



# Red Hat Process Automation Manager 7.5

## Business Central 設定およびプロパティの設定

ガイド



# Red Hat Process Automation Manager 7.5 Business Central 設定およびプロパティの設定

---

## ガイド

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Configuring\_Business\_Central\_settings\_and\_properties.ent file |.

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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、Red Hat Process Automation Manager 7.6 のさまざまな Business Central の機能をカスタマイズして、管理する方法を説明します。

## 目次

前書き .....	4
<b>第1章 ユーザーおよびグループの管理 .....</b>	<b>5</b>
1.1. ユーザーの作成 .....	5
1.2. ユーザーの編集 .....	6
1.3. グループの作成 .....	6
1.4. グループの編集 .....	6
<b>第2章 セキュリティー管理 .....</b>	<b>8</b>
2.1. セキュリティー管理プロバイダー .....	8
2.1.1. プロパティファイルを基にした Red Hat JBoss EAP セキュリティー管理プロバイダーの設定 .....	8
2.1.2. プロパティファイルと CLI モードを基にした Red Hat JBoss EAP セキュリティー管理プロバイダーの設定 .....	9
2.2. パーミッションおよび設定 .....	10
2.2.1. Business Central でのグループおよびロールのパーミッションの変更 .....	11
2.2.2. Business Central ホームページの変更 .....	12
2.2.3. 優先順位の設定 .....	12
<b>第3章 アーティファクトの管理 .....</b>	<b>13</b>
3.1. アーティファクトの表示 .....	13
3.2. アーティファクトのダウンロード .....	13
3.3. アーティファクトのアップロード .....	13
<b>第4章 データソース管理 .....</b>	<b>15</b>
4.1. データベースドライバーの追加 .....	15
4.2. データベースドライバーの編集 .....	15
4.3. データベースドライバーの削除 .....	15
4.4. データソースの追加 .....	16
4.5. データソースの編集 .....	16
4.6. データソースの削除 .....	16
<b>第5章 データセットのオーサリング .....</b>	<b>18</b>
5.1. データセットの追加 .....	18
5.2. データセットの編集 .....	19
5.3. データの再読み込み .....	19
5.4. キャッシュ .....	19
クライアントキャッシュ .....	20
バックエンドキャッシュ .....	20
<b>第6章 プロジェクト設定のカスタマイズ .....</b>	<b>21</b>
<b>第7章 アーティファクトリーポジトリのカスタマイズのプロパティ .....</b>	<b>23</b>
<b>第8章 言語のカスタマイズ設定 .....</b>	<b>24</b>
<b>第9章 プロセス管理のカスタマイズ .....</b>	<b>25</b>
<b>第10章 PROCESS DESIGNER のカスタマイズ .....</b>	<b>26</b>
<b>第11章 SSH キー .....</b>	<b>27</b>
11.1. デフォルトの SSH キーストア .....	27
11.2. カスタムの SSH キーストア .....	27
11.3. SSH キーの作成 .....	28
11.4. SSH キーストアを使用した SSH 公開鍵の登録 .....	28
11.5. SSH キーの削除 .....	29

第12章 BUSINESS CENTRAL でのサービスタスクの管理 .....	30
第13章 DASHBUILDER データのエクスポートおよびインポート .....	34
13.1. DASHBUILDER データのエクスポート .....	34
13.2. DASHBUILDER データのインポート .....	34
第14章 LDAP 接続 .....	36
14.1. LDAP USERGROUPCALLBACK 実装 .....	37
関連情報 .....	38
第15章 データベース接続 .....	39
15.1. データベースの USERGROUPCALLBACK 実装 .....	39
関連情報 .....	40
第16章 SETTINGS.XML ファイルを使用した MAVEN の設定 .....	41
関連情報 .....	41
第17章 GAV チェック管理 .....	42
17.1. GAV チェックおよび子の GAV エディションの設定 .....	42
17.2. 全プロジェクトの GAV チェックの設定 .....	42
第18章 PROCESS SERVER および BUSINESS CENTRAL での環境モードの設定 .....	44
第19章 GIT フックおよびリモート GIT リポジトリの統合 .....	45
19.1. POST-COMMIT の GIT フックの作成 .....	45
19.2. リモート GIT リポジトリのインポート .....	46
19.3. 既存のリモート GIT プロジェクトリポジトリ用の GIT フックの設定 .....	47
19.4. BUSINESS CENTRAL のシステムプロパティとしての GIT フックの設定 .....	48
19.5. リモート GIT リポジトリの統合 .....	49
19.6. GIT フックの終了コード .....	52
19.7. GIT フック通知のカスタマイズ .....	52
19.7.1. Business Central の Git フック通知 .....	53
19.7.2. Git フック通知の国際化サポート .....	53
第20章 BUSINESS CENTRAL のブランチでのロールベースアクセス制御 .....	55
20.1. ロールベースのブランチアクセスのカスタマイズ .....	55
第21章 プロセスインスタンスログの表示 .....	56
第22章 BUSINESS CENTRAL システムプロパティ .....	57
第23章 BUSINESS CENTRAL 使用時のパフォーマンスチューニングに関する考慮点 .....	63
付録A バージョン情報 .....	64



## 前書き

管理者は、管理者の **Settings** ページで以下をカスタマイズできます。

- **Roles:** ロールのホームページ、優先順位、パーミッションを設定します。
- **Groups:** グループのホームページ、優先順位、およびパーミッションを設定し、グループの作成および削除を行います。
- **Users:** ユーザーの作成や削除、ユーザーに対するグループおよびロールの追加または削除、およびユーザーパーミッションの表示を行います。
- **Artifacts:** M2 リポジトリーアーティファクトの表示、アーティファクトのアップロード、JAR ファイルの表示とダウンロードを行います。
- **Data Sources:** データソースおよびデータベースドライバーの追加、更新、または削除を行います。
- **Data Sets:** データセットの作成、変更、または削除を行います。
- **Projects:** ファイルエクスポートプロパティ、スペースプロパティ、デフォルト値、GAV 詳細プロパティなどプロジェクトの設定を表示し、編集します。
- **Artifact Repository:** アーティファクトリポジトリプロパティを管理します。
- **Artifact Repository:** Business Central の言語を設定します。
- **Process Administration:** Business Central でデフォルトのページネーションオプションを設定します。
- **Process Designer:** ダイアグラムエディタープロパティを設定します。
- **SSH Keys:** SSH キーを追加または削除します。
- **Service Tasks Administration:** デフォルトのサービスタスクを有効化または無効化して、カスタムのサービスタスクをアップロードします。
- **Profiles:** ワークベンチプロファイルを **Planner and Rules** または **Full** に設定します。

### 前提条件

- Red Hat JBoss Enterprise Application Platform 7.2.0 がインストールされている。詳細は『[Red Hat JBoss Enterprise Application Platform 7.2 インストールガイド](#)』を参照してください。
- Red Hat Process Automation Manager をインストールして実行している。詳細は、『[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](#)』を参照してください。
- Business Central に **admin** ユーザーロールでログインします。



## 第1章 ユーザーおよびグループの管理

Business Central は、ユーザー、グループ、およびロールのセキュリティー管理のエンティティーを 3 種類定義します。パーミッションは、ロールにもグループにも両方割り当てることができます。Business Central には以下のロールがあります。

- process-admin
- manager
- admin
- analyst
- developer
- user



### 注記

ロールレジストリーというアプリケーションのユーザーロールには、ロールの識別子がありますが、ユーザーグループにはありません。

Business Central を使用して、必要数のユーザーとグループを作成し、管理できます。ユーザーは、最低でもユーザー固有のロールを 1 つ以上割り当てて、Business Central にログインできるようにする必要があります。ユーザーの権限は、ユーザーが所属するグループとロールからのパーミッションにより異なります。ユーザーに複数のロールまたはグループが割り当てられている場合は、ロールまたはグループの優先順位が考慮される点に注意してください。

### 1.1. ユーザーの作成

ユーザーの権限および設定は、ユーザーに割り当てたロールと、ユーザーが属するグループで制御されます。Business Central で、ユーザーをいくつでも作成できます。



### 注記

プロセスエンジンまたは Process Server で **unknown** という名前のユーザーは作成しないでください。**unknown** ユーザーアカウントは、superuser のアクセス権限があるシステム名用に予約されています。**unknown** ユーザーアカウントでは、ログインしているユーザーがない場合に、SLA 違反リスナーに関連するタスクを実行します。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Users** を選択します。
2. **New user** をクリックし、ユーザー名を入力し、**Next** をクリックします。
3. ユーザーにロールを割り当てるには、**Roles** タブをクリックして、**Add Roles** をクリックし、任意のロールを選択してから、**Add to selected roles** をクリックします。
4. 必要に応じて、ユーザーにグループを割り当てるには、**Groups** タブをクリックして **Add to groups** をクリックし、任意のグループを選択してから **Add to selected groups** をクリックします。
5. **Create** をクリックします。

6. **Yes** をクリックして、ユーザーにパスワードを設定し、**Change** をクリックします。



#### 注記

ユーザーが Business Central にアクセスするために1つ以上のロールが必要です。

## 1.2. ユーザーの編集

Business Central **Settings** ページの **Users** オプションを使用して、ユーザーのグループとロールを変更できます。全ユーザーパーミッションは、ユーザーのグループとロールパーミッションを元にしています。ユーザーのパーミッションは、**Permissions** タブから確認できます。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Users** を選択します。
2. **All users** リストから、編集するユーザーをクリックします。右のペインにユーザーの情報が表示されます。
3. **Edit** をクリックして、以下のタスクを実行します。
  - ユーザーのグループを変更するには、**Groups** タブをクリックして、**Add to groups** をクリックします。次に、ユーザーを所属させるグループを選択して **Add to selected groups** をクリックし、**Save** をクリックします。
  - ユーザーのロールを変更するには、**Roles** タブをクリックし、**Add roles** をクリックします。ユーザーに割り当てるロールを選択して、**Add to selected roles** をクリックし、**Save** をクリックします。
  - ユーザーパーミッションを表示するには、**Permissions** タブをクリックして属性を展開します。
  - パスワードを変更するには、**Change Password** をクリックして、新規パスワードを入力し、**Change** をクリックします。
  - ユーザーを削除するには、**Delete** をクリックしてから、**Yes** をクリックし、削除を確認します。

