



Red Hat Process Automation Manager 7.7

Red Hat JBoss Web Server への Red Hat
Process Automation Manager のインストールお
よび設定

Red Hat Process Automation Manager 7.7 Red Hat JBoss Web Server への Red Hat Process Automation Manager のインストールおよび設定

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

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

本書では、Red Hat JBoss Web Server 5.2.0 に Red Hat Process Automation Manager 7.7 をインストールする方法を説明します。

目次

| | |
|--|----|
| 前書き | 3 |
| 第1章 RED HAT PROCESS AUTOMATION MANAGER とは | 4 |
| 1.1. RED HAT PROCESS AUTOMATION MANAGER コンポーネント | 4 |
| 1.2. RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー | 5 |
| 第2章 RED HAT PROCESS AUTOMATION MANAGER インストールファイルのダウンロード | 7 |
| 第3章 RED HAT PROCESS AUTOMATION MANAGER インストーラーの使用 | 8 |
| 3.1. インタラクティブモードでのインストーラーの使用 | 8 |
| 3.2. CLI モードでのインストーラーの使用 | 11 |
| 第4章 KIE SERVER ZIP ファイルのインストールと設定 | 14 |
| 4.1. ZIP ファイルからの KIE SERVER のインストール | 14 |
| 4.2. JDBC WEB SERVER データソースの設定 | 16 |
| 第5章 キーストアを使用したパスワードセキュリティの確保 | 20 |
| 第6章 KIE SERVER インストールの確認 | 23 |
| 第7章 ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのダウンロードおよびインストール | 24 |
| 第8章 スタンドアロン BUSINESS CENTRAL の実行 | 27 |
| 8.1. BUSINESS CENTRAL のシステムプロパティ | 28 |
| 第9章 RED HAT PROCESS AUTOMATION MANAGER の MAVEN 設定およびリポジトリ | 33 |
| 9.1. プロジェクト設定ファイル (POM.XML) を使用した MAVEN の設定 | 33 |
| 9.2. MAVEN 設定ファイルの修正 | 33 |
| 9.3. RED HAT PROCESS AUTOMATION MANAGER の MAVEN 依存関係の追加 | 34 |
| 9.4. オフラインで使用する MAVEN ミラーリポジトリの用意 | 37 |
| 第10章 GIT リポジトリからのプロジェクトのインポート | 40 |
| 第11章 LDAP と SSL の統合 | 41 |
| 付録A バージョン情報 | 42 |

前書き

本書では、JBoss Web Server に Red Hat Process Automation Manager 7.7 をインストールする方法を説明します。

前提条件

- 『[Red Hat Process Automation Manager インストールの計画](#)』の内容を確認している。
- Red Hat JBoss Web Server 5.2.0 がインストールされている。Red Hat JBoss Web Server のインストールに関する情報は『[Red Hat JBoss Web Server Installation Guide](#)』を参照してください。

第1章 RED HAT PROCESS AUTOMATION MANAGER とは

Red Hat Process Automation Manager は、ビジネスの自動化アプリケーションとマイクロサービスを作成する Red Hat ミドルウェアプラットフォームです。これにより、企業のビジネスユーザーと IT ユーザーが、ビジネスプロセスおよびポリシーを文書化、シミュレート、管理、自動化、およびモニターできます。ビジネスユーザーおよび IT ユーザーがより効果的に協力できるように設計されているため、ビジネスアプリケーションへの変更は簡単にすばやく行うことができます。

Red Hat JBoss Web Server は Tomcat をベースとしたエンタープライズレベルの Web サーバーで、中規模および大規模のアプリケーション用に設計されています。Red Hat JBoss Web Server は、Java Server Pages (JSP) および Java Servlet テクノロジー、PHP、ならびに CGI をデプロイするための単一プラットフォームを提供します。

Red Hat JBoss Web Server インストールに KIE Server およびヘッドレス Process Automation Manager コントローラーをインストールできます。または、スタンドアロンの Business Central JAR ファイルを実行できます。

本書では、Red Hat JBoss Web Server インスタンスに Red Hat Process Automation Manager をインストールする方法を説明します。

他の環境への Red Hat Process Automation Manager のインストール方法に関する説明は、以下のドキュメントを参照してください。

- [『Red Hat JBoss EAP 7.2 への Red Hat Process Automation Manager のインストールおよび設定』](#)
- [『IBM WebSphere Application Server への KIE Server のインストールおよび設定』](#)
- [『Oracle WebLogic Server サーバーへの KIE Server のインストールおよび設定』](#)
- [『Red Hat OpenShift Container Platform への Red Hat Process Automation Manager イミュータブルサーバー環境のデプロイメント』](#)
- [『Red Hat OpenShift Container Platform への Red Hat Process Automation Manager オペレーティング環境のデプロイ』](#)
- [『Red Hat OpenShift Container Platform への Red Hat Process Automation Manager フリーフォーム管理サーバー環境のデプロイ』](#)
- [『Operator を使用した Red Hat OpenShift Container Platform への Red Hat Process Automation Manager 環境のデプロイメント』](#)

サポートされるコンポーネントについては、以下のドキュメントを参照してください。

- [What is the mapping between Red Hat Process Automation Manager and the Maven library version?](#)
- [Red Hat JBoss Data Virtualization 7 でサポートされる構成](#)

1.1. RED HAT PROCESS AUTOMATION MANAGER コンポーネント

Red Hat Process Automation Manager は、Business Central および KIE Server で構成されます。

- Business Central は、ビジネスルールを作成して管理するグラフィカルユーザーインターフェースです。Business Central は、Red Hat JBoss EAP インスタンスまたは Red Hat OpenShift Container Platform (OpenShift) にインストールできます。

Business Central は、スタンドアロンの JAR ファイルとしても使用できます。Business Central スタンドアロンの JAR ファイルとして使用して、アプリケーションサーバーにデプロイせずに Business Central を実行できます。

- KIE Server では、ルール、およびその他のアーティファクトが実行されます。ルールのインスタンスを作成して実行し、計画の問題を解決します。KIE Server は、Red Hat JBoss EAP インスタンス、OpenShift、Oracle WebLogic Server インスタンス、IBM WebSphere Application Server インスタンスに、または Spring Boot アプリケーションの一部としてインストールできます。

KIE Server は、管理モードまたは非管理モードで動作するように設定できます。非管理モードの場合は、手動で KIE コンテナ (デプロイメントユニット) を作成および維持する必要があります。KIE コンテナは、プロジェクトの特定のバージョンです。管理モードの場合は、Process Automation Manager コントローラーが KIE Server の設定を管理し、ユーザーはコントローラーと対話形式で KIE コンテナを作成、維持します。

Red Hat JBoss Web Server インストールに KIE Server およびヘッドレス Process Automation Manager コントローラーをインストールできます。または、スタンドアロンの Business Central JAR ファイルを実行できます。

1.2. RED HAT PROCESS AUTOMATION MANAGER ロールおよびユーザー

Business Central または KIE Server にアクセスするには、サーバーを起動する前にユーザーを作成して適切なロールを割り当てます。本セクションは、利用可能な Red Hat Process Automation Manager ユーザーロールを説明します。



注記

admin、**analyst**、**developer**、**manager**、**process-admin**、**user**、および **rest-all** のロールは Business Central に予約されており、**kie-server** ロールは KIE Server に予約されています。このため、利用可能なロールは、インストールされているシステムが、Business Central、KIE Server、またはその両方かによって異なります。

- **admin**: **admin** ロールを持つユーザーは Business Central 管理者です。管理者は、ユーザーの管理や、リポジトリの作成、クローン作成、および管理ができます。アプリケーションで必要な変更すべてにアクセスできます。**admin** ロールを持つユーザーは、Red Hat Process Automation Manager の全領域にアクセスできます。
- **analyst**: **analyst** ロールを持つユーザーには、すべてのハイレベル機能へのアクセスがあり、プロジェクトのモデル化および実行が可能です。ただし、このユーザーは、**Design → Projects** ビューでスペースにコントリビューターを追加したり、スペースを削除したりできません。**Deploy → Execution Servers** ビューへのアクセスは管理者を対象にしており、**analyst** ロールを持つユーザーは利用できません。ただし、**Deploy** ボタンは、このユーザーが Library パースペクティブにアクセスする時に利用できます。
- **developer**: **developer** ロールを持つユーザーは、ほぼすべての機能にアクセスができ、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できます。アセットリポジトリを管理し、プロジェクトを作成、ビルド、およびデプロイでき、Red Hat CodeReady Studio を使用してプロセスを表示できます。**developer** ロールが割り当てられているユーザーには、新規リポジトリの作成やクローン作成などの、特定の管理機能は表示されません。
- **manager**: **manager** ロールを持つユーザーはレポートを表示できます。このユーザーは通常、ビジネスプロセス、そのパフォーマンス、ビジネスインジケター、その他のビジネス関連のレポートに関する統計に関心があります。このルールを持つユーザーがアクセスできるのはプロセスおよびタスクのレポートに限られます。

