



# Red Hat JBoss BPM Suite 6.3

## スタートガイド

Red Hat JBoss BPM Suite のスタートガイド



# Red Hat JBoss BPM Suite 6.3 スタートガイド

---

Red Hat JBoss BPM Suite のスタートガイド

## Red Hat Content Services

Klara Kufova

kkufova@redhat.com

Marek Czernek

mczernek@redhat.com

Tomas Radej

tradej@redhat.com

Vidya Iyengar

viyengar@redhat.com

## 法律上の通知

Copyright © 2016 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

This guide provides information for getting started with Red Hat JBoss BPM Suite.

## 目次

第1章 RED HAT JBOSS BPM SUITE のクイックスタート .....	3
第2章 インストールオプション .....	4
2.1. EAP6 バンドルのインストール	4
2.2. 汎用デプロイ可能バンドルのインストール	8
2.3. サーバーの起動	16
2.4. JAVA SECURITY MANAGER とパフォーマンス管理	17
第3章 BUSINESS CENTRAL へのログイン .....	19
第4章 HELLO WORLD プロジェクト .....	20
4.1. リポジトリ構造の作成	20
第5章 HELLO WORLD プロセス .....	22
5.1. ビジネスプロセスの作成	22
5.2. ビジネスプロセスのモデル化	22
5.3. 要素プロパティの定義	23
5.4. ビルドとデプロイ	24
5.5. ビジネスプロセスのインスタンス化	25
5.6. ビジネスプロセスの中止	25
第6章 HELLO WORLD ビジネスルール .....	27
6.1. ビジネスルールの作成	27
6.2. ビジネスルールタスクの追加	28
6.3. ビルドとデプロイ	28
6.4. ビジネスプロセスのインスタンス化	29
第7章 BAM .....	31
7.1. RED HAT JBOSS BPM SUITE DASHBUILDER へのアクセス	31
7.2. インスタンスの監視	31
第8章 RED HAT JBOSS DEVELOPER STUDIO .....	32
8.1. JBOSS CENTRAL	32
8.2. JBOSS DEVELOPER STUDIO プラグインのインストール	33
8.3. DROOLS ランタイムの設定	34
8.4. JBOSS BPM SUITE ランタイムの設定	34
8.5. JBOSS BPM SUITE サーバーの設定	35
8.6. GIT リポジトリから JBOSS DEVELOPER STUDIO へのプロジェクトのインポート	35
8.7. DROOLS プロジェクトの作成	38
8.8. JBOSS BPM SUITE プロジェクトの作成	39
第9章 BUSINESS RESOURCE PLANNER .....	40
9.1. BUSINESS RESOURCE PLANNER のインストール	40
9.2. BUSINESS RESOURCE PLANNER サンプルの実行	40
付録A 改訂履歴 .....	41



## 第1章 RED HAT JBOSS BPM SUITE のクイックスタート

このセクションでは、最小手順で Red Hat JBoss BPM Suite をダウンロード、インストールおよび実行します。詳細な手順や、他のインストール方法については、本セクションに続く内容を参照してください。

この手順では、ご使用のシステムに最小限サポートされる Java バージョンがインストールされていることを前提としています。インストーラーは Red Hat JBoss EAP サーバーをインストールしないので、既存の Red Hat JBoss EAP サーバー (バージョン 6.4 およびそれ以降) も必須となります。



### 注記

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、[access.redhat.com](https://access.redhat.com) から最新のパッチをダウンロードし、『Red Hat JBoss EAP Installation Guide』に記述された更新手順に従ってください。

### 手順1.1 Red Hat JBoss BPM Suite のクイックスタート

1. JBoss BPM Suite インストーラーを [access.redhat.com](https://access.redhat.com) から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-bpmsuite-6.3.0.GA-installer.jar
```

3. GUI インストーラーの指示に従い、インストール後に JBoss BPM Suite アプリケーションにアクセスするために必要なユーザー名とパスワードを書き留めておきます。
4. インストールが終了したら、コマンドプロンプトで JBoss BPM Suite をインストールした EAP ホームフォルダーに移動します。
5. 以下のコマンドを実行して、JBoss BPM Suite サーバーを起動します。

```
bin/standalone.sh
```

6. Web ブラウザーを開き、<http://localhost:8080/business-central/> にアクセスして Business Central にログインします。インストール時に作成した、JBoss BPM Suite アプリケーション用のユーザー名とパスワードの組み合わせを使用してログインします。

これで Red Hat JBoss BPM Suite が正常にインストールされ、稼働します。以下のことが実行可能となります。

ロールに関する知識を深める: 「[ロールの定義](#)」

新たなユーザーの作成: 「[ユーザーの作成](#)」

Hello World プロジェクトの作成: [4章 Hello World プロジェクト](#)

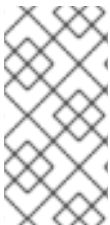
インストーラーを用いた詳細なインストール手順の確認: 「[インストールガイド](#)」

## 第2章 インストールオプション

Red Hat JBoss BPM Suite には、以下の2つのバージョンがあります。

- Red Hat JBoss Enterprise Application Platform (EAP) 6.4 にインストールするための実行可能 jar インストーラー。
- zip ファイルインストール。これ自体も2つのバージョンがあります。
  - **jboss-bpmsuite-6.3.0.GA-deployable-eap6.x.zip**: Red Hat JBoss Enterprise Application Platform でのデプロイメント用のバージョン。
  - **jboss-bpmsuite-6.3.0.GA-deployable-generic.zip**: Red Hat JBoss Web Server (EWS)、Apache Tomcat 6、および Apache Tomcat 7 上のデプロイメントに適した追加ライブラリーが含まれるデプロイ可能バージョン。

ご使用の環境に応じて、プロジェクトに最も適したインストールオプションを選択してください。



### 注記

Red Hat JBoss BPM Suite 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。基礎となる 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 が使用されるようにするには、"**-Dfile.encoding=UTF-8**" のシステムプロパティを使用してください。



### 重要

JBoss BPM Suite 6.1 以降では、Red Hat JBoss BPM Suite をインストールする前に JBoss EAP 6.4 以降がインストールされている必要があります。

## 2.1. EAP6 バンドルのインストール

本セクションでは、Red Hat JBoss Enterprise Application Platform (EAP) にデプロイ可能な Red Hat JBoss BPM Suite パッケージのインストールについて説明します。



### 注記

Red Hat JBoss BPM Suite のインストールでは Red Hat JBoss EAP 6.4 以上がサポート対象となります。

### 2.1.1. Red Hat JBoss EAP6 パッケージのダウンロード

JBoss Enterprise Application Platform 向けのデプロイ可能な Red Hat JBoss BPM Suite パッケージをダウンロードするには、以下の手順に従います。

1. [Red Hat カスタマーポータル](#)にアクセスし、ログインします。
2. ページ上部の **DOWNLOADS (ダウンロード)** をクリックします。
3. 開いた **製品のダウンロード** ページで、**Red Hat JBoss BPM Suite** をクリックします。
4. **Version** ドロップダウンメニューからバージョン **6.3.0** を選択します。



5. 開かれた **Software Downloads** ページで、**Red Hat JBoss BPM Suite 6.3.0 Deployable for EAP 6** の行にある **Download** をクリックします。

## 2.1.2. Red Hat JBoss EAP6 パッケージのインストール

### 新しい JBoss EAP インスタンス上でのインストール

未設定の JBoss EAP 向けの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 カスタマーポータル](#) からダウンロードした、JBoss EAP にデプロイ可能な zip パッケージを展開します。
2. JBoss EAP にデプロイ可能な zip パッケージを展開した後、JBoss EAP の `SERVER_HOME` ディレクトリーへマージします。



#### 警告

この手順は、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する必要があります。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. このプロセスでは、JBoss EAP の `SERVER_HOME` ディレクトリーの複数ファイルが上書きされますが、zip の展開処理でこれらファイルを上書きすることを許可する必要があります。
4. `SERVER_HOME/bin/product.conf` ファイルに文字列 `slot=bpms` が含まれていることを確認してください。

Red Hat Enterprise Linux では、zip ファイルをダウンロードしたディレクトリーで次のコマンドを実行すると、Red Hat JBoss BPM Suite zip ファイルの展開とサーバーディレクトリーへのマージを一度に行えます。

```
unzip -u jboss-bpmsuite-VERSION-TYPE.zip -d SERVER_HOME_PARENT_DIR
```

#### 例2.1 unzip コマンド

```
unzip -u jboss-bpmsuite-6.3.0.GA-deployable-eap6.x.zip -d  
/home/john/myServers/
```

サーバー起動時に、Red Hat JBoss BPM Suite がデプロイされます。

### 既存の JBoss EAP 設定上でのインストール



## 警告

以下の手順は JBoss BPM Suite をインストールするためのものであって、既存の JBoss BPM Suite 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 JBoss EAP に既存の JBoss BPM Suite 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 JBoss EAP 向けの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 カスタマーポータル](#)からダウンロードした、JBoss EAP にデプロイ可能な zip パッケージを展開します。
2. ダウンロードした zip アーカイブを展開しますが、ファイルを上書きしないでください。以下のファイルを手作業で `SERVER_HOME` ディレクトリーにマージします。
  - `jboss-eap-6.4/domain/configuration/*` - JBoss BPM Suite には JMS が必要なため、JBoss BPM Suite 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注意してください。
  - `jboss-eap-6.4/standalone/configuration/*` - JBoss BPM Suite には JMS が必要なため、JBoss BPM Suite ディストリビューションで提供されるすべての設定ファイル (特に `standalone.xml` と `standalone-ha.xml`) 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。
  - `jboss-eap-6.4/modules/layers.conf`
  - `jboss-eap-6.4/bin/product.conf`
3. ターゲットの JBoss EAP に競合する名前のデプロイメントが含まれないようにしてください。JBoss BPM Suite ディストリビューションから、フォルダー `jboss-eap-6.4/standalone/deployments` を `EAP_HOME` ディレクトリーにコピーします。
4. JBoss BPM Suite という名前の JBoss EAP モジュールレイヤーが存在しないことを確認し、フォルダー `jboss-eap-6.4/modules/system/layers/bpms` を EAP 6.4 フォルダーにコピーします。

### 2.1.3. ロールの定義

サーバーを起動して 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 BPM Suite で使用される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) を管理でき、アプリケーションに必要な変更を加えるため完全アクセスを持ちます。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。
- **developer:** `developer` は、ほとんどすべての機能にアクセスし、ルール、モデル、プロセスフロー、フォーム、およびダッシュボードを管理することができます。また、アセットリポジトリを管理したり、プロジェクトを作成、構築、およびデプロイしたり、RedHat

JBossDeveloperStudio を使用してプロセスを表示したりできます。ただし、**developer** ロールでは、新規リポジトリーの作成やクローンなど、一部の管理機能を使用できません。

- **analyst:** **analyst** ロールは、プロジェクトをモデル化および実行するため、すべての高レベル機能へアクセスできます。しかし、**analyst** ロールを持つユーザーは **オーサリング**→**管理** へアクセスできません。また、**デプロイ**→**アーティファクトリポジトリー** 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。しかし、**Project Editor** を使用すると、**Build & Deploy** ボタンを使用できます。
- **user:** ユーザーまたはビジネスユーザーは、特定プロセスを操作するために使用されるビジネスタスクリストの作業を行います。このロールを持つユーザーは、ダッシュボードへのアクセスやプロセスの管理が可能です。
- **manager:** マネージャーはシステムを閲覧し、ビジネスプロセスに関する統計、ビジネスプロセスのパフォーマンス、ビジネス指標、およびシステムのその他のレポート内容に関心があります。このロールを持つユーザーは **BAM** にのみアクセスできます。



### 注記

上記のロールは、ユーザー作成プロセスの間に入力します。ロールについての詳細情報については、[Chapter 2. Business Central configuration](#) を参照してください。ロールとタスクの関係の詳細については、[Appendix 11.5 User Task](#) を参照してください。

## 2.1.4. 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。

1. Unix システムの場合は、以下のコマンド

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

を実行します。Windows システムの場合は、以下のコマンド

