



Red Hat Process Automation Manager 7.6

Oracle WebLogic Server への Process Server の
インストールおよび設定

Red Hat Process Automation Manager 7.6 Oracle WebLogic Server への Process Server 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

本書は、Process Server 用に Oracle Weblogic Server を設定する方法と、Oracle サーバーインスタンスに Process Server 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 PROCESS SERVER | 4 |
| 第2章 ORACLE WEBLOGIC SERVER | 5 |
| 第3章 ORACLE WEBLOGIC SERVER のインストールおよび実行 | 6 |
| 第4章 PROCESS SERVER への ORACLE WEBLOGIC SERVER の設定 | 8 |
| 4.1. PROCESS SERVER のグループおよびユーザーの設定 | 8 |
| 4.2. ORACLE WEBLOGIC SERVER への JDBC データソースの設定 | 8 |
| 4.3. JAVA MESSAGE SERVICE (JMS) の設定 | 11 |
| 4.3.1. JMS サーバーの作成 | 11 |
| 4.3.2. JMS モジュールの作成 | 11 |
| 4.3.3. JMS 接続ファクトリーの作成 | 12 |
| 4.3.3.1. Process Server の JMS 接続ファクトリー | 12 |
| 4.3.4. JMS キューの作成 | 13 |
| 4.3.4.1. Process Server の JMS キュー | 13 |
| 4.4. ORACLE WEBLOGIC SERVER へのシステムプロパティの設定 | 14 |
| 4.5. ORACLE WEBLOGIC SERVER の停止および再起動 | 15 |
| 第5章 ORACLE WEBLOGIC SERVER への PROCESS SERVER のインストール | 17 |
| 5.1. ORACLE WEBLOGIC SERVER への PROCESS SERVER インストールの確認 | 17 |
| 第6章 ORACLE WEBLOGIC SERVER へのヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストールおよび実行 | 19 |
| 6.1. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのシステムプロパティの設定 | 20 |
| 6.2. インストールの検証 | 21 |
| 第7章 ORACLE WEBLOGIC SERVER での埋め込みプロセスエンジンまたはデジジョンエンジンの設定 | 23 |
| 第8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 | 28 |
| 第9章 次のステップ | 30 |
| 付録A バージョン情報 | 31 |

前書き

システム管理者は、Red Hat Process Server に Oracle WebLogic Server を設定し、Oracle サーバーインスタンスに Process Server をインストールできます。

前提条件

- Oracle WebLogic Server インスタンスのバージョン 12.2.1.3.0 以降がインストールされている。詳細なインストール手順は『[Oracle WebLogic Server 製品ページ](#)』を参照してください。
- Oracle WebLogic Server 管理コンソール (通常は <http://<HOST>:7001/console>) へのアクセスがある。

第1章 PROCESS SERVER

Process Server は、Red Hat Process Automation Manager のルールおよびその他のアーティファクトの保存先および、実行先のサーバーです。Process Server はスタンドアロンで組み込みのコンポーネントで、REST、Java Message Service (JMS)、または Java クライアントサイドアプリケーションで利用可能なインターフェースを介してルールをインスタンス化して実行し、ソルバーを介してプロセス、ジョブ、および Red Hat Business Optimizer 機能を管理できます。

Web でデプロイ可能な WAR ファイルとして作成することで、Process Server は Web コンテナであればどこにでもデプロイできます。Process Server の現在のバージョンには、Red Hat Decision Manager および Red Hat Process Automation Manager の両方に対するデフォルトの拡張機能が含まれます。

Process Server はメモリー消費が最小限でフットプリントが小さいため、クラウドインスタンスに簡単にデプロイできます。このサーバーの各インスタンスでは、複数のコンテナを開いてインスタンスを作成できるため、並行して複数のルールサービスを実行できます。

Process Server は、Oracle WebLogic Server、IBM WebSphere Application Server などのアプリケーションサーバーと統合して、Red Hat Process Automation Manager のアプリケーション管理を合理化できます。

第2章 ORACLE WEBLOGIC SERVER

Oracle WebLogic Server は、分散型 Java アプリケーションを作成するために API の標準セットを提供する Java EE アプリケーションサーバーで、データベース、メッセージングサービス、外部のエンタープライズシステムへの接続など、さまざまなサービスにアクセスできます。ユーザークライアントは、Web ブラウザークライアントまたは Java クライアントを使用してこれらのアプリケーションにアクセスします。

第3章 ORACLE WEBLOGIC SERVER のインストールおよび実行

Process Server に対応する多くの設定を適用するために、Oracle WebLogic Server をインストールして実行する必要があります。本セクションは、スタンドアロンの Oracle WebLogic Server ドメインに Oracle WebLogic Server をインストールして起動する方法を説明します。

