



# Red Hat Decision Manager 7.4

## Red Hat Decision Manager と Red Hat Fuse の 統合

ガイド



# Red Hat Decision Manager 7.4 Red Hat Decision Manager と Red Hat Fuse の統合

---

ガイド

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Integrating\_Red\_Hat\_Fuse\_with\_Red\_Hat\_Decision\_Manager.ent file |.

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

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

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

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

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

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

The OpenStack<sup>®</sup> 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.

## 概要

本書では、Apache Karaf および Red Hat JBoss Enterprise Application Platform で、Red Hat Fuse を Red Hat Decision Manager 7.4 と統合する方法を説明します。

---

## 目次

はじめに .....	3
第1章 RED HAT FUSE および RED HAT DECISION MANAGER .....	4
1.1. RED HAT DECISION MANAGER の KARAF 機能 .....	4
第2章 APACHE KARAF 上に FUSE を統合した RED HAT DECISION MANAGER のデシジョンエンジンおよびプロセスエンジン .....	6
2.1. KARAF 上の古くなった RED HAT DECISION MANAGER 機能 XML ファイルの削除 .....	6
2.2. XML ファイルを使用した KARAF への RED HAT DECISION MANAGER 機能のインストール .....	7
2.3. MAVEN を使用した KARAF への RED HAT DECISION MANAGER 機能のインストール .....	8
第3章 RED HAT DECISION MANAGER を使用した RED HAT JBOSS EAP 7.2 内の FUSE のインストール ...	10
第4章 KIE-CAMEL コンポーネント .....	12
付録A バージョン情報 .....	13



## はじめに

システム管理者は、Apache Karaf および Red Hat JBoss Enterprise Application Platform で、Red Hat Decision Manager と Red Hat Fuse を統合して、統合サービス間の通信を容易化します。

## 第1章 RED HAT FUSE および RED HAT DECISION MANAGER

Red Hat Fuse は、アジャイル統合ソリューションの一部である、分散型のクラウドネイティブ統合プラットフォームです。チームはこの分散アプローチを使用することで、必要に応じて統合サービスをデプロイできます。Fuse には、統合エキスパート、アプリケーション開発者、ビジネスユーザーなど、さまざまなユーザーに柔軟にサービスを提供でき、各自がデプロイメント、アーキテクチャー、ツールを選択できます。API 中心のコンテナベースのアーキテクチャーは、各サービスを切り離して、個別に作成、展開、デプロイできるようにします。その結果、企業全体のコラボレーションをサポートする統合ソリューションが実現できます。

Red Hat Decision Manager は、ビジネスルール管理、複合イベント処理、Decision Model & Notation (DMN) 実行、およびプランニングの問題を解決するための Red Hat Business Optimizer を組み合わせた、オープンソースの意思決定管理プラットフォームです。これにより、ビジネス上の意思決定を自動化し、そのロジックをビジネス全体で利用できるようにします。

ルール、デシジョンテーブル、DMN モデルなどのビジネスアセットはプロジェクト内で組織化され、Business Central リポジトリに保存されます。これにより、ビジネス全体で一貫性や透明性を維持し、監査を行えます。ビジネスユーザーは、IT 担当者からのサポートなしでビジネスロジックを編集できます。

Apache Karaf コンテナプラットフォームまたは Red Hat JBoss Enterprise Application Platform に Red Hat Fuse をインストールしてから、このコンテナに Red Hat Decision Manager をインストールし、設定してください。



### 重要

Red Hat Decision Manager 7.4 がサポートする Red Hat Fuse のバージョンについては、[Red Hat Process Automation Manager 7 でサポートされる設定](#) を参照してください。

### 1.1. RED HAT DECISION MANAGER の KARAF 機能

以下の表では、Red Hat Decision Manager の Karaf 機能を紹介します。

機能	説明
<b>drools-module</b>	Drools のコアとコンパイラーが含まれており、プレーン DRL から KIE ベースと KIE セッションを作成するのに使用します。また、実行可能モデルの実装も含まれています。永続性、プロセス、またはデシジョンテーブルを必要とせずに Drools を使用してルール評価ができます。
<b>drools-template</b>	Drools テンプレートが含まれています。
<b>drools-jpa</b>	プロセスまたはデシジョンテーブルを必要とせずに、Drools を使用して永続性 (Persistence) とトランザクション (Transaction) でルール評価を行います。 <b>drools-jpa</b> 機能には <b>drools-module</b> が含まれますが、 <b>droolsjbpn-hibernate</b> 機能をインストールするか、互換性のあるハイバネートバンドルがインストールされているようにしてください。