```
add-user.bat
```

を **bin** ディレクトリーから実行します。

2. ユーザータイプ入力のプロンプトが表示されたら **b** を入力して **Application User** を選択し、**Enter** キーを押します。
3. **Enter** を押してデフォルトのレルム (**ApplicationRealm**) を使用します。
4. 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。



### 注記

ユーザー名は既存のグループと競合しないようにしてください。たとえば、**admin** という名前のグループがすでにある場合、**admin** という名前のユーザーは作成しないでください。

5. 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。

6. ロール入力のプロンプトが表示されたら、ユーザーが必要となるロールをコンマ区切りのリストで入力します (「[ロールの定義](#)」を参照)。
7. ユーザーを追加することを確認します。
8. 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 (今後必要な場合)。

## 2.2. 汎用デプロイ可能バンドルのインストール

Red Hat JBoss Web Server (EWS) 上に Red Hat JBoss BPM Suite をインストールするには、製品の汎用デプロイ可能パッケージ (generic deployable package) を使用する必要があります。

EWS 上のインストールでは、汎用デプロイ可能パッケージに Red Hat JBoss EWS の一部ではない追加のトランザクションマネージャーとセキュリティーライブラリーが含まれます。

汎用デプロイ可能パッケージをインストールするには、次の zip ファイルが必要になります。

- **jboss-bpmsuite-6.3.0.GA-deployable-generic.zip**: このファイルには、**business-central.war**、**dashbuilder.war**、および **kie-server.war** の Web アプリケーションが含まれています。
- **jboss-bpmsuite-6.3.0.GA-engine.zip**: 汎用デプロイメントに必要なアプリケーションおよび他のライブラリーをエンジンに組み込むためのサポート対象実行エンジンライブラリーです。

### 2.2.1. 汎用デプロイ可能パッケージのダウンロード

JBoss Web Server 用の Red Hat JBoss BPM Suite 汎用デプロイ可能パッケージをダウンロードするには、以下の手順に従います。

1. [Red Hat カスタマーポータル](#)にアクセスし、ログインします。
2. **ダウンロード** をクリックします。
3. 製品一覧から **Red Hat JBoss BPM Suite** をクリックします。
4. **Version** のドロップダウンメニューから、バージョン **6.3.0** を選択します (まだ選択されていない場合)。
5. 表示される **Software Downloads** セクションで、**Red Hat JBoss BPM Suite 6.3.0 Deployable for All Supported Containers** の行に移動し、**Download** をクリックします。
6. また **Red Hat JBoss BPM Suite 6.3.0 Core Engine** ファイルの行にも移動し、**Download** をクリックして JBoss BPM Suite Core Engine ファイルをダウンロードします。

### 2.2.2. 汎用デプロイ可能パッケージのインストール

汎用デプロイ可能パッケージをインストールするには、基盤のプラットフォーム (Red Hat JBoss WS) をインストールした後に以下を設定する必要があります。

- データベースドライバーとトランザクションマネージャー (Bitronix) を設定します (「[Red Hat JBoss Web Server 2.1 \(Tomcat 7\) 用のトランザクションマネージャーの設定](#)」を参照)。
- Business Central アプリケーションを設定します。ユーザーとロールを設定し、永続性を設定します (「[Red Hat JBoss Web Server 2.1 \(Tomcat 7\) 用の Business Central の設定](#)」を参照)。
- Dashbuilder アプリケーションを設定します。ユーザーとロールを設定し、永続性を設定します (「[Red Hat JBoss Web Server 2.0 \(Tomcat 7\) への Dashbuilder の設定](#)」を参照)。

### 2.2.2.1. Red Hat JBoss Web Server 2.1 (Tomcat 7) 用のトランザクションマネージャーの設定

1. [Red Hat Customer Portal](#) からダウンロードした汎用デプロイ可能 zip パッケージを一時的な場所に展開します。この zip パッケージには、**business-central.war**、**dashbuilder.war**、および **kie-server.war** という 3 つの Web アプリケーションアーカイブが展開された形式で含まれています。これらのフォルダーの名前を変更して、**.war** 拡張子を削除します。
2. これらのフォルダーを、**\$TOMCAT\_DIR/webapps** フォルダー直下にコピーします。

これにより、**\$TOMCAT\_DIR/webapps/business-central**、**\$TOMCAT\_DIR/webapps/dashbuilder** および **\$TOMCAT\_DIR/webapps/kie-server** という 3 つの展開された形式のフォルダーが存在することになります。



#### 注記

**\$TOMCAT\_DIR** は、ご使用の web サーバーがあるホームディレクトリーを意味します。web サーバーのホームディレクトリーへの実際のパスに置き換えてください (例: `/home/john/jboss-ews-2.1/tomcat7/`)。

3. JBoss BPM Suite Engine ファイルアーカイブのコンテンツを、必要なライブラリーをコピーできる一時的な場所に展開します。展開後には、展開されたフォルダーと **lib** フォルダー下にあるすべてのコア JBoss BPM Suite ライブラリーが含まれるようになります。
4. トランザクションマネージャーをインストールします。



#### 警告

以下のセクションでは、トランザクションマネージャーである Bitronix の設定について説明しています。これは Red Hat では正式にサポートされていません。

以下のトランザクションマネージャー jar ライブラリーを **lib** フォルダーから **\$TOMCAT\_DIR/lib/** ディレクトリーにコピーします。

- **btm-VERSION.jar**
- **btm-tomcat55-lifecycle-VERSION.jar**

- `jta-VERSION.jar`
- `slf4j-api-VERSION.jar`
- `slf4j-jdk14-VERSION.jar`

さらに、以下のライブラリーをダウンロードし、`$TOMCAT_DIR/lib/` フォルダにコピーします。

- [javax.security.jacc-api.jar](#)

#### 5. ドライバーをデータベースにインストールします。

関連するデータベースドライバーが含まれる `jar` ファイルを `$TOMCAT_DIR/lib/` へコピーします。



#### 組み込みの H2 データベースへのドライバー

組み込みの H2 データベースを使用している場合は、ドライバーは `business-central/WEB-INF/lib/` にあります。

#### 6. トランザクションマネージャー設定ファイルを `$TOMCAT_DIR/conf/` に作成します。

- `btm-config.properties`

```
bitronix.tm.serverId=tomcat-btm-node0
bitronix.tm.journal.disk.logPart1Filename=${btm.root}/work/btm1.t
log
bitronix.tm.journal.disk.logPart2Filename=${btm.root}/work/btm2.t
log
bitronix.tm.resource.configuration=${btm.root}/conf/resources.pro
perties
```

- `resources.properties` (`resource.ds1.uniqueName` は後で `tomcat` リソース定義で使用されるデータソース名を定義します。この値を覚えておいてください)

ご使用の環境に一致するよう、以下の定義の値を変更してください。

#### 例2.2 H2 データソース定義

```
resource.ds1.className=bitronix.tm.resource.jdbc.lrc.LrcXADataS
ource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=10
resource.ds1.maxPoolSize=20
resource.ds1.driverProperties.driverClassName=org.h2.Driver
resource.ds1.driverProperties.url=jdbc:h2:file:~/jbpm
resource.ds1.driverProperties.user=sa
resource.ds1.driverProperties.password=
resource.ds1.allowLocalTransactions=true
```

#### 例2.3 MySQL 5.5 データソース定義

```
resource.ds1.className=com.mysql.jdbc.jdbc2.optional.MysqlXADat
```

```
aSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:mysql://localhost:3306/s
ampled
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

#### 例2.4 DB2 タイプ 4 のデータソース定義

```
resource.ds1.className=com.ibm.db2.jcc.DB2Driver
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:db2://localhost:50000/sa
mpledb
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

#### 例2.5 Oracle のデータソース定義

```
resource.ds1.className=oracle.jdbc.xa.client.OracleXADataSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:oracle:thin:@//localhost
:1521/bpms
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

#### 例2.6 Microsoft SQL サーバーのデータソース定義

```
resource.ds1.className=com.microsoft.sqlserver.jdbc.SQLServerDr
iver
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:sqlserver://localhost:14
33;databaseName=bpms;
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

7. コンテナの起動および停止時に **Bitronix** を起動および停止するため、**\$TOMCAT\_DIR/conf/server.xml** でトランザクションマネージャリスナーを設定します。

以下の要素を最後の **<Listener>** 要素として **<Server>** 要素に追加します。

```
<Listener
  className="bitronix.tm.integration.tomcat55.BTMLifecycleListener" />
```

8. **btm.root** システムプロパティと **bitronix** 設定ファイルが置かれる場所を定義します。

**\$TOMCAT\_DIR/bin/** 内に、以下の内容が含まれる読み取り可能な **setenv.sh** ファイルを作成します。

```
CATALINA_OPTS="-Xmx512M -XX:MaxPermSize=512m -
Djava.security.auth.login.config=$CATALINA_HOME/webapps/business-
central/WEB-INF/classes/login.config -Dbtm.root=$CATALINA_HOME -
Dbitronix.tm.configuration=$CATALINA_HOME/conf/btm-config.properties
-Dorg.jbpm.designer.perspective=RuleFlow -
Djbpm.tsr.jndi.lookup=java:comp/env/TransactionSynchronizationRegistr
y"
```

プロパティ **org.jbpm.designer.perspective** は **RuleFlow** に設定され、デザイナーのデフォルトのパーспекティブが **Full** ではなく **RuleFlow** になることを可能とします。該当する場合は、ファイル実行パーミッションを付与します。

git リポジトリの **ssh clone** が機能するには、**java.security.auth.login.config** プロパティを設定する必要があります。



## MICROSOFT WINDOWS システムの TOMCAT

Microsoft Windows システムでは、ファイルコンテンツの **\$CATALINA\_HOME** の値を同等の環境変数名に置き換えるか、以下の例のように絶対パスを使用して **setenv.bat** に値を追加します。

