



Red Hat JBoss BRMS 6.4

スタートガイド

Red Hat JBoss BRMS のスタートガイド

Red Hat JBoss BRMS 6.4 スタートガイド

Red Hat JBoss BRMS のスタートガイド

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

Emily Murphy

Gemma Sheldon

Michele Haglund

Mikhail Ramendik

Stetson Robinson

Vidya Iyengar

法律上の通知

Copyright © 2020 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, 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[®] 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 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

本ガイドには、Red Hat JBoss BRMS の使用を開始するための情報が記載されています。

目次

第1章 RED HAT JBOSS BRMS のクイックスタート	3
第2章 インストール	5
2.1. RED HAT JBOSS EAP 用 RED HAT JBOSS BRMS のダウンロード	5
2.2. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM への RED HAT JBOSS BRMS のインストール	5
2.2.1. Red Hat JBoss EAP 新規インスタンスへのインストール	5
2.2.2. 既存の Red Hat JBoss EAP 上でのインストール	6
2.3. ロールの定義	7
2.4. ユーザーの作成	7
2.5. サーバーの起動	8
2.6. JAVA SECURITY MANAGER とパフォーマンス管理	9
第3章 BUSINESS CENTRAL へのログイン	11
第4章 HELLO WORLD ルールの例	12
4.1. BUSINESS CENTRAL を使用した最初のルールの作成	12
4.2. REALTIME DECISION SERVER の設定	14
4.3. KIE サーバー JAVA クライアント API を使用したルールの実行	15
第5章 RED HAT JBOSS DEVELOPER STUDIO	20
5.1. RED HAT JBOSS DEVELOPER STUDIO プラグインのインストール	20
5.2. DROOLS ランタイムの設定	20
5.3. RED HAT JBOSS BRMS サーバーの設定	21
5.4. KIE NAVIGATOR	21
Kie Navigator の機能	23
Kie Navigator サーバーレベル	23
Kie Navigator 組織単位レベル	23
Kie Navigator リポジトリレベル	24
Kie Navigator プロジェクトレベル	24
5.5. GIT リポジトリから RED HAT JBOSS DEVELOPER STUDIO へのプロジェクトのインポート	24
5.6. DROOLS プロジェクトの作成	27
第6章 BUSINESS RESOURCE PLANNER	28
6.1. BUSINESS RESOURCE PLANNER のインストール	28
6.2. BUSINESS RESOURCE PLANNER サンプルの実行	28
付録A バージョン情報	29

第1章 RED HAT JBOSS BRMS のクイックスタート

本セクションでは、Red Hat JBoss BRMS をダウンロード、インストールおよび実行する方法について説明します。インストールの詳細な手順や他のインストール方法については、『Red Hat JBoss BRMS インストールガイド』を参照してください。

要件

- システムに Java のサポート対象バージョンがインストールされていること。Java のサポート対象バージョンについては、Red Hat カスタマーポータル[の「Red Hat JBoss BPM Suite 6 Supported Configurations」](#)を参照してください。
- Red Hat JBoss EAP サーバーインストールのバージョン 6.4 以降があること。インストーラーファイルを使用している場合は、Red Hat JBoss EAP 6 のバージョン 6.4.7 以降がインストールされているはずですが。



注記

Red Hat JBoss EAP サーバーを更新する必要があることがあります。インストーラーで更新の必要性が通知された場合は、Red Hat [カスタマーポータル](#) から最新のパッチをダウンロードし、『Red Hat JBoss EAP インストールガイド』に記述された更新手順に従ってください。

手順: Red Hat JBoss BRMS のクイックスタート

1. Red Hat カスタマーポータル[の「Software Downloads」](#)セクションからインストーラーをダウンロードします。
2. 次のコマンドを実行して、インストーラーを起動します。

```
java -jar jboss-brms-VERSION-installer.jar
```

3. インストールプロセスを完了させます。インストール後に Red Hat JBoss BRMS アプリケーションにアクセスするためのユーザー名とパスワードを書き留めます。
4. インストールが終了したら、Red Hat JBoss BRMS のインストール先である `EAP_HOME` ディレクトリーにコマンドプロンプトで移動します。
5. サーバーを起動します。

```
bin/standalone.sh
```

6. Web ブラウザーを開いて、<http://localhost:8080/business-central/> に移動し、インストール時に作成した Red Hat JBoss BRMS アプリケーションにアクセスする際のユーザー名とパスワードでログインします。

これで Red Hat JBoss BRMS が正常にインストールされ、稼働します。

次のステップ

- ロールに関する知識を深める (『[ロールの定義](#)』を参照)
- 追加ユーザーを作成する (『[ユーザーの作成](#)』を参照)

- Hello World ルールを作成する (「[4章 Hello World ルールの例](#)」を参照)

Red Hat JBoss EAP と他のコンテナでのインストールについての詳細情報は、『[Red Hat JBoss BRMS インストールガイド](#)』を参照してください。

第2章 インストール

本セクションでは、インストール済みの Red Hat JBoss EAP 6.4 (あるいはそれ以降) のインスタンスに Red Hat JBoss BRMS をインストールする方法について説明します。別のコンテナをお使いの場合は、『Red Hat JBoss BRMS インストールガイド』を参照してください。

「1章 Red Hat JBoss BRMS のクイックスタート」の指示に従っていただければ、JAR インストーラーを使用して Red Hat JBoss BRMS が既にインストールされています。その場合は、「次のステップ」の指示に従ってください。



注記

Red Hat JBoss BRMS は、UTF-8 エンコーディングで機能するように設計されています。JVM で別のエンコーディングシステムを使用すると、予期せぬエラーが発生する可能性があります。JVM で **UTF-8** を使用するには、「**-Dfile.encoding=UTF-8**」のシステムプロパティを使用してください。

2.1. RED HAT JBOSS EAP 用 RED HAT JBOSS BRMS のダウンロード

1. ブラウザーで Red Hat カスタマーポータル「[ダウンロード](#)」セクションを開きます。
2. 表示された「[製品のダウンロード](#)」ページで Red Hat JBoss BRMS をクリックして、ログインします。
3. [バージョン](#) のドロップダウンメニューからバージョン **6.4** を選択します。
4. **Red Hat JBoss BRMS 6.4.0 Deployable for EAP 6** を選択して、**Download** をクリックします。Red Hat JBoss EAP 7.0 を使用している場合は、**Red Hat JBoss BRMS 6.4.0 Deployable for EAP 7** を選択します。

2.2. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM への RED HAT JBOSS BRMS のインストール

2.2.1. Red Hat JBoss EAP 新規インスタンスへのインストール

Red Hat JBoss Enterprise Application Platform にデプロイ可能なパッケージをインストールするには、以下の手順に従います。