最新の詳細情報は『[Oracle WebLogic Server 製品ページ](#)』を参照してください。



注記

起動するサーバーと同じリスナーポートを使用する Oracle WebLogic Server インスタンスを実行している場合は、新たにサーバーを起動する前に、稼働しているサーバーを停止する必要があります。

手順

1. Oracle WebLogic Server の [Downloads page](#) から、Oracle WebLogic Server 12.2.1.3.0 以降をダウンロードします。
2. ターゲットシステムにサインインし、認定済み JDK がシステムにインストールされていることを確認します。インストーラーには認定済み JDK が必要です。システム要件は「[Oracle Fusion Middleware Systems Requirements and Specifications](#)」を参照してください。JDK のダウンロードは「[Planning an Installation of Oracle Fusion Middleware](#)」の「About JDK Requirements for an Oracle Fusion Middleware Installation」のセクションを参照してください。
3. インストールプログラムをダウンロードしたディレクトリーに移動します。
4. システムの JDK ディレクトリーから **java -jar** を実行して、インストールプログラムを起動します。以下の例を参照してください
UNIX ベースのシステムの場合:

```
/home/Oracle/jdk/jdk1.8.0_131/bin/java -jar fmw_12.2.1.3.0_wls_generic.jar
```

Windows オペレーティングシステムの場合:

```
C:\Program Files\Java\jdk1.8.0_131\bin\java -jar fmw_12.2.1.3.0_wls_generic.jar
```

この例の JDK の場所を、実際にシステムに保存されている JDK の場所に置き換えます。

5. インストールウィザードプロンプトに従い、インストールを完了します。
6. インストールが完了したら、端末からドメインディレクトリー (**WLS_HOME/user_projects/<DOMAIN_NAME>**) に移動します。以下は例となります。

```
WLS\user_projects\mydomain
```

7. 以下のいずれかのコマンドを実行して、Oracle WebLogic Server を起動します。
UNIX ベースのシステムの場合:

```
startWebLogic.sh
```

Windows オペレーティングシステムの場合:

■

```
startWebLogic.cmd
```

起動スクリプトが多数のメッセージを表示しますが、最後に以下のようなメッセージを表示します。

```
<Dec 8, 2017 3:50:42 PM PDT> <Notice> <WebLogicServer> <000360> <Server started in RUNNING mode>
```

8. Web ブラウザーで、以下の URL を開きます。

```
http://<HOST>:<PORT>/console
```

<HOST> は、ホストサーバーのシステム名または IP アドレスです。

<PORT> は、ホストサーバーが要求をリッスンしているポートのアドレス (デフォルトでは 7001) です。

たとえば、システムで起動している Oracle WebLogic Server のローカルインスタンスに対して管理コンソールを起動するには、以下の URL をブラウザーに入力します。

```
http://localhost:7001/console/
```

SSL (secure socket layer) を使用して管理コンソールを起動したら、**http** の後に **s** を追加して、**https://<HOST>:<PORT>/console** にする必要があります。

9. WebLogic 管理コンソールのログインページが表示されたら、管理者の認証情報を入力します。

第4章 PROCESS SERVER への ORACLE WEBLOGIC SERVER の設定

Oracle WebLogic Server に Process Server をデプロイする前に、システムプロパティー、セキュリティー設定、JMS 要件、その他の Oracle WebLogic Server プロパティーを設定する必要があります。この設定により、Process Server との最適な統合が促進されます。

前提条件

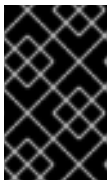
- Oracle WebLogic Server をインストールし、実行している。
- WebLogic 管理コンソールにログインしている。

4.1. PROCESS SERVER のグループおよびユーザーの設定

WebLogic 管理コンソールでユーザーを **kie-server** グループに割り当て、Oracle WebLogic Server で、コンテナ管理の認証メカニズムを有効にします。

手順

1. WebLogic 管理コンソールで **Security Realms** に移動します。
2. 必要なセキュリティーレームを選択するか、**New** をクリックして新しいセキュリティーレームを作成します。
3. **Users and Groups** → **Groups** → **New** の順に移動して、**kie-server** グループを作成します。
4. **Users** → **New** の順に移動し、新しいユーザーを作成します。
5. **server-user** などのユーザーとそのパスワードを入力し、**OK** をクリックします。



重要

選択したユーザー名が、ロールまたはグループの名前と競合しないようにします。たとえば、**kie-server** という名前のロールが存在する場合は、**kie-server** という名前のユーザーを作成しないでください。