```
set "CATALINA_OPTS=-Xmx512M -XX:MaxPermSize=512m -
Dbtm.root=C:/Tomcat -
Dbitronix.tm.configuration=C:/Tomcat/conf/btm-
config.properties -
Dorg.jbpm.designer.perspective=RuleFlow"
```

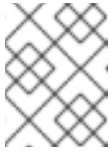
### 2.2.2.2. Red Hat JBoss Web Server 2.1 (Tomcat 7) 用の Business Central の設定

以下の手順に従って、**Business Central** を設定します。

1. **Valve** を設定して、Tomcat のユーザー設定が **Business Central** の Web アプリケーションでロードできるようにします。
  - a. **\$TOMCAT\_DIR/conf/tomcat-users.xml** でユーザーとロールを定義します。**Business Central** では、ユーザーが **admin** や **analyst** として指定されたロールを持つ必要があることに注意してください (ユーザーおよびロール定義の詳細は、Tomcat 7 のドキュメントを参照してください)。



以下のプログラム一覧は、**admin** および **analyst** ロールを追加し、**bpmsadmin** というユーザーにこれらのロールを割り当てる方法の例です。



### 注記

ユーザー名は既知のロールと競合しないようにしてください。たとえば、**admin** という名前のユーザーは作成しないでください。

```
<role rolename="admin"/>
<role rolename="analyst" />
<user username="bpmsadmin" password="P@ssw0rd"
roles="admin, analyst"/>
```

- b. **\$TOMCAT\_DIR/webapps/business-central/WEB-INF/lib/**にある **kie-tomcat-integration-VERSION.jar** を **\$TOMCAT\_DIR/lib/** に移動します (コピーはしないでください)。
- c. **\$TOMCAT\_DIR/webapps/business-central/WEB-INF/lib/**にある **jboss-jaxb-api-VERSION.jar** を **\$TOMCAT\_DIR/lib/** にコピーします。
- d. **\$TOMCAT\_DIR/conf/server.xml** 内で、Tomcat Valve 宣言を関連する **<host>** 要素に追加します。

```
<Valve className="org.kie.integration.tomcat.JACCValve" />
```

2. 基盤の H2 データベースが提供するデフォルト以外のデータソースを使用している場合は、永続性を設定する必要があります。デフォルトの H2 データベースを使用している場合は、これ以降の手順を省略できます。

この手順では、MySQL オプション向けに定義済みである bitronix の **resources.properties** ファイルの **uniqueName=jdbc/jbpm** で定義された JNDI 名 **jdbc/myDataSource** を用いてデータソースを設定します。

- a. **business-central/META-INF/context.xml** で、**<Resource>** 要素のデータソース JNDI 名を置き換えます。 **uniqueName** 属性は、**resources.properties** で設定された **resource.ds1.uniqueName** プロパティになります。

```
<Resource name="jdbc/myDataSource" uniqueName="jdbc/jbpm"
auth="Container" removeAbandoned="true"
factory="bitronix.tm.resource.ResourceObjectFactory"
type="javax.sql.DataSource"/>
```

- b. **business-central/WEB-INF/web.xml** で、**<res-ref-name>** 要素のデータソース JNDI 名を使用するデータソース名に置き換えます。

```
<resource-ref>
  <description>Console DS</description>
  <res-ref-name>jdbc/myDataSource</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

- c. **business-central/WEB-INF/classes/META-INF/persistence.xml** を変更しま

す。

H2以外のデータベースを使用している場合、このファイルでデータベースの **hibernate dialect** の名前を変更します。以下のコードは、**persistence.xml** の元のデータベース情報を示しています。

```
<property name="hibernate.dialect"
value="org.hibernate.dialect.H2Dialect"/>
```

この情報は次のように更新できます (以下は MySQL データベースの場合)。

```
<property name="hibernate.dialect"
value="org.hibernate.dialect.MySQLDialect"/>
```



### 注記

DB2 のダイアレクトは **org.hibernate.dialect.DB2Dialect** になります。AS/400 上の DB2 の場合は **org.hibernate.dialect.DB2400Dialect**、Oracle は **org.hibernate.dialect.Oracle10gDialect**、Microsoft SQL Server は **org.hibernate.dialect.SQLServerDialect** になります。

- d. **business-central/WEB-INF/classes/META-INF/persistence.xml** ファイルを変更して、JBoss BPM Suite プロセスエンジンが新しいデータベースを使用できるようにします。

以下のコードは、**persistence.xml** の元のデータソース情報を示しています。

```
<jta-data-source>java:comp/env/jdbc/jbpm</jta-data-source>
```

この値を、以前に定義したデータソースに変更します。

```
<jta-data-source>java:comp/env/jdbc/myDatasource</jta-data-source>
```

3. これで JBoss Web Server を起動し、Business Central にログインできるようになります。

- a. **\$TOMCAT\_HOME/bin** ディレクトリー内の **startup.sh** を実行します。

```
./startup.sh
```

- b. Web ブラウザーで <http://localhost:8080/business-central> にアクセスします。

- c. 以前に **tomcat-users.xml** ファイルで定義したユーザー名とパスワードを使ってログインします。

### 2.2.2.3. Red Hat JBoss Web Server 2.0 (Tomcat 7) への Dashbuilder の設定



## 注記

Red Hat JBoss Web Server 上で Dashbuilder を設定する前に、Business Central が「[Red Hat JBoss Web Server 2.1 \(Tomcat 7\) 用の Business Central の設定](#)」の説明にあるように正常にインストールされ、起動することを確認してください。Dashbuilder は履歴ログデータベース表を必要とし、これを提供できるのは Business Central のみであるためです。以下のステップを実行する前にこれらの表がデータベース内に存在しない場合は、初期化エラーが発生する可能性があります。

Red Hat JBoss Web Server 上で Dashbuilder を設定するには、以下の手順に従います。

1. `$TOMCAT_DIR/conf/tomcat-users.xml` でユーザーとロールを定義します。Dashbuilder ではユーザーが `admin` や `analyst` として指定されたロールを持つ必要があることに注意してください。これらのユーザーを既に Business Central で定義している場合は、再度定義する必要はありません。
2. `$TOMCAT_DIR/conf/server.xml` ファイルの以下の行をコメント解除して、Dashbuilder と Business Central の間のシングルサインオンを有効にします。

```
<Valve className="org.apache.catalina.authenticator.SingleSignOn" />
```

3. Business Central の設定と同様に、デフォルトおよび統合された H2 以外のデータベースを使用している場合は、永続性を設定する必要があります。

ここでは、bitronix の `resources.properties` ファイルの `uniqueName=jdbc/jbpm` で定義された JNDI 名 `jdbc/dashbuilderDS` を使ってデータソースを設定します。

- a. `dashbuilder/META-INF/context.xml` 内の `<Resource>` 要素のデータソース JNDI 名を置き換えます。 `uniqueName` 属性は `resources.properties` で設定された `resource.ds1.uniqueName` プロパティになります。

```
<Resource name="jdbc/dashbuilderDS" uniqueName="jdbc/jbpm"
  auth="Container" removeAbandoned="true"
  factory="bitronix.tm.resource.ResourceObjectFactory"
  type="javax.sql.DataSource"/>
```



## 注記

データベースによっては、ここで他のプロパティを定義する必要があることもあります。たとえば、Oracle 環境ではこのエントリは次のようになります。

```
<Resource name="jdbc/jbpm" uniqueName="jdbc/jbpm"
  auth="Container" removeAbandoned="true"
  factory="bitronix.tm.resource.ResourceObjectFactory"
  type="javax.sql.DataSource" username="username"
  password="password"
  driverClassName="oracle.jdbc.xa.client.OracleXADataSource"
  url="jdbc:oracle:thin:YOUR-URL:1521:YOUR-DB"
  maxActive="8" />
```

- b. `dashbuilder/WEB-INF/web.xml` で、`<res-ref-name>` 要素のデータソース JNDI 名に使用するデータソース名を追加します。

```
<resource-ref>
  <description>Dashboard Builder Datasource</description>
  <res-ref-name>jdbc/dashbuilderDS</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

- c. **dashbuilder/META-INF/context.xml** でトランザクションファクトリーを定義します。

```
<Transaction
  factory="bitronix.tm.BitronixUserTransactionObjectFactory"/>
```

- d. **<session-factory>** 要素の **dashbuilder/WEB-INF/etc/hibernate.cfg.xml** にあるデータソース JNDI 名を更新します。

```
<property
  name="connection.datasource">java:/comp/env/jdbc/dashbuilderDS</p
roperty>
```

4. Java Web サーバーを再起動し、これらの変更を適用します。再起動したら、**Business Central** 内からアクセスするか、直接 **http://localhost:8080/dashbuilder** にアクセスして **Dashbuilder** を使用します。

## 2.3. サーバーの起動

JBoss EAP 6 のバンドルインストールを使用して Red Hat JBoss BPM Suite をインストールした場合、2 通りのモードのいずれかでサーバーを起動できます。



### 注記

Red Hat Java Web Server に汎用デプロイ可能バージョンを使用して JBoss BPM Suite をインストールした場合は、ダウンロードおよびインストールの手順にサーバー起動の手順が含まれているため、以下の説明に従う必要はありません。

Red Hat JBoss BPM Suite に同梱されているデフォルトの起動スクリプト **standalone.sh** は、パフォーマンス重視の最適化がされています。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、**\$SERVER\_HOME/bin/** ディレクトリーへ移動します。
2. Unix 環境では以下を実行します。

```
./standalone.sh
```

Windows 環境では以下を実行します。

```
./standalone.bat
```

Red Hat JBoss BPM Suite には、セキュリティー重視で最適化されている **standalone-secure.sh** スクリプトも含まれています。このスクリプトでは、既知のセキュリティー脆弱性を保護するセキュリティーポリシーがデフォルトで適用されます。



## 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することに注意してください。セキュリティーとパフォーマンスのバランスは、個々の状況を判断して決定する必要があります。「[Java Security Manager とパフォーマンス管理](#)」を参照してください。

このスクリプトでサーバーをセキュアモードで実行する場合は、以下の手順に従います。

1. コマンドラインで、**\$SERVER\_HOME/bin/** ディレクトリーへ移動します。
2. Unix 環境では以下を実行します。

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

Windows 環境では以下を実行します。