- **process-admin: process-admin** ロールを持つユーザーは、ビジネスプロセス管理者です。ビジネスプロセス、ビジネスタスク、および実行エラーへの完全アクセスがあります。このユーザーは、ビジネスレポートを表示でき、タスク受信箱リストにアクセスできます。
- **user: user** ロールが割り当てられたユーザーは、タスクインボックスリストで作業できます。このリストには、現在実行中のプロセスの一部であるビジネスタスクなどが含まれます。このロールが割り当てられたユーザーは、タスクレポートやプロセスの表示、プロセスの管理ができます。
- **rest-all: rest-all** ロールを持つユーザーは、Business Central REST 機能にアクセスできます。
- **kie-server: kie-server** ロールを持つユーザーは、KIE Server REST 機能へのアクセスがあります。このロールは、Business Central で **Manage** ビューおよび **Track** ビューにアクセスするユーザーにとって必須となります。

第2章 RED HAT PROCESS AUTOMATION MANAGER インストールファイルのダウンロード

お使いの環境およびインストール要件に応じた Red Hat Process Automation Manager ディストリビューションをダウンロードします。

手順

1. Red Hat カスタマーポータルでの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
 - **製品:** Process Automation Manager
 - **バージョン:** 7.7
2. 任意のインストール方法に合わせて、以下の製品ディストリビューションのいずれかをダウンロードします。



注記

ダウンロードするのは、どれか1つのディストリビューションのみです。

- インストーラーを使用して Red Hat JBoss Web Server に Red Hat Process Automation Manager をインストールする場合は、**Red Hat Process Automation Manager 7.7.0 Installer (rhpm-installer-7.7.0.jar)** をダウンロードします。インストーラーのグラフィックユーザーインターフェースにしたがって、インストールプロセスを進めます。
- デプロイ可能な zip ファイルを使用して Red Hat JBoss Web Server に KIE Server をインストールするには、以下のファイルをダウンロードします。
 - **Red Hat Process Automation Manager 7.7.0 Add Ons (rhpm-7.7.0-add-ons.zip)**
 - **Red Hat Process Automation Manager 7.7.0 Maven Repository (rhpm-7.7.0-maven-repository.zip)**
ZIP ファイルにはグラフィカルユーザーインターフェースが必要ないため、SSH を使用して Red Hat Process Automation Manager をインストールできます。
- アプリケーションサーバーにデプロイせずに Business Central を実行するには、**Red Hat Process Automation Manager 7.7.0 Business Central Standalone (rhpm-7.7.0-business-central-standalone.jar)** をダウンロードします。

第3章 RED HAT PROCESS AUTOMATION MANAGER インストーラーの使用

本セクションでは、インストール JAR ファイルを使用して KIE Server およびヘッドレス Process Automation Manager コントローラーをインストールする方法説明します。JAR ファイルは、既存の Red Hat JBoss Web Server 5.2.0 サーバーインストールに Red Hat Process Automation Manager をインストールする実行ファイルです。インストーラーは、インタラクティブモードまたはコマンドラインインターフェース (CLI) モードで実行できます。

次のステップ:

以下のセクションのいずれかに記載される手順を行います。

- [「インタラクティブモードでのインストーラーの使用」](#)
- [「CLI モードでのインストーラーの使用」](#)

3.1. インタラクティブモードでのインストーラーの使用

Red Hat Process Automation Manager のインストーラーは、実行可能な JAR ファイルです。このインストーラーを使用して、既存の Red Hat JBoss Web Server 5.2.0 サーバーインストールに Red Hat Process Automation Manager をインストールできます。



注記

セキュリティ上の理由で、root 以外のユーザーでインストーラーを実行する必要があります。

前提条件

- サポート対象の JDK がインストールされている。サポート対象の JDK の一覧については、[「Red Hat Process Automation Manager 7 Supported Configurations」](#) を参照してください。
- バックアップを作成してある Red Hat JBoss Web Server 5.2.0 のサーバーインストールが利用できる。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。



注記

Tomcat への書き込み権限のあるユーザーとしてログインします。

- `$PATH` 環境変数に含まれている JAR バイナリー。Red Hat Enterprise Linux では、`java-$JAVA_VERSION-openjdk-devel` パッケージに含まれています。



注記

Red Hat Process Automation Manager 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 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ます。

手順

1. ターミナルウィンドウで、インストーラー JAR ファイルをダウンロードしたディレクトリーに移動し、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.7.0.jar
```

注記

Windows でインストーラーを実行すると、インストール時に管理者の認証情報が求められる場合があります。この要求を回避するには、インストールコマンドに **izpack.mode=privileged** オプションを追加します。

```
java -Dizpack.mode=privileged -jar  
rhpam-installer-7.7.0.jar
```

また、32 ビットの Java 仮想マシンでインストーラーを実行している場合には、メモリー不足になる可能性があります。この問題を防ぐには、以下のコマンドを実行します。

```
java -XX:MaxHeapSize=4g -jar  
rhpam-installer-7.7.0.jar
```

グラフィカルインストーラーにスプラッシュ画面と使用許諾契約書のページが表示されます。

2. **I accept the terms of this license agreement (本使用許諾契約書の内容に同意します)** をクリックし、**Next** をクリックします。
3. Red Hat Process Automation Manager をインストールする Red Hat JBoss Web Server 5.2.0 のホームを指定して、**Next** をクリックします。
4. インストールするコンポーネントを選択し、**Next** をクリックします。
Business Central は Red Hat JBoss Web Server にインストールすることができません。Business Central をインストールできるのは Red Hat JBoss EAP だけですが、KIE Server およびヘッドレス Process Automation Manager コントローラーを Red Hat JBoss Web Server にインストールできます。ヘッドレス Decision Manager コントローラーは、KIE Server を管理するために使用します。複数の KIE Server インスタンスを管理する場合は、ヘッドレス Process Automation Manager コントローラーをインストールしてください。
5. ユーザーを作成して、**Next** をクリックします。デフォルトでは、同じコンテナに Business Central と KIE Server の両方をインストールする場合は、新しいユーザーに **admin**、**kie-server** および **rest-all** ロールが割り当てられます。KIE Server REST ケーパビリティを使用するには、**kie-server** ロールが必要です。

注記

必ず、既存のユーザー、ロール、またはグループとは異なるユーザー名を指定してください。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。

パスワードは 8 文字以上で、数字と、英数字以外の文字をそれぞれ 1 文字以上使用する必要があります。ただし & の文字は使用できません。

ユーザー名とパスワードを書き留めておきます。Business Central および KIE Server にアクセスする時に必要になります。

6. **Installation Overview** ページで **Next** をクリックしてインストールを起動します。Installation Overview ページに、インストールされるコンポーネントの一覧が表示されます。
7. インストールが完了したら、**Next** をクリックします。
8. KIE Server がインストールされている場合には、**Component Installation** で **Configure Runtime** の手順が表示されます。**Configure Runtime Environment** ページで、デフォルトのインストールを実行するか、詳細設定を行うかを選択します。
Perform advanced configuration を選択した場合は、データベース設定、または特定の KIE Server オプションのカスタマイズが選択できます。
9. **JDBC Drive Configuration** ページで **Customize database settings** を選択した場合は、データソースの JDBC ドライバーのベンダーを選択し、ドライバの JAR ファイルを1つ以上選択し、**Next** をクリックします。
データソースは、アプリケーションサーバーなど、Java Database Connectivity (JDBC) クライアントを有効にするオブジェクトで、データベースへの接続を確立します。アプリケーションは、JNDI (Java Naming and Directory Interface) ツリーまたはローカルのアプリケーションコンテキストでデータソースを検索し、データベース接続を要求してデータを取得します。KIE Server にデータソースを設定して、サーバーと、指定したデータベースとの間で適切なデータ交換を行う必要があります。
10. **KIE Server Properties Configuration** で **Customize KIE Server properties** を選択した場合は、以下のいずれかを変更します。
 - **KIE Server ID**: KIE Server プロパティの名前
 - 無効にする KIE Server 機能の選択を解除します。
11. **Next** をクリックして、ランタイム環境を設定します。
12. 画面上部に **Processing finished** が表示されたら、**Next** をクリックしてインストールを完了します。
13. オプションで、**Generate Installation Script and Properties File** をクリックして、XML ファイルでインストールデータを保存し、**Done** をクリックします。
インストーラーは、今後のインストールを自動化する **auto.xml** ファイルと、ユーザーのパスワードと他の機密情報に関連する変数を保存する **auto.xml.variables** ファイルの2つのファイルを生成します。複数のシステムで **auto.xml** ファイルを使用して、元のインストールと同じ設定の同じタイプのサーバーに、Red Hat Process Automation Manager を簡単に繰り返しインストールできます。必要に応じて、**auto.xml** ファイルの **installpath** パラメーターを更新してください。XML ファイルでインストールを行うには、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.7.0.jar <path-to-auto.xml-file>
```

これで、インストーラーを使用して Red Hat Process Automation Manager がインストールできました。Business Central だけをインストールした場合は、この手順を繰り返して、別のサーバーに KIE Server をインストールします。



注記

Microsoft SQL Server を使用する場合は、データベースに適した正しいトランザクションの分離を設定されていることを確認してください。設定されていない場合には、デッドロックが発生する可能性があります。推奨の設定は、以下のステートメントを入力して、ALLOW_SNAPSHOT_ISOLATION と READ_COMMITTED_SNAPSHOT を ON にします。

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

3.2. CLI モードでのインストーラーの使用

コマンドラインインターフェース (CLI) を使用して Red Hat Process Automation Manager インストーラーを実行できます。



注記

セキュリティ上の理由で、root 以外のユーザーでインストーラーを実行する必要があります。

前提条件

- サポート対象の JDK がインストールされている。サポート対象の JDK の一覧については、「[Red Hat Process Automation Manager 7 Supported Configurations](#)」を参照してください。
- バックアップを作成してある Red Hat JBoss Web Server 5.2.0 のサーバーインストールが利用できる。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。