## 1.3. グループの作成

Business Central では、グループを使用してユーザーをまとめて、パーミッションを制御できます。任意の数のグループを作成できますが、グループには最低でもユーザーを1つ所属させる必要があります。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Groups** を選択します。
2. **New group** をクリックし、グループ名を入力し、**Next** をクリックします。
3. このグループに追加するユーザーを選択し、**Add selected users** をクリックします。新規作成したグループは、**All groups** に表示されます。

## 1.4. グループの編集

必要に応じて、ホームページ、優先順位、パーミッションなどグループの属性を編集できます。Business Central の **Settings** の **Groups** オプションから、グループの変更や削除が可能です。

## 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Groups** を選択します。
2. **All groups** 一覧から、編集するグループをクリックします。右のペインにユーザーの情報が表示されます。
3. **Home Page** リストからホームページを選択します。
4. **Priority** リストから優先順位を選択します。
5. **Permissions** セクションから、リソース属性を展開して、パーミッションを変更します。



### 注記

**Pages** パーミッション、**Editor** パーミッション、**Spaces** パーミッション、および **Projects** パーミッションに例外を追加できます。

6. **Save** をクリックして変更を適用します。

## 第2章 セキュリティー管理

セキュリティー管理とは、ユーザー、グループ、パーミッションを管理するプロセスです。Business Central セキュリティー管理ページから Business Central のリソースおよび機能へのアクセスを制御できます。

Business Central は、ユーザー、グループ、およびロールのセキュリティー管理のエンティティーを 3 種類定義します。パーミッションは、ロールにもグループにも両方割り当てることができます。ユーザーは、所属するグループおよびロールのパーミッションを継承します。

### 2.1. セキュリティー管理プロバイダー

レルムは、セキュリティー管理のコンテキストで各種アプリケーションリソースへのアクセスを制限します。レルムには、ユーザー、グループ、ロール、パーミッションに関する情報が含まれます。特定のレルムに対する具体的なユーザーおよびグループ管理サービスの実装は、セキュリティー管理プロバイダーと呼ばれます。

組み込みのセキュリティー管理プロバイダーがアプリケーションセキュリティーレルムの要件を満たさない場合は、独自のセキュリティー管理プロバイダーを構築して登録できます。



#### 注記

セキュリティー管理プロバイダーがインストールされていない場合は、セキュリティーレルムを管理するユーザーインターフェイスは利用できません。セキュリティー管理プロバイダーをインストールして設定した後に、セキュリティー管理ユーザーインターフェイスでユーザーおよびグループの管理機能は自動的に有効になります。

Business Central には、**application-users.properties** または **application-roles.properties** プロパティファイルの内容を基にレルムタイプをサポートする Red Hat JBoss EAP セキュリティー管理プロバイダーが含まれます。

#### 2.1.1. プロパティファイルを基にした Red Hat JBoss EAP セキュリティー管理プロバイダーの設定

独自の Red Hat JBoss EAP セキュリティー管理プロバイダーを構築して登録できます。プロパティファイルを基にして Red Hat JBoss EAP セキュリティー管理プロバイダーを使用するには、以下の手順を行います。

##### 前提条件

- Red Hat JBoss EAP がインストールされている。

##### 手順

- Red Hat JBoss EAP インスタンスの既存のユーザーまたはロールプロパティファイルを使用するには、以下の例で示すように、**EAP\_HOME/standalone/configuration/application-users.properties** および **EAP\_HOME/standalone/configuration/application-roles.properties** ファイルに以下のシステムプロパティを含めます。

```
<property name="org.uberfire.ext.security.management.wildfly.properties.realm"
value="ApplicationRealm"/>
<property name="org.uberfire.ext.security.management.wildfly.properties.users-file-path"
```

```
value="/standalone/configuration/application-users.properties"/>
<property name="org.uberfire.ext.security.management.wildfly.properties.groups-file-path"
value="/standalone/configuration/application-roles.properties"/>
```

以下の表は、これらのプロパティーの説明とデフォルト値を示しています。

表2.1 プロパティーファイルを基にする Red Hat JBoss EAP セキュリティー管理プロバイダー

プロパティー	説明	デフォルト値
<b>org.uberfire.ext.security.m anagement.wildfly.propert ies.realm</b>	レルムの名前このプロパ ティーは必須ではありません。	<b>ApplicationRealm</b>
<b>org.uberfire.ext.security.m anagement.wildfly.propert ies.users-file-path</b>	ユーザープロパティーファイ ルの絶対パス。このプロパ ティーは必須です。	<b>./standalone/configuration /application- users.properties</b>
<b>org.uberfire.ext.security.m anagement.wildfly.propert ies.groups-file-path</b>	グループプロパティーファイ ルの絶対パス。このプロパ ティーは必須です。	<b>./standalone/configuration /application- roles.properties</b>

- アプリケーションのルートディレクトリーに **security-management.properties** ファイルを作成します。たとえば、以下のファイルを作成します。

```
src/main/resources/security-management.properties
```

- security-management.properties** ファイルの値として、以下のシステムプロパティーおよびセキュリティープロバイダー名を入力します。

```
<property name="org.uberfire.ext.security.management.api.userManagementServices"
value="WildflyUserManagementService"/>
```

### 2.1.2. プロパティーファイルと CLI モードを基にした Red Hat JBoss EAP セキュリ ティー管理プロバイダーの設定

プロパティーファイルと CLI モードを基に Red Hat JBoss EAP セキュリティー管理プロバイダーを使用するには、以下の手順を行います。

#### 前提条件

- Red Hat JBoss EAP がインストールされている。

#### 手順

- Red Hat JBoss EAP インスタンスの既存のユーザーまたはロールプロパティーファイルを使用するには、以下の例で示すように、**EAP\_HOME/standalone/configuration/application-users.properties** および **EAP\_HOME/standalone/configuration/application-roles.properties** ファイルに以下のシステムプロパティーを含めます。

```
<property name="org.uberfire.ext.security.management.wildfly.cli.host" value="localhost"/>
<property name="org.uberfire.ext.security.management.wildfly.cli.port" value="9990"/>
```

```
<property name="org.uberfire.ext.security.management.wildfly.cli.user" value="
<USERNAME>"/>
<property name="org.uberfire.ext.security.management.wildfly.cli.password" value="
<USER_PWD>"/>
<property name="org.uberfire.ext.security.management.wildfly.cli.realm"
value="ApplicationRealm"/>
```

以下の表は、これらのプロパティの説明とデフォルト値を示しています。

表2.2 プロパティファイルと CLI モードを基にする Red Hat JBoss EAP セキュリティー管理プロバイダー

プロパティ	説明	デフォルト値
<b>org.uberfire.ext.security.management.wildfly.cli.host</b>	ネイティブ管理インターフェイスホスト。	<b>localhost</b>
<b>org.uberfire.ext.security.management.wildfly.cli.port</b>	ネイティブ管理インターフェイスポート。	<b>9990</b>
<b>org.uberfire.ext.security.management.wildfly.cli.user</b>	ネイティブ管理インターフェイスのユーザー名。	NA
<b>org.uberfire.ext.security.management.wildfly.cli.password</b>	ネイティブ管理インターフェイスのユーザーのパスワード。	NA
<b>org.uberfire.ext.security.management.wildfly.cli.realm</b>	アプリケーションのセキュリティーコンテキストで使用されるレルム。	<b>ApplicationRealm</b>

2. アプリケーションのルートディレクトリーに **security-management.properties** ファイルを作成します。たとえば、以下のファイルを作成します。

```
src/main/resources/security-management.properties
```

3. **security-management.properties** ファイルの値として、以下のシステムプロパティおよびセキュリティープロバイダー名を入力します。

```
<property name="org.uberfire.ext.security.management.api.userManagementServices"
value="WildflyCLIUserManagementService"/>
```

## 2.2. パーミッションおよび設定

パーミッションは、アプリケーション内の特定のリソースに関連するアクションを実行するためにユーザーに付与される権限です。たとえば、以下のパーミッションを指定できます。

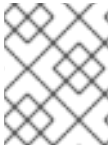
- ページを表示する。
- プロジェクトを保存する。
- リポジトリーを削除する。

- ダッシュボードを削除する。

パーミッションは、付与と拒否ができ、グローバルに設定することも、リソースを指定して設定することもできます。パーミッションを使用すると、リソースへのアクセス時のセキュリティーが保護され、アプリケーション内の機能をカスタマイズできます。

### 2.2.1. Business Central でのグループおよびロールのパーミッションの変更

Business Central では、個人ユーザーに対するパーミッションは変更できません。ただし、グループおよびロールのパーミッションは変更できます。変更したパーミッションは、変更したロールが割り当てられているか、変更したグループに所属するユーザーに適用されます。



#### 注記

ロールまたはグループへの変更は、そのロールまたはグループに関連のあるユーザーに加えられます。

#### 前提条件

- Business Central に **admin** ユーザーロールでログインします。

#### 手順

1. Business Central で **Security management** ページにアクセスするには、画面の右上隅にある **Admin** アイコンを選択します。
2. Business Central **Settings** ページで **Roles**、**Groups**、または **Users** をクリックします。クリックしたアイコンのタブに、**Security management** ページが開きます。
3. リストから編集するロールまたはグループをクリックします。全詳細が右側のペインに表示されます。
4. **Settings** セクションの **Home Page** または **Priority** を設定します。
5. **Permissions** セクションで、Business Central、ページ、エディター、スペース、プロジェクトのパーミッションを設定します。

#### 図2.1パーミッションの設定

admin settings

Home Page ⓘ

Priority ⓘ

Permissions

---

> Workbench ⓘ

Pages ⓘ

<input type="text" value="Read"/>	<input type="text" value="Update"/>	<input type="text" value="Delete"/>	<input type="text" value="Create"/>
-----------------------------------	-------------------------------------	-------------------------------------	-------------------------------------

[Add Exception](#)

Editors ⓘ

<input type="text" value="Read"/>
-----------------------------------

Spaces ⓘ

<input type="text" value="Read"/>	<input type="text" value="Update"/>	<input type="text" value="Delete"/>	<input type="text" value="Create"/>
-----------------------------------	-------------------------------------	-------------------------------------	-------------------------------------

Projects ⓘ

<input type="text" value="Read"/>	<input type="text" value="Update"/>	<input type="text" value="Delete"/>	<input type="text" value="Create"/>	<input type="text" value="Build"/>
-----------------------------------	-------------------------------------	-------------------------------------	-------------------------------------	------------------------------------

