



Red Hat JBoss Fuse 6.3

Integration Guide

Use Integration packs for Red Hat JBoss Fuse

Red Hat JBoss Fuse 6.3 Integration Guide

Use Integration packs for Red Hat JBoss Fuse

JBoss A-MQ Docs Team

Content Services

fuse-docs-support@redhat.com

Legal Notice

Copyright © 2016 Red Hat.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Use this guide to help you use the Integration packages on Red Hat JBoss Fuse.

Table of Contents

CHAPTER 1. INSTALL INTEGRATION PACK	3
1.1. LIMITED ENTITLEMENT TO BPM SUITE AND BRMS	3
1.2. PACK DEFINITIONS	3
1.3. INSTRUCTIONS FOR JBOSS FUSE ON JBOSS EAP	3
1.4. INSTRUCTIONS FOR JBOSS FUSE ON KARAF	4
CHAPTER 2. KIE-EAP-DISTRIBUTION-BPMS-LAYER PACKAGE	7
APPLY THE PACKAGE	7
CONTENTS OF THE JBOSS BPM SUITE LAYER INTEGRATION PACKAGE	7
JBOSS BPM SUITE CAN INVOKE CAMEL	7
CAMEL CAN INVOKE JBOSS BPM SUITE	7
JBOSS BPM SUITE CAN ORCHESTRATE SWITCHYARD SERVICES	8
SWITCHYARD CAN INVOKE JBOSS BPM SUITE	8
CAMEL CAN USE RULES	8
SWITCHYARD CAN USE RULES	9
OVERVIEW	9
CHAPTER 3. CONFIGURING MAVEN DEPENDENCIES	10
INTEGRATING APACHE CXF WITH KIE-CAMEL	10
CHAPTER 4. DOWNLOAD THE OFFLINE REPOSITORY	12

CHAPTER 1. INSTALL INTEGRATION PACK

The Integration packages for JBoss Fuse 6.3 contain libraries and components that can interact with other integration modules that are not included in the base JBoss Fuse 6.3 package.

1.1. LIMITED ENTITLEMENT TO BPM SUITE AND BRMS

A JBoss Fuse subscription includes a limited entitlement to use BPM Suite and BRMS in embedded form in SwitchYard (in the SwitchYard BPM component and in SwitchYard Rules component). *All other uses require a separate BPM Suite or BRMS subscription.*



NOTE

Fuse Service Works (FSW) subscriptions included an entitlement for use of embedded rules (BRMS) and BPM when used with SwitchYard. When JBoss Fuse and FSW merged, the entitlement was carried forward, but only for use with SwitchYard. All other uses (for example, from Camel) require a separate BPM Suite or BRMS subscription.

1.2. PACK DEFINITIONS

The JBoss Fuse Integration Packages contain the following components:

Package Name	Available Download
Fuse Integration distro for Karaf	<code>fuse-integration-karaf-distro-1.7.0.redhat-version.zip</code>
Fuse Integration distro for JBoss EAP	The <code>fuse-integration-eap-full-1.7.0.redhat-version-installer.zip</code> contains <code>fuse-integration-eap-installer-1.7.0.redhat-version.jar</code> and the <code>kie-eap-installer-6.5.0.Final-redhat-15.jar</code>
	The <code>fuse-integration-eap-installer-1.7.0.redhat-version.jar</code> is available for download on its own.
KIE JBoss Fuse Distributions BPMS layer	<code>kie-eap-installer-6.5.0.Final-redhat-15.jar</code> or the <code>kie-eap-distributions-bpms-layer-6.4.0.Final-redhat-8.zip</code>
Fuse Integration Pack Maven repository	<code>jboss-fuse-intpack-1.7.0.redhat-version-maven-repository.zip</code>

1.3. INSTRUCTIONS FOR JOSS FUSE ON JOSS EAP

Prerequisites

JBoss Fuse 6.3 must be installed on *Red Hat JBoss Enterprise Application Platform (JBoss EAP) 6.4*. See [JBoss EAP 6.4 Installation Guide](#) and [Installation on JBoss EAP](#) for installation instructions.

Identify the correct sequence of packages to install

When you are installing an Integration Pack over Fuse on JBoss EAP, you must match the version of the Integration Pack to the version of JBoss Fuse. For example, the JBoss Fuse version may be the GA version or a roll-up patch. If the JBoss Fuse version is the GA version, then the Integration Pack version must be the GA version. However, if the JBoss Fuse version is roll-up pack 3, the Integration Pack version must also be roll-up pack 3. See the [JBoss Fuse 6.3 Software Downloads](#) page on the Red Hat Customer Portal for available version numbers.

