



Red Hat Decision Manager 7.0

**Red Hat JBoss BRMS 6.4 から Red Hat
Decision Manager 7.0 への移行**

Red Hat Decision Manager 7.0 Red Hat JBoss BRMS 6.4 から Red Hat Decision Manager 7.0 への移行

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

法律上の通知

Copyright © 2018 Red Hat, Inc.

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.

概要

本書では、ユーザーデータおよび API を Red Hat JBoss BRMS 6.4 から Red Hat Decision Manager 7.0 に移行する方法を説明します。

目次

前書き	3
第1章 概要	4
第2章 DECISION CENTRAL でのプロジェクト移行作業	5
第3章 JAVA プロジェクトの移行	7
3.1. 移行した JAVA プロジェクトで発生するビルドエラーおよび実行エラーのトラブルシューティング	8
3.1.1. Drools モジュールにおける API 変更	8
3.1.2. Drools モジュールにおけるロジック変更	9
第4章 RED HAT DECISION MANAGER 7.0 における RED HAT BUSINESS OPTIMIZER の変更	11
4.1. DECISION CENTRAL での RED HAT BUSINESS OPTIMIZER アセットの設定の更新	11
第5章 DECISION SERVER のアップグレード	13
付録A バージョン情報	15

前書き

Red Hat Decision Manager 7.0 で使用したい Red Hat BRMS 6.4 プロジェクトがある場合は、始めにそのプロジェクトを移行する必要があります。

前提条件/事前作業

Red Hat Decision Manager 7.0 に移行したいアーティファクトが含まれる Red Hat BRMS 6.4 システムを保有している。

第1章 概要

BRMS バージョン 6.4 を使用していて Red Hat Decision Manager バージョン 7.0 をインストールした場合には、既存のプロジェクトを新たな製品に移行する必要があります。

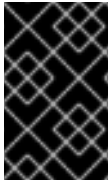
Business Central ワークベンチを使用して作成したアプリケーション (ここでは **デシジョンサービス** と呼びます) を、新たな等価コンポーネントである Decision Central に移行できます。この移行を完了するのに、ベースとなるコードを手作業で変更する必要はありません。

Java コード (たとえば Eclipse) で作成したデシジョンサービスの場合は、Red Hat Decision Manager 7.0 で使用するための修正が必要になります。それぞれのプロジェクトについて **pom.xml** ファイル内の依存関係を更新して、プロジェクトをビルドし直す必要があります。新バージョンでの変更によって、既知のビルドエラーが発生することがあります。その場合には、コードを修正しなければなりません。

既存の Java クライアントアプリケーションを移行する場合にも、それぞれのプロジェクトについて **pom.xml** ファイル内の依存関係を更新する必要があります。アプリケーションが埋め込まれた BRMS エンジン (Drools、OptaPlanner) を使用している場合は、この変更でエンジンも更新されます。アプリケーションが Decision Server を呼び出す場合は、API クライアントライブラリーが更新されます。

クライアントアプリケーションが Java クライアントライブラリーでは作成されておらず、REST API を使用して KIE サーバー (Decision Server) と通信する場合は、互換性のない小規模な API 変更に対応するために修正が必要となる場合があります。

KIE サーバーを停止して同じホスト上の新たな Decision Server を起動するだけで、古いサーバーを新たなサーバーに置き換えることができます。Decision Server 7.0 を使用して、BRMS 6.4 で作成された KJAR ファイルを実行することは可能です。ただし、最大のパフォーマンスを得るためには、プロジェクトを Red Hat Decision Manager 7.0 に移行して、ビルドし直してください。



重要

バージョン 6.4 より古い BRMS を使用している場合は、プロジェクトをバージョン 6.4 に移行してから 7.0 に移行してください。詳細については、『[Red Hat JBoss BPM Suite 6.4 Migration Guide](#)』を参照してください。


第2章 DECISION CENTRAL でのプロジェクト移行作業

BRMS 6.4 の Business Central ワークベンチでプロジェクトを作成している場合は、アセットリポジトリを Decision Central にインポートすることで、プロジェクトを Red Hat Decision Manager 7.0 に移行することができます。