```
./standalone-secure.bat
```



## 注記

インストーラーを使用して **JBoss BPM Suite** をインストールした場合は、セキュリティーポリシーを適用するオプションはインストール時に提供されています。インストーラーは別個の **standalone-secure.sh** スクリプトは提供しません。



## 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。

## 2.4. JAVA SECURITY MANAGER とパフォーマンス管理

前述のとおり、Red Hat JBoss BPM Suite 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 BPM Suite アプリケーションをデプロイする場合は、環境およびパフォーマンスマーカに留意する必要があります。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 BPM Suite アプリケーションをデプロイしてください。

- パフォーマンスが重要な高負荷の環境では、他のシステムで開発され、適切に評価されたアプリケーションのみをデプロイすることが推奨されます。また、このようなシステムでは、Analyst のロールを持つユーザーを作成しないことが推奨されます。このような対策が取られた場合、これらのシステムでは JSM を無効にしても安全であることから、パフォーマンスが低下しません。

- 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ーを強化するため、JSM を有効にすることが推奨されます。

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することは安全でないため、推奨されません。

## 第3章 BUSINESS CENTRAL へのログイン

サーバーが正常に起動したら、Business Central へログインします。

1. 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。
2. 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 プロジェクト

Red Hat JBoss BPM Suite の基本機能を示すために、本章では **Hello World** ビジネスプロセスを使用して **Hello World** プロジェクトをセットアップする方法について説明します。このビジネスプロセスは、**Hello World!** メッセージを表示し、実行を完了します。完全な **when-then** ルール機能を使用する他の例については、『Red Hat JBoss BPM Suite Development Guide』の [Getting Started with Rules and Facts](#) を参照してください。

以下の手順を実行します。

1. **Artifact** リポジトリに **Hello World** リポジトリを作成します。
2. **HelloWorld** ビジネスプロセス定義で **Hello World** プロジェクトを作成します。
3. グラフィカルなプロセスデザイナーツールを使用して、ビジネスプロセスで **Hello World** プロセスロジックをモデル化します。
4. ローカルで実行されている実行エンジンにプロジェクトを構築し、デプロイします。
5. **Hello World** プロセスを実行します。
6. プロセスの実行を監視します。

### 4.1. リポジトリ構造の作成

すべてのビジネス資産は、組織単位 (**Organizational Unit**) にあるリポジトリに存在します。組織単位は **Artifact** リポジトリのディレクトリーです。デフォルトでは、**Artifact** リポジトリには組織単位が含まれていません。そのため、ビジネスプロセスなどのビジネス資産を作成するには、**Artifact** リポジトリに組織単位を作成し、リポジトリ (**Git**) を作成する必要があります。作成後、リポジトリにプロジェクトを追加できます。プロジェクトには、コンテンツを論理「ディレクトリー」構造にすることが可能な任意のパッケージ構造を含めることができます。ビジネス資産は任意のパッケージに追加できます (**Artifact** リポジトリの詳細は、『Red Hat JBoss BPM Suite User Guide』を参照してください)。

リポジトリ構造を作成するには、以下の手順に従います。

1. Web ブラウザーで **Business Central** を開き、**admin** ロールを持つユーザー (たとえば、**helloworlduser**) としてログインします。
2. アーティファクトリポジトリに組織単位を作成します。
  - a. **オーサリング** → **管理** に移動します。
  - b. パースペクティブメニューで **組織単位** → **Manage Organizational Units** の順に選択します。
  - c. 表示された **組織単位マネージャー** ビューで **追加** をクリックします。

表示された **Add New Organizational Unit (新しい組織単位の追加)** ダイアログボックスで以下のようにユニットプロパティを定義し、**Ok** をクリックします。

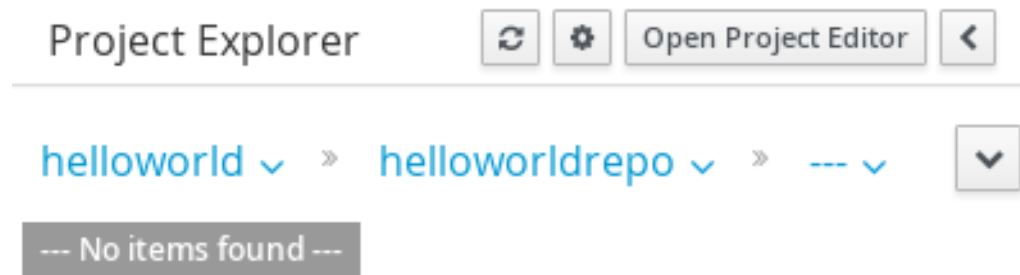
- 名前: **helloworld**
- 所有者: **helloworlduser**

3. **helloworld** 組織単位に新しいリポジトリを作成します。



- a. オーサリング → 管理 に移動します。
  - b. パースペクティブメニューでリポジトリ → 新規リポジトリ の順に選択します。
  - c. 表示された **New Repository** ダイアログボックスで以下のようにリポジトリプロパティを定義します。
    - リポジトリ名: **helloworldrepo**
    - 組織単位: **helloworld**
  - d. **Finish** をクリックします。
4. オーサリング → プロジェクトオーサリング に移動します。
  5. 組織単位ドロップダウンボックスの **Project Explorer** で **helloworld** を選択し、リポジトリドロップダウンボックスで **helloworldrepo** を選択します。

図4.1 Project Explorer の **helloworld** 組織グループで **helloworldrepo** リポジトリを選択



6. **helloworld** リポジトリでプロジェクトを作成します。
  - a. パースペクティブメニューで **新しいアイテム** → **プロジェクト** に移動します。
  - b. 表示された **New Project** ウィザードで以下のようにプロジェクトプロパティを定義します。
    - プロジェクト名: **HelloWorld**
    - グループ ID: **org.brms**
    - アーティファクト ID: **HelloWorld**
    - バージョン ID: **1.0**
  - c. **Finish** をクリックします。

## 第5章 HELLO WORLD プロセス

本章では、継続して Hello World の例を取り上げます。エンドツーエンドのビジネスプロセスを作成し、基本的な Hello World プロセスの作成について説明します。

### 5.1. ビジネスプロセスの作成

新しいビジネスプロセス定義を作成するには、以下の手順に従います。

1. プロジェクトオーサリングパースペクティブを表示します (オーサリング→プロジェクトオーサリング)。
2. 左側の Project Explorer ビューで **helloworld** 組織単位、**helloworldrepo** リポジトリ、および **HelloWorld** プロジェクトを選択します。
3. パースペクティブメニューで **新しいアイテム→ビジネスプロセス** とクリックし、プロセス定義の詳細を定義します。
  - **HelloWorld** をビジネスプロセス名として入力します。
  - **org.bpms.helloworld** パッケージを選択します。これにより、アーティファクトリポジトリ内でビジネスプロセス定義を作成する場所が定義されます。



#### パッケージの選択



必ず **org.bpms.helloworld** パッケージを選択してください。誤ったパッケージを選択するとデプロイメントに失敗します。

4. **OK** をクリックします。

作成されたプロセス定義のキャンバスを持つプロセスデザイナーが開かれます。

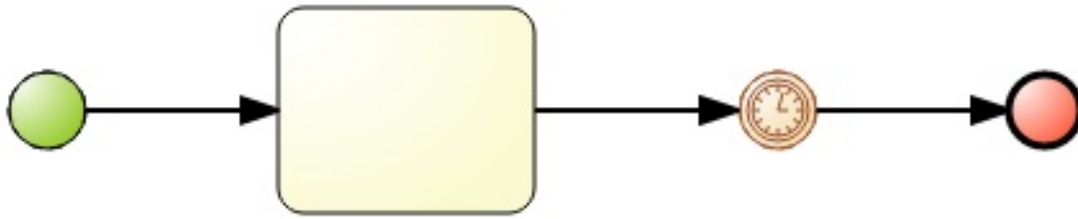
### 5.2. ビジネスプロセスのモデル化

ビジネスプロセス定義を作成した後、ビジネスプロセスデザイナーでビジネスプロセスを設計できます。ビジネスプロセスデザイナーは右側のタブに開かれます。プロジェクトエクスプローラーでタブを閉じてしまった場合は、**Business Processes** 下の **HelloWorld** をクリックして、ビジネスプロセスデザイナーを再度開きます。正しいパッケージ (**org.bpms.helloworld**) を選択するようにしてください。プロセスのコンテンツを設定するには、以下の手順に従います。

1. プロセス要素を持つ **Object Library** パレットを展開します。ビジネスプロセスデザイナータブの左上にある二重矢印のボタン (  ) をクリックします。
2. **Start Event** 要素がキャンバスに表示されます。
3. **Start Event** 要素をクリックします。クイックリンク項目がノードの近くに表示されます。  
**Task** アイコン (  ) をクリックし、外向きのシーケンスフローおよび **Start Event** へ紐付けされた **Task** 要素を作成します。
4. **Timer Event** を作成します。パレットから **Timer (Catching Intermediate Events)** の下にあるものをドラッグアンドドロップし、クイックリンクメニューを使用して **Task** を **Timer** 要素に紐付けします。
5. **Timer Event** 要素へ紐付けされる **End Event** 要素を作成します (クイックリンク機能を使用する

か、パレットからドラッグします)。

図5.1 HelloWorld プロセスのドラフト



### 5.3. 要素プロパティの定義

ここで、要素プロパティを定義する必要があります。

1. 右上隅にある二重矢印 (⏪) をクリックし、**Properties** タブを展開します。
2. キャンバスのどこかをクリックします。**Properties** タブに **HelloWorld** プロセスのプロパティが表示されます。必要なプロパティは事前定義されていますが、必要な場合は値を変更できます。
3. **Start** 要素と **End** 要素はオプションのプロパティのみを持ちますが、**Task** 要素ではタイプが必要となり、タイプ固有のプロパティを定義する必要があります。**Task** 要素をクリックし、必要なプロパティを定義します。

- 名前: **ScriptTask**
- タスクタイプ: **Script**
- スクリプト言語: **java**
- スクリプト: **System.out.println("Hello World!");**

**OK** をクリックします。

これで実行時にスクリプトを実行するタスクが定義されました。スクリプトは **Java** で定義され、**System.out.println("Hello World!")** メソッドを実行します。このメソッドは、サーバーの標準出力に **Hello World!** を書き込みます。デフォルトでは、サーバーの標準出力はサーバーが起動したコンソールになります。

4. 実行を検証できるようにするために、プロセスが実行時に待機するようタイマーイベントのプロパティを定義する必要があります。タイマー要素をクリックし、実行を待機する時間を定義します。
  - **Time Duration** を **1m** に設定し、**Ok** をクリックします。

この結果、プロセスはスクリプトタスクの実行後にタイマーイベントで1分間待機します。

5. 同じタブでタイマーイベントに名前を付け、プロセスを保存します。ビジネスプロセスエディターの上で **Save** をクリックします。

6. **Save this item** プロンプトが表示されます。**Check in comment** を入力します。



### 注記

**Check in comment** は、変更などの簡単な説明であり、アセットの保存時には毎回入力する必要があります。

7. プロセスデザイナーのツールバーにある  をクリックし、定義されたプロセスが有効であることを確認します。1つ以上の検証エラーがある要素は外側がオレンジ色になります。検証機能については、『Red Hat JBoss BPM Suite User Guide』を参照してください。

プロセス要素やそれらのプロパティに関する詳細は、『Red Hat JBoss BPM Suite User Guide』を参照してください。

## 5.4. ビルドとデプロイ

**Hello World** プロセスをインスタンス化する前に、プロジェクト全体をビルドし、実行サーバーへデプロイする必要があります。

1. **Business Central** のメインメニューで **オーサリング** → **プロジェクトオーサリング** に移動します。
2. **Project Explorer** で **Hello World** プロジェクトを見つけます。
3. プロジェクトエディターでプロジェクトを開きます。**Project Explorer** で **Open Project Editor** をクリックします。
4. **Project Screen** に正しいプロジェクト詳細が表示されていることを確認し、**Project Screen** ビューの右上隅にある **Build** → **Build & Deploy** を選択します。

図5.2 プロジェクトエディターでの HelloWorld プロジェクトプロパティ

Project: [HelloWorld:org.bpms:1.1] ▾ Save Delete Rename Copy Build ▾ ↕ ✕

Project Settings: Project General Settings ▾

### Project General Settings

Project Name: HelloWorld

Project Description: *Insert a project description for documentation purposes ...*

### Group artifact version

Group ID ⓘ: org.bpms  
Example: com.myorganization.myprojects


Artifact ID ⓘ: HelloWorld  
Example: MyProject

Version ⓘ: 1.1  
Example: 1.0.0

プロジェクトの構築および実行サーバーへのデプロイメントが完了し、インスタンス化が可能になったことを伝える緑色の通知が画面上部に表示されます。

## 5.5. ビジネスプロセスのインスタンス化

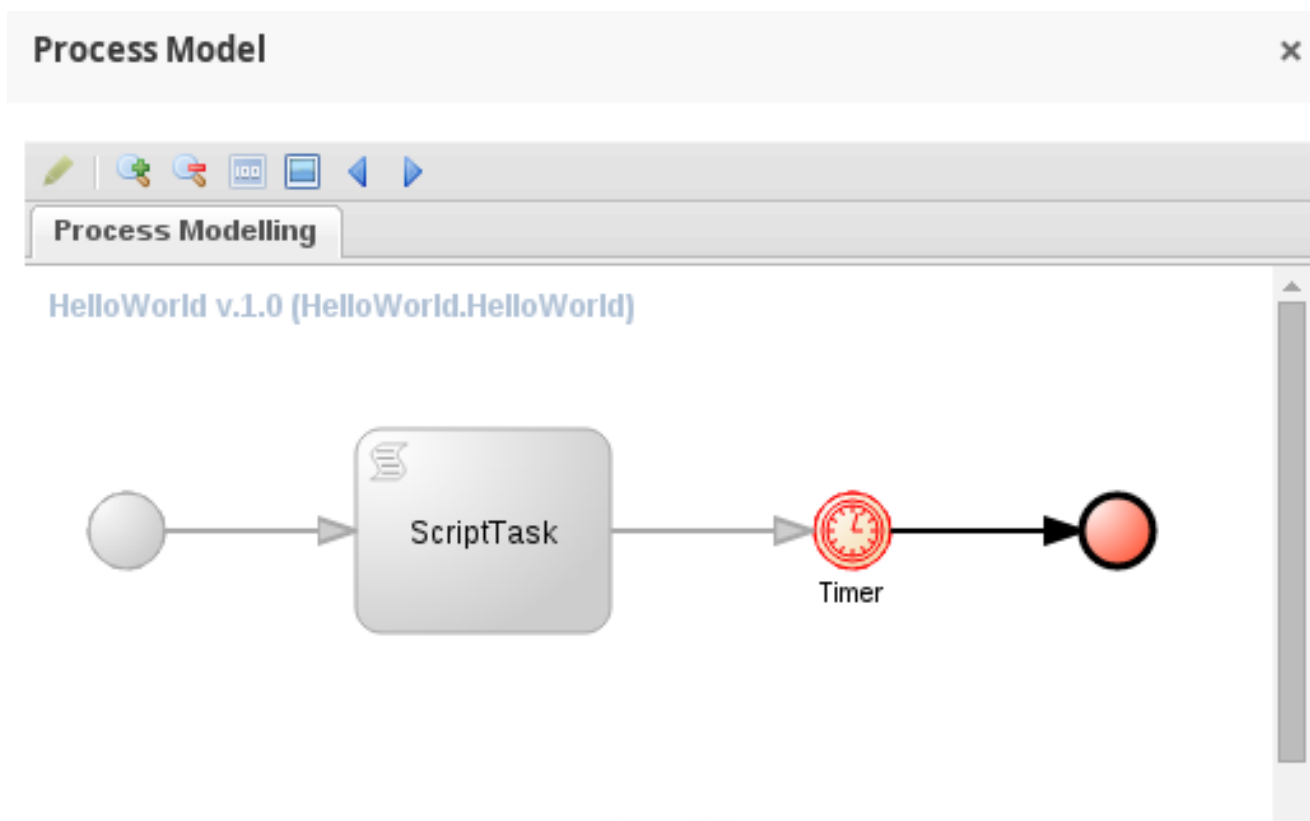
**HelloWorld** プロセスのインスタンスを作成する(つまり、ビジネスプロセスを実行する)には、以下の手順に従います。

1. メインメニューで **プロセス管理** → **プロセス定義** をクリックします。
2. 表示された **Process Definitions (プロセス定義)** タブで **HelloWorld** プロセスを見つけます。 **Refresh (更新)** ボタンをクリックしてリストにデプロイメントを表示する必要があることがあります。
3. プロセス定義エントリーの横にある **Start** (  ) をクリックし、ダイアログボックスで **Submit** をクリックして、プロセスのインスタンス化を確認します。

現在ログインしているユーザーがプロセス所有者となってプロセスがインスタンス化され、プロセスフォームが表示されます。定義されている場合は、プロセスのインスタンス化で、フォームを使ってユーザーからの入力を要求できます。詳細については、『Red Hat JBoss BPM Suite User Guide』を参照してください。

起動されたプロセスインスタンスの詳細を示す **Process Instance Details** ビューが表示されます。 **Hello World!** メッセージが標準出力に 2 回出力されます。通常、標準出力はサーバーが起動したターミナルエミュレーターになります。その後、プロセスインスタンスがタイマーイベントで待機します。 **Options** → **Process Model** をクリックして、現在の実行状態を確認します。

図5.3 HelloWorldのリアルタイム実行図: タイマーイベントの実行








## 5.6. ビジネスプロセスの中止

この時点で、**HelloWorld** プロセスのインスタンスは実行サーバー上で実行され、**Process Instances (プロセスインスタンス)** ビューで確認できます。このビューを表示するには、**Process Management (プロセス管理)** → **Process Instances (プロセスインスタンス)** に移動します。


このビューでは、選択されたプロセスの行をクリックして**インスタンスの詳細**を表示できます。また、プロセスインスタンスへのシグナル送信 (**Signal**) や、その中止 (**Abort**) など、基本的な管理アクションを実行することもできます。

ビジネスプロセスがインスタンス化されると、プロセスインスタンスはタイマーイベントを待機します。インスタンスを中止するには、インスタンス情報がある行で **Abort** をクリックします。インスタンスが **Active** プロセスインスタンスのリストで表示されなくなり、**Aborted** リストに表示されます。

#### 図5.4 中止されたプロセスインスタンスのリスト

Process Instances New Process Instance     

Active **x** Completed **x** **Aborted **x**** +

Filter Aborted Bulk Actions 

Select	Id	Name	Description	Version	Actions
<input type="checkbox"/>	3	HelloWorld	HelloWorld	1.0	

## 第6章 HELLO WORLD ビジネスルール

Red Hat JBoss BPM Suite には Red Hat JBoss BRMS が統合されているため、本章ではビジネスルールの仕組みと JBoss BPM Suite への統合方法について説明します。

特定のビジネスルールグループのビジネスルールを実行(チェック)する新しいビジネスルールタスクを **HelloWorld** プロセスに追加します。

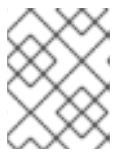
ここでは非常に単純な統合ケースを取り上げますが、実稼働環境では決定表などの高度な概念や技術が必要になることがあります。ビジネスルールや BRMS の詳細については、『Red Hat JBoss BRMS User Guide』を参照してください。

### 6.1. ビジネスルールの作成

ビジネスルールは **when-then** ステートメントで定義され、**true** の場合に何かが発生します(例: 18 歳以上の場合に特定のコンテンツにアクセスできる)。ビジネスルールは **DRL** ファイルに保存されます。

**HelloWorld** プロジェクトの **helloworldrule** ビジネスルールを定義するには、以下の手順に従います。

1. **Project Explorer** で **helloworld** 組織単位、**helloworldrepo** リポジトリ、**HelloWorld** プロジェクトおよび **default** パッケージを選択します。



#### パッケージの選択

必ず **default** パッケージを選択してください。誤ったパッケージを選択するとデプロイメントに失敗します。

2. **DRL** ファイルを作成します。
  - a. パースペクティブメニューで **New Item** → **DRL file** と選択します。
  - b. **Create new DRL file** ダイアログボックスで、リソース名を **helloworldrule** と定義し、パッケージが **org.bpms.helloworld** に設定されていることを確認します。
  - c. **OK** をクリックします。
3. **DRL** エディターで表示された **helloworldrule.drl** ファイルで、ルールを定義します。