6. 新たに作成したユーザーをクリックして、**Groups** タブに戻ります。
7. 選択ツールを使用して、**kie-server** グループを **Available** フィールドから **Chosen** フィールドに移動し、**Save** をクリックします。

4.2. ORACLE WEBLOGIC SERVER への JDBC データソースの設定

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、データ接続を要求してデータを取得します。Oracle WebLogic 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

- データベース接続を作成するのに使用する JDBC ドライバーが、データソースをデプロイする

すべてのサーバーにインストールされている。一部の JDBC ドライバーが Oracle WebLogic Server にインストールされている (DB2、Informix、MS SQL Server、および Sybase 用の WebLogic ブランドの DataDirect JDBC ドライバーなど)。JDBC ドライバーの詳細は、Oracle Help Center の「[Using JDBC Drivers with WebLogic Server](#)」を参照してください。

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:** 7.6
2. Red Hat Process Automation Manager 7.6.x Add-Ons をダウンロードします。x は現在のパッチバージョンです。
3. 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 **rhpmam-7.6.x-add-ons.zip** を展開します。
 - b. **TEMP_DIR/rhpmam-7.6-migration-tool.zip** を展開します。
 - c. 現在のディレクトリーから、**TEMP_DIR/rhpmam-7.6-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 jbpam < /ddl-scripts/postgresql/postgresql-jbpam-schema.sql
```

4. WebLogic 管理コンソールで、**Change Center** → **Lock & Edit** の順に移動します。
5. **Domain Structure** で、**Services** → **Data Sources** の順に移動します。
6. **Summary of Data Sources** ページで、**New** → **Generic Data Source** の順にクリックします。
7. **JDBC Data Sources Properties** ページで、以下の情報を入力するか、選択します。
 - **Name:** この JDBC データソースの名前を入力します。この名前は設定ファイル (**config.xml**) と管理コンソールでこのデータソースを参照するのに使用されます。
 - **JNDI Name:** JDBC データソースをバインドする JNDI パスを入力します。アプリケーションは、接続を予約する際に、この名前を使用して、JNDI ツリーでデータソースを検索します。
 - **Database Type:** 接続するデータベースの DBMS を選択します。DBMS が記載されていない場合は **Other** を選択します。
8. **Next** をクリックして、続行します。
9. データベースに接続するのに使用する **データベースドライバー** を選択します。この一覧には、選択した DBMS 用の一般的な JDBC ドライバーと、すでにインストールされているその他の JDBC ドライバーが含まれます。
10. **Transaction Options** ページで、**Supports Global Transactions** オプションを選択したままに

し、利用可能なトランザクションオプションから選択します。このチェックボックスの選択をはずし、このデータソースでグローバルトランザクションを無効(無視)することもできます。多くの場合、最適なデータ効率を得るために、このオプションは選択したままにする必要があります。

- **Two-Phase Commit:** このオプションを選択すると、標準の XA 処理が有効になります。このオプションは、XA JDBC ドライバーを選択して、データベース接続を作成する場合に限り使用できます。
- **Logging Last Resource:** このオプションを選択すると、LLR (Logging Last Resource) トランザクションの最適化を使用するグローバルトランザクションに非 XA JDBC 接続を使用できるようになります。このオプションは、Emulate Two-Phase Commit よりも推奨されません。このオプションは、非 XA JDBC ドライバーを選択してデータベース接続を作成する場合に限り使用できます。
- **Emulate Two-Phase Commit:** このオプションを選択すると XA JDBC 接続が有効になり、JTA を使用する分散トランザクションでの使用がエミュレートされます。このオプションは、アプリケーションがヒューリスティック条件を容認する場合に限り選択します。このオプションは、非 XA JDBC ドライバーを選択してデータベース接続を作成する場合に限り使用できます。
- **One-Phase Commit:** このオプションを選択すると、グローバルトランザクションで使用する非 XA 接続のみをトランザクションに使用します。このオプションは、非 XA JDBC ドライバーを選択してデータベースの接続を作成する場合に限り使用できます。

11. **Next** をクリックして、続行します。

12. **Connection Properties** ページで、以下のプロパティに値を入力します。

- **Service Name:** 接続するデータベースのサービス名を指定します。データソースが1つ以上提供されている場合は、各データソースに同じサービス名を指定する必要があります。このフィールドは、Oracle Real Application Clusters (RAC) に利用可能なサービスインスタンス接続ドライバーが1つ選択されている場合に限り使用できます。
- **Database Name:** 接続するデータベースの名前を入力します。データベースの名前要件は、JDBC ドライバーおよび DBMS によって異なります。
- **Host Name:** データベースをホストするサーバーの DNS 名または IP アドレスを入力します。Oracle GridLink サービスインスタンス接続を作成し、データソースを1つ以上提供している場合は、各データソースに対する接続は同じにする必要があります。
- **Port:** 接続要求に対してデータベースサーバーがリッスンするポートを入力します。
- **Database User Name:** データソースの各接続で使用するデータベースのユーザーアカウント名を入力します。
- **Password/Confirm Password:** データベースユーザーアカウントのパスワードを入力します。
- **oracle.jdbc.DRCPConnectionClass:** お使いの環境で必要な場合は、オプションで、DCRP (Database Resident Connection Pooling) 接続クラスを入力します。

13. **Next** をクリックして、続行します。

14. **Test Database Connection** ページで接続パラメーターを確認して、**Test Configuration** をクリックします。

Oracle WebLogic Server は、管理サーバーからデータベースへの接続を作成しようとします。接続テストの結果がページ上部に表示されます。テストに失敗した場合は、設定エラーを修正して、再度テストしてみてください。

15. **Next** をクリックして続行します。選択した JDBC ドライバーが管理サーバーにインストールされていない場合は、この手順を省略できます。
16. **Select Targets** ページで、データソースをデプロイするサーバーまたはクラスターを選択し、**Finish** をクリックします。