注記

Tomcat への書き込み権限のあるユーザーとしてログインします。

- `$PATH` 環境変数に含まれている JAR バイナリー。Red Hat Enterprise Linux では、`java-$JAVA_VERSION-openjdk-devel` パッケージに含まれています。



注記

Red Hat Process Automation Manager 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 を使用するようには、`"-Dfile.encoding=UTF-8"` のシステムプロパティを使用します。

手順

1. ターミナルウィンドウにおいて、インストーラーファイルをダウンロードしたディレクトリに移動し、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.7.0.jar -console
```

コマンドラインの対話プロセスが開始し、使用許諾契約書が表示されます。

```
press 1 to continue, 2 to quit, 3 to redisplay.
```

2. 使用許諾契約書を読んで **1** を入力し、Enter キーを押して続行します。

```
Specify the home directory of one of the following servers: Red Hat JBoss EAP 7.2 or Red Hat JBoss Web Server 5.2.0
```

3. 既存の Red Hat JBoss Web Server 5.2.0 インストールの親ディレクトリーを入力します。インストーラーが、指定したインストール場所を確認します。**1** を入力して確認し、続行します。
4. インストーラーの指示に従って、インストールを完了します。



注記

ユーザー名およびパスワードを作成する場合は、指定したユーザー名が既存のロールまたはグループの名前と競合しないようにしてください。たとえば、**admin** という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。

パスワードは 8 文字以上で、数字と、英数字以外の文字をそれぞれ 1 文字以上使用する必要があります。ただし **&** の文字を使用することは **できません**。

ユーザー名とパスワードを書き留めておきます。Business Central および KIE Server にアクセスする時に必要になります。

5. インストールが完了すると、以下のメッセージが表示されます。

```
Would you like to generate an automatic installation script and properties file?
```

6. **y** を入力してインストールデータが含まれる XML ファイルを作成します。あるいは、**n** を入力してインストールを完了します。**y** を入力すると、XML ファイルのパスを指定するよう求められます。
7. パスを入力するか、Enter キーを押して提案されたパスを了承します。インストーラーは、今後のインストールを自動化する **auto.xml** ファイルと、ユーザーのパスワードと他の機密情報に関連する変数を保存する **auto.xml.variables** ファイルの 2 つのファイルを生成します。複数のシステムで **auto.xml** ファイルを使用して、元のインストールと同じ設定の同じタイプのサーバーに、Red Hat Process Automation Manager を簡単に繰り返しインストールできます。必要に応じて、**auto.xml** ファイルの **installpath** パラメーターを更新してください。XML ファイルでインストールを行うには、以下のコマンドを入力します。

```
java -jar rhpam-installer-7.7.0.jar <path-to-auto.xml-file>
```

8. KIE Server だけをインストールした場合は、この手順を繰り返して、別のサーバーにヘッドレス Process Automation Manager コントローラーをインストールします。



注記

Microsoft SQL Server を使用する場合は、データベースに適した正しいトランザクションの分離を設定されていることを確認してください。設定されていない場合には、デッドロックが発生する可能性があります。推奨の設定は、以下のステートメントを入力して、ALLOW_SNAPSHOT_ISOLATION と READ_COMMITTED_SNAPSHOT を ON にします。

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON  
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

第4章 KIE SERVER ZIP ファイルのインストールと設定

カスタマーポータルでの Red Hat Process Automation Manager 7.7.0 Add Ons(**rhpm-7.7.0-add-ons.zip**) から入手できる **rhpm-7.7.0-kie-server-jws.zip** ファイルを使用して KIE Server をインストールし、Java Database Connectivity (JDBC) の Web server データソースを Red Hat JBoss Web Server に設定します。

4.1. ZIP ファイルからの KIE SERVER のインストール

KIE Server はビジネスアセットのランタイム環境を提供し、アセットリポジトリ (ナレッジストア) に保存されたデータにアクセスします。ZIP ファイルを使用して既存の Red Hat JBoss Web Server 5.1.0 サーバーインスタンスに KIE Server をインストールすることができます。



注記

インストーラー JAR ファイルを使用した KIE Server のインストール方法は「[3章 Red Hat Process Automation Manager インストーラーの使用](#)」を参照してください。

- [2章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) の説明に従い、以下のファイルがダウンロードされている。
 - Red Hat Process Automation Manager 7.7.0 Add Ons(**rhpm-7.7.0-add-ons.zip**)
 - Red Hat Process Automation Manager 7.7.0 Maven Repository(**rhpm-7.7.0-maven-repository.zip**)
- バックアップを作成してある Red Hat JBoss Web Server 5.2.0 サーバーインストールが利用できる。Red Hat JBoss Web Server インストールのベースディレクトリーは **JWS_HOME** とする。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。

手順

1. **rhpm-7.7.0-add-ons.zip** ファイルを展開します。
2. 展開した **rhpm-7.7.0-add-ons.zip** ファイルから、以下のファイルを展開します。
 - **rhpm-7.7.0-kie-server-jws.zip**
 - **rhpm-7.7.0-process-engine.zip**

以下の説明では、展開した **rhpm-7.7.0-kie-server-jws.zip** ファイルが含まれるディレクトリーは **JWS_TEMP_DIR** と呼ばれ、展開した **rhpm-7.7.0-process-engine.zip** ファイルが含まれるディレクトリーは **ENGINE_TEMP_DIR** と呼ばれます。

3. **JWS_TEMP_DIR/rhpm-7.7.0-kie-server-jws/kie-server.war** ディレクトリーを **JWS_HOME/tomcat/webapps** ディレクトリーにコピーします。



注記

コピーする Red Hat Process Automation Manager デプロイメントの名前が、Red Hat JBoss Web Server インスタンスの既存デプロイメントと競合しないことを確認します。

4. **kie-server.war** フォルダから **.war** 拡張子を削除します。
5. **kie-tomcat-integration-7.33.0.Final-redhat-00002.jar** ファイルを **ENGINE_TEMP_DIR** ディレクトリから **JWS_HOME/tomcat/lib** ディレクトリに移動します。
6. **jboss-jacc-api-<VERSION>.jar**、**slf4j-api-<VERSION>.jar** および **slf4j-jdk14-<VERSION>.jar** ファイルを **ENGINE_TEMP_DIR/lib** ディレクトリから **JWS_HOME/tomcat/lib** ディレクトリに移動します。<VERSION> は **lib** ディレクトリのバージョンのアーティファクトファイル名に置き換えます。
7. 最後の Valve 定義の後にある **JWS_HOME/tomcat/conf/server.xml** ファイルの **<host>** 要素に、以下の行を追加します。

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

8. テキストエディターで **JWS_HOME/tomcat/conf/tomcat-users.xml** ファイルを開きます。
9. **JWS_HOME/tomcat/conf/tomcat-users.xml** ファイルにユーザーおよびロールを追加します。以下の例で、<ROLE_NAME> は Red Hat Process Automation Manager でサポートされるロールです。<USER_NAME> および <USER_PWD> は、任意のユーザー名およびパスワードです。

```
<role rolename="<ROLE_NAME>"/>
<user username="<USER_NAME>" password="<USER_PWD>" roles="<ROLE_NAME>"/>
```

ユーザーに複数のロールを割り当てる場合は、以下の例に示すようにロールをコンマで区切ります。

```
<role rolename="admin"/>
<role rolename="kie-server"/>
<user username="rhpamUser" password="user1234" roles="admin,kie-server"/>
```

10. **JWS_HOME/tomcat/bin** ディレクトリで以下の手順の1つを実行します。

- Linux または UNIX の場合は、以下の内容の **setenv.sh** ファイルを作成します。

```
CATALINA_OPTS="-Xmx1024m -Dorg.jboss.logging.provider=jdk"
```

- Windows の場合は、以下の内容の **setenv.bat** ファイルを作成します。

```
set CATALINA_OPTS=-Xmx1024m -Dorg.jboss.logging.provider=jdk
```

注記

Microsoft SQL Server を使用する場合は、データベースに適した正しいトランザクションの分離を設定されていることを確認してください。設定されていない場合には、デッドロックが発生する可能性があります。推奨の設定は、以下のステートメントを入力して、ALLOW_SNAPSHOT_ISOLATION と READ_COMMITTED_SNAPSHOT を ON にします。