6. 変更するパーミッションのリソースタイプの横にある矢印をクリックして展開します。

7. 必要に応じて、リソースタイプに例外を追加するには、**Add Exception** をクリックしてから、必要なパーミッションを設定します。



#### 注記

Business Central のリソースタイプには、例外を追加できません。

8. **Save** をクリックします。

### 2.2.2. Business Central ホームページの変更

ホームページは、Business Central にログインすると表示されるページです。デフォルトでは、ホームページは **Home** に設定されます。ロールとグループ別に異なるホームページを指定できます。

#### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Roles** または **Groups** を選択します。
2. ロールまたはグループを選択します。
3. **Home Page** リストからページを選択します。
4. **Save** をクリックします。



#### 注記

そのロールまたはグループには、ページをホームページにする前に、そのページへの読み取りアクセスが必要です。

### 2.2.3. 優先順位の設定

ユーザーは、複数のロールを持ち、複数のグループに所属します。優先順位の設定は、ロールまたはグループの優先順を決定します。

#### 前提条件

- Business Central に **admin** ユーザーロールでログインします。

#### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Roles** または **Groups** を選択します。
2. ロールまたはグループを選択します。
3. 優先順位メニューから優先順位を選択し、**Save** をクリックします。



#### 注記

ユーザーに、設定が競合するロールが割り当てられているか、グループに所属している場合は、一番高い優先順位を持つロールまたはグループを設定します。



## 第3章 アーティファクトの管理

Business Central の **Artifacts** ページからアーティファクトを管理できます。アーティファクトリーポジトリーは、ローカルの Maven リポジトリーで、インストールごとに Maven リポジトリーは1つのみとなっています。Business Central は、**Sonatype Nexus™**、**Apache Archiva™**、**JFrog Artifactory™** などの Maven リポジトリーソリューションを使用することを推奨します。

**Artifacts** ページでは、Maven リポジトリーの全アーティファクトをすべて表示します。アーティファクトは、Maven リポジトリーにアップロードできます。



### 注記

**Artifacts** リポジトリーにアップロードできるのは、JAR ファイル、KJAR ファイル、および **pom.xml** ファイルのみです。

### 3.1. アーティファクトの表示

**Artifacts** ページからローカルの maven リポジトリーのコンテンツをすべて確認できます。

#### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Artifacts** を選択します。
2. **Open** をクリックしてアーティファクトの詳細を表示します。
3. **Ok** をクリックして **Artifacts** ページに戻ります。

### 3.2. アーティファクトのダウンロード

Business Central のリポジトリーからプロジェクトのローカルストレージにアーティファクトをダウンロードして保存できます。

#### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Artifacts** を選択します。
2. **Download** をクリックします。
3. アーティファクトを保存するディレクトリーを参照します。
4. **Save** をクリックします。

### 3.3. アーティファクトのアップロード

ローカルストレージから Business Central のプロジェクトにアーティファクトをアップロードできます。

#### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Artifacts** を選択します。
2. **Upload** をクリックします。

3. **Choose File** をクリックして、アーティファクトをアップロードするディレクトリーに移動して選択します。
4. **Upload** をクリックします。



#### 注記

Maven 以外のアーティファクトを使用する場合は、先に **mvn deploy** コマンドを使用してアーティファクトを Maven リポジトリーにデプロイしてから Business Central のアーティファクト一覧を更新します。

## 第4章 データソース管理

Business Central ではデータソースの管理機能があり、データソースを定義してデータベースにアクセスできます。これらのデータソースは、データセットなど、他の Business Central コンポーネントが使用します。また、データベースドライバーは、データソースと対象のデータベースの間の通信を有効にするために使用します。

**Data Source Authoring** ページから、Business Central にデータソースとデータベースのドライバーを追加できます。



### 注記

Business Central にはデフォルトのデータソースが含まれており、これは使用可能ですが、編集または削除することができません。

### 4.1. データベースドライバーの追加

Business Central に新規データベースドライバーを追加できます。

#### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Data Sources** を選択します。
2. **DataSource Explorer** ペインで **Add Driver** をクリックします。New driver ウィンドウが開きます。
3. **New driver** ウィンドウで、データベースドライバーの **Name**、**Driver Class Name**、**Group Id**、**Artifact Id**、および **Version** を入力します。
4. **Finish** をクリックして Business Central にドライバーを追加します。

### 4.2. データベースドライバーの編集

**Driver Definition** ペインからデータベースドライバーのプロパティを更新できます。

#### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Data Sources** を選択します。
2. **DataSource Explorer** ペインで編集するドライバーを選択します。
3. **Driver Definition** ペインで、**Name** フィールド、**Driver Class Name** フィールド、**Group Id** フィールド、**Artifact Id** フィールド、および **Version** フィールドに必要な変更を加えます。
4. **Update** をクリックします。
5. **Yes** をクリックしてドライバーへの変更を保存します。

### 4.3. データベースドライバーの削除

Business Central の **Data Source Definition** ペインからデータベースドライバーを削除できます。

## 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Data Sources** を選択します。
2. **DataSource Explorer** ペインで削除するドライバーを選択します。
3. **Remove** をクリックします。
4. **Delete** をクリックしてドライバーを削除します。

## 4.4. データソースの追加

**Data Sources Authoring** ページから Business Central に新しいデータソースを追加できます。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Data Sources** を選択します。
2. **DataSource Explorer** ペインで **Add DataSource** をクリックします。**New data source** ウィンドウが開きます。
3. **New data source** ウィンドウで、データソースの **Name**、データベースの **Connection URL**、**User** と **Password**、ならびに **Driver** を入力します。
4. **Test Connection** をクリックして、データベースへの接続を確認します。
5. **Finish** をクリックして Business Central へデータソースを追加します。

## 4.5. データソースの編集

Business Central でデータソースのプロパティを編集し、データベースへの接続をテストできます。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Data Sources** を選択します。
2. **DataSource Explorer** ペインで編集するデータソースを選択します。
3. **Data Source Definition** ペインで、**Name** フィールド、**Connection URL** フィールド、**User** フィールド、**Password** フィールド、および **Driver** フィールドに必要な変更を加えます。
4. **Test Connection** をクリックして、データベースへの接続を確認します。
5. **Update** をクリックします。
6. **Save** をクリックして、データソースへの変更を確定します。

## 4.6. データソースの削除

Business Central の **DataSource Explorer** ペインから既存のデータソースを削除できます。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Data Sources** を選択します。
2. **DataSource Explorer** ペインで削除するデータソースを選択します。
3. **Remove** をクリックします。
4. **Delete** をクリックして、データソースの削除を確定します。

## 第5章 データセットのオーサリング

データセットは情報の関連セットの集まりです。データベース、Microsoft Excel ファイル、メモリーなど、さまざまな方法で保存できます。データセット定義は、Business Central メソッドにデータセットへのアクセス、読み取り、および解析を指示します。Business Central はデータを保存しません。データが保存される場所にかかわらず、データセットへのアクセスを定義できます。

たとえば、データベースにデータが保存されると、有効なデータセットには、SQL クエリーの結果として、データベース全体またはデータベースのサブセットが含まれます。いずれの場合も、データは、情報を表示する Business Central のレポーティングコンポーネントの入力情報として使用されます。

データセットにアクセスするには、データセット定義を作成および登録する必要があります。このデータセットの定義では、データセットの場所と、その場所へのアクセス、読み取り、および解析の方法、ならびにデータセットが含まれるコラムを指定します。



### 注記

Data Sets ページは、admin ロールを持つユーザーにのみ表示されます。

### 5.1. データセットの追加

外部データソースからデータを取得して、レポーティングコンポーネントでデータを使用するデータセットを作成できます。

#### 手順

1. Business Central で、**Admin** → **Data Sets** に移動します。 **Data Set Explorer** ページが開きます。
2. **New Data Set** をクリックして、以下のプロバイダタイプから1つ選択します。
  - **Bean**: Java クラスからデータセットを生成するのに使用します。
  - **CSV**: リモートまたはローカルの CSV ファイルからデータセットを生成するのに使用します。
  - **SQL**: ANSI-SQL 準拠データベースからデータセットを生成するのに使用します。
  - **Elastic Search**: Elastic Search ノードからデータセットを生成するのに使用します。
  - **Execution Server**: Execution Server のカスタムのクエリー機能を使用してデータセットを生成します。



### 注記

KIE Server は、このオプションを使用して設定する必要があります。

3. **Data Set Creation Wizard** を完了し、**Test** をクリックします。



### 注記

設定手順は、選択するプロバイダーにより異なります。

4. **Save** をクリックします。

## 5.2. データセットの編集

既存のデータセットを編集し、レポートコンポーネントに取得したデータが最新になっていることを確認します。

### 手順

1. Business Central で、**Admin** → **Data Sets** に移動します。 **Data Set Explorer** ページが開きます。
2. **Data Set Explorer** ペインで、編集するデータセットを検索し、**Edit** をクリックします。
3. **Data Set Editor** ペインで、適切なタブを使用して必要に応じてデータを編集します。タブは、選択するデータセットプロバイダーの種類によって異なります。  
たとえば、**CSV** データプロバイダーの編集には、以下の変更が適用できます。
  - **CSV Configuration**: データセット定義の名前、ソースファイル、区切り記号などのプロパティを変更できます。
  - **Preview**: データのプレビューを使用できます。 **CSV Configuration** タブで **Test** をクリックすると、システムはデータセットのルックアップコールを実行し、データが利用可能な場合はプレビューが表示されます。 **Preview** タブには2つのサブタブがあります。
    - **Data columns**: どの列をデータセット定義に追加するかを指定できます。
    - **Filter**: 新しいフィルターを追加できます。
  - **Advanced**: 以下の設定を管理できます。
    - **Caching**: 詳細は「[キャッシュ](#)」を参照してください。
    - **Cache life-cycle**: データセット (またはデータ) を再読み込みされるまでの間隔を指定できます。バックエンドデータに変更が加えられると、**Refresh on stale data**機能は、キャッシュしたデータを再読み込みします。
4. 必要な変更を行ったら、**Validate** をクリックします。
5. **Save** をクリックします。