前提条件/事前作業

1. BRMS 6.4 Business Central において、該当プロジェクトが含まれるリポジトリの名前を探します。デフォルトのリポジトリ名は **repository1** です。
2. BRMS 6.4 Business Central を停止します。
3. **.niogit** ディレクトリーを探します。デフォルトのシステムでは、その場所は **\$EAP_home/bin/.niogit** です。この場所は、**\$EAP_home/standalone/configuration/standalone.xml** ファイルのシステムプロパティー **org.uberfire.nio.git.dir** で変更できます。システムプロパティーは、Business Central を起動するコマンドで **-D** オプションを指定して設定することもできます。
4. 別の JBoss EAP システムを使用して、BRMS 6.4 Business Central と同じホスト上に Red Hat Decision Manager (Decision Central を含む) をインストールします。手順については『[Red Hat Decision Manager のオンプレミスインストール](#)』を参照してください。

手順

1. Red Hat Decision Manager 7.0 Decision Central にログインします。
2. メインメニューで **Projects** を選択します。
3.  アイコンをクリックし、**Import Project** を選択します。
4. **Repository URL** フィールドに、以下のフォーマットで URL を入力します。

```
file://<path>/niogit/<repository>.git
```

たとえば、**.niogit** ディレクトリーの場所が **/opt/eap7.0/bin/.niogit** で、**repository1** リポジトリからプロジェクトをインポートする場合、URL は以下のようになります。

```
file:///opt/eap7.0/bin/.niogit/repository1.git
```

5. **Import** をクリックします。
6. インポートするプロジェクトを選択して **Import** をクリックします。
7. **NoRemoteRepositoryException** の例外が表示される場合があります。その場合は、以下のステップを実施します。
 - a. シェルのターミナルで、インポートするリポジトリのサブディレクトリー (**.niogit** の下位ディレクトリー) に移動します
 - b. リモートのリストを表示します (**git remote -v**)。
 - c. リストにあるすべてのリモートを削除します (**git remote remove <remote-name>**)。

例:

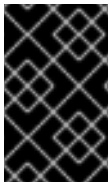
```
$ /opt/eap7.0/bin/.niogit/repository1.git
$ git remote -v
origin https://github.com/guvnorngtestuser1/guvnorng-playground
(fetch)
origin https://github.com/guvnorngtestuser1/guvnorng-playground
(push)
$ git remote remove origin
```

続いて、もう一度プロジェクトをインポートします (必要であれば、新しいリポジトリに別途リモートを追加することができます)。

8. インポートおよびインデックス化が完了するのを待ちます。
9. インポートするリポジトリが他にあれば、手順を繰り返します。

結果

プロジェクトが Decision Central にインポートされます。 **Build & Deploy** ボタンを使用して、通常どおりビルドおよびデプロイできます。



重要

Java でプロジェクトを作成してから Business Central ワークベンチにインポートしている場合は、プロジェクトを Java プロジェクトとして移行しなければならない場合があります。手順については「[3章 Java プロジェクトの移行](#)」を参照してください。

第3章 JAVA プロジェクトの移行

Business Central ワークベンチでは作成されていない Java プロジェクトの場合は、`pom.xml` ファイル内の依存関係を修正し、ビルドに関する問題を解決する必要があります。このプロセスは、Java で書かれたデシジョンサービスとクライアントアプリケーションで共通です。

前提条件/事前作業

Red Hat Decision Manager 7.0 用の Maven リポジトリをダウンロードし、ローカル Maven システムが利用できる状態にする必要があります (ローカルまたはリモートリポジトリとして)。

手順

1. プロジェクトの `pom.xml` ファイルを開きます。
2. 以下のグループに属するすべての依存関係から、`<version>` タグを削除します。
 - `org.kie`
 - `org.drools`
 - `org.jbpm`
 - `org.optaplanner`
3. `<dependencyManagement>` セクション内の `<dependencies>` タグの下に、以下の依存関係を追加します。

```
<dependency>
  <groupId>org.jboss.bom.rhdm</groupId>
  <artifactId>rhdm-platform-bom</artifactId>
  <version>${rhdm.version}</version>
  <scope>import</scope>
  <type>pom</type>
</dependency>
```