```
ALTER DATABASE <DBNAME> SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE <DBNAME> SET READ_COMMITTED_SNAPSHOT ON
```

4.2. JDBC WEB SERVER データソースの設定

Java Database Connectivity (JDBC) は、Java で記述されたプログラムを、一般的なデータベースのデータに接続するのに使用する API 仕様です。データソースは、アプリケーションサーバーなど、JDBC クライアントを有効にするオブジェクトで、データベースへの接続を確立します。アプリケーションは、JNDI (Java Naming and Directory Interface) ツリーまたはローカルのアプリケーションコンテキストでデータソースを検索し、データベース接続を要求してデータを取得します。KIE Server にデータソースを設定して、サーバーと、指定したデータベースとの間で適切なデータ交換を行う必要があります。

前提条件

- Red Hat JBoss Web Server に Red Hat Process Automation Manager がインストールされている。
- [2章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) の記載通りに、Red Hat Process Automation Manager 7.7.x Maven リポジトリ (**rhpm-7.7.0-maven-repository.zip**) および Red Hat Process Automation Manager 7.7.x Add-Ons (**rhpm-7.7.0-add-ons.zip**) ファイルをダウンロードしている。
- 以下のサポート対象のデータベースおよび Hibernate 方言の1つを設定します。
 - DB2: **org.hibernate.dialect.DB2Dialect**
 - MSSQL: **org.hibernate.dialect.SQLServer2012Dialect**
 - MySQL: **org.hibernate.dialect.MySQL5InnoDBDialect**
 - MariaDB: **org.hibernate.dialect.MySQL5InnoDBDialect**
 - Oracle: **org.hibernate.dialect.Oracle10gDialect**
 - PostgreSQL: **org.hibernate.dialect.PostgreSQL82Dialect**
 - PostgreSQL plus: **org.hibernate.dialect.PostgresPlusDialect**
 - Sybase: **org.hibernate.dialect.SybaseASE157Dialect**

手順

1. 以下の手順を実行して、データベースを準備します。
 - a. **TEMP_DIR** などの一時ディレクトリーに **rhpm-7.7.0-add-ons.zip** を展開します。
 - b. **TEMP_DIR/rhpm-7.7.0-migration-tool.zip** を展開します。
 - c. 現在のディレクトリーから、**TEMP_DIR/rhpm-7.7.0-migration-tool/ddl-scripts** ディレクトリーに移動します。このディレクトリーには、複数のデータベースタイプの DDL スクリプトが含まれています。
 - d. 使用するデータベースに、お使いのデータベースタイプの DDL スクリプトをインポートします。以下に例を示します。

```
psql jbpmsql < /ddl-scripts/postgresql/postgresql-jbpmsql-schema.sql
```

2. **rhpam-7.7.0-maven-repository.zip** のオフライン Maven リポジトリファイルを展開します。
3. 展開したオフラインの Maven リポジトリから **JWS_HOME/tomcat/lib** フォルダにコピーします。**VERSION** は対象のライブラリーのバージョンに置き換えます。

```
org/jboss/spec/javax/transaction/jboss-transaction-api_1.2_spec/{VERSION}/jboss-transaction-api_1.2_spec-{VERSION}.jar
org/jboss/integration/narayana-tomcat/{VERSION}/narayana-tomcat-{VERSION}.jar
org/jboss/narayana/jta/narayana-jta/{VERSION}/narayana-jta-{VERSION}.jar
org/jboss/jboss-transaction-spi/{VERSION}/jboss-transaction-spi-{VERSION}.jar
```

4. データベースの JDBC ドライバーを **JWS_HOME/tomcat/lib** フォルダにコピーします。
5. **JWS_HOME/tomcat/conf/context.xml** ファイルで、プーリングしている XA データソースを設定します。



注記

以下の例のプロパティで、お使いのデータベースサーバーに該当しない場合があります。JDBC ドライバーのドキュメントを参照して、どのプロパティを設定するか判断してください。

- a. プーリング機能なしで XA データソースを設定します。この XA データソースは、対象のデータベースに新規接続を作成するときに使用します。以下の例では、XA データソースは **xads** で、変数は [表4.1「XA データソースの変数」](#) に定義されています。

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
  <Resource
    auth="Container"
    databaseName="${datasource.dbName}"
    description="XA Data Source"
    factory="org.apache.tomcat.jdbc.naming.GenericNamingResourcesFactory"
    loginTimeout="0"
    name="xads"
    uniqueName="xads"
    portNumber="${datasource.port}"
    serverName="${datasource.hostname}"
    testOnBorrow="false"
    type="${datasource.class}"
    url="${datasource.url}"
    URL="${datasource.url}"
    user="${datasource.username}"
    password="${datasource.password}"
    driverType="4"
    schema="${datasource.schema}"
  />
</Context>
```

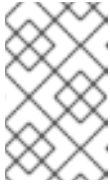
表4.1 XA データソースの変数

| 変数 | 説明 |
|-----------------------|---|
| <datasource.dbName> | データベース名 |
| <datasource.port> | データベースのポート番号 |
| <datasource.hostname> | データベースホスト名 |
| <datasource.class> | JDBC ドライバーの XADataSource クラス |
| <datasource.url> | JDBC データベース接続の URL。データベースによっては、URL プロパティは url 、他のデータベース (H2 データベースなど) では、このプロパティは URL です。 |
| <datasource.username> | データベース接続のユーザー名 |
| <datasource.password> | データベース接続のパスワード |
| <datasource.schema> | データベーススキーマ |

- b. 新規接続の作成時に XA データソースに依存するプーリングデータソースを設定します。以下の例では、データソースは **poolingXaDs** で、<datasource.username> はデータベース接続のユーザー名、<datasource.password> はデータベース接続のパスワードとなっています。

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
<Resource
name="poolingXaDs"
uniqueName="poolingXaDs"
auth="Container"
description="Pooling XA Data Source"
factory="org.jboss.narayana.tomcat.jta.TransactionalDataSourceFactory"
testOnBorrow="true"
transactionManager="TransactionManager"
transactionSynchronizationRegistry="TransactionSynchronizationRegistry"
type="javax.sql.XADataSource"
username="{datasource.username}"
password="{datasource.password}"
xaDataSource="xads"
/>
</Context>
```

データソースが **java:comp/env/poolingXaDs** の JNDI 名で取得でき、次の手順に記載されているように、**org.kie.server.persistence.ds** システムプロパティを使用して KIE Server に渡すことができます。

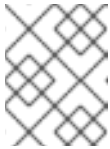


注記

プーリングのデータソース設定は、特に **TransactionManager** や **TransactionSynchronizationRegistry** など、**kie-server** アプリケーションの **context.xml** ファイルで以前に設定された追加リソースに依存します。

6. KIE Server をがデータソースを使用するように設定します。

a. テキストエディターで以下のスクリプトのいずれかを開きます。



注記

setenv.sh または **setenv.bat** スクリプトはすでに存在しますが、存在しない場合には作成してください。

- Linux または Unix の場合:

```
JWS_HOME/tomcat/bin/setenv.sh
```

- Windows の場合:

```
JWS_HOME/tomcat/bin/setenv.bat
```

b. 以下のプロパティを **CATALINA_OPTS** に追加します。 **<hibernate.dialect>** はお使いのデータベースの Hibernate 方言に置き換えます。

```
CATALINA_OPTS="-Xmx1024m  
-Dorg.jboss.logging.provider=jdk  
-Dorg.kie.server.persistence.ds=java:comp/env/poolingXaDs  
-Dorg.kie.server.persistence.tm=JBossTS  
-Dorg.kie.server.persistence.dialect=${<hibernate.dialect>}"
```

第5章 キーストアを使用したパスワードセキュリティの確保

キーストアを使用して、Business Central と KIE Server の間の通信に使用するパスワードを暗号化できます。コントローラーと KIE Server のパスワードを暗号化する必要があります。Business Central と KIE Server を別のアプリケーションサーバーにデプロイする場合には、いずれのアプリケーションサーバーもキーストアを使用する必要があります。

Java Cryptography Extension KeyStore (JCEKS) は、対称鍵をサポートするので、キーストアには JCEKS を使用してください。JDK インストールに含まれる KeyTool を使用して、新しい JCEKS を作成します。



注記

KIE Server が JCEKS で設定されていない場合には、KIE Server のパスワードはシステムプロパティにプレーンテキスト形式で保存されます。

前提条件

- Red Hat JBoss Web Server に KIE Server がインストールされている。
- Java 8 以降がインストールされている。

手順

1. テキストエディターで **JWS_HOME/tomcat/conf/tomcat-users.xml** ファイルを開きます。
2. **kie-server** ロールを割り当てた KIE Server ユーザーを **JWS_HOME/tomcat/conf/tomcat-users.xml** ファイルに追加します。以下の例では、**<USER_NAME>** と **<PASSWORD>** は、任意のユーザー名とパスワードに置き換えてください。

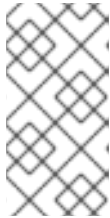
```
<role rolename="kie-server"/>
<user username="<USER_NAME>" password="<PASSWORD>" roles="kie-server"/>
```

3. keytool を使用して JCEKS を作成するには、Java 8 のホームディレクトリーで以下のコマンドを実行します。

```
$<JAVA_HOME>/bin/keytool -importpassword -keystore <KEYSTORE_PATH> -keypass
<ALIAS_KEY_PASSWORD> -alias <PASSWORD_ALIAS> -storepass
<KEYSTORE_PASSWORD> -storetype JCEKS
```

上記の例では、以下の変数を置き換えてください。

- **<KEYSTORE_PATH>**: キーストアの保存先のパス
 - **<KEYSTORE_PASSWORD>**: キーストアのパスワード
 - **<ALIAS_KEY_PASSWORD>**: エイリアスで保存した値にアクセスする時に使用するパスワード
 - **<PASSWORD_ALIAS>**: プロセスへのエントリーに使用するエイリアス
4. プロンプトが表示されたら、作成した KIE Server ユーザーのパスワードを入力します。
 5. システムプロパティを設定するには、**JWS_HOME/tomcat/bin** ディレクトリーで以下の手順の1つを実行し、変数は以下の表で記載されているように置き換えます。



注記

Business Central またはスタンドアロンのコントローラーが Red Hat JBoss Web Server とは別のインスタンスにインストールされている場合には、**kie.keystore.key.server.alias** と **kie.keystore.key.server.pwd** のプロパティを **CATALINA_OPTS** に追加しないでください。

- Linux または UNIX の場合は、以下の内容の **setenv.sh** ファイルを作成します。