```
rule "helloworldrule"
ruleflow-group "helloworldgroup"
when
then
    System.out.println("Hello World!");
end
```

このルールは **when** 条件を定義しません。つまり、チェックされる(実行するため呼び出される)と常に **true** になり、**Hello World!** フレーズが出力されます。

4. **Save** をクリックします。
5. **Save this item** プロンプトが表示されます。**Check in comment** を入力し、**Save** をクリックします。

## 6.2. ビジネスルールタスクの追加

ビジネスルールタスクは、特定のルールフローグループに属するルールをチェックするタスクのことです。


ビジネスルールタスクをプロセスに追加するには、以下の手順に従います。

1. ビジネスプロセスをプロセスデザイナーで開きます。**Project Explorer** で **HelloWorld** プロジェクトと **org.bpmns.helloworld** パッケージを選択し、**BUSINESS PROCESSES** をクリックして、**HelloWorld** プロセスを選択します。



### パッケージの選択

必ず **org.bpmns.helloworld** パッケージを選択してください。誤ったパッケージを選択すると、デプロイメントに失敗します。

2. ビジネスプロセスデザイナーで表示された **HelloWorld** プロセスで、プロセス要素とともに **Object Library** パレットを展開します。ビジネスプロセスデザイナータブの左上隅にある二重矢印のボタン (  ) をクリックします。
3. **Tasks** メニューを展開し、ビジネスルールタスクを右側のキャンバスへドラッグアンドドロップします。

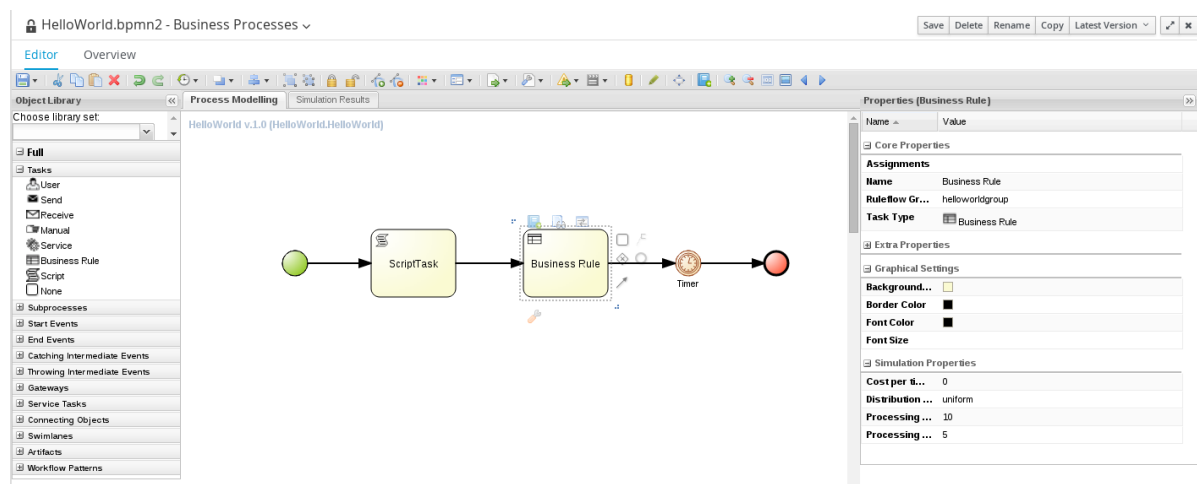
フローの紐付けを調整し、タスクをワークフローへ統合します。

4. ビジネスルールタスクを選択し、そのプロパティを **Properties** パネルで定義します。

- 名前: **BusinessRule**
- ルールフローグループ: **helloworldgroup**

Ruleflow Group プロパティは、タスクの実行時にチェックされるルールのグループを定義します。この例では、**helloworldrule.dr1** ファイルで定義された **helloworldrule** ルールのみが、**helloworldgroup** グループに存在します。

図6.1 HelloWorld プロセスのビジネスルールタスクおよびそのプロパティ



5. **Save** ボタンをクリックし、変更を保存します。

## 6.3. ビルドとデプロイ



プロジェクト全体を構築し、実行サーバーへデプロイします。

1. **Business Central** のメインメニューで **オーサリング** → **プロジェクトオーサリング** に移動します。
2. **Project Explorer** で **HelloWorld** プロジェクトを見つけます。
3. プロジェクトエディターでプロジェクトを開きます。**Project Explorer** で **Open Project Editor** をクリックします。
4. **Project Screen** に正しいプロジェクト詳細が表示されていることを確認し、**Project Screen** ビューの右上隅にある **Build** → **Build & Deploy** をクリックします。

図6.2 プロジェクトエディターでの HelloWorld プロジェクトプロパティー

Project: [HelloWorld:org.bpms:1.1] v

Save Delete Rename Copy Build ↕ ✕

Project Settings: Project General Settings v

Project General Settings

Project Name: HelloWorld

Project Description: Insert a project description for documentation purposes ...

Group artifact version

Group ID ⓘ: org.bpms  
Example: com.myorganization.myprojects

Artifact ID ⓘ: HelloWorld  
Example: MyProject

Version ⓘ: 1.1  
Example: 1.0.0

プロジェクトのビルドおよび実行サーバーへのデプロイメントが完了し、インスタンス化が可能になっ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

以前のバージョンの **HelloWorld** デプロイメントは、ビジネスルールタスクがある新バージョンに置き換えられます。両方のデプロイメントを保持したい場合は、プロジェクトエディターでプロジェクトのバージョン番号を変更します。

## 6.4. ビジネスプロセスのインスタンス化

**HelloWorld** プロセスのインスタンスを作成する（つまり、ビジネスプロセスを実行）には、以下の手順に従います。

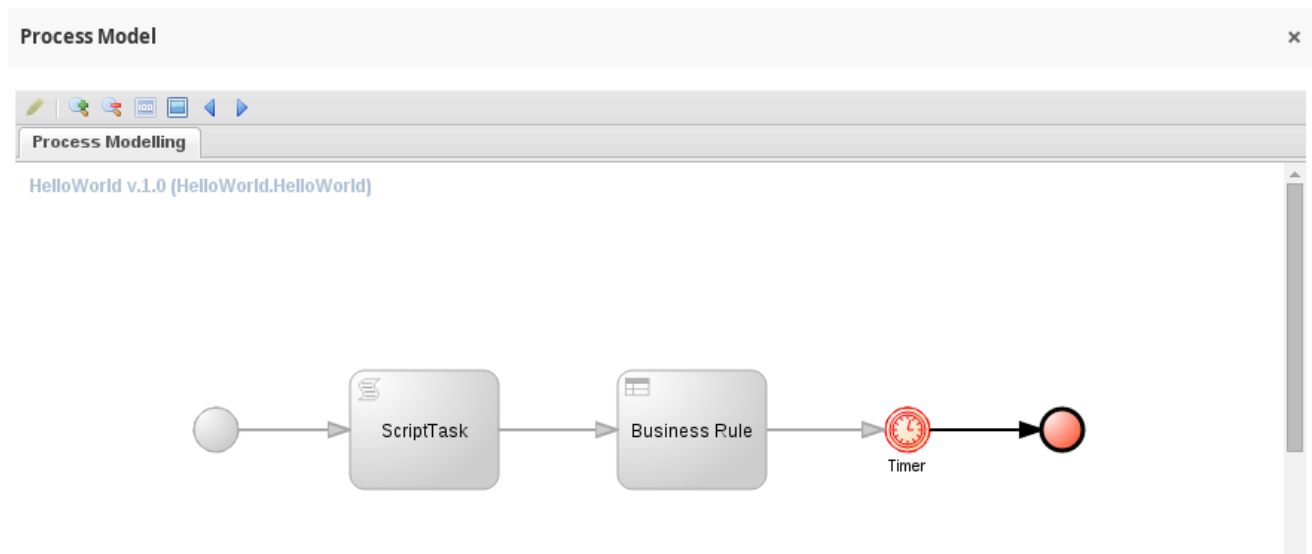
1. メインメニューで **プロセス管理** → **プロセス定義** をクリックします。
2. 表示された **Process Definitions (プロセス定義)** タブで **HelloWorld** を見つけます。**Refresh (更新)** をクリックしてリストにデプロイメントを表示する必要があることがあります。

3. プロセス定義エントリーの横にある **Start**() をクリックし、ダイアログボックスで **Submit** をクリックして、プロセスのインスタンス化を確認します。

現在ログインしているユーザーがプロセス所有者となってプロセスがインスタンス化され、プロセスフォームが表示されます (定義されている場合、プロセスのインスタンス化で、フォームを使ってユーザーからの入力を要求できます。詳細は『Red Hat JBoss BPM Suite User Guide』を参照してください)。

起動されたプロセスインスタンスの詳細を示す **Process Instance Details** ビューが表示されます。 **Hello World!** メッセージが標準出力に 2 回出力されます。通常、標準出力はサーバーが起動したターミナルエミュレーターになります。その後、プロセスインスタンスがタイマーイベントで待機します。 **Options** → **Process Model** を選択して、現在の実行状態を確認します。

図6.3 HelloWorld のリアルタイム実行図: タイマーイベントの実行



## 第7章 BAM

### 7.1. RED HAT JBOSS BPM SUITE DASHBUILDER へのアクセス

Dashbuilder は、Red Hat JBoss BPM Suite に含まれるビジネスアクティビティ監視用の Web ベース ユーザーインターフェースです。Business Central から Dashbuilder にアクセスするには、**Dashboards (ダッシュボード) → Process & Task Dashboards (プロセス & タスクダッシュボード)** と選択します。

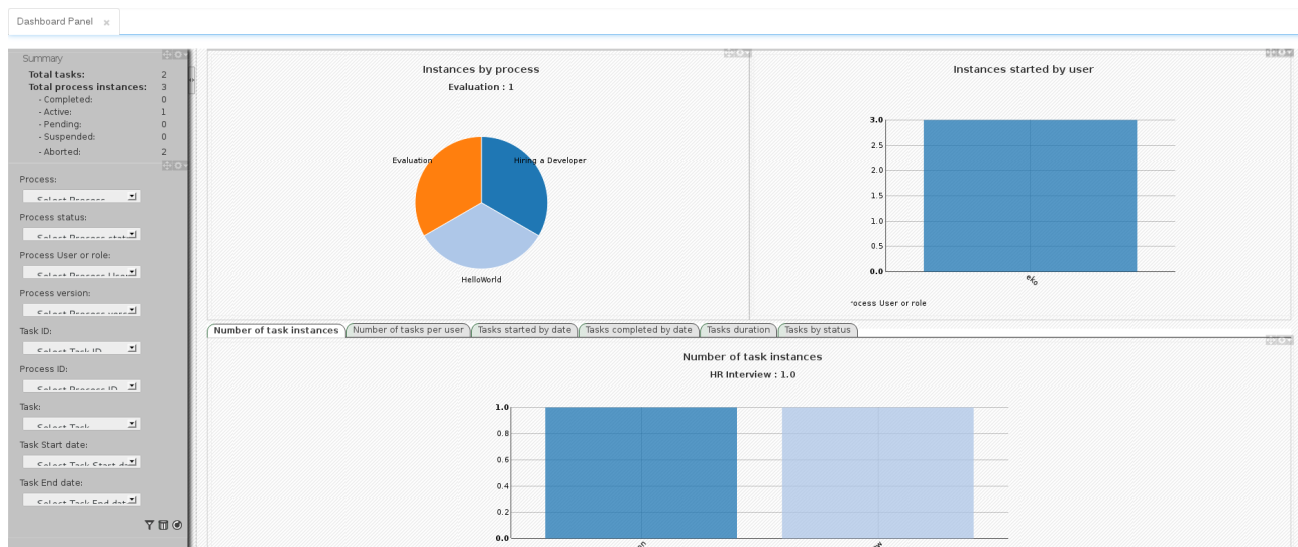
表示されるダッシュボードには、左側で選択されたランタイムデータの統計が表示されます。Dashbuilder で独自のダッシュボードを作成するには、**Dashboards (ダッシュボード) → Business Dashboards (ビジネスダッシュボード)** とクリックして Dashbuilder を表示します。

### 7.2. インスタンスの監視

Dashbuilder は、実行エンジン上のランタイムデータ (プロセスインスタンスおよびタスク) の状態を監視できる特別なダッシュボードを提供します。

このデータを表示するには、**Dashboards (ダッシュボード) → Process & Task Dashboard (プロセス & タスクダッシュボード)** と選択します。

図7.1 プロセスおよびタスクダッシュボード



左側のパネルで統計を表示するエンティティを選択すると、右側の表とデータが更新されます。

## 第8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。最新版は、Red Hat カスタマーポータル <https://access.redhat.com> から入手してください。JBoss Developer Studio は Red Hat JBoss BRMS および Red Hat JBoss BPM Suite 用のツールと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ベースになっています。そのため、JBoss BRMS プラグインは Drools プラグインと呼ばれ、JBoss BPM Suite プラグインは jBPM プラグインと呼ばれます。

インストールおよび設定の手順については、『Red Hat JBoss Developer Studio』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、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。

### 8.1. JBOSS CENTRAL

JBoss Developer Studio 8.0 が先に起動している場合、JBoss Central はワークベンチのメインウィンドウに表示されます。**Start from scratch** 下にあるメニューオプションを選択すると、JBoss Central から新しいプロジェクトを作成できます。サンプルプロジェクトを起動するには、**Start from a sample** 下のリンクを選択します。

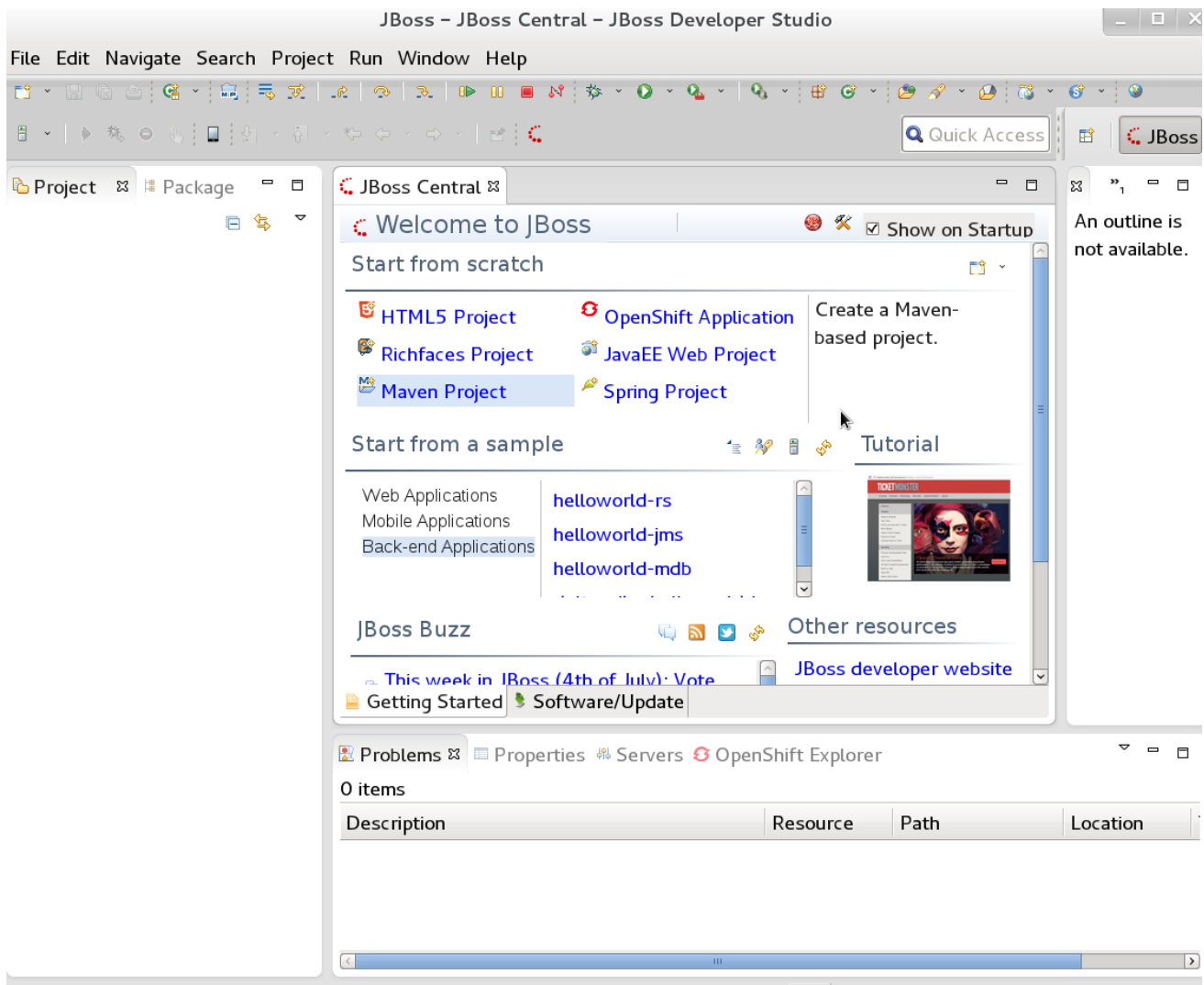


### 注記

GWT3 の問題のため、JBDS では GWT3 を使用しないように設定してください。以下のよう、システムプロパティを **0** に設定することで、これが可能になります。