ここで、`${rhdm.version}` は、Red Hat Decision Manager 7.0 用 Maven リポジトリの `org.jboss.bom.rhdm:rhdm-platform-bom` アーティファクトのバージョンです。リポジトリに入り、`maven-repository/org/jboss/bom/rhdm/rhdm-platform-bom` ディレクトリに移動して、アーティファクトのバージョンを確認できます。このディレクトリのサブディレクトリの名前だけを、バージョンとして指定します (例: `7.0.0.GA-redhat-1`)。

4. コードに Drools CDI アノテーション (`@KReleaseId`、`@KContainer`、`@KBase`、`@KSession`) が使用されている場合は、以下の依存関係も追加します。

```
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-cdi</artifactId>
</dependency>
```

アノテーションを処理する CDI 拡張子が別のモジュールを使用して定義されるようになったので、この依存関係が必要です。この依存関係がないと、アノテーションは機能しません。

5. (必要に応じて) 古い **org.guvnor** 依存関係を、新しい **org.uberfire** 依存関係に置き換えます。

古い **org.guvnor** 依存関係:

```
<dependency>
  <groupId>org.guvnor</groupId>
  <artifactId>guvnor-rest-client</artifactId>
</dependency>
```

新しい **org.uberfire** 依存関係:

```
<dependency>
  <groupId>org.uberfire</groupId>
  <artifactId>uberfire-rest-client</artifactId>
</dependency>
```

6. プロジェクトアーティファクトのバージョンを更新し、**pom.xml** ファイルを保存します。
7. 通常の Maven ビルドを使用してプロジェクトをビルドし直します。
8. ビルドエラー、またはその後に実行エラーが発生する場合は、[「移行した Java プロジェクトで発生するビルドエラーおよび実行エラーのトラブルシューティング」](#)で説明する情報を元に、それらの問題を解決します。

3.1. 移行した **JAVA** プロジェクトで発生するビルドエラーおよび実行エラーのトラブルシューティング

以下の情報を元に、Red Hat Decision Manager 7.0 用に移行したプロジェクトの再ビルド後に発生するビルドエラーまたは実行エラーを解決します。

3.1.1. Drools モジュールにおける **API** 変更

BRMS 6.4 から Red Hat Decision Manager 7.0 への移行により、以下の Drools モジュールに互換性のない変更が含まれます。

- **kie-api**: KIE グループからの全プロジェクト用のメイン API。変更のほとんどは、以下のクラスに関するものです。
 - **org.kie.api.task.***
 - **org.kie.api.executor.***
 - **org.kie.api.concurrent.***
 - **org.kie.api.builder.***
 - **org.kie.api.command.***
 - **org.kie.api.runtime.***
- **drools-core**: ルールエンジン。以下のクラスが変更されます。
 - **org.drools.core.command.***

- `org.drools.core.common.*`
- **kie-server-api**: KIE Server (Decision Server) 用の標準 API (コマンド、モデル等を含む)。以下のクラスが変更されます。
 - `org.kie.server.api.commands.*`
 - `org.kie.server.api.marshalling.*`
 - `org.kie.server.api.model.*`
 - `org.kie.server.api.rest.RestURI` (定数がわずかに変更され、先頭の / が省略されます)
- **kie-server-controller-api**: KIE Server Controller 用の API。以下のクラスが変更されま
す。
 - `org.kie.server.controller.api.service.*`
- **kie-server-controller-rest**: KIE Server Controller 用の REST API。以下のクラスが変更
されます。
 - `org.kie.server.controller.rest.RestSpecManagementServiceImpl`
 - `org.kie.server.controller.rest.RestKieServerControllerImpl`
- **kie-server-client**: KIE Server Client。以下のクラスが変更されます。
 - `org.kie.server.client.SolverServicesClient`
 - `org.kie.server.client.UIServicesClient`
 - `org.kie.server.client.admin.ProcessAdminServicesClient`
 - `org.kie.server.client.ProcessServicesClient`
 - `org.kie.server.client.QueryServicesClient`
 - `org.kie.server.client.JobServicesClient`
 - `org.kie.server.client.UserTaskServicesClient`
 - `org.kie.server.client.KieServicesClient`
 - `org.kie.server.client.KieServicesConfiguration`