1. 「[Red Hat JBoss EAP 用 Red Hat JBoss BRMS のダウンロード](#)」でダウンロードした ZIP アーカイブを Red Hat JBoss Enterprise Application Platform ホームディレクトリー (**EAP_HOME**、**jboss-eap-6.4** ディレクトリー) の **親** ディレクトリーに移動します。
2. ダウンロードした ZIP アーカイブを展開し、**EAP_HOME** ディレクトリー (**jboss-eap-6.4**) にマージします。



警告

この手順は、Red Hat JBoss EAP のインストールに使用したユーザーアカウントと同じアカウントで実行します。このアカウントは、スーパーユーザーアカウント **以外** である必要があります。

3. プロンプトが出たら、**EAP_HOME** ディレクトリーにあるファイルをダウンロードした ZIP アーカイブで上書きします。

2.2.2. 既存の Red Hat JBoss EAP 上でのインストール



警告

以下の手順は Red Hat JBoss BRMS をインストールするためのものであって、既存の JBoss BRMS インスタンスを更新するものではありません。ターゲットとなる EAP に既存の JBoss BRMS インストールがないことを確認してください。

設定済みの Red Hat JBoss EAP にデプロイ可能なパッケージをインストールするには、以下の手順に従います。

1. 「[Red Hat JBoss EAP 用 Red Hat JBoss BRMS のダウンロード](#)」でダウンロードした Red Hat JBoss EAP 用のデプロイ可能な ZIP パッケージを展開します。
2. ダウンロードした zip アーカイブを解凍しますが、ファイルを上書きしないでください。手作業で以下のファイルを **EAP_HOME** ディレクトリー (**jboss-eap-6.4**) にマージします。
 - **jboss-eap-6.4/domain/configuration/*** (Red Hat JBoss BRMS には JMS が必要なため、Red Hat JBoss BRMS ディストリビューションで提供される **domain.xml** にあるすべてのプロファイルに JMS がデフォルトで追加されることに留意してください)
 - **jboss-eap-6.4/standalone/configuration/*** (Red Hat JBoss BRMS には JMS が必要なため、Red Hat JBoss BRMS ディストリビューションで提供されるすべてのプロファイルの設定ファイル (**standalone.xml** および **standalone-ha.xml** を含む) に JMS がデフォルトで追加されることに注意してください)



警告

この作業は、必ず EAP をインストールしたユーザーアカウントで実行するようにしてください。スーパーユーザーアカウントは使用しないでください。

- Red Hat JBoss BRMS ディストリビューションから **jboss-eap-6.4/standalone/deployments** を **EAP_HOME** ディレクトリーにコピーします。



注記

Red Hat JBoss EAP にすでにデプロイメントがある場合は、現在のデプロイメントが Red Hat JBoss BRMS のデプロイメントと同じ名前にならないようにしてください。

2.3. ロールの定義

サーバーを起動し Business Central へログインする前に、ユーザーアカウントを作成する必要があります。本セクションでは、Red Hat JBoss BRMS で使用されるさまざまなユーザーロールについて説明します。

- **admin:** admin ロールを持つユーザーは、アプリケーションの管理者です。管理者は、ユーザーやリポジトリ (作成およびクローン) を管理し、アプリケーションに必要な変更を加えるための完全アクセスがあります。管理者はシステム内のすべてのエリアにアクセスできます。
- **analyst:** analyst ロールは、プロジェクトをモデル化するため、すべての機能へアクセスできますが、analyst ロールを持つユーザーは **Authoring** → **Administration** へアクセスできません。また、**Deployment** → **Artifact Repository** ビューなど、開発者を対象とする低レベル機能の一部にもアクセスできませんが、Project Editor を使用すると、**Build & Deploy** ボタンを使用できます。

ロールに関する詳しい情報は、『Red Hat JBoss BPM Suite Administration and Configuration Guide』の「[Access Control](#)」セクションを参照してください。ロールとタスクの対話に関する詳しい情報は、『Red Hat JBoss BPM Suite User Guide』の「[User Task](#)」セクションを参照してください。

遭遇する可能性のあるその他のロールを以下に示します。

- REST API のアクセス制御ロール。詳しい情報は、『Red Hat JBoss BPM Suite Development Guide』の「[Control of REST API](#)」セクションを参照してください。
- Intelligent Process Server **kie-server** ロール。詳しい情報は、『Red Hat JBoss BPM Suite Development Guide』の「[Intelligent Process Server and Realtime Decision Server](#)」のパートを参照してください。
- 管理リポジトリ機能の **kiemgmt** ロール。『Red Hat JBoss BPM Suite Administration and Configuration Guide』の「[Managing Assets](#)」セクションを参照してください。



注記

上記のロールは、ユーザー作成のプロセス中に入力します。

2.4. ユーザーの作成

新規ユーザーの追加を開始するには、Unix システムでは **add-user.sh** スクリプトを、Windows システムでは **add-user.bat** ファイルを EAP **bin** ディレクトリーから実行する必要があります。

手順: 新規ユーザーの作成

1. **EAP_HOME** ディレクトリーに移動します。
2. Unix システムでは、以下のコマンドを実行します。

```
bin/add-user.sh
```

Windows システムでは以下を実行します。

```
bin\add-user.bat
```

3. **b** を入力して、アプリケーションユーザーを選択し、**Enter** を押します。
4. **Enter** を押して、デフォルトのレルム (**ApplicationRealm**) を確定します。
5. ユーザー名入力のプロンプトが表示されたら、ユーザー名を入力し (例: **helloworlduser**)、確認します。



重要

すでにあるロールやグループと選択したユーザー名が **競合しない** ようにしてください。

たとえば、**admin** と呼ばれるロールがある場合は、**admin** という名前のユーザーは作成 **しないで** ください。

6. パスワード入力のプロンプトが表示されたらユーザーのパスワードを入力し (例: **HelloWorld@123**)、再入力します。



注記

パスワードは 8 文字以上で、アルファベットの大文字と小文字 (A-Z、a-z)、1 文字以上の数字 (0-9)、および 1 文字以上の特殊文字 (~!@#\$%^*()-_+=) が含まれる必要があります。

7. ロール入力のプロンプトが表示されたら、ユーザーが必要とするロールをコンマ区切りのリストで入力します。詳細は、「[ロールの定義](#)」を参照してください。
Business Central のユーザーには、**analyst** または **admin** ロールが割り当てられている必要があります。
8. ユーザーを追加することを確認します。
9. 次のプロンプトで **yes** を入力してクラスタリングを有効にします。

2.5. サーバーの起動

サーバーは、2 つのモードのいずれかで起動できます。

- **standalone.sh** スクリプトを使用した standalone モード。
- **standalone-secure.sh** スクリプトを使用した standalone-secure モード。