## 5.3. データの再読み込み

データの再読み込み機能を使用すると、データセット (またはデータ) を再読み込みされるまでの間隔を指定できます。バックエンドデータに変更が加えられると、**Refresh on stale data**機能は、キャッシュしたデータを再読み込みします。

## 5.4. キャッシュ

Business Central は、インメモリーデータを使用してデータセットを保存し、データ操作を実行するキャッシュメカニズムを提供します。データのキャッシュにより、ネットワークトラフィック、リモートシステムのペイロード、処理時間が減ります。パフォーマンスの問題を回避するには、Decision Central にキャッシュを設定します。

データセットを生成するデータルックアップ呼び出しの場合、キャッシュメソッドは、データルックアップ呼び出しが実行される場所と結果のデータセットが格納される場所を決定します。データのルックアップコールの例としては、ロケールパラメーターを **Urban** として設定するすべての住宅ローンアプリケーションが挙げられます。

Business Central データセット機能には、キャッシュレベルが2つあります。

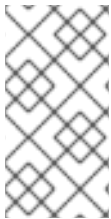
- クライアントレベル
- バックエンドレベル

### クライアントキャッシュ

キャッシュを有効にすると、データセットはルックアップ操作時に Web ブラウザーにキャッシュされ、その後のルックアップ操作ではバックエンドへの要求が実行されません。グループ化、集計、フィルターリング、並べ替えなどのデータセット操作は Web ブラウザーで処理されます。クライアントのキャッシュは、データセットのサイズが小さい場合 (例: データが 10 MB より少ない) にのみ有効になります。データセットが大きい場合は、パフォーマンスの低下や断続的なフリーズなどのブラウザー問題が発生する場合があります。クライアントのキャッシュは、ストレージシステムへの要求などの、バックエンド要求の数を減らします。

### バックエンドキャッシュ

キャッシュが有効な場合に、デシジョンエンジンはデータセットをキャッシュします。これにより、リモートのストレージシステムへのバックエンドの要求数が減ります。データセットの全操作は、インメモリーデータを使用してデシジョンエンジンで実行されます。バックエンドキャッシュは、データセットのサイズが頻繁に更新され、インメモリーに保存されて処理される場合に限り有効です。バックエンドキャッシングは、リモートストレージを使用した低レイテンシー接続問題が発生している状況でも有効です。



### 注記

バックエンドキャッシュの設定は、**Data Set Editor** の **Advanced** タブに常に表示されるわけではありません。これは、インメモリーデシジョンエンジンでデータのルックアップ操作を解決するのに、Java および CSV のデータプロバイダーはバックエンドキャッシュに依存するためです (データセットはメモリー内に存在する必要があります)。



## 第6章 プロジェクト設定のカスタマイズ

プロジェクトにはアセットが保存され、スペースの一部になります。スペースでは、複数のプロジェクトを保持できます。

たとえば、組織には HR、給与、エンジニアリング、R&D など、多くの部署が含まれます。各部署はスペースにマッピングされ、部署ごとに独自のプロジェクトが割り当てられます。

最初から新規プロジェクトを作成するか、既存の Git リポジトリからプロジェクトをクローンできます。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Projects** を選択します。**Projects** ページが開きます。
2. **Project Preferences** ペインで変更する設定をクリックします。以下の設定を利用できます。
  - **File exporting**: この設定には以下のプロパティーが含まれます。

表6.1 ファイルエクスポートのプロパティー

フィールド	説明
PDF の向き	PDF の向きが横向きか、縦向きかを指定します。
PDF の単位	PDF の単位が PT、MM、CN、または IN のいずれかを指定します。
PDF ページ形式	PDF ページ形式が A[0-10]、B[0-10]、または C[0-10] のいずれかに指定します。

- **Spaces**: この設定には以下のプロパティーが含まれます。

表6.2 スペースのプロパティー

フィールド	説明
名前	スペースのデフォルト名。存在しない場合は自動的に作成されます。
所有者	スペースのデフォルトの所有者。存在しない場合は自動的に作成されます。
グループ ID	スペースのデフォルトグループ ID。存在しない場合は自動的に作成されます。
エイリアス (単数)	スペースのカスタマイズエイリアス (単数) を指定します。

フィールド	説明
エイリアス (複数)	スペースのカスタマイズエイリアス (複数) を指定します。

- **Default values:** この設定には以下のプロパティが含まれます。

表6.3 デフォルト値のプロパティ

フィールド	説明
バージョン	クイック設定オプションを使用してプロジェクトを作成する場合のプロジェクトのデフォルトバージョン番号
説明	クイック設定オプションを使用してプロジェクトを作成する場合のプロジェクトのデフォルトの説明
ブランチ	Git リポジトリを使用時に使用するデフォルトのブランチ

- **Advanced GAV preferences:** この設定には以下のプロパティが含まれます。

表6.4 GAV 詳細設定のプロパティ

フィールド	説明
GAV の競合チェックを無効にしますか？	GAV の競合チェックを有効化するか、無効化するか指定します。この機能を無効にすると、複数のプロジェクトに同じ GAV (グループ ID、アーティファクト、バージョン) を指定できます。
GAV の子エディションを許可しますか？	GAV エディションに対する子/サブプロジェクトを許可するかどうかを指定します。



### 注記

重複する GAV の検出は、**Development Mode** のプロジェクトでは無効になっています。Business Central のプロジェクトで重複する GAV 検出を有効にするには、プロジェクトの **Settings** → **General Settings** → **Version** に移動して、**Development Mode** オプションを **OFF** (該当する場合) に切り替えます。

3. **Save** をクリックします。

## 第7章 アーティファクトリポジトリーのカスタマイズのプロパティ

場合によっては、ドメインモデルの JAR ファイルをビルドするのに、プロジェクトで外部の依存関係を解決する必要があります。リポジトリーには、以下の機能を持つアーティファクトが必要です。

- リポジトリーは、Maven リポジトリーである
- 全スナップショットにはタイムスタンプが含まれる
- アセットはほぼ、ローカルのハードドライブに保存されている

デフォルトでは、アーティファクトのリポジトリーは `$WORKING_DIRECTORY/repositories/kie` に含まれます。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Artifact Repository** を選択します。**Artifact Repository** ページが開きます。
2. いずれかを選択して、**Properties** セクションに情報を入力します。
3. **Save** をクリックします。

## 第8章 言語のカスタマイズ設定

Business Central の **Settings** ページで、言語を変更できます。Business Central は以下の言語をサポートします。

- 英語
- ドイツ語
- スペイン語
- フランス語
- 日本語
- ポルトガル語
- 簡体字中国語

デフォルトの言語は英語です。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Lauguages** を選択します。**Language Selector** が開きます。
2. **Language** リストから希望の言語を選択します。
3. **OK** をクリックします。

## 第9章 プロセス管理のカスタマイズ

Business Central では、**Process Administration** ページの **Default items per page** プロパティを編集して、デフォルトのページオプションをカスタマイズできます。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Process Administration** を選択します。
2. **Properties** セクションから **Default items per page** プロパティを更新して、**Save** をクリックします。



### 注記

ページごとに表示するアイテムは 10、20、50、または 100 のいずれかを指定できます。

## 第10章 PROCESS DESIGNER のカスタマイズ

Business Central では、**Settings** ページからダイアグラムエディターのプロパティを編集して、Process Designer をカスタマイズできます。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Process Designer** を選択します。
2. **Properties** セクションで、以下のプロパティのいずれかを更新します。
  - **Auto hide category panel** チェックボックスを選択して、自動的にカテゴリーツールバーパネルが非表示になるようにします。
  - 描画エリアの幅を設定するには、**Drawing area width** フィールドで、2800 から 5600 までの整数値を入力します。
  - 描画エリアの高さを設定するには、**Drawing area height** フィールドで、1400 から 2800 までの整数値を入力します。
  - 高解像度のディスプレイを使用しており、文字やオブジェクトがぼやけて表示される場合には、**Enable HiDPI** チェックボックスを選択します。このオプションはデフォルトで無効になっています。
3. **Save** をクリックします。

## 第11章 SSH キー

Business Central には SSH キーストアサービスがあり、ユーザーの SSH 認証を有効にします。また、設定可能なデフォルトの SSH キーストアと拡張可能な API (カスタムの実装) が含まれており、複数の SSH の公開鍵形式をサポートします。

SSH 公開鍵を登録するには、Business Central の **Settings** ページから **SSH Keys** オプションにアクセスできます。

### 11.1. デフォルトの SSH キーストア

Business Central に含まれるデフォルトの SSH キーストアでは、ユーザーの公開鍵を保存するのにファイルベースのストレージメカニズムが採用されています。デフォルトでは、Business Central は **\*.security** ディレクトリーを root ディレクトリーとして使用します。ただし、別のディレクトリーを参照するように、**appformer.ssh.keys.storage.folder** システムプロパティーの値を設定することで、カスタムのストレージパスを使用できます。

SSH 公開鍵は **{securityFolderPath}/pkeys/{userName}/** ディレクトリー構造に保存されます。

各 SSH 公開鍵は以下のファイルで設定されており、これらのファイルはストレージフォルダーに配置されています。

- **{keyId}.pub**: このファイルには SSH 公開鍵のコンテンツが含まれます。ファイル名で、システムのロジックキー ID がわかるので、ファイル名がランタイム時に変更されないようにしてください。以下に例を示します。

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAQDmak4Wu23RZ6XmN94bOsqecZxuTa4RRhhQm
HmTZjMB7HM57/90u/B/gB/GhsPEu1nAXL0npY56tT/MPQ8vRm2C2W9A7CzN5+z5yyL3W01Y
Zy3kzslk77CjULjfhrcfQSL3b2sPG5jv5E5/nyC/swSytucwT/PE7aXTS9H6cHIKUdYPzlt94SHoBx
WRIK7PjI9d+eLB+hmDzvbVa1ezu5a8yu2kcHi6Nxxf15iRj2rsceDTp0imC1jMoC6ZDfBvZSxL9F\
TMwFdNmTIJveBtv9nAbnAvIWiiS0VOKdj1s3GxBxeZYAcKbcsK9sJzusptk5dxGsG2Z8vInaglN
6OaOQ7b7tcomzCYYwviGQ9gRX8sGsVrw39gsDIGYP2tA4bRr7ecHnlNg1b0HCchA5+QCdk
4Hbz1UrnHmPA2Lg9c3Wgm2qedvQdVJXuS3mlwYOqL40aXPs6890PvFJUlpivsZnF50djPnws
MxJZEf1HdTXgZD1Bh54ogZf7czyUNfkNkE69yJDbTHjpQd0cKUQnu9tVxqmBzhX31yF4VcsMe
ADcf2Z8wIA3n4LZnC/GwonYlq5+G93zJpFOkPhme8c2XuPuCXF795lsxyJ8SB/AlwPJAhEtm0y
0s0l1I4eWqxsDxkBOgN+ivU0cZrVMssHJEJb4o0FLf7iHhOW56/iMdD9w== userName
```

- **.{keyId}.pub.meta**: このファイルには、JSON 形式のキーメタデータが含まれます。キーにメタデータが含まれない場合は、新規のメタデータファイルが動的に生成されます。以下に例を示します。