```
# Don't use GTK 3
export SWT_GTK3=0
```

## 8.1 JBoss Central



[D]

## 8.2. JBOSS DEVELOPER STUDIO プラグインのインストール

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

### 手順8.1 JBoss Developer Studio の Drools および jBPM プラグインのインストール

1. JBoss Developer Studio を起動します。
2. **Help** → **Install New Software** と選択します。
3. **Add** をクリックして **Add Repository** メニューに入ります。
4. **Name** フィールドの横でソフトウェアサイトの名前を指定し、**Location** フィールドに URL <https://devstudio.jboss.com/updates/8.0/integration-stack/> 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、再度 **Next** をクリックします。

7. ライセンスの内容を確認し、該当のラジオボタンを押して内容に同意して、**Finish** をクリックします。
8. プラグインのインストールが完了したら、JBoss Developer Studio を再起動します。

### 8.3. 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、ビジネス資産のコンパイルおよび実行に必要なライブラリを提供します。

#### 手順8.2 JBoss BRMS ランタイムの設定

1. **jboss-brms-VERSION-engine.zip** アーカイブにあるランタイム jar ファイルを展開します。これは、[Red Hat Customer Portal](#) からダウンロードできます。
2. JBoss Developer Studio メニューから **Window** を選択し、**Preferences** をクリックします。
3. **Drools** → **Installed Drools Runtimes** と選択します。
4. **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、手順1で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 JBDS に登録します。
5. 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。
6. **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、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す。

### 8.4. JBOSS BPM SUITE ランタイムの設定

Red Hat JBoss Developer Studio で JBoss BPM Suite 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す。

JBoss BPM Suite の汎用デプロイ可能 zip アーカイブを [Red Hat Customer Portal](#) から以前にダウンロードした場合、ランタイムを構成する jar ファイルは **jboss-bpms-engine.zip** アーカイブにあります。

#### 手順8.3 jBPM ランタイムの設定

1. JBoss Developer Studio メニューから **Window** を選択し、**Preferences** をクリックします。
2. **jBPM** → **Installed jBPM Runtimes** と選択します。
3. **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、ランタイムのあるディレクトリーに移動します。
4. **OK** をクリックし、新しいランタイムを選択してから再度 **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、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す。

## 8.5. JBOSS BPM SUITE サーバーの設定

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

### 手順8.4 サーバーの設定

1. **Window** → **Open Perspective** → **Other** を選択して **jBPM** ビューを開き、**jBPM** を選択し、**OK** をクリックします。
2. **Window** → **Show View** → **Other...** の順に選択して **Server** → **Servers** を選択し、サーバービューを追加します。
3. サーバーパネルを右クリックし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 6.4+** と選択してサーバーを定義し、**Next** をクリックします。
5. **Browse** ボタンをクリックしてホームディレクトリーを設定します。**JBoss BPM Suite** をインストールした **JBoss EAP 6.4** のインストールディレクトリーを選択します。
6. **Name** フィールドにサーバー名を指定します。設定ファイルが設定されていることを確認してから **Finish** をクリックします。

## 8.6. GIT リポジトリーから JBOSS DEVELOPER STUDIO へのプロジェクトのインポート



### 注記

これは、JBoss Developer Studio と連携するのに必要でない追加機能です。

JBoss Developer Studio は、セントラル Git 資産リポジトリーへ接続するように設定できます。リポジトリーにはルール、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 リポジトリーをインポートするか、リモート Git リポジトリーをクローンすることができます。

### 手順8.5 リモート Git リポジトリーのクローン

1. サーバータブから Red Hat JBoss BRMS/BPM Suite サーバー (どちらか該当する方) を選択し、**Start** アイコンをクリックして起動します。
2. 同時に、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 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
```