17. WebLogic 管理コンソールのメインメニューに戻り、**Change Center** → **Activate Changes** の順に選択します。

Oracle WebLogic Server データソースの詳細は、Oracle Help Center の『[JDBC Data Sources for Oracle WebLogic Server](#)』を参照してください。

4.3. JAVA MESSAGE SERVICE (JMS) の設定

Java Message Service (JMS) は、Process Server が、Oracle WebLogic Server、IBM WebSphere Application Server などのアプリケーションサーバーとメッセージを交換するのに使用する Java API です。Process Server を経由して JMS メッセージを送受信するようにアプリケーションサーバーを設定し、2 台のサーバー間でコラボレーションが適切に行われるようにします。

4.3.1. JMS サーバーの作成

JMS を使用するには JMS サーバーを作成する必要があります。

手順

1. WebLogic 管理コンソールで、**Services** → **Messaging** → **JMS Servers** の順に移動します。
2. **New** をクリックして、JMS サーバーを新たに作成します。
3. JMS サーバーの名前を入力して、**Next** をクリックします。
4. Process Server デプロイメントに使用するターゲットサーバーを選択します。
5. **Finish** をクリックします。

4.3.2. JMS モジュールの作成

接続ファクトリー、キューなどの JMS リソースを格納する JMS モジュールを作成する必要があります。

前提条件

- JMS サーバーを作成している。

手順

1. WebLogic 管理コンソールで、**Services** → **Messaging** → **JMS Modules** の順に移動します。
2. **New** をクリックしてモジュールを作成します。
3. モジュール名を入力し、**Next** をクリックします。

4. Process Server デプロイメントに使用するターゲットサーバーを選択し、**Finish** をクリックします。
5. 新たに作成したモジュール名をクリックし、**Subdeployments** をクリックします。
6. **New** をクリックして、モジュールのサブデプロイメントを作成します。
7. サブデプロイメントの名前を入力し、**Next** をクリックします。
8. チェックボックスを選択して、事前に作成した JMS サーバーを選択します。
9. **Finish** をクリックして、サブデプロイメントの設定を完了します。

4.3.3. JMS 接続ファクトリーの作成

Process Server でメッセージングを有効にするには、メッセージの送受信に JMS 接続ファクトリーをいくつか作成する必要があります。

前提条件

- JMS サーバーを作成している。
- JMS モジュールを作成している。

手順

1. WebLogic 管理コンソールで、**Services** → **Messaging** → **JMS Modules** の順に移動して、JMS モジュールの一覧を確認します。
2. 事前に作成したモジュールを選択し、**New** をクリックして新しい JMS リソースを作成します。
3. **Connection Factory** を選択し、**Next** をクリックします。
4. 以下に示す必須の各接続ファクトリーに、接続ファクトリー名 (例: **KIE.SERVER.REQUEST**) および JNDI 名 (例: **jms/cf/KIE.SERVER.REQUEST**) を入力し、**Next** をクリックします。接続ファクトリーが、JMS モジュールにデフォルトで割り当てるサーバーを自動的に選択します。
5. **Finish** をクリックして接続ファクトリーを追加し、必要な各ファクトリーに対して上記手順を繰り返します。

4.3.3.1. Process Server の JMS 接続ファクトリー

以下は、Process Server で JMS メッセージングを有効にするのに必要な Java Message Service (JMS) 接続ファクトリーになります。

表4.1 Process Server に必要な JMS 接続ファクトリー

| 名前 | デフォルト値 | 用途 |
|--|----------------------------------|-------------------------|
| KIE.SERVER.REQUEST ST | jms/cf/KIE.SERVER.REQUEST | 全要求を Process Server へ送信 |

| 名前 | デフォルト値 | 用途 |
|----------------------------|-----------------------------------|------------------------------|
| KIE.SERVER.RESPONSE | jms/cf/KIE.SERVER.RESPONSE | Process Server が生成する応答をすべて受信 |
| KIE.SERVER.EXECUTOR | jms/cf/KIE.SERVER.EXECUTOR | Process Server エグゼキューターサービス |

4.3.4. JMS キューの作成

JMS キューは、ポイントツーポイントメッセージング (point-to-point messaging) の宛先エンドポイントになります。Process Server で JMS メッセージングを有効にするには JMS キューをいくつか作成する必要があります。

前提条件

- JMS サーバーを作成している。
- JMS モジュールを作成している。

手順

1. WebLogic 管理コンソールで、**Services** → **Messaging** → **JMS Modules** の順に移動して、JMS モジュールの一覧を確認します。
2. 事前に作成したモジュールを選択し、**New** をクリックして新規 JMS リソースを作成します。
3. **Queue** を選択し、**Next** をクリックします。
4. 以下に示す必須の各キューに、キューの名前 (例: **KIE.SERVER.REQUEST**) および JNDI 名 (例: **jms/KIE.SERVER.REQUEST**) を入力して、**Next** をクリックします。
5. JMS サーバーに接続する JMS モジュールのサブデプロイメントを選択します。
6. **Finish** をクリックしてキューを追加し、必要な各キューに上記手順を繰り返します。

4.3.4.1. Process Server の JMS キュー

以下は、Process Server で JMS メッセージングを有効にするのに必要な Java Message Service (JMS) キューになります。

表4.2 Process Server に必要な JMS キュー

| 名前 | デフォルト値 | 用途 |
|----------------------------|--------------------------------|------------------------------|
| KIE.SERVER.REQUEST | jms/KIE.SERVER.REQUEST | 全要求を Process Server へ送信 |
| KIE.SERVER.RESPONSE | jms/KIE.SERVER.RESPONSE | Process Server が生成する応答をすべて受信 |

| 名前 | デフォルト値 | 用途 |
|----------------------------|--------------------------------|-----------------------------|
| KIE.SERVER.EXECUTOR | jms/KIE.SERVER.EXECUTOR | Process Server エグゼキューターサービス |

4.4. ORACLE WEBLOGIC SERVER へのシステムプロパティの設定

Process Server をデプロイする前に、Oracle WebLogic Server に、本セクションに記載するシステムプロパティを設定します。

手順

1. Java Virtual Machine (JVM) のメモリーサイズを増やすには、以下のシステムプロパティを設定します。