Install the Fuse Integration Pack on JBoss Fuse on JBoss EAP

The integration Fuse Integration Pack pack must be installed on the Fuse/EAP instance of JBoss EAP.

The JBoss EAP installation directory is referred to as **EAP_HOME** in these instructions.

1. Make sure that the JBoss EAP server is stopped. See [chapter "Start and Stop the Application Server" in "Installation on JBoss EAP"](#).
2. Unpack the downloaded Integration Pack file, **fuse-integration-eap-installer-distro-version.zip**, into a temporary location.
3. Navigate to **EAP_HOME** and install the KIE EAP package using the following command:

```
java -jar temporary location/kie-eap-installer-version.jar
```

4. Stay in **EAP_HOME** and install the Fuse integration package using the following command:

```
java -jar temporary location/fuse-integration-eap-installer-version.jar
```

Run the Quickstarts

The Fuse integration quickstarts can be found in **EAP_HOME/quickstarts/fuse-integration**.

1. Start the instance of JBoss EAP where the integraton pack has been installed. See [chapter "Start and Stop the Application Server" in "Installation on JBoss EAP"](#)
2. Navigate to **EAP_HOME/quickstarts/fuse-integration**.
3. Navigate to the directory for the quickstart you want to run. Follow the instructions in the README file to execute the quickstart. Each quickstart has different instructions.

1.4. INSTRUCTIONS FOR JOSS FUSE ON KARAF

Install the Integration Pack on JBoss Fuse on Karaf

The Integration Pack pack must be installed on an instance of JBoss Fuse on Karaf. Make sure that the server is stopped.



IMPORTANT

The Integration Pack can only be installed in a standalone Karaf container. The Integration Pack is *not* supported in Fabric.

1. Download **fuse-integration-karaf-distro-1.7.0.redhat-version.zip** and unpack the .zip files on to JBoss Fuse on Karaf.
2. Start JBoss Fuse.
3. In Karaf, add Karaf BRMS assemblies:

```
features:addurl mvn:org.apache.karaf.assemblies.features/brms-features/2.4.0.redhat-630283/xml/features
```

4. Add Drools if it has not already been added:

```
features:addurl mvn:org.drools/drools-karaf-features/6.5.0.Final-redhat-15/xml/features-fuse
```

5. Add the JBoss Fuse integration features files:

```
features:addurl mvn:org.jboss.integration.fuse/karaf-features/1.7.0.redhat-630036/xml/features
```

6. Install the core features by entering the following command:

```
features:install feature-name
```

and replacing *feature-name* with the feature name one at a time:

```
fuse-bxms-switchyard-bpm
fuse-bxms-switchyard-common-knowledge
fuse-bxms-switchyard-rules
```

7. Add the JBoss Fuse integration quickstart features files:

```
features:addurl mvn:org.jboss.integration.fuse.quickstarts/karaf-features/1.7.0.redhat-630036/xml/features
```

8. Install the quickstart features by entering the following command:

```
features:install feature-name
```

and replacing *feature-name* with the feature name one at a time:

```
fuse-bxms-quickstart-switchyard-bpm-service
fuse-bxms-quickstart-switchyard-rules-camel-cbr
fuse-bxms-quickstart-switchyard-rules-interview
fuse-bxms-quickstart-switchyard-rules-interview-container
fuse-bxms-quickstart-switchyard-rules-interview-dtable
fuse-bxms-quickstart-switchyard-library
fuse-bxms-quickstart-switchyard-rules-multithread
```

```
fuse-bxms-quickstart-switchyard-rules-csv
fuse-bxms-quickstart-switchyard-helpdesk-ds
fuse-bxms-quickstart-switchyard-helpdesk
fuse-bxms-quickstart-switchyard-drools-channels
fuse-bxms-quickstart-camel-blueprint-drools-decision-table
fuse-bxms-quickstart-camel-spring-drools-decision-table
fuse-bxms-quickstart-camel-jbpm-workitems
```

The Fuse integration quickstarts can be found in ***fuse_install_dir/quickstarts/fuse-integration***. Navigate to the directory for the quickstart you want to run. Follow the instructions in the README file to execute the quickstart. Each quickstart has different instructions.