注記

サーバーをドメインモードで起動する場合は、対応するスクリプトはそれぞれ **domain.sh** と **domain-secure.sh** になります。

デフォルトの起動スクリプトは **standalone.sh** になります。これは、パフォーマンス重視で最適化されています。パフォーマンスモードでサーバーを実行する場合は、以下の手順に従います。

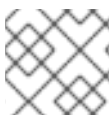
1. コマンドラインで **EAP_HOME** ディレクトリーに移動します。
2. Unix 環境 の場合は以下を実行します。

```
bin/standalone.sh
```

Windows 環境の場合は以下を実行します。

```
bin\standalone.bat
```

standalone-secure.sh スクリプトは、セキュリティー重視で最適化されています。このスクリプトでは、既知の脆弱性を保護するセキュリティーポリシーが適用されます。



注記

実稼働環境では **standalone-secure.sh** スクリプトの使用が推奨されます。



警告

セキュリティーマネージャーを使用すると、パフォーマンスが著しく低下します。どちらのスクリプトを使用するかは、個々の状況を判断して決定してください。「[Java Security Manager とパフォーマンス管理](#)」を参照してください。

セキュアモードでサーバーを起動するには、以下の手順に従います。

1. コマンドラインで **EAP_HOME** ディレクトリーに移動します。
2. Unix 環境 の場合は以下を実行します。

```
bin/standalone-secure.sh
```

Windows 環境の場合は以下を実行します。

```
bin\standalone-secure.bat
```



注記

JAR インストーラーを使用して Red Hat JBoss BRMS をインストールした場合は、インストール中にセキュリティーポリシーの適用を選択できます。インストーラーでは別個の **standalone-secure.sh** スクリプトは提供されません。

2.6. JAVA SECURITY MANAGER とパフォーマンス管理

Red Hat JBoss BRMS で MVEL スクリプトの評価をサンドボックス化できるように Java Security Manager (JSM) を有効にすると、高負荷環境でのパフォーマンスが低下します。Red Hat JBoss BRMS

アプリケーションをデプロイする場合は、ご自身の環境でのパフォーマンスニーズを考慮してください。以下のガイドラインを使用して、セキュアで高パフォーマンスな Red Hat JBoss BRMS アプリケーションをデプロイしてください。

高負荷環境でパフォーマンス重視の場合は、

- 他のシステムで開発され、適切なレビューを経ているアプリケーションをデプロイします。
- システム上で **analyst** ロールのあるユーザーを作成しないでください。

これらの対策が取られた場合には、パフォーマンスが低下しないので JSM を無効にした状態のままでも安全です。

高負荷にならないテストおよび開発環境や、ルールやプロセスのオーサリングが外部ネットワークに公開される環境では、

- MVEL の評価を適切にサンドボックス化してセキュリティを強化するため、JSM を有効にします。
- JSM が無効な状態で、**analyst** ロールを持つユーザーに Business Central コンソールへのログインを許可しないでください。これは安全で **ない** ため、**推奨されません**。

第3章 BUSINESS CENTRAL へのログイン

Business Central にログインするには、以下を実行します。

1. Red Hat JBoss BRMS でサーバーを起動します。
2. Web ブラウザーで <http://localhost:8080/business-central> にアクセスします。ドメイン名から実行するようユーザーインターフェースが設定されている場合は、<http://www.example.com:8080/business-central> のように **localhost** をドメイン名に置き換えます。
3. インストール中に作成されたユーザー認証情報を使用してログインします (例: ユーザー名 **helloworlduser** とパスワード **HelloWorld@123**)。

第4章 HELLO WORLD ルールの例

本章では、単純なビジネスルールで構成される Hello World プロジェクトをセットアップすることにより、Red Hat JBoss BRMS の基本的な機能を説明します。

本章では、以下の作業を行います。

- リポジトリの作成
- プロジェクトの作成
- ルールの作成
- ナレッジベースの作成
- プロジェクトのビルドおよびデプロイ

4.1. BUSINESS CENTRAL を使用した最初のルールの作成

前提条件

このルールの例を実行するには Red Hat JBoss BRMS と Maven が正常にインストールされている必要があります。Red Hat JBoss BRMS を起動し、Business Central にログインしなければなりません。

手順: Business Central を使用した最初のルールの作成および実行

1. Business Central のメインメニューで **Authoring** → **Administration** をクリックし、続いて **Organizational Unit** → **Manage Organizational Units** をクリックします。
2. **追加** をクリックします。
3. **Add New Organizational Unit** ダイアログボックスが表示されたら、以下のようにユニットプロパティを定義します。
 - **Name: Hello World**
 - **Owner: finance-dept@company.org****OK** をクリックします。
4. パースペクティブメニューで **Repositories** → **New repository** をクリックします。
5. **New Repository** ダイアログボックスが表示されたら、以下のようにリポジトリプロパティを定義します。
 - **Repository Name: EmployeeRepo**
 - **Organizational Unit** ドロップダウンメニューで **Hello World** を選択します。**Finish** をクリックします。
6. **Authoring** → **Project Authoring** に移動します。
7. 組織単位ドロップダウンボックスの Project Explorer で **Hello World** を選択し、リポジトリドロップダウンボックスで **EmployeeRepo** を選択します。
8. パースペクティブメニューで **New Item** → **Project** に移動します。

9. **New Project** ウィザードが表示されたら、プロジェクトの名前を指定します (例: **MyProject**)。ウィザードの **Group Artifact Version** 部分で、以下のようにプロジェクトの Maven プロパティを定義します。

- **Group ID: org.brms**
- **Artifact ID: MyProject**
- **Version ID: 1.0.0**

Finish をクリックします。

10. パースペクティブメニューで **New Item** → **Data Object** に移動します。

11. **Create new Data Object** ダイアログボックスが表示されたら、以下のようにオブジェクト名とパッケージの値を指定します。

- **Data Object: Person**
- **Package: org.brms.myproject**

OK をクリックします。

12. 新しく作成された **Person** データオブジェクトの **Person** ウィンドウが表示されたら、**add field** をクリックして **New field** ダイアログを開きます。Id フィールドで変数名を追加し、Type フィールドで変数のデータ型を選択して、必要なすべての変数を定義するまで **Create and continue** をクリックします。以下に例を示します。

- **Id: firstName**
Type: **String**
- **Id: lastName**
Type: **String**
- **Id: hourlyRate**
Type: **Integer**
- **Id: wage**
Type: **Integer**

最後の変数になったら **Create** をクリックし、続いて **Save** をクリックします。

13. パースペクティブメニューで **New Item** → **DRL File** をクリックします。

14. **Create new** ダイアログボックスで、以下のようにルールファイルの名前とパッケージ名を指定します。

- **DRL file name MyRule**
- **Package: org.brms.myproject**

OK をクリックします。

15. **MyRule.drl** の DRL エディターが表示されたら、ルールを作成します。

```
package org.brms.myproject;  
rule "MyRule"  
when
```

```

Person(hourlyRate*wage > 100)
Person(name : firstName, surname : lastName)

then
  System.out.println( "Hello" + " " + name + " " + surname + "!" );
  System.out.println( "You are rich!" );

end