```
USER_MEM_ARGS=-Xms512m -Xmx1024m
```

JVM メモリーサイズを増やさないと、Process Server のデプロイ時に Oracle WebLogic Server がフリーズするか、エラーが発生します。

2. Oracle WebLogic Server インスタンスに、以下の Process Server システムプロパティを指定します。

表4.3 Process Server のシステムプロパティ

| 名前 | 値 | 説明 |
|---|--|---|
| kie.server.jms.queues.response | jms/KIE.SERVER.RESPONSE | Process Server が使用する応答用の JMS キューの JNDI 名。 |
| org.kie.server.domain | OracleDefaultLoginConfiguration | JMS の使用時にユーザーの認証に使用される JAAS LoginContext ドメイン。 |
| org.kie.server.persistence.ds | jdbc/jbpm | Process Server のデータソースの JNDI 名。 |
| org.kie.server.persistence.tm | org.hibernate.service.jta.platform.internal.WeblogicJtaPlatform | Hibernate プロパティを設定するためのトランザクションマネージャプラットフォーム。 |
| org.kie.server.persistence.dialect | 例: org.hibernate.dialect.H2Dialect | 使用する Hibernate 方言を指定します。データソースに従って設定します。 |
| org.kie.executor.jms.queue | jms/KIE.SERVER.EXECUTOR | Process Server へのジョブエグゼキューターの JMS キュー。 |

| 名前 | 値 | 説明 |
|--------------------------------------|---|---|
| <code>org.kie.executor.jms.cf</code> | <code>jms/cf/KIE.SERVER.EXECUTOR</code> | Process Serve へのジョブエグゼキューターの JMS 接続ファクトリー。 |
| <code>org.kie.server.router</code> | 例: <code>http://localhost:9000</code> | (オプション) クラスター化された Process Server 環境で、アプリケーションサーバーがその一部である1つ以上の Process Server ルーター (Smart Router) の URL を1つ以上指定します。 |

3. `JAVA_OPTIONS` 環境編集に同じプロパティ値を設定します。