CHAPTER 2. KIE-EAP-DISTRIBUTION-BPMS-LAYER PACKAGE

The Integration Pack contains a .zip file which contains JBoss EAP modules for *Red Hat JBoss BPM Suite* (JBoss BPM Suite) and *Red Hat JBoss BRMS* (JBoss BRMS) integration.



WARNING

JBoss BPM Suite and JBoss BRMS are separate products. The integration package can only be used if you have a subscription to the product you want to integrate with.

APPLY THE PACKAGE

The package will be applied by the SwitchYard/BRMS Integration Pack installer jar.

CONTENTS OF THE JOSS BPM SUITE LAYER INTEGRATION PACKAGE

The following table shows the relationships between different components in JBoss BPM Suite and JBoss BRMS, and JBoss Fuse. Expansion of the table entries can be found below the table.

Table 2.1. Relationship Between JBoss BPM Suite/JBoss BRMS components and Fuse components

	Camel	SwitchYard
JBoss BPM Suite	JBoss BPM can invoke Camel	JBoss BPM Suite can orchestrate SwitchYard services
	Camel can invoke JBoss BPM Suite	SwitchYard can invoke JBoss BPM Suite
Rules	Rules does not use Camel.	Rules does not use SwitchYard
	Camel can use Rules	SwitchYard can call Rules

JBoss BPM SUITE CAN INVOKE CAMEL

JBoss BPM Suite has a Camel WorkItemHandler. See [JBoss BPM Suite Development Guide](#) for more information.

CAMEL CAN INVOKE JOSS BPM SUITE

There are three interfaces that can be used to invoke JBoss BPM Suite from Camel:

- [IMS](#) – see [JBoss BPM Suite Development Guide](#) [IMS section](#)

- [JMS - see JBoss BPM Suite Development Guide - JMS section](#)
- [REST - JBoss BPM Suite Development Guide - REST API section](#)
- [Java - see JBoss BPM Suite Development Guide - Remote Java API section](#)

JBoss BPM Suite Can Orchestrate SwitchYard Services

The JBoss BPM Suite process definition must be embedded in the SwitchYard application, not on a remote JBoss BPM Suite server.

JBoss BPM Suite will not execute on a remote JBoss BPM Suite server. It will only run via jars provided in the integration pack.

See the provided quickstart `EAP_HOME/quickstarts/fuse-integration/switchyard-bpm-service` for an example.

SwitchYard Can Invoke JBoss BPM Suite

The following scenarios have are supported:

- Invoke/signal local JBoss BPM Suite process defined as a SwitchYard service.
- Invoke/signal remote JBoss BPM Suite process defined as a SwitchYard service.
- Invoke/signal a remote JBoss BPM Suite process *and* insert/execute rules locally as a SwitchYard service using rules definitions resolved as KIE artifacts using Maven.

JBoss BPM Suite process can be embedded in the SwitchYard application or it can be on a JBoss BPM Suite server on a different JVM.

If the process is embedded, the engine used must be the .jar file that is installed as part of the Integration pack. If the JBoss BPM Suite engine is running locally in the JBoss Fuse JVM, then it can only be used to orchestrate SwitchYard services and can not be used to perform other tasks.

If the JBoss BPM Suite process is not embedded, the invocation must occur via the SwitchYard JBoss BPM Suite component. In this case, the user does not need to manage the interaction.

Camel Can Use Rules

You can insert facts and execute rules locally in JBoss Fuse using a Camel route. Rule definitions are resolved as KIE artifacts using Maven.

Camel can invoke Business Rules when the Karaf container is used. This functionality is not available on JBoss Fuse on JBoss EAP. It will be made available in a future release.

For use by JBoss BPM Suite, the Rules must be in a Maven repository where the Kie scanner can find them. The Maven repository can be on the same machine or a remote machine. If the repository is on a remote machine, the rules will be added to the local repository during rules resolution process.

The Rules engine can be placed on the same machine as Camel is installed on, or on a remote JBoss BRMS execution server. For the local engine, binaries are provided by the Integration Pack. For the remote engine, the rules can be invoked using a mechanism such as the REST interface provided by JBoss BRMS.

SWITCHYARD CAN USE RULES

You can insert facts and execute rules locally as a SwitchYard service using rules definitions deployed with the application. This makes the rules static and will require a rebuild to change them.

Alternatively you can insert facts and execute rules locally as a SwitchYard service using rules definitions resolved as KIE artifacts using Maven.. This repository can be either on the same machine as JBoss Fuse, or on a remote machine. Rules stored in a Maven repository can be updated at runtime using Kie scanning.