```
set CATALINA_OPTS="
-Dkie.keystore.keyStoreURL=<KEYSTORE_URL>
-Dkie.keystore.keyStorePwd=<KEYSTORE_PWD>
-Dkie.keystore.key.server.alias=<KEY_SERVER_ALIASES>
-Dkie.keystore.key.server.pwd=<KEY_SERVER_PWD>
-Dkie.keystore.key.ctrl.alias=<KEY_CONTROL_ALIASES>
-Dkie.keystore.key.ctrl.key.ctrl.pwd=<KEY_CONTROL_PWD>
```

- Windows の場合は、以下の内容の **setenv.bat** ファイルを作成します。

```
set CATALINA_OPTS="
-Dkie.keystore.keyStoreURL=<KEYSTORE_URL>
-Dkie.keystore.keyStorePwd=<KEYSTORE_PWD>
-Dkie.keystore.key.server.alias=<KEY_SERVER_ALIASES>
-Dkie.keystore.key.server.pwd=<KEY_SERVER_PWD>
-Dkie.keystore.key.ctrl.alias=<KEY_CONTROL_ALIASES>
-Dkie.keystore.key.ctrl.pwd=<KEY_CONTROL_PWD>
```

表5.1 KIE Server JCEKS を読み込む時に使用するシステムプロパティ

| システムプロパティ | プレースホルダー | 説明 |
|--------------------------------------|-----------------------|---|
| kie.keystore.keyStoreURL | <KEYSTORE_URL> | 使用する JCEKS の URL。例: file:///home/kie/keystores/keystore.jceks |
| kie.keystore.keyStorePwd | <KEYSTORE_PWD> | JCEKS のパスワード |
| kie.keystore.key.server.alias | <KEY_SERVER_ALIASES> | パスワードの保存先となる REST サービスのキーのエイリアス |
| kie.keystore.key.server.pwd | <KEY_SERVER_PWD> | 保存したパスワードを使用する REST サービスのエイリアスのパスワード |
| kie.keystore.key.ctrl.alias | <KEY_CONTROL_ALIASES> | パスワードの保存先のデフォルトの REST Process Automation Controller のキーのエイリアス |
| kie.keystore.key.ctrl.pwd | <KEY_CONTROL_PWD> | 保存したパスワードを使用する、デフォルトの REST Process Automation Controller のエイリアスのパスワード |

6. KIE Server を起動して、設定を検証します。

第6章 KIE SERVER インストールの確認

KIE Server が正しくインストールされていることを確認します。

前提条件

- KIE Server がインストールされ、設定されている。

手順

1. KIE Server を起動するには、**JWS_HOME/tomcat/bin** ディレクトリーからターミナルを開いて以下のコマンドのいずれかを入力します。
 - Linux または UNIX ベースのシステムの場合:

```
┆ $ ./startup.sh
```
 - Windows の場合:

```
┆ startup.bat
```
2. 数分後に **JWS_HOME/tomcat/logs** ディレクトリーのファイルを確認し、エラーが発生している場合には修正します。
3. Web ブラウザーに **http://localhost:8080/kie-server/services/rest/server** を入力し、KIE Server が正常に動作していることを確認します。
4. **tomcat-users.xml** ファイルに保存したユーザー名とパスワードを入力します。

第7章 ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのダウンロードおよびインストール

KIE Server は、管理モードまたは非管理モードで動作するように設定できます。非管理モードの場合は、手動で KIE コンテナ (デプロイメントユニット) を作成および維持する必要があります。管理モードの場合は、Process Automation Manager コントローラーが KIE Server の設定を管理し、ユーザーはコントローラーと対話形式で KIE コンテナを作成して維持します。

Process Automation Manager コントローラーは Business Central と統合します。Business Central をインストールする場合は、**Execution Server** ページを使用して KIE コンテナを作成および維持します。ただし、Business Central をインストールしない場合は、ヘッドレス Process Automation Manager コントローラーをインストールし、REST API または KIE Server Java Client API を使用してそのコントローラーと対話します。

前提条件

- 「[2章 Red Hat Process Automation Manager インストールファイルのダウンロード](#)」に従って **Red Hat Process Automation Manager 7.7.0 Add Ons (rhpmam-7.7.0-add-ons.zip)** ファイルがダウンロードされている。
- Red Hat JBoss Web Server 5.2.0 サーバーインストールが利用できる。Red Hat JBoss Web Server インストールのベースディレクトリーは **JWS_HOME** とする。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。

手順

1. **rhpmam-7.7.0-add-ons.zip** ファイルを展開します。**rhpmam-7.7.0-controller-jws.zip** ファイルは展開したディレクトリーにあります。
2. **rhpmam-7.7.0-controller-jws.zip** アーカイブを一時ディレクトリーに展開します。以下の例では、このディレクトリーを **TEMP_DIR** とします。
3. **TEMP_DIR/rhpmam-7.7.0-controller-jws.zip/controller.war** ディレクトリーを **JWS_HOME/tomcat/webapps** ディレクトリーにコピーします。



注記

コピーする Red Hat Process Automation Manager デプロイメントの名前が、Red Hat JBoss Web Server インスタンスの既存デプロイメントと競合しないことを確認します。

4. **controller.war** フォルダーから **.war** 拡張子を削除します。
5. **TEMP_DIR/rhpmam-7.7.0-controller-jws/SecurityPolicy/** ディレクトリーのコンテンツを **JWS_HOME/bin** にコピーします。
6. ファイルの上書きを確認するメッセージが表示されたら、**Yes** を選択します。
7. **JWS_HOME/tomcat/conf/tomcat-users.xml** ファイルに **kie-server** ロールおよびユーザーを追加します。以下の例で、**<USER_NAME>** と **<PASSWORD>** は、任意のユーザー名とパスワードです。

```
<role rolename="kie-server"/>
<user username="<USER_NAME>" password="<PASSWORD>" roles="kie-server"/>
```

8. KIE Server を実行するインスタンスの **JWS_HOME/tomcat/bin** ディレクトリーで、以下のタスクの1つを実行します。

- Linux または UNIX の場合は、以下の内容の **setenv.sh** ファイルを作成します。

```
CATALINA_OPTS="-Xmx1024m
-Dorg.jboss.logging.provider=jdk
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.id=<KIE_SERVER_ID>
-Dorg.kie.server.location=http://<HOST>:<PORT>/kie-server/services/rest/server
-Dorg.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller"
```

- Windows の場合は、以下の内容の **setenv.bat** ファイルを作成します。

```
set CATALINA_OPTS=-Xmx1024m -Dorg.jboss.logging.provider=jdk
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.id=<KIE_SERVER_ID>
-Dorg.kie.server.location=http://<HOST>:<PORT>/kie-server/services/rest/server
-Dorg.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller
```

上記の例では、以下の変数に置き換えてください。

- <CONTROLLER_USER>** および **<CONTROLLER_PWD>** を、この手順の最初の方で定義した **kie-server** ロールを持つユーザー名とパスワードに置き換えます。
- <KIE_SERVER_ID>** を一意の識別子に置き換えます。
- <CONTROLLER_HOST>:<CONTROLLER_PORT>** は、コントローラーの IP アドレス (ホストとポート) に置き換えます。KIE Server とコントローラーで同じサーバーを使用する場合は、**<CONTROLLER_HOST>:<CONTROLLER_PORT>** は **localhost:8080** に置き換えます。

9. ヘッドレス Process Automation Manager コントローラーを実行しているインスタンスの **JWS_HOME/tomcat/bin** ディレクトリーに、以下の内容で読み取り可能な **setenv.sh** ファイルを作成します。ここで、**<USERNAME>** は KIE Server ユーザーで、**<USER_PWD>** はそのユーザーのパスワードになります。

```
CATALINA_OPTS="-Dorg.kie.server.user=<USERNAME> -Dorg.kie.server.pwd=
<USER_PWD>"
```

10. ヘッドレス Process Automation Manager コントローラーを起動するには、**JWS_HOME/tomcat/bin** ディレクトリーに以下のいずれかのコマンドを実行します。

- Linux または UNIX ベースのシステムの場合:

```
$ ./startup.sh
```

- Windows の場合:

```
startup.bat
```

11. 数分後に **JWS_HOME/tomcat/logs** ディレクトリーを確認し、エラーが発生している場合には修正します。
12. ヘッドレス Process Automation Manager コントローラーが正常に動作していることを確認するには、Web ブラウザーに **http://<CONTROLLER_HOST>:<CONTROLLER_PORT>/controller/rest/controller/management/servers** を入力します。KIE Server とコントローラーで同じサーバーを使用している場合は、<CONTROLLER_HOST>:<CONTROLLER_PORT> は **localhost:8080** になります。
13. **tomcat-users.xml** ファイルに保存したユーザー名とパスワードを入力します。

第8章 スタンドアロン BUSINESS CENTRAL の実行

Business Central スタンドアロン JAR ファイルを使用して、Red Hat JBoss EAP などのアプリケーションサーバーにデプロイせずに Business Central を実行できます。



注記

Red Hat は、Red Hat Enterprise Linux にインストールした場合に限りこのインストールタイプをサポートします。

前提条件

- [2章 Red Hat Process Automation Manager インストールファイルのダウンロード](#) に従って Red Hat Process Automation Manager 7.7.0 Business Central Standalone (**rhpmam-7.7.0-business-central-standalone.jar**) ファイルがダウンロードされている。

手順

1. ディレクトリーを作成し、**rhpmam-7.7.0-business-central-standalone.jar** ファイルをこのディレクトリーに移動します。
2. ターミナルウィンドウで、スタンドアロン JAR ファイルを含むディレクトリーに移動します。
3. **application-users.properties** ファイルを作成し、管理者ユーザーを含めます。この Business Central インスタンスが KIE Server の Process Automation Manager コントローラーになる場合は、Process Automation Manager コントローラーユーザーを含めます。以下に例を示します。