```

保存 をクリックします。

ルールに関する詳しい説明は、『Red Hat JBoss BPM Suite Development Guide』の「[All About Rules](#)」の部分を参照してください。

16. 以下の手順は独自のナレッジベースを作成する方法を示しています。ナレッジセッションに関する詳細については、『Red Hat JBoss BPM Suite Development Guide』の「[KIE Sessions](#)」の章を参照してください。
 - a. **Open Project Editor** をクリックします。
 - b. **Project Settings: Project General Settings** → **Knowledge Base Settings: Knowledge bases and sessions** をクリックします。
 - c. **Add** をクリックし、ナレッジベースの名前を入力します。
この例では **myBase** と入力します。
 - d. **Make Default** をクリックします。
 - e. **Knowledge Sessions** で **Add** をクリックします。
 - f. セッションの名前を入力します (例: **mySession**)。 **Default** にチェックを入れ、 **stateless** 状態を選択します。
 - g. 右上隅の **Save** をクリックします。
17. **Open Project Editor** をクリックし、続いて **Build & Deploy** をクリックします。プロジェクトのビルドおよび Execution Server へのデプロイメントが正常に完了したことを伝える緑色の通知が画面上部に表示されます。



注記

ビルドが失敗したことを示す赤い通知が表示された場合は、Red Hat JBoss BRMS コンソールにビルドの失敗に関する情報が表示されます。これはアプリケーションをトラブルシューティングする場合に重要な情報になります。プロジェクトをビルドする前に、すべてのアセットが保存されていることを確認します。

4.2. REALTIME DECISION SERVER の設定

Realtime Decision Server インスタンスを登録し、REST API からルールを実行します。本セクションでは、以下の点について説明します。

- Realtime Decision Server の登録方法
- Realtime Decision Server への新しいコンテナの追加方法

手順: Realtime Decision Server の登録

1. **kie-server** ロールの新規ユーザーを作成します。
コマンドラインで **EAP_HOME/bin/** に移動し、以下のコマンドを実行します。

```
~]$ ./add-user.sh -a --user kieserver --password kieserver1! --role kie-server,rest-all
```

2. **EAP_HOME/standalone/configuration/standalone.xml** を開き、**system-properties** タグに以下のプロパティを追加します。

```
<property name="org.kie.server.user"
  value="kieserver"></property>
<property name="org.kie.server.pwd"
  value="kieserver1!"></property>
<property name="org.kie.server.location"
  value="http://localhost:8080/kie-server/services/rest/server"></property>
<property name="org.kie.server.controller"
  value="http://localhost:8080/business-central/rest/controller"></property>
<property name="org.kie.server.controller.user"
  value="kieserver"></property>
<property name="org.kie.server.controller.pwd"
  value="kieserver1!"></property>
<property name="org.kie.server.id"
  value="local-server-123"></property>
```

3. **EAP_HOME/bin/** に移動し、以下のコマンドを実行します。

```
~]$ ./standalone.sh
```

4. Business Central において、**Deploy** → **Execution Servers** をクリックします。
ウィンドウには **local-server-123** と表示されるはずですが、**KIE CONTAINERS** で **Add Container** をクリックします。
5. **New Container** ダイアログが開くので、以下の情報を入力します。
 - Name: **myContainer**
 - **Search** をクリックし、**MyProject-1.0.0.jar** の横にある **Select** をクリックします。
 - **Finish** をクリックしてコンテナを作成します。
6. **myContainer** を選択し、**Start** をクリックしてコンテナを起動します。

4.3. KIE サーバー JAVA クライアント API を使用したルールの実行

ルールを実行するには、Realtime Decision Server に要求を送信する必要があります。本セクションでは、KIE サーバー Java クライアント API を使用して REST API で要求を送信する方法について説明します。サーバーに要求を送信するには任意のクライアントを使用できます。



注記

Maven インストールが Red Hat JBoss BRMS オンラインリポジトリを使用するよう設定されていることを確認します。Maven の設定の詳細については、『[Red Hat JBoss BPM Suite インストールガイド](#)』の「[オンラインリポジトリを使用するための Maven を設定](#)」セクションを参照してください。

1. 任意のディレクトリーに移動し、以下のコマンドを実行します。

```
~]$ mvn archetype:generate -DgroupId=com.MyProject.app -DartifactId=my-app -
DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

これにより、**my-app** という名前のディレクトリーが以下の構造で作成されます。

```
my-app
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |   |   |-- com
    |   |   |   |-- MyProject
    |   |   |   |   |-- app
    |   |   |   |   |   |-- App.java
    |   |-- test
    |   |   |-- java
    |   |   |   |-- com
    |   |   |   |   |-- MyProject
    |   |   |   |   |   |-- app
    |   |   |   |   |   |   |-- AppTest.java
```

2. Maven プロジェクトが使用するライブラリーを宣言します。**my-app/pom.xml** を編集して Red Hat JBoss BRMS 依存関係を設定します。Bill of Material (BOM) および依存関係管理の詳細については、『[Red Hat JBoss BPM Suite Development Guide](#)』の「[Dependency Management](#)」セクションを参照してください。

```
<!-- Declaration of BOM -->
<dependencyManagement>
<dependencies>
<dependency>
<groupId>org.jboss.bom.brms</groupId>
<artifactId>jboss-brms-bpmsuite-platform-bom</artifactId>
<version>6.4.0.GA-redhat-2</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
<dependencies>
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<scope>test</scope>
</dependency>
<!-- Realtime Decision Server dependencies -->
<dependency>
```

```

    <groupId>org.kie.server</groupId>
    <artifactId>kie-server-client</artifactId>
  </dependency>
<dependency>
  <groupId>org.kie.server</groupId>
  <artifactId>kie-server-api</artifactId>
</dependency>
<!-- Red Hat JBoss BRMS dependency. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-core</artifactId>
</dependency>

<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-simple</artifactId>
</dependency>
<!-- Your project -->
<dependency>
  <groupId>org.brms</groupId>
  <artifactId>MyProject</artifactId>
  <version>1.0.0</version>
</dependency>
</dependencies>

```

注記

最後の依存関係は Red Hat JBoss BRMS でビルドしたプロジェクトです。Maven アプリケーションであるクライアントコードのモデルクラス (この例では **org.brms.myproject.Person**) はサーバー側 (Red Hat JBoss BRMS) に一致する必要があります。一致しないと、要求の送信時にルールが一致しくなくなります。これを実現する最も簡単な方法は、両側でモデルクラスを共有することです。これは、モデルクラスを Maven 依存関係として追加することにより実現できません。

したがって、Business Central で新しいプロジェクトを作成する場合に、以下の属性が入力した属性に一致するようにします。

- **groupId**
- **artifactId**
- **version**

3. **my-app/src/test/java/com/MyProject/app/AppTest.java** Java クラスを探します。このクラスは、Maven によりデフォルトで作成されます。
4. **AppTest.java** を編集します。以下のインポートを追加します。