ここで挙げるクラスに関するビルドエラーが発生する場合は、「[List of API changes between BRMS 6.4 and Decision Manager 7.0](#)」に書かれた詳細なレポートを確認してください。レポートには API 変更の詳細が記載されています。この情報を元に、コードを変更に対応させることができます。

3.1.2. Drools モジュールにおけるロジック変更

Drools モジュールに以下のロジック変更が加えられました。

- Red Hat BRMS 6.4 では、ルールが累積パターンの `sum` 関数を実行する場合、返される結果は入力のデータタイプにかかわらず必ず `double` 型でした。Red Hat Decision Manager 7.0 では、

sum 関数を実行する入力のデータタイプが維持されます。この機能拡張により、sum 関数からより正確な結果が得られます。以下の例では、accumulate 関数で得られる結果のデータタイプは **double** 型ではなく **Long** 型になります。

```
Long(...) from accumulate(..., sum($p.getLongWeight()))
```

Red Hat BRMS 6.4 プロジェクトのルールに累積パターンの sum 関数が含まれている場合は、この関数を探して確認してください。Red Hat BRMS 6.4 ではこの関数は double 型のデータタイプを返していましたが、Red Hat Decision Manager 7.0 では入力値のデータタイプを返す点に注意してください。

- 累積パターンと一致するファクトがない場合、Red Hat Decision Manager 7.0 では min および max accumulate 関数は **+/-Integer.MAX_VALUE** ではなく **null** を返します。したがって、BRMS 6.4 の場合とは異なり、ルールの累積は一致せずルールは実行されません。

第4章 RED HAT DECISION MANAGER 7.0 における RED HAT BUSINESS OPTIMIZER の変更

Red Hat Business Optimizer は、Red Hat Decision Manager における埋め込みプランニングエンジンで、計画問題を最適化します。Red Hat Business Optimizer は、定期的に更新されるコミュニティの OptaPlanner プロジェクトをベースにしており、最新の Red Hat Business Optimizer 機能に合わせてコード変更が必要になる場合があります。最新の OptaPlanner 変更と移行要件の概要は「[OptaPlanner upgrade recipe archive](#)」を参照してください。OptaPlanner のバージョン 7.0 から 7.6 へのアップグレード情報は、Red Hat JBoss BRMS 6.4 から Red Hat Decision Manager 7.0 へのアップグレードに適しています。

Red Hat Decision Manager 7.0 では、Decision Central で Red Hat Business Optimizer 設定をいくつか更新して、最新の OptaPlanner 変更を適用する必要があります。

4.1. DECISION CENTRAL での RED HAT BUSINESS OPTIMIZER アセットの設定の更新

Decision Central にソルバー設定アセット (`.solver.xml` ファイル) またはソリューション関連のデータオブジェクトが存在する場合は、Red Hat Decision Manager 7.0 でこのアセットにいくつかの更新を行い、最新の Red Hat Business Optimizer 変更を適用する必要があります。

前提条件/事前作業

Decision Central データが、Red Hat JBoss BRMS から Red Hat Decision Manager 7.0 へ移行している。移行手順は「[2章 Decision Central でのプロジェクト移行作業](#)」を参照してください。

手順

1. Red Hat Decision Manager 7.0 の Decision Central にログインします。
2. **Menu** → **Design** → **Projects** に移動し、プロジェクト名を選択します。
3. サーバー設定アセット (`.solver.xml` ファイル) が存在する場合は開きます。
4. ソルバー設定デザイナーで、変更せずに **Save** をクリックします。この手順は、Red Hat Decision Manager 7.0 でソルバー設定アセットに新しいコードを生成するのに必要です。その他のソルバー設定アセットでもこの手順を行います。
5. 必要に応じて、**Project Explorer** (左側のメニュー) の **Data Objects** の下で、**Planning Solution** として設定したデータオブジェクト (`.java` ファイル) を開きます。
このデータオブジェクトにこの設定が選択されていることを確認する場合は、データオブジェクトデザイナーの右側にある OptaPlanner アイコンをクリックします。**Planning Solution** が選択されていないと、この手順は適用されません。
6. データオブジェクトデザイナーの **general properties** で、**Superclass** ドロップダウンオプションを **Nothing selected** に設定します。この設定は、Red Hat Decision Manager 7.0 の Red Hat Business Optimizer では必要なくなりました。
7. データオブジェクトデザイナーの右側で、OptaPlanner アイコンをクリックして、**Planner Settings** を表示します。**Planning Solution** オプションが選択されているはずですが (選択されていないと、この手順は適用されません)。
8. **No selected** を選択し、**Planning Solution** を再度選択して、**Solution Score Type** を指定します。この手順は、Red Hat Decision Manager 7.0 の計画問題に新しいコードを生成するために必要です。