```
{
  "name": "Key",
  "creationDate": "Oct 10, 2018 10:10:50 PM",
  "lastTimeUsed": "Oct 11, 2018 12:11:23 PM"
}
```

### 11.2. カスタムの SSH キーストア

要件に合わせてデフォルトの SSH キーストアを拡張し、カスタマイズできます。**appformer.ssh.keystore** システムプロパティーで、使用する SSH サービスの Java クラス名を指定してください。このプロパティーが定義されていない場合や、不正な値が含まれる場合には、デフォ

ルトの SSH キーストアが読み込まれます。



### 注記

SSH キーストアのカスタム実装を作成するには、Java クラスが **uberfire-ssh-api** モジュールで定義した **org.uberfire.ssh.service.backend.keystore.SSHKeyStore** クラスを実装する必要があります。

## 11.3. SSH キーの作成

SSH キーを Business Central に追加または登録する前に、お使いのシステムで SSH キーを生成する必要があります。

### 手順

1. システムでコマンド端末を開きます。
2. 以下の例のように、**ssh-keygen** コマンドを実行して、SSH キーを作成します。 **<user\_login>** はユーザー名に置き換えてください。

```
ssh-keygen -t rsa -b 4096 -C "<user_login>"
```



### 注記

Business Central キーストアでサポートされる SSH キーは **ssh-rsa**、**ssh-dss**、**ecdsa-sha2-nistp256**、**ecdsa-sha2-nistp384**、および **ecdsa-sha2-nistp521** です。

3. 以下の例のように、プロンプトが表示されたら、Enter キーを押して、デフォルトのキーファイルの場所を確定します。 **<user\_login>** はユーザー名に置き換えてください。

```
Enter a file in which to save the key (/home/<user_login>/.ssh/id_rsa): [Press enter]
```

4. コマンドプロンプトで、パスフレーズを入力して確定します。

```
Enter passphrase (empty for no passphrase): [Type a passphrase]
Enter same passphrase again: [Type passphrase again]
```

5. **ssh-agent** を起動します。

```
eval "$(ssh-agent -s)"
Agent pid <any-number-here>
```

6. 新しい SSH 秘密鍵を **ssh-agent** に追加します。違う名前のキーを使用する場合は、コード内の **id\_rsa** を置き換えます。

```
ssh-add ~/.ssh/id_rsa
```

## 11.4. SSH キーストアを使用した SSH 公開鍵の登録

新規作成した SSH 公開鍵は、Business Central キーストアに登録する必要があります。



## 手順

1. システムでコマンド端末を開きます。
2. 以下の例のように、**cat** コマンドを実行します。**id\_rsa** はキーの名前に置き換えます。

```
cat ~/.ssh/id_rsa.pub
```

3. SSH 公開鍵のコンテンツをコピーします。
4. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**SSH Keys** を選択します。
5. **SSH Keys** ページで **Add SSH Key** をクリックします。
6. **Add SSH Key** ウィンドウで **Name** フィールドに名前を入力し、SSH 公開鍵のコンテンツを **Key** フィールドにコピーします。



### 注記

Name および Key は必須のフィールドです。

7. **Add SSH Key** をクリックしてキーを登録します。

## 11.5. SSH キーの削除

Business Central の **SSH Keys** ページから SSH キーを削除できます。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**SSH Keys** を選択します。
2. **SSH Keys** ページで削除する SSH キーの削除アイコンをクリックします。
3. **Delete SSH Key** をクリックして、削除を確定します。

## 第12章 BUSINESS CENTRAL でのサービスタスクの管理

サービスタスク (作業アイテム) とは、複数のビジネスプロセスまたは Business Central の全プロジェクトの間でカスタマイズして再利用できるタスクのことです。Red Hat Process Automation Manager は、Business Central のサービスタスクのリポジトリで、サービスタスクセットを提供します。デフォルトのサービスタスクを有効化または無効化して、カスタムのサービスタスクを Business Central にアップロードし、適切なプロセスにこのタスクを実装できます。



### 注記

Red Hat Process Automation Manager には、サポートされるカスタムタスクの限定セットが含まれています。Red Hat Process Automation Manager に含まれていないカスタムタスクはサポートされません。

### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択して、**Service Tasks Administration** を選びます。  
このページは、サービスタスクのインストール設定や、Business Central 全体にあるプロジェクトのプロセスで利用可能なサービスタスクを表示します。このページで有効にしたサービスタスクは、プロジェクトレベルの設定で利用できます。プロジェクトレベルの設定で、プロセスで使用する各サービスタスクをインストールできます。サービスタスクをプロジェクトにインストールする方法は、**Service Tasks Administration** ページの **Settings** で有効化または無効化したグローバル設定により決まります。
2. **Settings** で、各設定を有効化または無効化して、ユーザーがプロジェクトレベルでインストールするときに、利用可能なサービスタスクを実装する方法を決定します。  
以下のサービスタスクの設定が利用できます。
  - **Install as Maven artifact** ファイルがない場合には、サービスタスクの JAR ファイルを Maven リポジトリにアップロードし、Business Central で設定します。
  - **Install service task dependencies into project** サービスタスクの依存関係をプロジェクトの **pom.xml** ファイルに追加します。このファイルでタスクがインストールされます。
  - **Use version range when installing service task into project** プロジェクトの依存関係として追加するサービスタスクの固定バージョンではなく、バージョン範囲を使用します。たとえば、**7.16.0.Final** ではなく **[7.16,)** です。
3. 必要に応じて利用可能なサービスタスクを有効化または無効化します (**ON** または **OFF** に設定)。有効化したサービスタスクは、Business Central の全プロジェクトのプロジェクトレベル設定に表示されます。

図12.1 サービスタスクおよびサービスタスク設定の有効化

Service Tasks Administration

Settings

**Install as Maven artifact**  ON  
Instructs if enabled service tasks should be installed into Maven repository

**Install service task dependencies into project**  ON  
Instructs that service task dependencies are added as project dependencies upon installation

**Use version range when installing service task into a project**  OFF  
Instructs that a version range will be used when installing service task in projects

[Add Service Task](#)

	<b>BusinessRuleTask</b>	Execute business rule or service tasks Execute a business rule task	<input checked="" type="checkbox"/> ON <input type="checkbox"/> OFF
	<b>CamelCXFConnector</b>	Use Apache Camel connectors in your processes Connect to a JAX-WS service hosted in CXF	<input type="checkbox"/> OFF <input type="checkbox"/> ON
	<b>CamelFTPConnector</b>	Use Apache Camel connectors in your processes Access remote file system over FTP	<input type="checkbox"/> OFF <input type="checkbox"/> ON
	<b>CamelFTPSConnector</b>	Use Apache Camel connectors in your processes Access remote file system over FTPS	<input type="checkbox"/> OFF <input type="checkbox"/> ON
	<b>CamelFileConnector</b>	Use Apache Camel connectors in your processes Access file systems and process files	<input type="checkbox"/> OFF <input type="checkbox"/> ON
	<b>CamelGenericConnector</b>	Use Apache Camel connectors in your processes Send payload to a Camel endpoint	<input type="checkbox"/> OFF <input type="checkbox"/> ON
	<b>CamelJMSConnector</b>	Use Apache Camel connectors in your processes Send message to a JMS Queue or Topic	<input type="checkbox"/> OFF <input type="checkbox"/> ON
	<b>CamelSQLConnector</b>	Use Apache Camel connectors in your processes Execute SQL query at a Camel endpoint and retrieve results	<input type="checkbox"/> OFF <input type="checkbox"/> ON
	<b>CamelXSLTConnector</b>	Use Apache Camel connectors in your processes Process a message using an XSLT template	<input type="checkbox"/> OFF <input type="checkbox"/> ON
	<b>DecisionTask</b>	Execute business rule or service tasks Execute a DMN decision task	<input checked="" type="checkbox"/> ON <input type="checkbox"/> OFF
	<b>Email</b>	Send an email Send email	<input checked="" type="checkbox"/> ON <input type="checkbox"/> OFF
	<b>JMSSendTask</b>	Send JSM messages Send JMS Message	<input checked="" type="checkbox"/> ON <input type="checkbox"/> OFF
	<b>Rest</b>	Perform REST calls Perform a Rest call	<input checked="" type="checkbox"/> ON <input type="checkbox"/> OFF
	<b>ServiceTask</b>	Execute business rule or service tasks Execute a service task	<input checked="" type="checkbox"/> ON <input type="checkbox"/> OFF
	<b>WebService</b>	Perform Webservice operations Perform a Webservice call	<input checked="" type="checkbox"/> ON <input type="checkbox"/> OFF

4. カスタムのサービスタスクを追加するには、**Add Service Task** をクリックして、適切な JAR ファイルを参照し、**Upload** アイコンをクリックします。JAR ファイルには、**@Wid** のアンテーションを指定したワークアイテムハンドラーの実装を含める必要があります。
5. Business Central ですべての必須サービスタスクを設定した後に、プロジェクトの **Settings** → **Service Tasks** ページに移動すると、有効化したサービスタスクで利用可能なものが表示されます。
6. サービスタスクごとに、**Install** をクリックして、対象のプロジェクトのプロセスでタスクを利用できるようにするか、**Uninstall** をクリックして、プロジェクトのプロセスからタスクを除外します。

7. サービスタスクのインストール時に追加情報を求められた場合には、必要な情報を入力して、もう一度 **Install** をクリックします。  
サービスタスクの必須パラメーターは、タスクのタイプにより異なります。たとえば、ルールとデシジョンタスクにはアーティファクトの GAV 情報 (グループ ID、アーティファクト ID、およびバージョン) が、メールタスクにはホストとポートアクセスの情報が、REST タスクには API の認証情報が必要です。他のサービスタスクでは、追加のパラメーターが必要でない場合もあります。

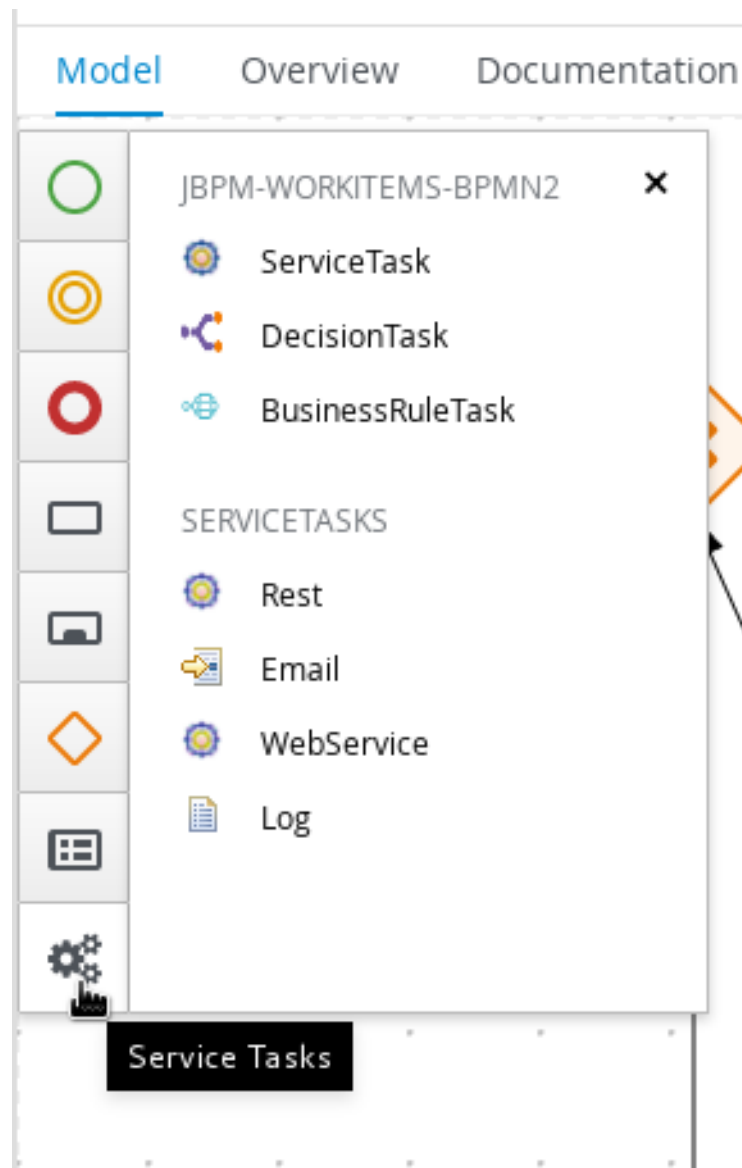
図12.2 プロセスで使用するためのサービスタスクのインストール

The screenshot shows the 'Mortgage\_Process' settings page in Red Hat Business Central. The 'Service Tasks' section is active, displaying a list of tasks available for installation. The tasks are:

Task Name	Description	Action
BusinessRuleTask	Execute a business rule task	Uninstall
DecisionTask	Execute a DMN decision task	Uninstall
Email	Send email	Install
JMSSendTask	Send JMS Message	Install
Rest	Perform a Rest call	Install
ServiceTask	Execute a service task	Uninstall
WebService	Perform a Webservice call	Uninstall

8. **Save** をクリックします。
9. プロジェクトページに戻り、プロジェクトのビジネスプロセスを選択または追加します。プロセスデザイナーパレットで **Service Tasks** オプションを選択すると、有効化してインストールした、利用可能なサービスタスクが表示されます。

図12.3 プロセスデザイナーでのインストール済みサービスタスクのアクセス



## 第13章 DASHBUILDER データのエクスポートおよびインポート

Dashbuilder は Business Central に統合されているダッシュボードおよびレポートツールで、データセットエディターとコンテンツマネージャーのページで使用します。データタイプには3種類あります。

- データセット
- パースペクティブ
- ナビゲーション

Business Central では、Dashbuilder データは Zip ファイルとしてインポートおよびエクスポートできます。



### 重要

この機能は管理者ユーザーのみが利用できます。

### 13.1. DASHBUILDER データのエクスポート

Dashbuilder のデータはすべて、Business Central から ZIP ファイルとしてエクスポートできます。

#### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Dashbuilder Data Transfer** を選択します。
2. **Dashbuilder Data Transfer** ページで **Export** をクリックします。  
すべての Databuilder データを含む **export.zip** ファイルがダウンロードされます。**export.zip** ファイル構造はデータタイプで区切られています。以下に例を示します。