```

import org.kie.api.command.BatchExecutionCommand;
import org.kie.server.api.model.ServiceResponse;
import org.kie.server.client.KieServicesClient;
import org.kie.server.client.KieServicesConfiguration;
import org.kie.server.client.KieServicesFactory;
import org.kie.server.client.RuleServicesClient;

```

```
import org.kie.api.KieServices;
import org.kie.api.command.Command;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import org.brms.myproject.Person;
import org.drools.core.command.impl.GenericCommand;
```

testApp() メソッドを探し、ナレッジベースをロードし、以下のコードを追加してルールを実行します。

```
Person p1 = new Person();
p1.setFirstName("Anton");
p1.setLastName("RedHat");
p1.setHourlyRate(11);
p1.setWage(20);

String url = "http://localhost:8080/kie-server/services/rest/server";
String username = "kieserver";
String password = "kieserver1!";
String container = "myContainer";
String session = "mySession";

KieServicesConfiguration config = KieServicesFactory
    .newRestConfiguration(url, username, password);
Set<Class<?>> allClasses = new HashSet<Class<?>>();
allClasses.add(Person.class);
config.addExtraClasses(allClasses);

KieServicesClient client = KieServicesFactory.newKieServicesClient(config);
RuleServicesClient ruleClient = client.getServiceClient(RuleServicesClient.class);
List<GenericCommand<?>> commands = new ArrayList<GenericCommand<?>>();

commands.add((GenericCommand<?>) KieServices.Factory
    .get().getCommands().newInsert(p1,"Person Insert ID"));
commands.add((GenericCommand<?>) KieServices.Factory
    .get().getCommands().newFireAllRules("fire-identifier"));

BatchExecutionCommand batchCommand = KieServices.Factory
    .get().getCommands().newBatchExecution(commands,session);
ServiceResponse<String> response = ruleClient.executeCommands(container,
    batchCommand);
System.out.println(response.getResult());
```

デフォルトのマージャーが JAXB であることに注意してください (別のものに設定されていない限り)。つまり、使用する予定のカスタムクラスでコンテキストを設定する必要があります。この例では、カスタムクラスは **Person.class** になります。この例では **stateless** セッションが使用されるため、単一のコマンドが実行されると kie セッションは破棄されます。このため、**BatchCommand** オブジェクトに複数のコマンドをラップする必要があります。

マージャータイプを変更するには、以下のいずれかのコマンドを入力します。

```
config.setMarshallingFormat(MarshallingFormat.JSON);
```

```
config.setMarshallingFormat(MarshallingFormat.XSTREAM);
```

次に、以下のインポートを追加します。

```
import org.kie.server.api.marshalling.MarshallingFormat;
```

5. **my-app** に移動し、コマンドラインから以下のコマンドを実行します。

```
~]$ mvn clean install
```

Maven 出力と Red Hat JBoss BRMS 出力があることに注意してください。Red Hat JBoss BRMS コンソールで期待される出力は以下のとおりです。