The Rules engine is installed on the JBoss Fuse server as part of the integration pack installation.

OVERVIEW

Kie-Camel is an Apache Camel component (endpoint) that integrates with KIE (Drools). It allows you to specify a KIE module (using maven GAV) which you can pull into the route and execute. Also, It enables you to specify portions of the message body as facts. For more details about the kie-camel component, see [Apache Camel Integration](#)

CHAPTER 3. CONFIGURING MAVEN DEPENDENCIES

This chapter provides information for using the Apache CXF component with Kie-Camel.

INTEGRATING APACHE CXF WITH KIE-CAMEL

To support both EAP and Karaf containers, kie-camel should support Apache CXF 2.7.x and 3.x versions. Each version might contain different packages, classes, and methods. In your project, if you are using both kie-camel and Apache CXF, the application might display a few classpath collision errors. To avoid this, Red Hat recommends that you exclude the artifacts whose version does not match the container version, and replace them with artifacts whose version matches the version of Apache CXF in your container.

- EAP 6.4.x: You can define `eap.cxf.version` in properties with the Apache CXF version that is in your EAP container.

```
<dependency>
  <groupId>org.jboss.integration.fuse</groupId>
  <artifactId>kie-camel</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.apache.cxf</groupId>
      <artifactId>cxf-core</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.apache.cxf</groupId>
  <artifactId>cxf-core</artifactId>
  <version>${eap.cxf.version}</version>
</dependency>
```

- Fuse on Karaf: Define `karaf.cxf.version` in properties with Apache CXF version that is in your Fuse/Karaf container.

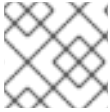
```
<dependency>
  <groupId>org.jboss.integration.fuse</groupId>
  <artifactId>kie-camel</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.apache.cxf</groupId>
      <artifactId>cxf-rt-rs-security-oauth</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.cxf</groupId>
      <artifactId>cxf-rt-frontend-jaxws</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.cxf</groupId>
      <artifactId>cxf-rt-frontend-jaxrs</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.cxf</groupId>

```

```
        <artifactId>cxfrtfeaturesclustering</artifactId>
    </exclusion>
    <exclusion>
        <groupId>org.apache.cxf</groupId>
        <artifactId>cxfrtbindingssoap</artifactId>
    </exclusion>
</exclusions>
</dependency>
<dependency>
    <groupId>org.apache.cxf</groupId>
    <artifactId>cxfrtrssecurityoauth</artifactId>
    <version>${karaf.cxf.version}</version>
</dependency>
<dependency>
    <groupId>org.apache.cxf</groupId>
    <artifactId>cxfrtfrontendjaxws</artifactId>
    <version>${karaf.cxf.version}</version>
</dependency>
<dependency>
    <groupId>org.apache.cxf</groupId>
    <artifactId>cxfrtfrontendjaxrs</artifactId>
    <version>${karaf.cxf.version}</version>
</dependency>
<dependency>
    <groupId>org.apache.cxf</groupId>
    <artifactId>cxfrtfeaturesclustering</artifactId>
    <version>${karaf.cxf.version}</version>
</dependency>
<dependency>
    <groupId>org.apache.cxf</groupId>
    <artifactId>cxfrtbindingssoap</artifactId>
    <version>${karaf.cxf.version}</version>
</dependency>
```

CHAPTER 4. DOWNLOAD THE OFFLINE REPOSITORY

A Maven repository that contains the artifacts required for the Integration Pack is available for download. It is an optional download and is provided for when an offline Maven repository is used.



NOTE

This repository is useful only if you are using offline Maven repositories.

To use an offline Maven repository:

1. In the Red Hat customer portal, go to <https://access.redhat.com/jbossnetwork/restricted/listSoftware.html?product=jboss.fuse&downloadType=distributions>.
2. Select the Fuse version you are using. This displays the downloads for that Fuse version.
3. Select the offline Maven repository that meets your needs.
4. Click **Download**.
5. Unpack the downloaded zip file into an accessible location.
6. Add entries for the unzipped repositories to Maven's **settings.xml** file. Base your entry on the code below.

```
<repository>
  <id>BPMS-repo</id>
  <name>Red Hat JBoss Fuse 6.3.0</name>
  <url>file://file-location/BPMS-repo/maven-repository/</url>
  <layout>default</layout>
  <releases>
    <enabled>>true</enabled>
    <updatePolicy>never</updatePolicy>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
    <updatePolicy>never</updatePolicy>
  </snapshots>
</repository>
```