automation Manager, go to **`$EAP_HOME/standalone/configuration/standalone-full.xml`**.
4. Open **`standalone-full.xml`** and under the under the **`<system-properties>`** tag, set the **`kie.maven.settings.custom`** property to the full path name of the **`settings.xml`** file. For example:

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

5. Start or restart Business Central and Process Server.

3.2.2. Starting a service in Process Server

If you have configured a Process Server to use a Maven repository, you can use an API call to start a KIE container (deployment unit) and the services in it.

Procedure

To load a service into a KIE container in the Process Server and start it, run the following command to send an API request:

```
$ curl --user "<username>:<password>" -H "Content-Type: application/json" -X PUT -d '{"container-id" : "<containerID>","release-id" : {"group-id" : "<groupID>","artifact-id" : "<artifactID>","version" : "<version>"}}' http://<serverhost>:<serverport>/kie-server/services/rest/server/containers/<containerID>
```

Replace the following values:

- **<username>:** The user name of a user who can log on to Process Server and administer it.
- **<password>:** The password for the user who can log on to Process Server and administer it.
- **<containerID>:** The identifier for the KIE container (deployment unit). You can use any random identifier but it must be the same in both places in the command (the URL and the data).
- **<groupID>**, **<artifactID>**, **version:** The project GAV values.
- **<serverhost>:** The host name for the Process Server, or **`localhost`** if you are running the command on the same host as the Process Server.
- **<serverport>:** The port number for the Process Server.

For example:

```
curl --user "rhpamAdmin:password@1" -H "Content-Type: application/json" -X
PUT -d '{"container-id" : "kie1","release-id" : {"group-id" :
"org.kie.server.testing","artifact-id" : "container-crud-tests1","version"
: "2.1.0.GA"}}' http://localhost:39043/kie-
server/services/rest/server/containers/kie1
```

3.2.3. Stopping and removing a service in Process Server

You can use an API call to stop and remove a KIE container (deployment unit) with services on Process Server. You need the container identifier that was configured when the container was started.

Procedure

To stop and remove a KIE container with services on Process Server, run the following command to send an API request:

```
$ curl --user "<username>:<password>" -X DELETE http://<serverhost>:
<serverport>/kie-server/services/rest/server/containers/<containerID>
```

Replace the following values:

- **<username>:** The user name of a user who can log on to Process Server and administer it.
- **<password>:** The password for the user who can log on to Process Server and administer it.
- **<containerID>:** The identifier for the KIE container (deployment unit). You can use any random identifier but it must be the same in both places in the command (the URL and the data).
- **<serverhost>:** The host name for the Process Server, or **`localhost`** if you are running the command on the same host as the Process Server.
- **<serverport>:** The port number for the Process Server.

For example:

```
curl --user "rhpamAdmin:password@1" -X DELETE http://localhost:39043/kie-
server/services/rest/server/containers/kie1
```

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on: Friday, July 20, 2018.