```
16:26:56,119 INFO [stdout] (http-/127.0.0.1:8080-5) Hello Anton RedHat!  
16:26:56,119 INFO [stdout] (http-/127.0.0.1:8080-5) You are rich!
```

第5章 RED HAT JBOSS DEVELOPER STUDIO

Red Hat JBoss Developer Studio は Eclipse をベースにした JBoss 統合開発環境 (IDE) です。最新の Red Hat JBoss Developer Studio は Red Hat カスタマーポータル「[ダウンロード](#)」セクションから入手してください。Red Hat JBoss Developer Studio は Red Hat JBoss BRMS および Red Hat JBoss BPM Suite 用のツールおよびインターフェースを持つプラグインを提供します。これらのプラグインはコミュニティーバージョンの製品が基になっています。Red Hat JBoss BRMS プラグインは Drools プラグインと呼ばれ、BPM Suite プラグインは jBPM プラグインと呼ばれます。

Red Hat JBoss Developer Studio のインストールまたは設定の手順は、**Red Hat JBoss Developer Studio** のドキュメントを参照してください。



警告

複数バイトの名前を処理する方法における問題のため、Red Hat JBoss Developer Studio のインスタンスは、エンコーディングが UTF-8 に設定されたファイルで開始する必要があります。`$JBDS_HOME/studio/jbdevstudio.ini` ファイルに "**Dfile.encoding=UTF-8**" というプロパティを追加することでこれが可能になります。

5.1. RED HAT JBOSS DEVELOPER STUDIO プラグインのインストール

Red Hat JBoss Developer Studio の Drools プラグインは更新サイトより入手できます。

手順: Drools Red Hat JBoss Developer Studio プラグインのインストール

1. Red Hat JBoss Developer Studio を起動します。
2. **Help** → **Install New Software** をクリックします。
3. **Add** をクリックして **Add Repository** メニューに入ります。
4. **Name** フィールドの横にソフトウェアサイトの名前を指定し、**Location** フィールドに <https://devstudio.jboss.com/10.0/stable/updates/integration-stack/> の URL を追加します。
5. **OK** をクリックします。
6. 利用可能なオプションから **JBoss Business Process and Rule Development** 機能を選択して、**Next** をクリックし、もう一度 **Next** をクリックします。
7. ライセンス内容を確認して、該当のラジオボタンを押してライセンス内容に同意し、**Finish** をクリックします。
8. インストールが完了したら、Red Hat JBoss Developer Studio を再起動します。

5.2. DROOLS ランタイムの設定

Red Hat JBoss Developer Studio で Red Hat JBoss BRMS プラグインを使用するには、ランタイムを設定する必要があります。

ランタイムは、このソフトウェアの特定のリリースを表す JAR ファイルを集めたもので、ビジネスアセットのコンパイルおよび実行に必要なライブラリーを提供します。

手順: Red Hat JBoss BRMS Runtime の設定

1. Red Hat カスタマーポータルでの「[ダウンロード](#)」セクションを開きます。
2. Red Hat JBoss BRMS をクリックしてログインします。
3. Red Hat JBoss BRMS 6.4.0 Core Engine をダウンロードします。
4. **jboss-brms-VERSION-engine.zip** に格納されているランタイム JAR ファイルを展開します。
5. Red Hat JBoss Developer Studio メニューより **Window** → **Preferences** をクリックします。
6. **Drools** → **Installed Drools Runtimes** を選択します。
7. **Add...** をクリックして、新しいランタイムの名前を入力し **Browse** をクリックして、展開したランタイムファイルのあるディレクトリーに移動します。OK をクリックして、選択したランタイムを Red Hat JBoss Developer Studio に登録します。
8. 横にあるチェックボックスをクリックして、作成したランタイムをデフォルトの Drools ランタイムとして指定します。
9. OK をクリックします。既存のプロジェクトがある場合はダイアログボックスが表示され、Red Hat JBoss Developer Studio を再起動してランタイムを更新する必要があることが示されません。

5.3. RED HAT JBOSS BRMS サーバーの設定

Red Hat JBoss BRMS サーバーを実行するように Red Hat JBoss Developer Studio を設定できます。

手順: サーバーの設定

1. **Window** → **Perspective** → **Open Perspective** → **Other** と選択した後に **Drools** を選んで Drools ビューを開き、OK をクリックします。
2. **Window** → **Show View** → **Other...** と選択した後に、**Server** → **Servers** とクリックして、**Server** のビューを追加します。
3. **Servers** パネルを右クリックしてサーバーメニューを開き、**New** → **Server** を選択します。
4. **JBoss Enterprise Middleware** → **JBoss Enterprise Application Platform 6.1+** を選択し、**Next** をクリックして、サーバーを定義します。