3. JBoss Developer Studio で **File** → **Import...** と選択し、**Git** フォルダーを選択します。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。
4. リポジトリーソースに **Clone URI** を選択し、**Next** をクリックします。
5. 次のウィンドウで Git リポジトリーの詳細を入力し、**Next** をクリックします。

## 図8.2 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: ssh ▼

Port:

Authentication

User:

Password:

Store in Secure Store

?
< Back
Next >
Cancel
Finish

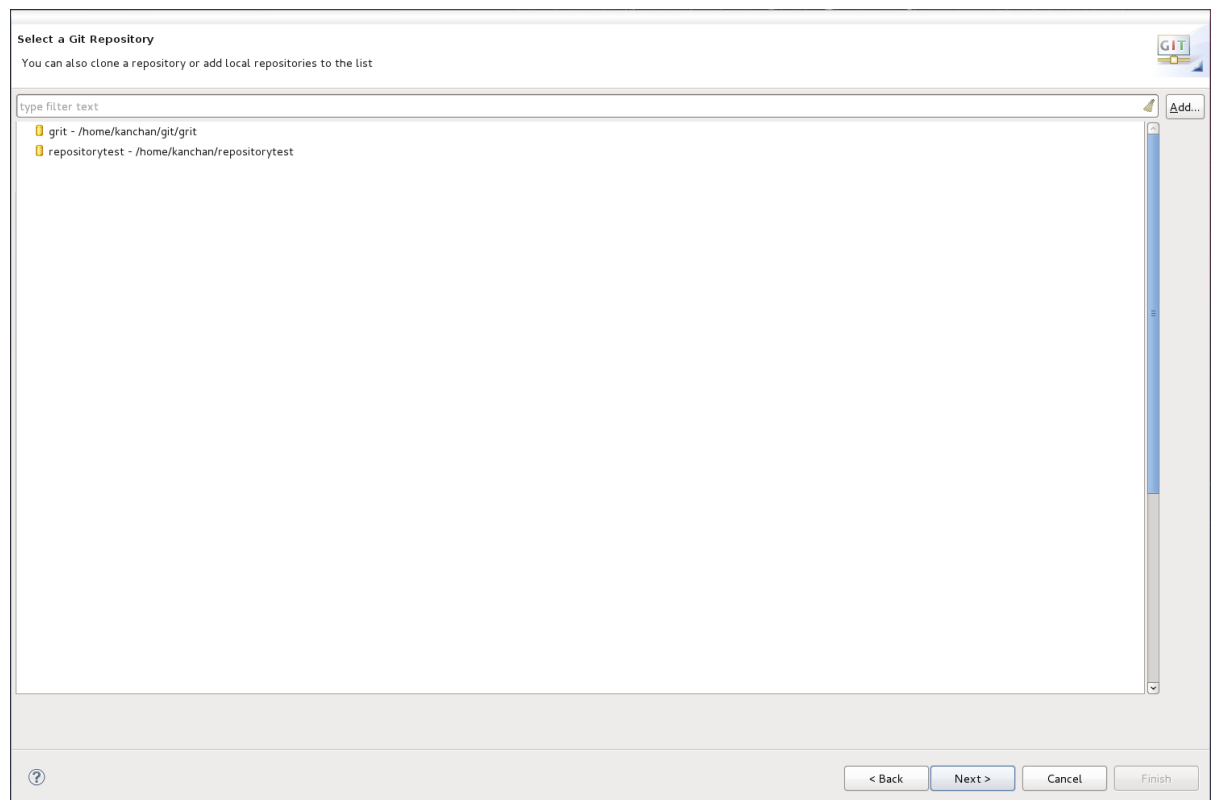
6. 次のウィンドウでインポートするブランチを選択し、**Next** をクリックします。
7. このプロジェクトのローカルストレージを定義するために、空でないディレクトリーを入力 (または選択) して設定を変更し、**Next** をクリックします。
8. 次のウィンドウでプロジェクトを一般プロジェクトとしてインポートし、**Next** をクリックします。このプロジェクトに名前を付け、**Finish** をクリックします。