機能	説明
<b>drools-decisiontable</b>	デシジョンテーブルと合わせて、Drools を使用します。
コアエンジン JAR および <b>kie-ci</b>	Red Hat Decision Manager と KIE スキャナー ( <b>kie-ci</b> ) を使用して、Maven リポジトリから kJAR をダウンロードします。
<b>kie-camel</b>	Fuse と Red Hat Decision Manager を統合する Apache Camel エンドポイントである <b>kie-camel</b> コンポーネントを提供します。
<b>kie-spring</b>	<b>kie-spring</b> コンポーネントをインストールし、XML タグを使用して KIE セッションのリスナーを設定可能にします。

## 第2章 APACHE KARAF 上に FUSE を統合した RED HAT DECISION MANAGER のデジジョンエンジンおよびプロセスエンジン

Apache Karaf は、スタンドアロンで、オープンソースのランタイム環境です。OSGi Alliance の OSGi 標準に基づいています。Karaf は、高度なクラ出力デバッグサポートを備えた OSGi バンドルを通じてモジュール化のサポートを提供します。Karaf コンテナでは、依存関係の複数のバージョンを並行してデプロイできます。ホットコードスワップを使用すると、コンテナをシャットダウンせずにモジュールをアップグレードまたは置き換えることができます。

Red Hat Decision Manager は、Karaf 機能を使用して、Karaf で FUSE と統合されます。これらの機能を使用して、Karaf 上の FUSE 向けに Red Hat Decision Manager の個別のコンポーネントをインストールできます。

機能ファイルは XML ファイルで、このファイルを使用して特定の機能向けにどの DOSGI バンドルをインストールするかを指定します。以下の機能 XML ファイルにより、Red Hat Decision Manager と Fuse on Karaf の統合を容易にします。

- **rhba-features-<FUSE-VERSION>-features.xml**  
このファイルは、Karaf にインストールした Fuse に含まれており、<FUSE-VERSION> は Fuse のバージョンに置き換えます。このファイルは、**system/org/jboss/fuse/features/rhba-features** ディレクトリーの Karaf システムリポジトリに保存されます。このファイルには、Red Hat Decision Manager 機能をインストールするための前提条件が含まれています。
- **kie-karaf-features-7.23.0.Final-redhat-00002-features-fuse.xml**  
このファイルは Red Hat Decision Manager に含まれており、Red Hat Decision Manager 機能を提供します。この機能により、Red Hat Fuse にデプロイできる OSGi 機能が決まります。OSGi ユーザーは、このファイルから機能をインストールして、Red Hat Decision Manager を Fuse にインストールし、それぞれのアプリケーションで使用できます。この機能ファイルは、Red Hat Process Automation Manager で配布されるオンラインおよびオフラインの Maven リポジトリに配置されています。この機能ファイルは、Red Hat Process Automation Manager で配布されるオンラインおよびオフラインの Maven リポジトリに配置されています。このファイルのグループ ID、アーティファクト ID、およびバージョン (GAV) は、**org.kie:kie-karaf-features:7.23.0.Final-redhat-00002** です。

### 2.1. KARAF 上の古くなった RED HAT DECISION MANAGER 機能 XML ファイルの削除

お使いの環境に、以前の Red Hat Decision Manager 機能 XML ファイル (例: **kie-karaf-features-<VERSION>-features.xml**) が含まれている場合には、このファイルと関連するファイルすべてを削除してから、最新の機能 XML ファイルをインストールする必要があります。

#### 前提条件

- 以前の機能 XML ファイルが Apache Karaf の環境に存在する。

#### 手順

1. 以下のコマンドを実行して、使用環境に以前の Red Hat Decision Manager 機能 XML ファイルが含まれているかどうかを確認します。

```
$ JBossFuse:karaf@root> feature:repo-list
$ JBossFuse:karaf@root> feature:list
```

- 以下のコマンドを入力します。<FUSE\_HOME> は Fuse のインストールディレクトリーに置き換えて、Red Hat Fuse コンソールを起動します。

```
$ ./<FUSE_HOME>/bin/fuse
```

- 以下のコマンドを入力します。<FEATURE\_NAME> は、アンインストールする機能の名前に置き換えて、以前の機能 XML ファイルを使用するアプリケーションまたは機能をアンインストールします。

```
JBossFuse:karaf@root> features:uninstall <FEATURE_NAME>
```

以下の例では、機能の削除方法を紹介します。

```
JBossFuse:karaf@root> features:uninstall drools-module
JBossFuse:karaf@root> features:uninstall jbpm
JBossFuse:karaf@root> features:uninstall kie-ci
```

- Karaf の home で、**drools**、**kie**、または **jbpm** を使用するバンドルへの参照を検索します。以下の例では、**grep** を使用してこれらのコンポーネントを検索する方法を示しています。

```
karaf@root> list -t 0 -s | grep drools
karaf@root> list -t 0 -s | grep kie
karaf@root> list -t 0 -s | grep jbpm
```