```
dashbuilder/datasets/definitions/dataset-example1.csv
dashbuilder/datasets/definitions/dataset-example1.dset
dashbuilder/datasets/definitions/dataset-example2.csv
dashbuilder/datasets/definitions/dataset-example2.dset
dashbuilder/datasets/readme.md
dashbuilder/perspectives/page1/perspective_layout
dashbuilder/perspectives/page1/perspective_layout.plugin
dashbuilder/perspectives/page2/perspective_layout
dashbuilder/perspectives/page2/perspective_layout.plugin
dashbuilder/perspectives/readme.md
dashbuilder/navigation/navigation/navtree.json
dashbuilder/navigation/readme.md
VERSION
```

### 13.2. DASHBUILDER データのインポート

アーカイブが以下の例と同じように設定されている場合には、Zip ファイルで Dashbuilder データを Business Central にインポートできます。

```
dashbuilder/datasets/definitions/dataset-example1.csv
dashbuilder/datasets/definitions/dataset-example1.dset
```

```
dashbuilder/datasets/definitions/dataset-example2.csv
dashbuilder/datasets/definitions/dataset-example2.dset
dashbuilder/datasets/readme.md
dashbuilder/perspectives/page1/perspective_layout
dashbuilder/perspectives/page1/perspective_layout.plugin
dashbuilder/perspectives/page2/perspective_layout
dashbuilder/perspectives/page2/perspective_layout.plugin
dashbuilder/perspectives/readme.md
dashbuilder/navigation/navigation/navtree.json
dashbuilder/navigation/readme.md
VERSION
```

## 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Dashbuilder Data Transfer** を選択します。



### 警告

既存システムのデータが上書きされないように、新規インストールした Red Hat Process Automation Manager にのみ、Dashbuilder のデータをインポートするようにしてください。

2. **Dashbuilder Data Transfer** ページで、**Choose File** アイコンをクリックします。
3. インポートする ZIP ファイルに移動し、ファイルを選択します。
4. **Upload** アイコンをクリックします。
5. **Import** をクリックします。

## 第14章 LDAP 接続

Business Central は、ユーザータスクサービスがユーザー、グループ、ロールの情報を LDAP サービスから直接取得できるように、Red Hat Process Automation Manager で LDAP サーバー専用の **UserGroupCallback** 実装を提供します。

以下の LDAP **UserGroupCallback** 実装プロパティは、設定可能です。

表14.1 LDAP UserGroupCallback プロパティ

プロパティ	説明
<b>ldap.bind.user</b>	LDAP サーバーへの接続に使用するユーザー名  指定がない場合はこのプロパティは任意となり、LDAP サーバーは匿名アクセスを受け入れます。
<b>ldap.bind.pwd</b>	LDAP サーバーへの接続に使用するパスワード  指定がない場合はこのプロパティは任意となり、LDAP サーバーは匿名アクセスを受け入れます。
<b>ldap.user.ctx</b>	ユーザー情報を含む LDAP のコンテキスト
<b>ldap.role.ctx</b>	グループおよびロールの情報を含む LDAP のコンテキスト
<b>ldap.user.roles.ctx</b>	ユーザーグループおよびロールのメンバーシップ情報を含む LDAP のコンテキスト  指定がない場合はこのプロパティは任意となり、代わりに <b>ldap.role.ctx</b> プロパティを使用します。
<b>ldap.user.filter</b>	ユーザー情報検索用のフィルター  このプロパティは通常、代入キー {0} が含まれており、パラメーターと置き換えられます。
<b>ldap.role.filter</b>	グループおよびロールの情報の検索フィルター  このプロパティは通常、代入キー {0} が含まれており、パラメーターと置き換えられます。
<b>ldap.user.roles.filter</b>	ユーザーグループおよびロールのメンバーシップ情報を検索するためのフィルター  このプロパティは通常、代入キー {0} が含まれており、パラメーターと置き換えられます。



プロパティ	説明
<b>ldap.user.attr.id</b>	LDAP に含まれるユーザー ID の属性名  指定がない場合はこのプロパティは任意となり、代わりに <b>uid</b> プロパティを使用します。
<b>ldap.roles.attr.id</b>	LDAP 内のグループおよびロール ID の属性名  指定がない場合はこのプロパティは任意となり、代わりに <b>cn</b> プロパティを使用します。
<b>ldap.user.id.dn</b>	DN のユーザー ID。ロールの検索前にユーザー DN をクエリーするようにコールバックを指示します。これは任意で、デフォルトは <b>false</b> です。
<b>java.naming.factory.initial</b>	初期コンテキストファクトリークラス名。デフォルトは <b>com.sun.jndi.ldap.LdapCtxFactory</b>
<b>java.naming.security.authentication</b>	認証タイプ。可能な値は、 <b>none</b> 、 <b>simple</b> 、および <b>strong</b> です。デフォルトは <b>simple</b> です。
<b>java.naming.security.protocol</b>	<b>ssl</b> など、使用するセキュリティープロトコル
<b>java.naming.provider.url</b>	LDAP url (デフォルトは <b>ldap://localhost:389</b> 。プロトコルが <b>ssl</b> に設定されている場合は <b>ldap://localhost:636</b> )

## 14.1. LDAP USERGROUPCALLBACK 実装

以下のいずれかの方法で、該当の LDAP プロパティを設定して、LDAP **UserGroupCallback** 実装を使用できます。

- プログラム: 該当の **LDAPUserGroupCallbackImpl** プロパティでプロパティのオブジェクトをビルドし、同じプロパティオブジェクトをパラメーターとして使用し、**LDAPUserGroupCallbackImpl** を作成します。  
以下に例を示します。