## 手順8.6 ローカル Git リポジトリのインポート

1. サーバータブから Red Hat JBoss BRMS/BPM Suite サーバー (どちらか該当する方) を選択し、**Start** アイコンをクリックして起動します。
2. JBoss Developer Studio で **File** → **Import...** と選択し、**Git** フォルダーを選択します。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。
3. リポジトリソースとして **Existing local repository** を選択し、**Next** をクリックします。

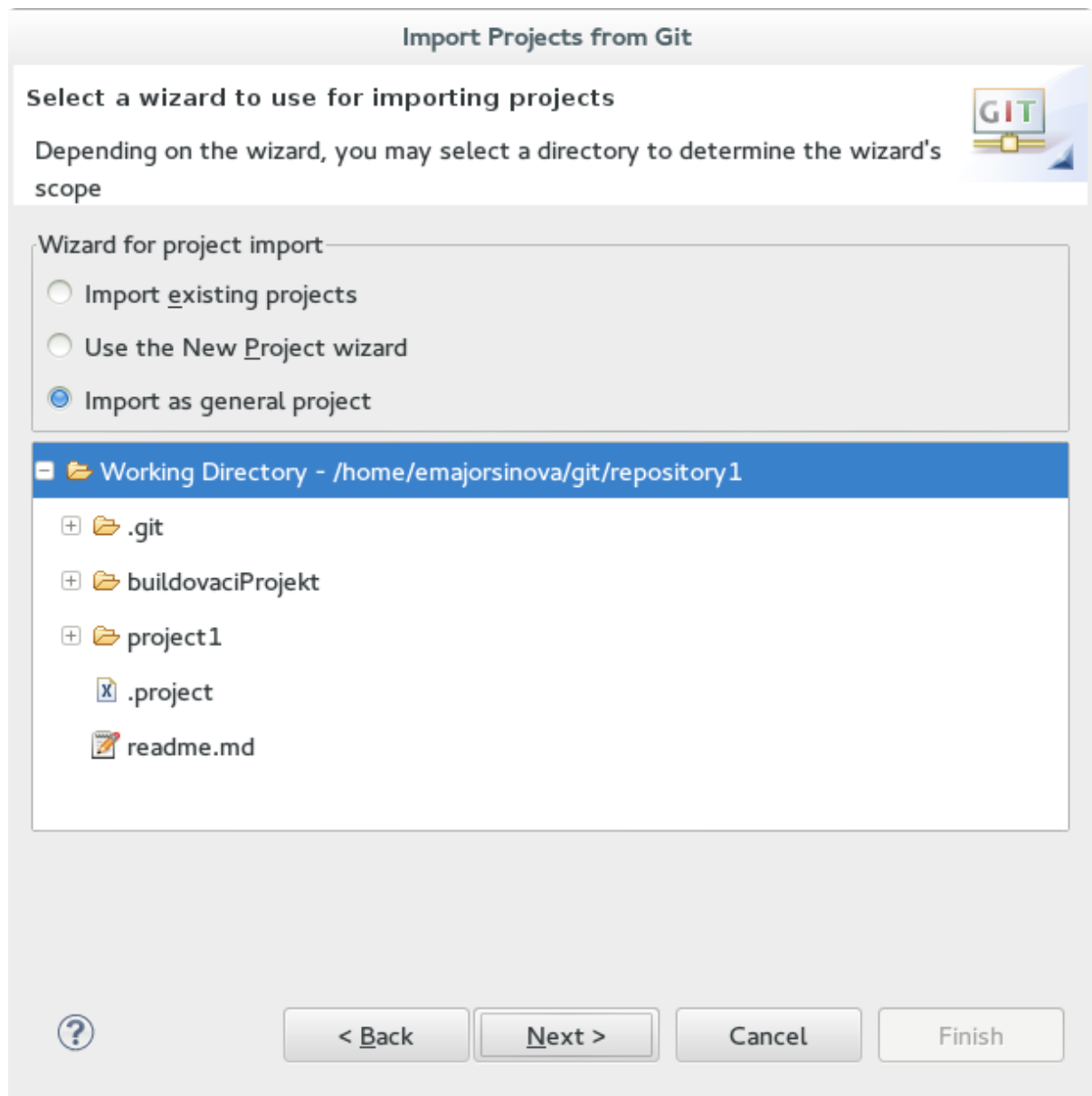


図8.3 Git リポジトリの詳細



4. 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。
5. ダイアログボックスが開かれ、**Wizard for project import group** から **Import as general project** のラジオボタンを選択し、**Next** をクリックします。プロジェクトに名前を付け、**Finish** をクリックします。

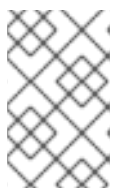
図8.4 プロジェクトインポートのウィザード



## 8.7. DROOLS プロジェクトの作成

### 手順8.7 新しい RedHat JBossDeveloperStudio プロジェクトの作成

1. メインメニューから **File** → **New** → **Project** を選択します。  
**Drools** → **Drools Project** と選択し、**Next** をクリックします。
2. プロジェクト名を **Project name:** テキストボックスに入力し、**Next** をクリックします。



#### 注記

JBossDeveloperStudio は、サンプル HelloWorld ルールファイルをプロジェクトに追加するオプションを提供します。**Next** をクリックしてこのデフォルトを許可し、以下の手順でサンプルプロジェクトをテストします。

3. **Drools** ランタイムを選択します (またはデフォルトを使用します)。

4. **Drools 6.0.x** と互換性のあるコードを選択します。**GroupID**、**ArtifactID**、および **Version** を入力し、**Finish** をクリックします。
5. プロジェクトをテストするには、主なメソッドを含む **Java** ファイルを右クリックして、**Run** → **run as** → **Java Application** の順に選択してください。

Console タブに出力が表示されます。

## 8.8. JBOSS BPM SUITE プロジェクトの作成

### 手順8.8 RedHat JBossDeveloperStudio での新しい JBoss BPM Suite プロジェクトの作成

1. メインメニューから **File** → **New** → **Project** を選択します。  
**jBPM** → **jBPM Project** と選択し、**Next** をクリックします。
2. プロジェクト名を **Project name:** テキストボックスに入力し、**Next** をクリックします。



#### SAMPLE HELLOWORLD RULE FILE

JBossDeveloperStudio は、サンプル HelloWorld ルールファイルプロジェクトに追加するオプションを提供します。**Next** をクリックしてこのデフォルトを許可し、以下の手順でサンプルプロジェクトをテストします。

3. **jBPM ランタイム** を選択します (またはデフォルトを使用します)。
4. **Generate code compatible with jBPM 6 or above** を選択し、**Finish** をクリックします。
5. プロジェクトをテストするには、主なメソッドを含む **Java** ファイルを右クリックして、**Run** → **run as** → **Java Application** の順に選択してください。

Console タブに出力が表示されます。

## 第9章 BUSINESS RESOURCE PLANNER

**Business Resource Planner** はライトウェイトの組み込み可能な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。通常の Java™ プログラマーがプランニングの問題を効率的に解決できるようにし、最適化ヒューリスティックおよびメタヒューリスティックと、大変効率的なスコア計算を組み合わせます。

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便利です。

- **従業員/患者の勤務表:** プランナーを使用して看護婦の勤務時間割を作成でき、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。

### 9.1. BUSINESS RESOURCE PLANNER のインストール

1. [Red Hat Customer Portal](#) に移動し、ユーザークレデンシャルでログインします。
2. **Downloads (ダウンロード)** → **Product Downloads (製品のダウンロード)** を選択します。
3. 開いた **Product Downloads (製品のダウンロード)** ページで、**Red Hat JBoss BPM Suite** をクリックします。
4. **Version** ドロップダウンメニューからバージョン **6.3** を選択します。
5. **Red Hat JBoss BPM Suite 6.3 Business Resource Planner** を選択し、**Download (ダウンロード)** をクリックします。

### 9.2. 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、以下のコマンドを実行します。

```
./runExamples.sh
```

Windows 環境では、以下のコマンドを実行します。

```
./runExamples.bat
```

3. 開かれた GUI アプリケーションサンプルよりサンプルを1つ選択し、任意の IDE で実行します。

## 付録A 改訂履歴

<b>改訂 6.3.0-5.1</b> 翻訳ファイルを XML ソースバージョン 6.3.0-5 と同期	<b>Fri May 27 2016</b>	<b>Red Hat Localization Services</b>
<b>改訂 6.3.0-5</b> ライブリンクでビルド	<b>Thu May 5 2016</b>	<b>Tomas Radej</b>
<b>改訂 6.3.0-4</b> ブートストラッピングのリンク	<b>Thu May 5 2016</b>	<b>Tomas Radej</b>
<b>改訂 6.3.0-3</b> すべてのブックを再ビルド	<b>Thu Apr 28 2016</b>	<b>Tomas Radej</b>
<b>改訂 6.3.0-2</b> すべてのブックを再ビルド	<b>Thu Apr 28 2016</b>	<b>Tomas Radej</b>
<b>改訂 6.3.0-1</b> JBoss BPM Suite リリース 6.3.0 の初回ビルド	<b>Thu Apr 28 2016</b>	<b>Tomas Radej</b>