```
rhpmamAdmin=password1
controllerUser=controllerUser1234
```

4. **application-roles.properties** ファイルを作成し、**application-users.properties** ファイルに含まれるユーザーにロールを割り当てます。以下に例を示します。

```
rhpmamAdmin=admin
controllerUser=kie-server
```

5. 以下の内容で **application-config.yaml** 設定ファイルを作成します。<APPLICATION_USERS> は **application-users.properties** ファイルへのパスで、<APPLICATION_ROLES> は **application-roles.properties** ファイルへのパスです。

```
thorntail:
  security:
    security-domains:
      other:
        classic-authentication:
          login-modules:
            myloginmodule:
              code: org.kie.security.jaas.KieLoginModule
              flag: optional
              module: deployment.business-central-webapp.war
    management:
      security-realms:
        ApplicationRealm:
          local-authentication:
```

```

default-user: local
allowed-users: local
skip-group-loading: true
properties-authentication:
  path: <APPLICATION_USERS>
plain-text: true
properties-authorization:
  path: <APPLICATION_ROLES>
datasource:
management:
  wildfly:
    admin: admin

```

6. 以下のコマンドを入力します。

```
java -jar rhpam-7.7.0-business-central-standalone.jar -s application-config.yaml
```

また、このコマンドに **-D<property>=<value>** パラメーターを追加して、Business Central でサポートされる任意のプロパティを設定することもできます。以下は例になります。

```
java -jar rhpam-7.7.0-business-central-standalone.jar -s application-config.yaml -
D<property>=<value> -D<property>=<value>
```

詳細は、「[Business Central のシステムプロパティ](#)」を参照してください。

8.1. BUSINESS CENTRAL のシステムプロパティ

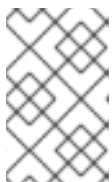
このセクションに記載の Business Central のシステムプロパティは **standalone*.xml** ファイルに渡されます。または、スタンドアロン Business Central のインストール時に、以下のコマンドで、本セクションに列記するプロパティを使用することができます。