```

JAVA_OPTIONS="-Dkie.server.jms.queues.response=jms/KIE.SERVER.RESPONSE
-Dorg.kie.server.domain=OracleDefaultLoginConfiguration
-Dorg.kie.executor.jms.cf=jms/cf/KIE.SERVER.EXECUTOR
-Dorg.kie.executor.jms.queue=jms/KIE.SERVER.EXECUTOR
-Dorg.kie.server.persistence.ds=jdbc/jbpm
-Dorg.kie.server.persistence.tm=org.hibernate.service.jta.platform.internal.WeblogicJtaPlatform
-Dorg.kie.server.persistence.dialect=org.hibernate.dialect.H2Dialect
// Optional server router, for clustered server environment
-Dorg.kie.server.router=http://localhost:9000

```

4.5. ORACLE WEBLOGIC SERVER の停止および再起動

Oracle WebLogic Server で必要なシステムプロパティをすべて設定したら、Oracle サーバーを停止して再起動し、設定が適用されていることを確認します。

手順

1. WebLogic 管理コンソールで、**Change Center** → **Lock & Edit**の順に移動します。
2. **Domain Structure** で **Environment** → **Servers** → **Control**の順にクリックします。
3. 停止するサーバーを選択して、**Shutdown** をクリックします。
4. **When Work Completes** を選択してサーバーを正常に停止するか、**Force Shutdown Now** を選択して継続中のタスクの完了を待たずに直ちにサーバーを停止します。
5. **Server Life Cycle Assistant** ペインで **Yes** をクリックして、シャットダウンを完了します。
6. シャットダウンが完了したら、端末でドメインディレクトリーに移動し、`WLS_HOME/user_projects/<DOMAIN_NAME>` に移動します。以下は例となります。

```
WLS\user_projects\mydomain
```

7. 以下のいずれかのコマンドを実行して Oracle WebLogic Server を再起動し、新しい設定を適用します。

UNIX ベースのシステムの場合:

```
startWebLogic.sh
```

Windows オペレーティングシステムの場合:

```
startWebLogic.cmd
```

8. Web ブラウザーで管理コンソール (例: <http://localhost:7001/console/>) を開き、認証情報を使用してログインします。

第5章 ORACLE WEBLOGIC SERVER への PROCESS SERVER のインストール

Oracle WebLogic Server に必要なシステムプロパティをすべて設定したら、Oracle WebLogic Server に Process Server をインストールして、Red Hat Process Automation Manager アプリケーション管理を合理化します。

前提条件

- 「[4章 Process Server への Oracle WebLogic Server の設定](#)」の説明通りに、Oracle WebLogic Server インスタンスが設定されている。

手順

- 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: 7.6
- Red Hat Process Automation Manager 7.6.0 Process Server for All Supported EE7 Containers をダウンロードします。
- ダウンロードした `rhpm-7.6.0-kie-server-ee7.zip` ファイルを一時ディレクトリーに展開します。
- WebLogic 管理コンソールで **Deployments** に移動し、既存のアプリケーションをすべて表示します。
- Install** をクリックします。
- `rhpm-7.6.0-kie-server-ee7.zip` ファイルをダウンロードして展開した一時ディレクトリーに移動し、`rhpm-7.6.0-kie-server-ee7/kie-server.war` に移動します。
- kie-server.war** ファイルを選択し、**Next** をクリックして続行します。
- ターゲットとするスタイルとして **Install this deployment as an application** を選択し、**Next** をクリックします。
- アプリケーション名を **kie-server** に設定し、セキュリティモデルを **DD Only** に設定します。残りのオプションはデフォルトのままにし、**Next** をクリックして続行します。
- Additional Configuration** セクションで **No, I will review the configuration later** を選択し、**Finish** をクリックします。

5.1. ORACLE WEBLOGIC SERVER への PROCESS SERVER インストールの確認

Oracle WebLogic Server に Process Server をインストールしたら、インストールが成功していることを確認します。

前提条件

- 「[4章 Process Server への Oracle WebLogic Server の設定](#)」の説明通りに、Oracle WebLogic Server インスタンスが設定されている。
- [5章 Oracle WebLogic Server への Process Server のインストール](#) の記載通りに Process Server がインストールされている。

手順

1. Web ブラウザーに Process Server の URL (**`http://<HOST>:<PORT>/kie-server/services/rest/server`**) を入力します。
2. Process Server が実行していることを確認します。
Process Server が実行していない場合は、Oracle WebLogic Server インスタンスを停止して再起動して、Process Server の URL にアクセスしてみてください。

第6章 ORACLE WEBLOGIC SERVER へのヘッドレス PROCESS AUTOMATION MANAGER コントローラーのインストールおよび実行

Process Server REST API または Java Client API を使用して Process Server に接続するには、Oracle Weblogic Server 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。ヘッドレス Process Automation Manager コントローラーは、一元化された方法で Process Server 設定を管理するため、このコントローラーを使用してコンテナの作成および維持を行い、サーバーレベルのタスクを実行できます。

前提条件

- 「[4章 Process Server への Oracle WebLogic Server の設定](#)」の説明通りに、Oracle WebLogic Server インスタンスが設定されている。
- Process Server が Oracle WebLogic Server インスタンスにインストールされている。
- 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ある。

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:** 7.6
2. Red Hat Process Automation Manager 7.6.0 Add-Ons をダウンロードします。
3. ダウンロードした **rhpmam-7.6.0-add-ons.zip** ファイルを一時ディレクトリーに展開します。
4. WebLogic 管理コンソールで、**Security Realms** → **Users and Groups** の順に移動します。
5. 事前に作成した **kie-server** グループに移動して、ヘッドレス Process Automation Manager コントローラーのユーザー (**controller** など) を作成します。続いて、このユーザーのパスワードを設定して **OK** をクリックします。グループとユーザーの作成方法の詳細は、[「Process Server のグループおよびユーザーの設定」](#) を参照してください。
6. **Deployments** に移動して、既存のアプリケーションをすべて表示します。
7. **Install** をクリックします。
8. **rhpmam-7.6.0-add-ons.zip** ファイルをダウンロードして展開した一時ディレクトリーに移動し、**rhpmam-7.6.0-add-ons/rhpmam-7.6-controller-ee7.zip/controller.war** に移動します。
9. **controller.war** ファイルを選択し、**Next** をクリックして続行します。
10. ターゲットとするスタイルとして **Install this deployment as an application** を選択し、**Next** をクリックします。
11. アプリケーション名を **controller** のままにし、セキュリティーモデルを **DD Only** に設定します。残りのオプションはデフォルトのままにし、**Next** をクリックして続行します。

12. **Additional Configuration** セクションで **No, I will review the configuration later** を選択し、**Finish** をクリックします。

6.1. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーのシステムプロパティーの設定

ヘッドレス Process Automation Manager コントローラーをインストールしたら、本セクションで紹介するシステムプロパティーをアプリケーションサーバーに設定して、ヘッドレス Process Automation Manager コントローラーとの適切な接続を有効にします。



注記

実稼働環境で最適な結果を得るには、Process Server とヘッドレス Process Automation Manager コントローラーを別のサーバーにインストールします。開発環境の場合は、Process Server とヘッドレス Process Automation Manager コントローラーを同じサーバーにインストールします。いずれの場合も、ヘッドレス Process Automation Manager コントローラーがインストールされているすべてのアプリケーションサーバーでこのプロパティーを変更します。

前提条件

- Process Server およびヘッドレス Process Automation Manager コントローラーがアプリケーションサーバーインスタンスにインストールされている。

手順

1. ヘッドレス Process Automation Manager コントローラーがインストールされているアプリケーションサーバーインスタンスに、以下の JVM プロパティー値を指定します。

表6.1 ヘッドレス Process Automation Manager コントローラーに必要なプロパティー

| 名前 | 要件 |
|----------------------------|--|
| org.kie.server.user | kie-server ロールが割り当てられているユーザー |
| org.kie.server.pwd | org.kie.server.user プロパティーに指定したユーザーのパスワード |

2. Process Server がインストールされているアプリケーションサーバーインスタンスに、以下の JVM プロパティー値を指定します。

表6.2 ヘッドレス Process Automation Manager コントローラーがインストールされている場合に Process Server に必要なプロパティー

| 名前 | 要件 |
|---------------------------------------|---|
| org.kie.server.controller.user | kie-server ロールが割り当てられているユーザー |
| org.kie.server.controller.pwd | org.kie.server.controller.user プロパティーに指定したユーザーのパスワード |

| 名前 | 要件 |
|----------------------------------|--|
| org.kie.server.id | Process Server インストールの ID または名前 (rhdm700-decision-server-1 など) |
| org.kie.server.location | Process Server の URL (http://<HOST>:<PORT>/kie-server/services/rest/server) |
| org.kie.server.controller | ヘッドレス Process Automation Manager コントローラーの URL (http://<HOST>:<PORT>/controller/rest/controller) |

<HOST> は、Process Server ホストの ID または名前です (例: **localhost** または **192.7.8.9**)。

<PORT> は Process Server ホストのポートです (例: **7001**)。

6.2. インストールの検証

ヘッドレス Process Automation Manager コントローラーをインストールし、必要なシステムプロパティとロール要件をアプリケーションサーバーに定義したら、ヘッドレス Process Automation Manager コントローラーが正しく動作していることを確認します。

前提条件

- Process Server およびヘッドレス Process Automation Manager コントローラーがアプリケーションサーバーインスタンスにインストールされている。
- アプリケーションサーバーで、ヘッドレス Process Automation Manager コントローラーに必要なシステムプロパティおよびロール要件をすべて設定している。

手順

端末で以下のコマンドを実行して、ヘッドレス Process Automation Manager コントローラーが動作していることを確認します。