```
import org.kie.api.PropertiesConfiguration;
import org.kie.api.task.UserGroupCallback;
...
Properties properties = new Properties();
properties.setProperty(LDAPUserGroupCallbackImpl.USER_CTX, "ou=People,dc=my-domain,dc=com");
properties.setProperty(LDAPUserGroupCallbackImpl.ROLE_CTX, "ou=Roles,dc=my-domain,dc=com");
properties.setProperty(LDAPUserGroupCallbackImpl.USER_ROLES_CTX, "ou=Roles,dc=my-domain,dc=com");
properties.setProperty(LDAPUserGroupCallbackImpl.USER_FILTER, "(uid={0})");
properties.setProperty(LDAPUserGroupCallbackImpl.ROLE_FILTER, "(cn={0})");
```

```
properties.setProperty(LDAPUserGroupCallbackImpl.USER_ROLES_FILTER, "(member={0})");
```

```
UserGroupCallback ldapUserGroupCallback = new
LDAPUserGroupCallbackImpl(properties);
```

```
UserGroupCallbackManager.getInstance().setCallback(ldapUserGroupCallback);
```

- 宣言設定: アプリケーションのルートに **jbpm.usergroup.callback.properties** ファイルを作成するか、システムプロパティとしてファイルの場所を指定します。以下に例を示します。

#### **-Djbpm.usergroup.callback.properties=FILE\_LOCATION\_ON\_CLASSPATH**

ユーザータスクサーバーの起動時に LDAP コールバックを登録するようにしてください。

以下に例を示します。

```
#ldap.bind.user=
#ldap.bind.pwd=
ldap.user.ctx=ou\=People,dc\=my-domain,dc\=com
ldap.role.ctx=ou\=Roles,dc\=my-domain,dc\=com
ldap.user.roles.ctx=ou\=Roles,dc\=my-domain,dc\=com
ldap.user.filter=(uid\={0})
ldap.role.filter=(cn\={0})
ldap.user.roles.filter=(member\={0})
#ldap.user.attr.id=
#ldap.roles.attr.id=
```

## 関連情報

- [ロールおよびユーザー](#)
- [Red Hat Single Sign-On Server Administration Guide](#)
- [「LDAP ログインドメインの定義」](#)
- [「LDAP ログインモジュール」](#)
- [「LDAPExtended ログインモジュール」](#)
- [「AdvancedLDAP ログインモジュール」](#)
- [「AdvancedAdLDAP ログインモジュール」](#)
- [「LDAP connectivity options」](#)
- [「LDAPUsers ログインモジュール」](#)

## 第15章 データベース接続

Business Central は、Red Hat Process Automation Manager でデータベースサーバー専用の **UserGroupCallback** 実装を提供し、ユーザータスクサービスを有効にします。ユーザータスクサービスを使用して、ユーザーやグループ (ロール) の情報を直接データベースから取得できるようにします。

以下のデータベースの **UserGroupCallback** 実装プロパティを設定することができます。

表15.1 データベースの UserGroupCallback プロパティ

プロパティ	説明
<b>db.ds.jndi.name</b>	接続に使用するデータソースの JNDI 名
<b>db.user.query</b>	ユーザーの存在を確認する
<b>db.user.roles.query</b>	特定のユーザーのグループを収集する
<b>db.roles.query</b>	グループの存在を確認する

### 15.1. データベースの USERGROUPCALLBACK 実装

データベースの **UserGroupCallback** 実装では、必須のデータベースを作成する必要があります。以下のいずれかの方法で、該当のデータベースプロパティを設定し、この実装を使用できます。

- プログラム: 該当の **DBUserGroupCallbackImpl** プロパティでプロパティのオブジェクトをビルドし、プロパティオブジェクトで、パラメーターとして **DBUserGroupCallbackImpl** を作成します。  
以下に例を示します。

```
import static org.jbpm.services.task.identity.DBUserGroupCallbackImpl.DS_JNDI_NAME;
import static
org.jbpm.services.task.identity.DBUserGroupCallbackImpl.PRINCIPAL_QUERY;
import static org.jbpm.services.task.identity.DBUserGroupCallbackImpl.ROLES_QUERY;
import static
org.jbpm.services.task.identity.DBUserGroupCallbackImpl.USER_ROLES_QUERY;
...
props = new Properties();
props.setProperty(DS_JNDI_NAME, "jdbc/jbpm-ds");
props.setProperty(PRINCIPAL_QUERY, "select userId from Users where userId = ?");
props.setProperty(ROLES_QUERY, "select groupId from UserGroups where groupId = ?");
props.setProperty(USER_ROLES_QUERY, "select groupId from UserGroups where userId =
?");

callback = new DBUserGroupCallbackImpl(props);
```

- 宣言設定: アプリケーションのルートに **jbpm.usergroup.callback.properties** ファイルを作成するか、システムプロパティとしてファイルの場所を指定します。  
以下に例を示します。

**-Djbpm.usergroup.callback.properties=FILE\_LOCATION\_ON\_CLASSPATH**

ユーザータスクサーバーの起動時にデータベースコールバックを登録するようにしてください。

以下に例を示します。

```
System.setProperty("jbpm.usergroup.callback.properties",
"/jbpm.usergroup.callback.db.properties");
callback = new DBUserGroupCallbackImpl(true);
...
db.ds.jndi.name = jdbc/jbpm-ds
db.user.query = select userId from Users where userId = ?
db.roles.query = select groupId from UserGroups where groupId = ?
db.user.roles.query = select groupId from UserGroups where userId = ?
```

## 関連情報

- [ロールおよびユーザー](#)

## 第16章 SETTINGS.XML ファイルを使用した MAVEN の設定

Java アプリケーション開発は、Apache Maven ビルド自動化ツールを使用して、ソフトウェアプロジェクトをビルドし、管理します。Maven は Project Object Model (POM) 設定の XML ファイルを使用して、プロジェクトプロパティとビルドプロセスの両方を定義します。

Maven はレポジトリを使用して Java ライブラリー、プラグイン、および他のビルドアーティファクトを格納します。レポジトリはローカルまたはリモートのいずれかになります。ローカルレポジトリは、ローカルマシンにキャッシュされたリモートレポジトリからアーティファクトをダウンロードしたものです。リモートレポジトリは、**http://** (HTTP サーバーにある場合) や **file://** (ファイルサーバーにある場合) などの一般的なプロトコルを使用してアクセスされる他のレポジトリです。デフォルトのレポジトリは、パブリックのリモート Maven 2 Central Repository となっています。Maven は、settings.xml ファイルを変更して設定できます。グローバルの Maven 設定は、**M2\_HOME/conf/settings.xml** ファイルで、ユーザーレベルの設定は **USER\_HOME/.m2/settings.xml** ファイルで実行可能です。

### 関連情報

- [「Business Central および Process Server への外部 Maven レポジトリの設定」](#)
- [「Red Hat Process Automation Manager プロジェクトの Maven でのパッケージ化およびデプロイ」](#)
- [Red Hat Process Automation Manager の Maven 設定およびレポジトリ](#)
- [Maven を使ったシステム統合](#)
- [Welcome to Apache Maven](#)
- [Apache Maven Project - Introduction to Repositories](#)
- [Apache Maven Parent POMs Reference](#)

## 第17章 GAV チェック管理

Business Central では、プロジェクトは **グループ ID**、**アーティファクト ID**、および **バージョン (GAV)** の Maven 命名規則で識別されます。GAV の値は、プロジェクトとプロジェクトバージョンを区別し、特定のプロジェクトとの依存関係を識別します。

デフォルトでは、Business Central は GAV の重複を検出します。この機能は、**admin** ロールを持つユーザーにより無効にできます。

### 17.1. GAV チェックおよび子の GAV エディションの設定

以下の手順では、Business Central での GAV チェックの設定方法を説明します。

#### 手順

1. Business Central で、**Menu** → **Design** → **Projects** に移動して、プロジェクト名をクリックします。
2. プロジェクトウィンドウで、**Settings** タブをクリックします。
3. **General Settings** タブで以下のタスクを実行します。
  - 他のプロジェクトで同じ GAV を使用できるようにするには、**Disable GAV conflict check** チェックボックスを選択します。
  - このプロジェクトに GAV エディションを指定できるようにするには、**Allow child GAV edition** チェックボックスを選択します。
4. **Save** をクリックします。



#### 注記

**Reset** をクリックして、すべての変更を元に戻すことができます。

5. **Save** をクリックして、変更を確定します。



#### 注記

重複する GAV の検出は、**Development Mode** のプロジェクトでは無効になっていません。Business Central で重複する GAV 検出を有効にするには、プロジェクトの **Settings** → **General Settings** → **Version** に移動して、**Development Mode** オプションを **OFF** (該当する場合) に切り替えます。

### 17.2. 全プロジェクトの GAV チェックの設定

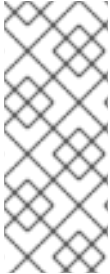
以下の手順では、Business Central の全プロジェクトに GAV チェックを設定する方法を説明します。また、システムの起動時に GAV チェックを無効にすることも可能です。

#### 手順

1. Business Central で、画面の右上隅にある **Admin** アイコンを選択し、**Projects** を選択します。**Projects** ウィンドウが開きます。
2. **Advanced GAV preferences** タブで以下のタスクのいずれかを実行します。

- 他のプロジェクトで同じ GAV を使用できるようにするには、**Disable GAV conflict check** チェックボックスを選択します。
- このプロジェクトに GAV エディションを指定できるようにするには、**Allow child GAV edition** チェックボックスを選択します。

3. **Save** をクリックします。



#### 注記

Business Central の起動時に、**org.guvnor.project.gav.check.disabled** システムプロパティを **true** に設定して、重複した GAV の削除機能を無効にすることも可能です。