```
java -jar rhpam-7.7.0-business-central-standalone.jar -s application-config.yaml -D<property>=
<value> -D<property>=<value>
```

このコマンドでは、**<property>** をリストのプロパティに、**<value>** をそのプロパティに割り当てる値に置き換えます。

- **org.uberfire.nio.git.dir**: KIE Server Git ディレクトリーの場所。
- **org.uberfire.nio.git.dirname**: KIE Server Git ディレクトリーの名前。デフォルト値: **.niogit**
- **org.uberfire.nio.git.proxy.ssh.over.http**: SSH が HTTP プロキシを使用するかどうかを指定します。デフォルト値: **false**
- **http.proxyHost**: HTTP プロキシのホスト名。デフォルト値: **null**
- **http.proxyPort**: HTTP プロキシのポート (整数値)。デフォルト値: **null**
- **org.uberfire.nio.git.proxy.ssh.over.https**: SSH が HTTPS プロキシを使用するかどうかを指定します。デフォルト値: **false**
- **https.proxyHost**: HTTPS プロキシのホスト名。デフォルト値: **null**
- **https.proxyPort**: HTTPS プロキシのポート (整数値)。デフォルト値: **null**

- **org.uberfire.nio.git.http.enabled**: HTTP デーモンを有効または無効にします。デフォルト値: **true**
- **org.uberfire.nio.git.http.host**: HTTP デーモンが有効な場合に、このデーモンはホストの識別子としてこのプロパティを使用します。これは、HTTP 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTP は、継続してサブレットコンテナに依存します。デフォルト値: **localhost**
- **org.uberfire.nio.git.http.hostname**: HTTP デーモンが有効な場合に、このデーモンはホスト名の識別子としてこのプロパティを使用します。これは、HTTP 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTP は、継続してサブレットコンテナに依存します。デフォルト値: **localhost**
- **org.uberfire.nio.git.http.port**: HTTP デーモンが有効な場合に、このデーモンはポート番号としてこのプロパティを使用します。これは、HTTP 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTP は、継続してサブレットコンテナに依存します。デフォルト値: **8080**
- **org.uberfire.nio.git.https.enabled**: HTTPS デーモンを有効または無効にします。デフォルト値: **false**
- **org.uberfire.nio.git.https.host**: HTTPS デーモンが有効な場合に、このデーモンはホストの識別子としてこのプロパティを使用します。これは、HTTPS 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTPS は、継続してサブレットコンテナに依存します。デフォルト値: **localhost**
- **org.uberfire.nio.git.https.hostname**: HTTPS デーモンが有効な場合に、このデーモンはホスト名の識別子としてこのプロパティを使用します。これは、HTTPS 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTPS は、継続してサブレットコンテナに依存します。デフォルト値: **localhost**
- **org.uberfire.nio.git.https.port**: HTTPS デーモンが有効な場合に、このデーモンはポート番号としてこのプロパティを使用します。これは、HTTPS 経由で Git リポジトリにアクセスする方法を表示するときに使用する参考属性です。HTTPS は、継続してサブレットコンテナに依存します。デフォルト値: **8080**
- **org.uberfire.nio.git.daemon.enabled**: Git デーモンを有効または無効にします。デフォルト値: **true**。
- **org.uberfire.nio.git.daemon.host**: Git デーモンが有効な場合は、このプロパティをローカルホストの識別子として使用します。デフォルト値: **localhost**。
- **org.uberfire.nio.git.daemon.hostname**: Git デーモンが有効な場合は、このプロパティをローカルホスト名の識別子として使用します。デフォルト値: **localhost**
- **org.uberfire.nio.git.daemon.port**: Git デーモンが有効な場合は、このプロパティをポート番号として使用します。デフォルト値: **9418**。
- **org.uberfire.nio.git.http.sslVerify**: Git リポジトリの確認をする SSL 証明書を有効または無効にします。デフォルト値: **true**



注記

デフォルトポートまたは割り当てられたポートが既に使用されている場合は、別のポートが自動的に選択されます。ポートが利用可能であることを確認し、詳細についてはログをチェックします。

- **org.uberfire.nio.git.ssh.enabled**: SSH デーモンを有効または無効にします。デフォルト値: **true**。
- **org.uberfire.nio.git.ssh.host**: SSH デーモンが有効な場合は、このプロパティをローカルホスト識別子として使用します。デフォルト値: **localhost**。
- **org.uberfire.nio.git.ssh.hostname**: SSH デーモンが有効な場合は、このプロパティをローカルホスト名の識別子として使用します。デフォルト値: **localhost**
- **org.uberfire.nio.git.SSH.port**: SSH デーモンが有効な場合は、このプロパティをポート番号として使用します。デフォルト値: **8001**



注記

デフォルトポートまたは割り当てられたポートが既に使用されている場合は、別のポートが自動的に選択されます。ポートが利用可能であることを確認し、詳細についてはログをチェックします。

- **org.uberfire.nio.git.ssh.cert.dir**: ローカルの証明書が保存される **.security** ディレクトリーの場所。デフォルト: 作業ディレクトリー。
- **org.uberfire.nio.git.ssh.passphrase**: SCP スタイルの URL (例: **git@github.com:user/repository.git**) を持つ Git リポジトリーのクローンを作成する場合に、オペレーティングシステムの公開キーストアにアクセスするためのパスフレーズ。
- **org.uberfire.nio.git.ssh.algorithm**: SSH で使用されるアルゴリズム。デフォルト値: **RSA**
- **org.uberfire.nio.git.ssh.ciphers**: コンマ区切りの暗号化の文字列。利用可能な暗号化は **aes128-ctr**、**aes192-ctr**、**aes256-ctr**、**arcfour128**、**arcfour256**、**aes192-cbc**、**aes256-cbc** です。このプロパティを使用しない場合は、すべての暗号化が読み込まれます。
- **org.uberfire.nio.git.ssh.macs**: コンマ区切りのメッセージ認証コード (MAC) の文字列。利用可能な MAC は **hmac-md5**、**hmac-md5-96**、**hmac-sha1**、**hmac-sha1-96**、**hmac-sha2-256**、**hmac-sha2-512** です。このプロパティを使用しない場合は、すべての MAC が読み込まれます。



注記

RSA、または DSA 以外のアルゴリズムを使う場合は、Bouncy Castle JCE ライブラリーを使用するようにアプリケーションサーバーを設定します。

- **org.uberfire.metadata.index.dir**: Lucene の **.index** ディレクトリーが保存される場所。デフォルト値: 作業ディレクトリー
- **org.uberfire.ldap.regex.role_mapper**: LDAP プリンシパル名をアプリケーションのロール名にマッピングするのに使用する regex パターン。プリンシパルの値がロール名に一致する場合はアプリケーションのロール名に置き換えられるため、変数ロールはパターンの一部でなければならない点に注意してください。
- **org.uberfire.sys.repo.monitor.disabled**: 設定モニターを無効にします。無効にした場合の影響を正しく理解していない場合は、無効にしないでください。デフォルト値: **false**
- **org.uberfire.secure.key**: パスワードの暗号化で使用するパスワード。デフォルト値: **org.uberfire.admin**

- **org.uberfire.secure.alg**: パスワードの暗号化で使用する暗号化アルゴリズム。デフォルト値: **PBEWithMD5AndDES**
- **org.uberfire.domain**: uberfire が使用するセキュリティードメイン名。デフォルト値: **ApplicationRealm**
- **org.guvnor.m2repo.dir**: Maven リポジトリディレクトリーが保存される場所。デフォルト値: **<working-directory>/repositories/kie**
- **org.guvnor.project.gav.check.disabled**: グループ ID、アーティファクト ID、およびバージョン (GAV) のチェックを無効にします。デフォルト値: **false**
- **org.kie.build.disable-project-explorer**: Project Explorer で選択したプロジェクトの自動ビルドを無効にします。デフォルト値: **false**
- **org.kie.verification.disable-dtable-realtime-verification**: デシジョンテーブルのリアルタイム確認および検証を無効にします。デフォルト値: **false**。
- **org.kie.server.controller**: Process Automation Manager コントローラーへの接続に使用される URL (例: **ws://localhost:8080/business-central/websocket/controller**)
- **org.kie.server.user**: Process Automation Manager コントローラーから KIE Server ノードへの接続時に使用するユーザー名。このプロパティーは、この Business Central システムを Process Automation Manager コントローラーとして使用する場合に限り必要になります。
- **org.kie.server.pwd**: Process Automation Manager コントローラーから KIE Server ノードに接続する際に使用するパスワード。このプロパティーは、この Business Central システムを Process Automation Manager コントローラーとして使用する場合に限り必要になります。
- **kie.maven.offline.force**: Maven のオフライン動作を強制します。true に設定すると、オンラインの依存関係解決が無効になります。デフォルト値: **false**



注記

このプロパティーは、Business Central にのみ使用してください。他のコンポーネントとランタイム環境を共有する場合は、設定を分離して、Business Central にだけ適用してください。

- **org.uberfire.gzip.enable**: **GzipFilter** 圧縮フィルターで Gzip の圧縮を有効にするか、または無効にします。デフォルト値: **true**
- **org.kie.workbench.profile**: Business Central プロファイルを選択します。許容値は、**FULL** または **PLANNER_AND_RULES** です。プリフィックス **FULL_** で、プロファイルを設定し、管理者設定にこのプロファイルの設定を表示されないようにします。デフォルト値: **FULL**
- **org.appformer.m2repo.url**: Business Central は依存関係を検索する時に、Maven リポジトリのデフォルトの場所を使用します。デフォルト値は、**http://localhost:8080/business-central/maven2** など、Business Central 内の Maven リポジトリを参照します。このプロパティーは、Business Central が起動する前に設定してください。デフォルト値: 内部の **m2** リポジトリへのファイルパス
- **appformer.ssh.keystore**: クラス名を指定して Business Central で使用する、カスタムの SSH キーストアを定義します。このプロパティーが指定されていない場合はデフォルトの SSH キーストアを使用します。

- **appformer.ssh.keys.storage.folder**: デフォルトの SSH キーストアを使用する場合にこのプロパティはユーザーの SSH 公開鍵の保存フォルダーを定義します。このプロパティが指定されていない場合には、この公開鍵は Business Central の **.security** フォルダーに保存されます。
- **appformer.experimental.features**: 実験的機能のフレームワークを有効にします。デフォルト値: **false**
- **org.kie.demo**: GitHub 外部へのデモアプリケーションのクローン作成を有効にします。
- **org.kie.workbench.controller**: Process Automation Manager コントローラーとの接続に使用する URL。例: **ws://localhost:8080/kie-server-controller/websocket/controller**
- **org.kie.workbench.controller.user**: Process Automation Manager コントローラーのユーザー。デフォルト値: **kieserver**
- **org.kie.workbench.controller.pwd**: Process Automation Manager コントローラーのパスワード。デフォルト値: **kieserver1!**
- **org.kie.workbench.controller.token**: Process Automation Manager コントローラーとの接続に使用するトークン文字列
- **kie.keystore.keyStoreURL**: Java Cryptography Extension KeyStore (JCEKS) の読み込みに使用する URL。例: **file:///home/kie/keystores/keystore.jceks**。
- **kie.keystore.keyStorePwd**: JCEKS に使用するパスワード
- **kie.keystore.key.ctrl.alias**: デフォルトの REST Process Automation Manager コントローラーに使用するキーのエイリアス
- **kie.keystore.key.ctrl.pwd**: デフォルトの REST Process Automation Manager コントローラーのエイリアスのパスワード
- **org.jbpm.wb.forms.renderer.ext**: Business Central と KIE Server のフォームのレンダリングを切り替えます。デフォルトでは、フォームのレンダリングは Business Central が行います。デフォルト値: **false**
- **org.jbpm.wb.forms.renderer.name**: Business Central と KIE Server のレンダリングフォームを切り替えることができます。デフォルト値: **workbench**

第9章 RED HAT PROCESS AUTOMATION MANAGER の MAVEN 設定およびリポジトリー

Red Hat Process Automation Manager プロジェクトの作成時に、Business Central は Business Central 用に設定された Maven リポジトリーを使用します。Maven project object model (POM) ファイル (**pom.xml**) を変更することで、Maven グローバルまたはユーザー設定を使用し、すべての Red Hat Process Automation Manager プロジェクトに対して Red Hat Process Automation Manager の公開リポジトリーから依存関係を取得するように指示できます。また、Business Central と KIE Server が外部の Maven リポジトリーを使用するか、Maven のミラーを準備してオフラインで使用できるように設定できます。

Red Hat Process Automation Manager のパッケージ化およびデプロイメントのオプションに関する詳細は、『[Red Hat Process Automation Manager プロジェクトのパッケージ化およびデプロイ](#)』を参照してください。

9.1. プロジェクト設定ファイル (POM.XML) を使用した MAVEN の設定

Maven を使用して Red Hat Process Automation Manager プロジェクトをビルドおよび管理するには、POM ファイル (**pom.xml**) を作成および設定する必要があります。このファイルにはプロジェクトの設定情報が含まれます。詳細は「[Apache Maven Project](#)」を参照してください。

手順

1. Maven プロジェクトを生成します。**pom.xml** ファイルは、Maven プロジェクトの作成時に自動的に生成されます。
2. **pom.xml** ファイルを編集して、追加の依存関係および新たなリポジトリーを追加します。プロジェクトのコンパイルおよびパッケージング時に、Maven がすべての JAR ファイルおよび依存関係にある JAR ファイルを Maven リポジトリーからダウンロードします。

pom.xml ファイルのスキーマは、http://maven.apache.org/maven-v4_0_0.xsd から取得できます。POM ファイルの詳細は「[Apache Maven Project](#)」を参照してください。

9.2. MAVEN 設定ファイルの修正

Red Hat Process Automation Manager では、Maven の **settings.xml** ファイルを使用して Maven の実行を設定します。**settings.xml** ファイルにプロファイルを作成およびアクティベートし、Red Hat Process Automation Manager プロジェクトが使用する Maven リポジトリーを宣言します。

Maven の **settings.xml** ファイルの詳細は Apache Maven Project の「[Setting Reference](#)」を参照してください。

手順

1. **settings.xml** ファイルでは、Red Hat Process Automation Manager プロジェクトが使用するリポジトリーを宣言します。通常、これは [online Red Hat Process Automation Manager Maven repository](#)、または Red Hat カスタマーポータルからダウンロードする Red Hat Process Automation Manager Maven リポジトリーおよび使用するカスタムアーティファクト用のリポジトリーのいずれかです。
2. たとえば `<SETTINGS_FILE_PATH>` は **settings.xml** ファイルのパスである `kie.maven.settings.custom=<SETTINGS_FILE_PATH>` プロパティを指定して、Business Central または KIE Server が **settings.xml** ファイルを使用するように設定されていることを確認します。

Red Hat JBoss Web Server で、KIE Server の場合は **-Dkie.maven.settings.custom=<SETTINGS_FILE_PATH>** を **setenv.sh** (Linux) または **setenv.bat** (Windows) ファイルの **CATALINA_OPTS** セクションに追加します。スタンドアロンの Business Central の場合は、以下のコマンドを入力します。

```
java -jar rhpam-7.7.0-business-central-standalone.jar -s application-config.yaml -
Dkie.maven.settings.custom=<SETTINGS_FILE_PATH>
```

9.3. RED HAT PROCESS AUTOMATION MANAGER の MAVEN 依存関係の追加

Red Hat Process Automation Manager プロジェクトで適切な Maven 依存関係を使用するには、プロジェクトの **pom.xml** ファイルに Red Hat Business Automation の BOM (bill of materials) ファイルを追加します。Red Hat Business Automation BOM は、Red Hat Decision Manager と Red Hat Process Automation Manager の両方に適用されます。BOM ファイルを追加すると、提供される Maven リポジトリから、推移的依存関係の適切なバージョンがプロジェクトに含められます。

Red Hat Business Automation BOM (Bill of Materials) についての詳細情報は、[「What is the mapping between Red Hat Process Automation Manager and the Maven library version?」](#) を参照してください。

手順

1. Red Hat Business Automation BOM を **pom.xml** ファイルで宣言します。