この例は、上記のコマンドからの出力です。

250	Active	80	7.19.0.201902201522	org.drools.canonical-model
251	Active	80	7.19.0.201902201522	org.drools.cdi
252	Active	80	7.19.0.201902201522	org.drools.compiler

- 以下のコマンドを入力します。**BUNDLE\_ID** は、検索で返されたバンドル ID に置き換えて、以前の手順で検出されたバンドルを削除します。

```
karaf@root> osgi:uninstall BUNDLE_ID
```

- 次のコマンドを入力して、古くなった **drools-karaf-features** の URL を削除します。

```
karaf@root> features:removeurl
mvn:org.kie/kie-karaf-features/VERSION.Final-redhat-VERSION/xml/features
```

- Fuse を再起動します。

## 2.2. XML ファイルを使用した KARAF への RED HAT DECISION MANAGER 機能のインストール

Karaf に Red Hat Decision Manager 機能をインストールして、Red Hat Decision Manager プロセス向けに動的ランタイム環境を作成します。

### 前提条件

- Apache Karaf コンテナで Red Hat Fuse が利用できるようになっている。Apache Karaf への Fuse のインストール手順については、[Installing Red Hat Fuse on the Apache Karaf container](#) を参照してください。
- 「[Karaf 上の古くなった Red Hat Decision Manager 機能 XML ファイルの削除](#)」に記載されているように、以前の Red Hat Decision Manager 機能 XML ファイルが削除されている。

## 手順

Red Hat Decision Manager の機能をインストールするには、次のコマンドを入力します。

```
$ JBossFuse:karaf@root> feature:install <FEATURE_NAME>
```

### 注記

**org.drools.osgi.spring.OsgiKModuleBeanFactoryPostProcessor** の代わりに **org.kie.spring.KModuleBeanFactoryPostProcessor** を使用して、OSGi 環境の KIE 要素を後処理します。

**kie-spring** 機能をインストールする前に **drools-module** 機能をインストールしないようにしてください。先にインストールしてしまうと、**drools-compiler** バンドルにより、**kie-spring** がエクスポートしたパッケージが検出されなくなります。

これらの機能を間違った順番でインストールした場合は、**osgi:refresh drools-compiler\_bundle\_ID** を実行して、**drools-compiler** が強制的に **Import-Package** メタデータをリビルドするようにします。

このコマンドでは、**<FEATURE\_NAME>** は、「[Red Hat Decision Manager の Karaf 機能](#)」に記載の機能の1つに置き換えます。

## 2.3. MAVEN を使用した KARAF への RED HAT DECISION MANAGER 機能のインストール

必要に応じて、Apache Karaf 上にある Fuse で Red Hat Decision Manager をインストールして、統合サービスをデプロイします。

### 前提条件

- Apache Karaf インストールに Red Hat Fuse 7.3 または 7.4 が存在する。インストール手順については、[Installing Red Hat Fuse on the Apache Karaf container](#)を参照してください。
- 「[Karaf 上の古くなった Red Hat Decision Manager 機能 XML ファイルの削除](#)」の説明のように、以前の機能 XML ファイルが削除されている。

## 手順

1. Maven リポジトリを設定するには、テキストエディターで **FUSE\_HOME/etc/org.ops4j.pax.url.mvn.cfg** ファイルを開きます。
2. <https://maven.repository.redhat.com/ga/> リポジトリが **org.ops4j.pax.url.mvn.repositories** 変数に存在することを確認します。必要に応じて、追加してください。



## 注記

エントリーは、コンマ、スペース、およびバックスラッシュ (, \) で区切ります。バックスラッシュを追加すると、強制的に改行されます。

3. Fuse を起動するには以下のコマンドを入力します。<FUSE\_HOME> は Fuse のインストールディレクトリーに置き換えます。

```
$ ./FUSE_HOME/bin/fuse
```

4. インストールの前提条件が含まれる機能ファイルに参照を追加するには、以下のコマンドを入力します。<FUSE\_VERSION> は、インストールする Fuse のバージョンに置き換えます。

```
$ feature:repo-add mvn:org.jboss.fuse.features/rhba-features/<FUSE-VERSION>/xml/features
```

5. 以下のコマンドを入力して、Red Hat Decision Manager 機能の XML ファイルへの参照を追加します。

```
$ JBossFuse:karaf@root> features:addurl mvn:org.kie/kie-karaf-features/VERSION/xml/features-fuse
```

現在の **drools-karaf-features** バージョンをチェックするには、[Red Hat Process Automation Manager 7 Supported Configurations](#) ページを参照してください。

6. 以下のコマンドを入力して、Red Hat Decision Manager 機能の XML ファイルで提供される機能をインストールします。このコマンドでは、<FEATURE\_NAME> は、「[Red Hat Decision Manager の Karaf 機能](#)」に記載の機能の1つに置き換えます。

```
JBossFuse:karaf@root> features:install <FEATURE_NAME>
```