5. **JBoss EAP 6.4 Runtime** をクリックして **Create new runtime (next page)** を選択します。Next をクリックします。
6. **Browse** をクリックしてホームディレクトリーを設定します。Red Hat JBoss BRMS をインストールした Red Hat JBoss EAP 6.4 のインストールディレクトリーを選択します。
7. **Name** フィールドにサーバーの名前を指定して、設定ファイルが設定されていることを確認し、**Finish** をクリックします。

5.4. KIE NAVIGATOR

Kie Navigator を使用すると、Red Hat JBoss BRMS サーバーコンテンツの閲覧、変更、およびデプロイが可能になります。つまり、Red Hat JBoss Developer Studio と Red Hat JBoss BRMS のシームレスな統合ができるようになります。

Kie Navigator の設定

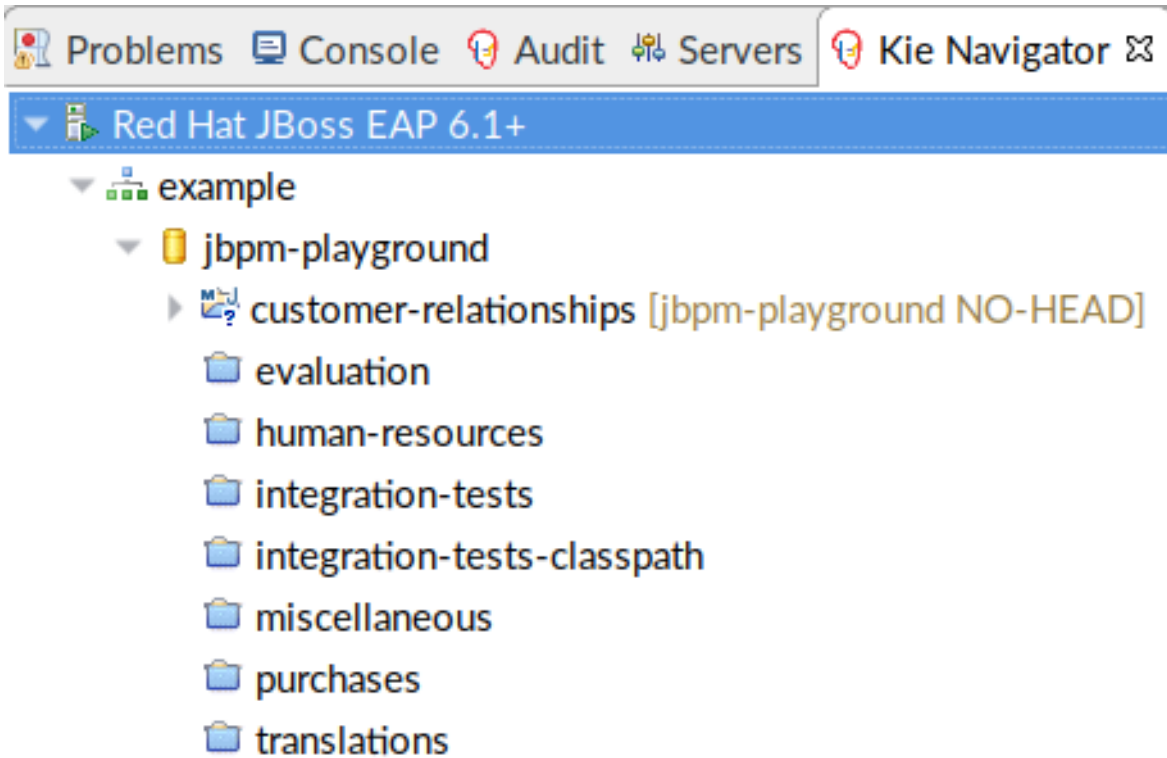
1. Red Hat JBoss Developer Studio で **Window** → **Show View** → **Other...** とクリックします。
2. **Kie Navigator** と入力し、**Kie Navigator** オプションを選択します。
3. Red Hat JBoss Developer Studio で Red Hat JBoss BRMS サーバーが設定されていない場合は、この設定を求められます。詳細は、「[Red Hat JBoss BRMS サーバーの設定](#)」を参照してください。
4. 使用するサーバー名を右クリックして、**Properties** を選びます。
5. kie のロールがあるユーザー名とパスワードを入力します。例えば、**kie-server** となります。
6. 以下を入力します。
 - **KIE Application Name: business-central**
 - **HTTP Port: 8080**
 - **Git Port: 8001**



注記





上記の値はデフォルト値です。Git または Business Central ポートのデフォルト値を変更している場合は、その値を使用してください。

7. **Use default Git Repository Path** にチェックを入れ、**Apply** をクリックします。
8. **Servers** タブでサーバーを起動し、稼働するまで待機します。
9. **Kie Navigator** タブで使用するサーバーを右クリックし、**Refresh** をクリックします。Red Hat JBoss BRMS git リポジトリのコンテンツが確認できます。Red Hat JBoss Developer Studio にインポートするプロジェクトをダブルクリックします。




Kie Navigator の機能

Kie Navigator ツリーでは、以下の 4 つのレベルを確認できます。

- サーバー ()
- 組織単位 ()
- リポジトリ ()
- プロジェクト ()

Kie Navigator サーバーレベル

 を右クリックすると、以下のオプションが選択できます。

Refresh

REST を実行してツリー全体を更新します。


Create Organization

新規の組織単位を作成します。

Properties

サーバーのプロパティを表示します。

Kie Navigator 組織単位レベル

 を右クリックすると、以下のオプションが選択できます。

Add Repository...

別の組織単位に関連付けられていないリポジトリを追加します。

Create Repository...

リポジトリを作成します。


Delete Organization...

選択した組織単位を削除します。

Properties

選択した組織単位のプロパティを表示します。

Kie Navigator リポジトリレベル

 を右クリックすると、以下のオプションが選択できます。

Import Repository

リポジトリをクローンして、Git Repository ビューで利用可能にします。このメニューのアクションは、リポジトリがクローンされていない場合に利用可能になります。

Create Project...

プロジェクトを作成します。

Remove Repository...

選択したリポジトリを削除します。


Show in Git Repository View

Git Repositories ビューを開いて、選択したリポジトリをハイライト表示します。

Properties

選択したリポジトリのプロパティを表示します。

Kie Navigator プロジェクトレベル

 を右クリックすると、以下のオプションが選択できます。

Import Project

Red Hat JBoss Developer Studio にプロジェクトをインポートします。

Delete Project...

選択したプロジェクトを削除します。

Properties

プロジェクトのプロパティを表示します。

5.5. GIT リポジトリから RED HAT JBOSS DEVELOPER STUDIO へのプロジェクトのインポート



注記

この追加機能は、Red Hat JBoss Developer Studio での作業には必須ではありません。

Red Hat JBoss Developer Studio は、中央の Git アセットリポジトリへ接続するように設定できます。リポジトリにはルール、モデル、関数、およびプロセスが保存されます。

ローカルの Git リポジトリをインポートするか、リモート Git リポジトリをクローンすることができます。

手順: リモート GIT リポジトリのクローン

1. **Server** タブから Red Hat JBoss BRMS サーバーを選択し、Start アイコンをクリックして起動します。

- セキュアシェルサーバーが稼働していない場合は、以下のコマンドを使用してこれを起動します。コマンドは、Linux および Mac 環境に固有のもので、これらのプラットフォームで `sshd` がすでに起動していると、このコマンドに失敗しますが、無視しても問題ありません。