```
<dependencyManagement>
<dependencies>
<dependency>
<groupId>com.redhat.ba</groupId>
<artifactId>ba-platform-bom</artifactId>
<version>7.7.0.redhat-00002</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
<dependencies>
<!-- Your dependencies -->
</dependencies>
```

2. **<dependencies>** タグでお使いのプロジェクトに必要な依存関係を宣言します。製品の BOM をプロジェクトにインポートしたら、ユーザー向け製品依存関係のバージョンが定義されるため、**<dependency>** 要素のサブ要素 **<version>** を指定する必要はありません。ただし、**<dependency>** 要素を使用して、プロジェクトで使用する依存関係を宣言する必要があります。
3. Business Central に作成されない標準のプロジェクトでは、お使いのプロジェクトに必要な依存関係をすべて指定します。Business Central に作成するプロジェクトでは、基本的なデシジョンエンジンとプロセスエンジンの依存関係が Business Central に自動的に提供されます。
 - 標準的な Red Hat Process Automation Manager プロジェクトでは、使用する機能に応じて、以下の依存関係を宣言します。

埋め込みプロセスエンジン依存関係

```
<!-- Public KIE API -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<!-- Core dependencies for process engine -->
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-flow</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-flow-builder</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-bpmn2</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-runtime-manager</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-persistence-jpa</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-query-jpa</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-audit</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<!-- Dependency needed for default WorkItemHandler implementations. -->
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-workitems-core</artifactId>
</dependency>

<!-- Logging dependency. You can use any logging framework compatible with slf4j. -->
<dependency>
  <groupId>ch.qos.logback</groupId>
```

```
<artifactId>logback-classic</artifactId>
<version>${logback.version}</version>
</dependency>
```

- CDI を使用する Red Hat Process Automation Manager プロジェクトでは、通常、以下の依存関係を宣言します。

CDI が有効化されたプロセスエンジンの依存関係

```
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-kie-services</artifactId>
</dependency>

<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-services-cdi</artifactId>
</dependency>
```

- 標準的な Red Hat Process Automation Manager プロジェクトでは、以下の依存関係を宣言します。

埋め込みデシジョンエンジン依存関係

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

<!-- Dependency for persistence support. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-persistence-jpa</artifactId>
</dependency>

<!-- Dependencies for decision tables, templates, and scorecards.
For other assets, declare org.drools:business-central-models-* dependencies. -->
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-decisiontables</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-templates</artifactId>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-scorecards</artifactId>
</dependency>
```



```
<!-- Dependency for loading KJARs from a Maven repository using KieScanner. -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
</dependency>
```

- KIE Server を使用するには、以下の依存関係を宣言します。

クライアントアプリケーション KIE Server の依存関係

```
<dependency>
  <groupId>org.kie.server</groupId>
  <artifactId>kie-server-client</artifactId>
</dependency>
```

- Red Hat Process Automation Manager にリモートクライアントを作成するには、以下の依存関係を宣言します。

クライアントの依存関係

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

- ルール、プロセス定義など、アセットを含む JAR ファイルを作成する場合は、お使いの Maven プロジェクトのパッケージングの種類を **kjar** とし、**org.kie:kie-maven-plugin** を使用して、**<project>** 要素に置かれた **kjar** パッケージングタイプを処理します。以下の例の **`\${kie.version}** は、[「What is the mapping between Red Hat Process Automation Manager and the Maven library version?」](#) に記載されている Maven ライブラリーのバージョンです。

```
<packaging>kjar</packaging>
<build>
  <plugins>
    <plugin>
      <groupId>org.kie</groupId>
      <artifactId>kie-maven-plugin</artifactId>
      <version>${kie.version}</version>
      <extensions>>true</extensions>
    </plugin>
  </plugins>
</build>
```

9.4. オフラインで使用する MAVEN ミラーリポジトリの用意

公開インターネットへの送信アクセスが、Red Hat Process Automation Manager のデプロイメントに設定されていない場合には、必要なアーティファクトすべてのミラーが含まれる Maven リポジトリを用意して、このリポジトリを使用できるようにする必要があります。



注記

Red Hat Process Automation Manager デプロイメントがインターネットに接続されている場合には、この手順を実行する必要はありません。

前提条件

- 公開インターネットへの送信アクセスが設定されているコンピューターが利用できる。

手順

1. 公開インターネットに送信アクセスができるコンピューターで、以下のアクションを実行します。
 - a. Red Hat カスタマーポータルの [Software Downloads](#) ページから、利用可能な **rhpam-7.7.0-offliner.zip** の製品配信可能ファイルをダウンロードします。
 - b. **rhpam-7.7.0-offliner.zip** ファイルの内容を任意のディレクトリーに展開します。
 - c. ディレクトリーに移動し、以下のコマンドを入力します。

```
./offline-repo-builder.sh offliner.txt
```

このコマンドは、**repository** サブディレクトリーを作成し、必要なアーティファクトをこのサブディレクトリーにダウンロードします。

一部のダウンロードが失敗したことを示すメッセージが表示された場合は、同じコマンドを再度実行してください。ダウンロードが再び失敗する場合は、Red Hat サポートに連絡してください。

2. Business Central 外でサービスを開発し、追加の依存関係がある場合には、ミラーリポジトリーにその依存関係を追加します。サービスを Maven プロジェクトとして開発した場合には、以下の手順を使用し、これらの依存関係を自動的に用意します。公開インターネットへに送信接続できるコンピューターで、この手順を実行します。
 - a. ローカルの Maven キャッシュディレクトリー (`~/.m2/repository`) のバックアップを作成して、ディレクトリーを削除します。
 - b. **mvn clean install** コマンドを使用してプロジェクトのソースをビルドします。
 - c. すべてのプロジェクトで以下のコマンドを入力し、Maven を使用してプロジェクトで生成したすべてのアーティファクトのランタイムの依存関係をすべてダウンロードするようにします。

```
mvn -e -DskipTests dependency:go-offline -f /path/to/project/pom.xml --batch-mode -Djava.net.preferIPv4Stack=true
```

`/path/to/project/pom.xml` は、プロジェクトの **pom.xml** ファイルへの正しいパスに置き換えます。

3. **repository** サブディレクトリーのコンテンツを、Red Hat Process Automation Manager をデプロイしたコンピューターのディレクトリーにコピーします。このディレクトリーがオフラインの Maven ミラーリポジトリーになります。

4. 「[Maven 設定ファイルの修正](#)」の説明に従い、Red Hat Process Automation Manager デプロイメント向けに、**settings.xml** ファイルを作成して設定します。
5. **settings.xml** ファイルで以下を変更します。
 - **<profile>** タグの下で、**<repositories>** または **<pluginRepositoeres>** タグがない場合には、必要に応じてこれらのタグを追加します。
 - **<repositories>** の下に、以下のシーケンスを追加します。

```
<repository>
  <id>offline-repository</id>
  <url>file:///path/to/repo</url>
  <releases>
    <enabled>>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
```

/path/to/repo は、ローカルの Maven ミラーリポジトリーのディレクトリーに対する完全パスに置き換えます。

- **<pluginRepositories>** の下に、以下のシーケンスを追加します。

```
<repository>
  <id>offline-plugin-repository</id>
  <url>file:///path/to/repo</url>
  <releases>
    <enabled>>true</enabled>
  </releases>
  <snapshots>
    <enabled>>false</enabled>
  </snapshots>
</repository>
```

/path/to/repo は、ローカルの Maven ミラーリポジトリーのディレクトリーに対する完全パスに置き換えます。

第10章 GIT リポジトリからのプロジェクトのインポート

Git は分散バージョンの管理システムであり、リビジョンをコミットオブジェクトとして実装します。変更をリポジトリに保存すると、Git リポジトリに新規コミットオブジェクトが作成されます。

Business Central は Git を使用してプロジェクトデータ (ルールやプロセスなどのアセットを含む) を格納します。Business Central でプロジェクトを作成すると、Business Central に埋め込まれている Git リポジトリに追加されます。他の Git リポジトリにプロジェクトがある場合は、Business Central スペースから、そのプロジェクトを Business Central の Git リポジトリにインポートできます。

前提条件

- Red Hat Process Automation Manager プロジェクトが外部の Git リポジトリに存在している。
- 外部の Git リポジトリへの読み取りアクセスに必要な認証情報がある。

手順

1. Business Central で **Menu** → **Design** → **Projects** の順にクリックします。
2. プロジェクトをインポートするスペースを選択または作成します。デフォルトのスペースは **MySpace** です。
3. プロジェクトをインポートするには、次のいずれかを実行します。
 - **Import Project** をクリックします。
 - ドロップダウンリストから **Import Project** を選択します。
4. **Import Project** ウィンドウに、インポートするプロジェクトが含まれる Git リポジトリの URL および認証情報を入力し、**Import** をクリックします。プロジェクトが Business Central の Git リポジトリに追加され、現在のスペースで利用できるようになります。

第11章 LDAP と SSL の統合

Red Hat Process Automation Manager を使用して、RH-SSO を介して LDAP および SSL を統合できます。詳細は『[Red Hat Single Sign-On Server Administration Guide](#)』を参照してください。

付録A バージョン情報

本ドキュメントの最終更新日: 2020 年 3 月 18 日 (水)