9. データオブジェクトデザイナーで **Save** をクリックし、**Planning Solution** として設定したその他のデータオブジェクトに同じ変更を加えます。

第5章 DECISION SERVER のアップグレード

BRMS 6.4 KIE Server と同じホスト上に新たなサーバーをインストールして、Red Hat Decision Manager 7.0 Decision Server にアップグレードすることができます。既存の KJAR ファイルを使用することは可能です。ただし、最大のパフォーマンスを得るためには、プロジェクトをアップグレードしてビルドし直してください。

手順

1. BRMS 6.4 KIE Server と同じホスト上に Red Hat Decision Manager 7.0 Decision Server をインストールします。その際には、別の JBoss EAP システムを使用します。手順については『[Red Hat Decision Manager のオンプレミスインストール](#)』を参照してください。
2. BRMS 6.4 KIE Server と同じポートにバインドするように、Red Hat Decision Manager 7.0 Decision Server を設定します (デフォルトの設定は同じです)。
3. 既存のデシジョンサービス (KJAR ファイル) を含む Maven リポジトリを使用するように、Decision Server を設定します。以下の手法のいずれかを使用します。
 - BRMS 6.4 KIE Server で使用していたものと同じ KJAR ファイルを使用します。Maven 設定をコピーするには、`kie.maven.settings.custom` システムプロパティの値を、古いバージョンの `$EAP_home/standalone/configuration/standalone.xml` ファイルから新しいバージョンの同じファイルにコピーします。
 - 既存のデシジョンサービスを新しいバージョンにアップグレードします。手順については「[2章 Decision Central でのプロジェクト移行作業](#)」および「[3章 Java プロジェクトの移行](#)」を参照してください。新たな KJAR ファイルを Maven リポジトリにアップロードします。続いて、そのリポジトリを使用するように Decision Server を設定します。
 - a. Maven リポジトリの `settings.xml` ファイルを作成します。
 - b. Red Hat Decision Manager 7.0 Decision Server の `$EAP_home/standalone/configuration/standalone.xml` ファイルにおいて、`<system-properties>` タグの `kie.maven.settings.custom property` を `settings.xml` ファイルのフルパス名に設定します。以下に例を示します。

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

この場合、KJAR ファイルのバージョンを変更する必要があります。開始するサービスのバージョンに正しく変更します。既存クライアントとの互換性を最大限維持するために、引き続き同じコンテナ名を使用することができます。
4. BRMS 6.4 KIE Server を停止します。
5. Red Hat Decision Manager 7.0 Decision Server を起動します。
6. API を使用して必要なデシジョンサービスを開始します。手順については『[デシジョンサービスのパッケージングおよびデプロイメント](#)』を参照してください。

結果

BRMS 6.4 KIE Server と同じ URL 上で Red Hat Decision Manager 7.0 Decision Server が動作し、同じサービスを提供します。

以下に示す API 変更に対応するため、REST API を使用するクライアントアプリケーションを変更しなければならない場合があります。

- ServiceResponse ラッパーが Planner サービスレスポンスから削除されました。



重要

Decision Central に接続するように Decision Server を設定することもできます。Decision Central はさまざまな Decision Server を管理し、それらのサーバー上に同一または異なるデシジョンサービスをデプロイすることができます。手順については『[デシジョンサービスのパッケージングおよびデプロイメント](#)』を参照してください。

付録A バージョン情報

本書の最終更新日: 2018 年 8 月 31 日