```
/sbin/service sshd start
```

- JBoss Developer Studio で **File** → **Import...** とクリックし、Git フォルダーに移動します。Git フォルダーを開き、**Projects from Git** を選択して **Next** をクリックします。
- Clone URI** としてリポジトリのソースを選択して、**Next** をクリックします。
- 次のウィンドウで Git リポジトリの詳細を入力して **次へ** をクリックします。

図5.1 Git リポジトリの詳細

Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI: Local File...

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

Store in Secure Store

? < Back Next > Cancel Finish

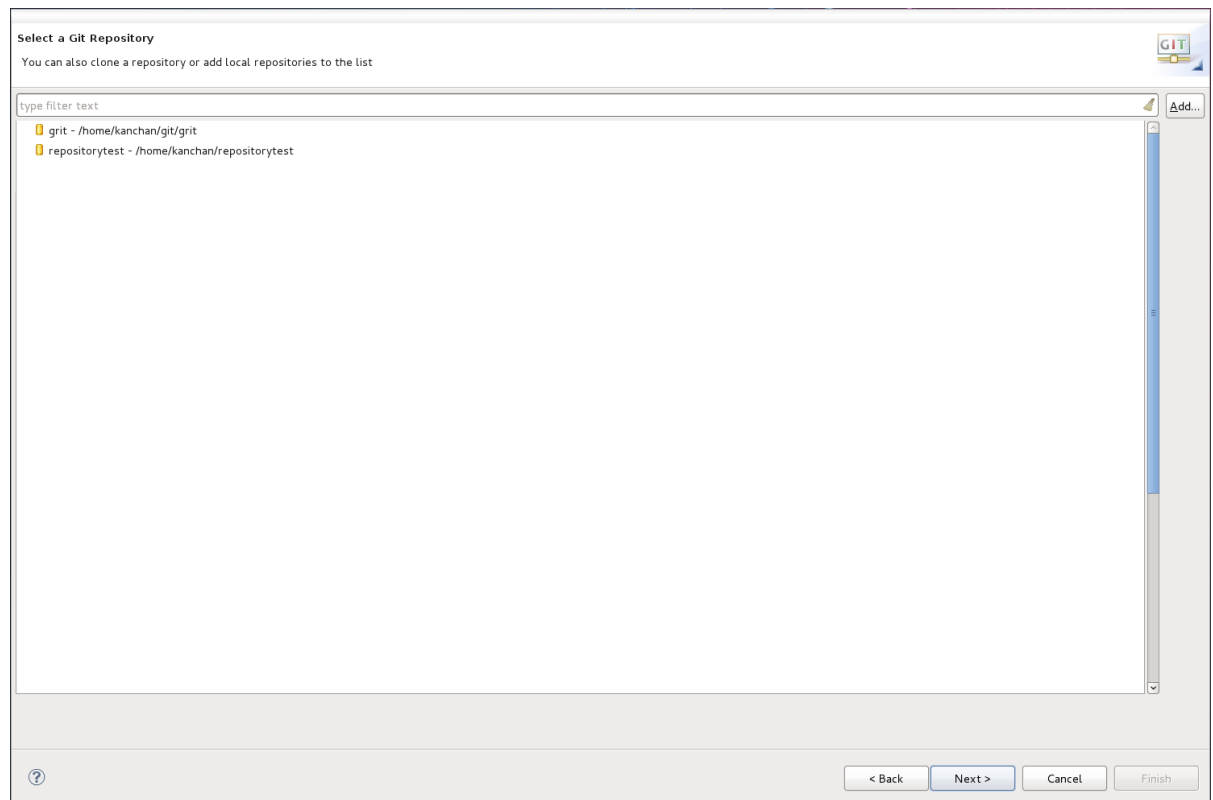
- 次のウィンドウでインポートするブランチを選択し、**Next** をクリックします。
- このプロジェクトのローカルストレージを定義するために、空でないディレクトリを入力 (または選択) して設定を変更し、**Next** をクリックします。

- 次のウィンドウでプロジェクトを一般プロジェクトとしてインポートし、**Next** をクリックします。このプロジェクトに名前を付け、**Finish** をクリックします。

手順: ローカル GIT リポジトリのインポート

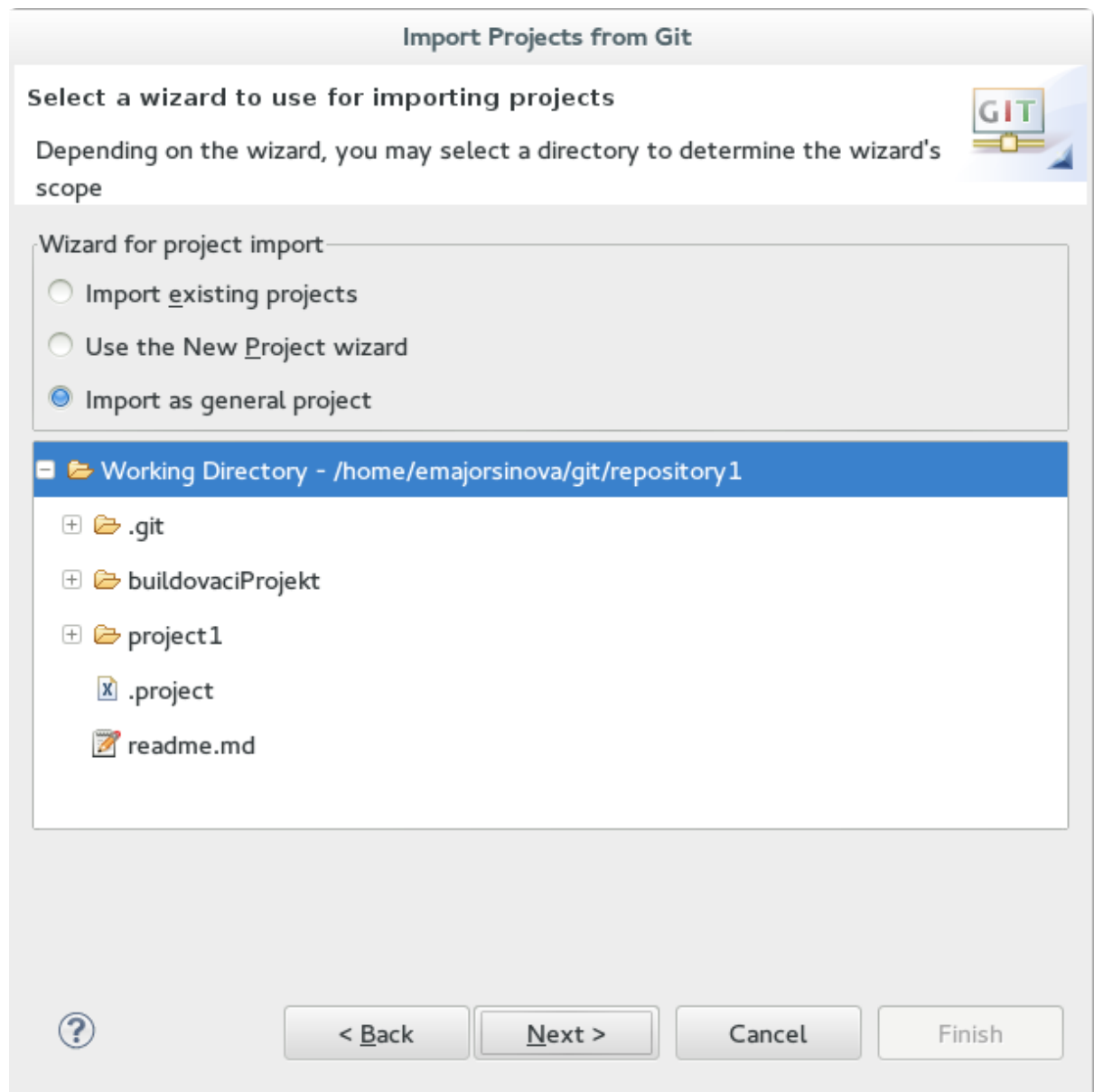
- Server** タブから Red Hat JBoss BRMS サーバーを選択し、Start アイコンをクリックして起動します。
- JBoss Developer Studio で **File** → **Import...** とクリックし、Git フォルダーに移動します。Git フォルダーを開き、**Projects from Git** を選択して **Next** をクリックします。
- Existing local repository** としてリポジトリのソースを選択して、**Next** をクリックします。

図5.2 Git リポジトリの詳細



- 使用可能なリポジトリリストから設定するリポジトリを選択し、**Next** をクリックします。
- 表示されたダイアログで、**Wizard for project import group** から **Import as general project** のラジオボタンを選択して、**Next** をクリックします。プロジェクトの名前を指定して **Finish** をクリックしてください。

図5.3 プロジェクトインポートのウィザード



5.6. DROOLS プロジェクトの作成

手順: Red Hat JBoss Developer Studio 新規プロジェクトの作成

1. メインメニューから **File** → **New** → **Project** をクリックします。
2. **Drools** → **Drools Project** を選択して、**Next** をクリックします。
3. ここで、2つ目のオプションを選択します。Red Hat JBoss Developer Studio では、Red Hat JBoss BRMS の例があるプロジェクトが作成されます。**Next** をクリックします。
4. **Project name:** テキストボックスにプロジェクト名を入力して **Finish** をクリックします。

プロジェクトをテストするには、以下を実行します。

1. **src/main/java** ディレクトリーに移動し、**com.sample** パッケージを展開します。
2. Java クラスを右クリックして、**Run As** → **Java Application** を選択します。
Console タブに出力が表示されます。

第6章 BUSINESS RESOURCE PLANNER

Business Resource Planner は組み込み可能な軽量プランニングエンジンで、プランニングの問題を最適化します。最適化のためのヒューリスティック法およびメタヒューリスティック法を非常に効率的なスコア計算と組み合わせ、Java™ 開発者がプランニングの問題を効率的に解決できるようにします。

このプランナーは、次のようなさまざまなユースケースの解決に役立ちます。

- **従業員勤務表/患者の名簿:** プランナーを使用して看護婦の勤務時間割を作成し、患者のベッド管理を追跡できます。
- **学校の時間割:** プランナーは、授業、コース、試験、および会議のプレゼンテーションの計画を容易にします。
- **工場の計画:** プランナーは、自動車の組み立てライン、機械の待機計画、および作業員のタスク計画を追跡します。
- **在庫の削減:** プランナーを使用すると、紙や金属などの資源の消費を削減し、無駄を最小限にすることができます。

6.1. BUSINESS RESOURCE PLANNER のインストール

1. Red Hat カスタマーポータル [「ダウンロード」](#) セクションを開きます。
2. Red Hat JBoss BRMS をクリックしてログインします。
3. バージョン のドロップダウンメニューからバージョン 6.4 を選択します。
4. Red Hat JBoss BRMS 6.4.0 Business Resource Planner を選択して、Download をクリックします。

6.2. BUSINESS RESOURCE PLANNER サンプルの実行

1. コマンドラインで `examples/` ディレクトリーに移動します。
2. Unix 環境では以下を実行します。

```
./runExamples.sh
```

Windows 環境では以下を実行します。

```
runExamples.bat
```

3. 表示された Examples GUI アプリケーションよりサンプルを1つ選択し、任意の IDE で実行します。

付録A バージョン情報

Documentation last updated on: Wednesday, Oct 23, 2019.