```
curl -X GET "http://<HOST>:<PORT>/controller/rest/controller/management/servers" -H "accept: application/xml" -u '<CONTROLLER>:<CONTROLLER_PWD>'
```

<HOST> は、Process Server ホストの ID または名前です (例: **localhost** または **192.7.8.9**)。

<PORT> は Process Server ホストのポートです (例: **7001**)。

<CONTROLLER> および <CONTROLLER_PWD> は、本セクションで作成したユーザーの認証情報です。

このコマンドにより、Process Server インスタンスに関する情報が返されます。



注記

あるいは、Process Server Java API Client を使用して、ヘッドレス Process Automation Manager コントローラーにアクセスすることもできます。

ヘッドレス Process Automation Manager コントローラーが実行していない場合は、アプリケーションサーバーインスタンスを停止して再起動し、ヘッドレス Process Automation Manager コントローラーの URL または API へ再度アクセスしてみてください。

第7章 ORACLE WEBLOGIC SERVER での埋め込みプロセスエンジンまたはデシジョンエンジンの設定

埋め込みエンジンとは、デシジョンとビジネスプロセスを実行できるようにする軽量のワークフローおよびルールエンジンです。埋め込みエンジンは、Red Hat Process Automation Manager アプリケーションに含めるか、OpenShift、Kubernetes および Docker 経由でサービスとしてデプロイできます。また、API を使用して Red Hat Process Automation Manager アプリケーションにエンジンを埋め込むか、または contexts and dependency injection (CDI) サービスセットとして、埋め込むことができます。

Red Hat Process Automation Manager アプリケーションで埋め込みされたエンジンを使用する予定の場合には、Red Hat Business Automation 部品表 (BOM) ファイルをプロジェクトの **pom.xml** ファイルに追加して、プロジェクトに、Maven の依存関係を追加する必要があります。Red Hat Business Automation BOM は、Red Hat Decision Manager および Red Hat Process Automation Manager に適用されます。Red Hat Business Automation BOM に関する詳細は、「[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>7.6.0.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。
 - 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>

```

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>

```

- Process 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の Process Server 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>
```

- プロジェクトで永続サポートのあるプロセスエンジンまたはデシジョンエンジンを使用する場合には、Red Hat Business Automation BOM ファイルから **version.org.hibernate-4ee7** プロパティをコピーして、**pom.xml** ファイルの **dependencyManagement** セクションに以下の Hibernate の依存関係を宣言する必要があります。

Hibernate の依存関係

```
<!-- hibernate dependencies -->
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-entitymanager</artifactId>
      <version>${version.org.hibernate-4ee7}</version>
    </dependency>

    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>${version.org.hibernate-4ee7}</version>
    </dependency>
  </dependencies>
</dependencyManagement>
```

■

第8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 Process 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 Process Server のパスワードを暗号化する必要があります。Business Central と Process 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

Process Server が JCEKS で設定されていない場合には、Process 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

- Process Server が Oracle WebLogic Server にインストールされている。
- 「[Process Server のグループおよびユーザーの設定](#)」の記載通りに、**kie-server** ロールを割り当てた Process Server ユーザーが作成されている。
- Java 8 以降がインストールされている。

手順

1. 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
2. プロンプトが表示されたら、作成した Process Server ユーザーのパスワードを入力します。
 3. 以下の表に記載のシステムプロパティを設定します。

表8.1 Process 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 |
|-----------|----------|----|
|-----------|----------|----|

| 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/key store.jceks |
| kie.keystore.keyStorePwd | <KEYSTORE_PWD> | JCEKS のパスワード |
| kie.keystore.key.server.alias | <KEY_SERVER_ALI AS> | 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_AL IAS> | 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_P WD> | 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 |

4. Process Server を起動して、設定を検証します。

第9章 次のステップ

- 『[Getting started with decision services](#)』
- 『[Designing a decision service using guided decision tables](#)』

付録A バージョン情報

本書の最終更新日: 2019 年 10 月 31 日 (木)