7. 次のコマンドを入力して、インストールを確認します。

```
$ JBossFuse:karaf@root> feature:list
```

機能が正常にインストールされると、ステータスは **started** になります。

## 第3章 RED HAT DECISION MANAGER を使用した RED HAT JBOSS EAP 7.2 内の FUSE のインストール

Red Hat Decision Manager を使用して Red Hat JBoss EAP に Fuse をインストールし、必要に応じて統合サービスをデプロイします。

### 前提条件

- Red Hat JBoss Enterprise Application Platform 7.2 に Red Hat Decision Manager がインストールされていること。インストールの説明は、[Red Hat JBoss EAP 7.2 への Red Hat Decision Manager のインストールおよび設定](#) を参照してください。

### 手順

- Red Hat Decision Manager をインストールした Red Hat JBoss Enterprise Application Platform 7.2 コンテナに Red Hat Fuse をインストールします。インストールの説明は、[Install Fuse 7.3 on JBoss EAP 7.2](#) を参照してください。
- テキストエディターで、Fuse のホームディレクトリーにある **pom.xml** を開きます。
- 以下の例のように **pom.xml** ファイルを編集して、**kie-camel** コンポーネントの依存関係を含めて、統合プロジェクトを作成します。

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-core</artifactId>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
  <exclusions>
    <exclusion>
      <groupId>aopalliance</groupId>
      <artifactId>aopalliance</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.kie.server</groupId>
  <artifactId>kie-server-api</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.jboss.spec.javax.xml.bind</groupId>
      <artifactId>jboss-jaxb-api_2.3_spec</artifactId>
    </exclusion>
    <exclusion>
      <groupId>javax.activation</groupId>
      <artifactId>activation</artifactId>
    </exclusion>
  </exclusions>
```

```
</dependency>
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-bpmn2</artifactId>
</dependency>
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-camel</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.apache.cxf</groupId>
      <artifactId>cxf-core</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.camel</groupId>
      <artifactId>camel-cxf</artifactId>
    </exclusion>
    <exclusion>
      <groupId>org.apache.camel</groupId>
      <artifactId>camel-cxf-transport</artifactId>
    </exclusion>
    <exclusion>
      <groupId>com.thoughtworks.xstream</groupId>
      <artifactId>xstream</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.kie.server</groupId>
  <artifactId>kie-server-client</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.jboss.spec.javax.ws.rs</groupId>
      <artifactId>jboss-jaxrs-api_2.0_spec</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

## 第4章 KIE-CAMEL コンポーネント

**kie-camel** コンポーネントは、Fuse と Red Hat Decision Manager が統合された Red Hat Fuse が提供する Apache Camel エンドポイントです。このコンポーネントを使用して、ルートにするプルして実行できる Maven グループ ID、アーティファクト ID、バージョン (GAV) の識別子を使用して、Red Hat Decision Manager モジュールを指定できます。また、ファクトとしてメッセージボディーの一部を指定できます。埋め込みエンジンや Decision Server で、**kie-camel** コンポーネントを使用できます。

### 埋め込みエンジン

このシナリオでは、KIE エンジンが Fuse 統合プロジェクトと同じコンテナで実行されます。エンジン間の通信には、KIE コマンドを使用できます。エンジン間の通信には、KIE コマンドを使用できます。Camel プロデューサーを作成するには、以下の URI を使用します。

```
kie-local:kie-session-name?action=execute
```

たとえば、次のコマンドを入力して、Spring の Camel ルートを初期化します。

```
<from uri="direct:runCommand" />
<to uri="kie-local:kie-session1?action=execute"/>
```

### Decision Server

このシナリオでは、**kie-camel** コンポーネントは Decision Server REST API を使用して Decision Server に接続します。こうすることでユーザーは、Decision Server API を使用して Decision Server と通信できます。プロデューサーを作成するには、以下の URI を使用します。

```
kie:http://username:password@kie-server-url`
```

たとえば、次のコマンドを入力して、Spring の Camel ルートを初期化します。

```
<from uri="direct:runCommand" />
<to uri="kie:http://user:psswd@localhost:8080/kie-server-services/services/rest/server"/>
```

メッセージには以下のヘッダーが含まれます。

表4.1 メッセージのヘッダーと説明

ヘッダー	説明
<b>CamelKieClient</b>	Decision Server クライアント (必須)
<b>CamelKieOperation</b>	Decision Server クライアント (必須)
<b>CamelKieParameterName</b>	クライアントメソッドパラメーターの値 (任意)
<b>CamelKieBodyParam</b>	メッセージボディーを保存するメソッドパラメーター (任意)



## 付録A バージョン情報

本書の最終更新日: 2021年11月15日(月)