```
$ ~/EAP_HOME/bin/standalone.sh -c standalone-full.xml  
-Dorg.guvnor.project.gav.check.disabled=true
```

## 第18章 PROCESS SERVER および BUSINESS CENTRAL での環境モードの設定

Process Server は、**production** (実稼働) モードと **development** (開発) モードでの実行が設定可能です。開発モードでは、柔軟な開発ポリシーが提供され、小規模な変更の場合はアクティブなプロセスインスタンスを維持しながら、既存のデプロイメントユニット (KIE コンテナ) を更新できます。また、大規模な変更の場合は、アクティブなプロセスインスタンスを更新する前に、デプロイメントユニットの状態をリセットすることも可能です。実稼働モードは、各デプロイメントで新規デプロイメントユニットが作成される実稼働環境に最適です。

開発環境では、**Deploy** をクリックすると、(該当する場合) 実行中のインスタンスを **中止** することなくビルドした KJAR ファイルを Process Server にデプロイすることができます。または **Redeploy** をクリックして、ビルドした KJAR ファイルをデプロイして実行中のインスタンスを中止することもできます。ビルドした KJAR ファイルを次回にデプロイまたは再デプロイすると、以前のデプロイメントユニット (KIE コンテナ) が同じターゲット Process Server で自動的に更新されます。

実稼働環境では、Business Central の **Redeploy** オプションは無効になり、**Deploy** をクリックして、ビルドした KJAR ファイルを Process Server 上の新規デプロイメントユニット (KIE コンテナ) にデプロイすることのみが可能です。

### 手順

1. Process Server 環境モードを設定するには、**org.kie.server.mode** システムプロパティを **org.kie.server.mode=development** または **org.kie.server.mode=production** に設定します。
2. Business Central のプロジェクトにデプロイメントの動作を設定するには、プロジェクトの **Settings** → **General Settings** → **Version** に移動して、**Development Mode** オプションを切り替えます。



### 注記

デフォルトでは、Process Server と Business Central の新規プロジェクトはすべて、開発モードになっています。

**Development Mode** がオンのプロジェクトや、実稼働モードの Process Server に手動で **SNAPSHOT** バージョンの接尾辞を追加したプロジェクトをデプロイすることはできません。



## 第19章 GIT フックおよびリモート GIT リポジトリの統合

Git フックは、**git commit** や **git push** などの Git イベントの前後に実行するバッシュスクリプトです。Business Central では、Git フックを使用して、イベントが発生するたびに、リポジトリが指定のアクションをトリガーするように設定できます。Git フックの詳細は、[Customizing Git Hooks](#) を参照してください。

Business Central と、リモート Git リポジトリを統合するには、post-commit の Git フックを使用します。こうすることで、Business Central とリモートリポジトリの間のコンテンツの複製を自動化できます。たとえば、Business Central プロジェクトに加えた変更をリモートの Git リポジトリに複製する、リアルタイムのバックアップストラテジーを実装できます。



### 注記

Business Central は、post-commit の Git フックのみをサポートします。

post-commit の Git フックは、コミットするたびに同期操作として実行します。Business Central は、コミット後の Bash が完了するまで待機して、リポジトリでの他の書き込み操作が行われないようにします。

### 19.1. POST-COMMIT の GIT フックの作成

post-commit の Git フックスクリプトファイルを作成して、そのファイルに含まれるコードを実行するか、Java プログラムなどの別のファイルからコードを実行できます。

#### 手順

1. **post-commit** Git フックファイルを作成します。

```
$ touch post-commit
```

2. **post-commit** ファイルのパーミッションを **755** に設定します。

```
$ chmod 755 post-commit
```

3. 以下のように、**#!/bin/bash** と必要なコードを **post-commit** ファイルに追加します。

- すべての変更をリモートリポジトリにプッシュするには、以下のコマンドを実行します。

```
#!/bin/bash
git push origin +master
```

- メッセージをログに記録するには、以下を実行します。

```
#!/bin/bash
echo 'Hello World'
```

- 別のファイルのコードを実行するには、以下を実行します。

```
#!/bin/bash
java -jar _EAP_HOME_/bin/.niogit/<SPACE>/<PROJECT_NAME>.git/hooks/git-push.jar
```



## 注記

post-commit の Git フックを使用して Java コードを実行するには、以下の Java ライブラリーを使用する必要があります。

- [JGit](#): 内部の Business Central Git リポジトリと対話するのに使用します。
- [GitHub API for Java](#): GitHub との通信に使用します。

post-commit の Git フックと Java コードの例に関する情報は、[Business Central post-commit Git Hooks Integration](#) を参照してください。


## 19.2. リモート GIT リポジトリのインポート

リモートの Git リポジトリを Business Central にインポートし、post-commit の Git フックを設定して、そのリモートリポジトリに変更を自動的にプッシュできます。

### 前提条件

- Red Hat JBoss EAP 7.2 サーバーインスタンスに Red Hat Process Automation Manager がインストールされている。
- 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 Git リポジトリへの読み取りアクセスに必要な認証情報。
- (Windows の場合) Cygwin は、インストール時に追加される Git パッケージでインストールします。Cygwin `/bin` フォルダへのパスは、お使いの環境の `PATH` 変数に追加されます。たとえば、`C:\cygwin64\bin` です。Cygwin のインストールに関する詳細は、[Installing and Updating Cygwin Packages](#) を参照してください。

### 手順

1. Business Central で、**Menu** → **Projects** に移動します。
2. Git 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。
3. 画面の右側の  をクリックして、**Import Project** を選択します。
4. **Import Project** ウィンドウで、Git リポジトリの URL を入力します (例: [https://github.com/USERNAME/REPOSITORY\\_NAME.git](https://github.com/USERNAME/REPOSITORY_NAME.git) および Git リポジトリの認証情報)。
5. **Import** をクリックします。  
プロジェクトを Business Central の Git リポジトリに追加すると、スペースで使用できるようになります。



## 重要

SCP スタイルの SSH URL の代わりに、HTTPS または Git プロトコルを使用します。Business Central は基本的な SSH URL をサポートしないため、この URL を使用する場合はエラーが発生します。

公開 SSH キーを Git プロバイダーに設定する必要があります。

Git リポジトリは、Red Hat Process Automation Manager のバージョンと互換性のある KJAR が1つだけ含まれる KJAR プロジェクトでなければなりません。KJAR コンテンツは、リポジトリのルートに配置する必要があります。

6. コマンド端末で、プロジェクトのリポジトリ Git ディレクトリーにある **hooks** ディレクトリーに移動します。以下に例を示します。

```
$ cd _EAP_HOME_/bin/.niogit/<SPACE>/<PROJECT_NAME>.git/hooks
```

7. 以下のように、リモート Git リポジトリに変更をプッシュする **post-commit** ファイルを作成します。以下に例を示します。

```
#!/bin/sh
git push origin +master
```

post-commit の Git フックの作成に関する情報は、「[post-commit の Git フックの作成](#)」を参照してください。

8. 必要に応じて、設定が正常に行われたことを確認するには、Business Central のガイド付きルールを作成します。
  - a. Business Central で **Menu → Projects → Add Asset → Guided Rule** に移動します。
  - b. **Create new Guided Rule** ページで必要な情報を入力します。
  - c. **OK** をクリックします。  
Business Central は、リモートリポジトリにすべての変更を自動的にプッシュします。

## 関連資料

- [Customizing Git - Git Hooks](#)

## 19.3. 既存のリモート GIT プロジェクトリポジトリ用の GIT フックの設定

既存のリモート Git リポジトリプロジェクトがある場合には、その既存のプロジェクトのリモート Git リポジトリに post-commit の Git フックを作成し、リモート Git リポジトリを Business Central に統合できます。

### 前提条件

- Red Hat JBoss EAP 7.2 サーバーインスタンスに Red Hat Process Automation Manager がインストールされている。
- 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 Git リポジトリへの読み取りアクセスに必要な認証情報。

- (Windows オペレーティングシステムの場合) インストール時に追加される Git パッケージで Cygwin がインストールされており、Cygwin `/bin` ディレクトリーへのパスがお使いの環境の **PATH** 変数に追加されている。たとえば、**C:\cygwin64\bin** です。Cygwin のインストールに関する詳細は、[Installing and Updating Cygwin Packages](#) を参照してください。

## 手順

1. コマンド端末で、プロジェクトのリポジトリー Git ディレクトリーにある **hooks** ディレクトリーに移動します。以下に例を示します。

```
$ cd _EAP_HOME_/bin/.niogit/<SPACE>/<PROJECT_NAME>.git/hooks
```

2. 以下のように、リモート Git リポジトリーに変更をプッシュする **post-commit** ファイルを作成します。以下に例を示します。

```
#!/bin/sh
git push origin +master
```

post-commit の Git フックの作成に関する情報は、「[post-commit の Git フックの作成](#)」を参照してください。

3. 必要に応じて、設定が正常に行われたことを確認するには、Business Central のガイド付きルールを作成します。
  - a. Business Central で **Menu** → **Projects** → **Add Asset** → **Guided Rule** に移動します。
  - b. **Create new Guided Rule** ページで必要な情報を入力します。
  - c. **OK** をクリックします。  
Business Central は、リモートリポジトリーにすべての変更を自動的にプッシュします。

## 19.4. BUSINESS CENTRAL のシステムプロパティとしての GIT フックの設定

既存の Git リポジトリープロジェクトがない場合や、post-commit の Git フックを多数のプロジェクトリポジトリーに適用する場合は、**org.uberfire.nio.git.hooks** システムプロパティの値に、フックファイルが含まれるディレクトリーを指定できます。このディレクトリーは Git リポジトリーにコピーされます。



### 注記

**org.uberfire.nio.git.hooks** システムプロパティを指定した場合は、すべての Business Central 内部リポジトリーおよびプロジェクトリポジトリーで post-commit の Git フックが使用されます。スクリプトでは完全修飾パスのみを使用する必要があります。

### 前提条件

- Red Hat JBoss EAP 7.2 サーバーインスタンスに Red Hat Process Automation Manager がインストールされている。
- (Windows オペレーティングシステムの場合) インストール時に追加される Git パッケージで Cygwin がインストールされており、Cygwin `/bin` ディレクトリーへのパスがお使いの環境の **PATH** 変数に追加されている。たとえば、**C:\cygwin64\bin** です。Cygwin のインストールに関する詳細は、[Installing and Updating Cygwin Packages](#) を参照してください。

## 手順

- ローカルシステムのディレクトリーに post-commit の Git フックを作成します。  
post-commit の Git フックの作成に関する情報は、「[post-commit の Git フックの作成](#)」を参照してください。
- org.uberfire.nio.git.hooks** システムプロパティーの値に、フックファイルが含まれるディレクトリーを指定するには、以下のいずれかのタスクを実行します。

- **org.uberfire.nio.git.hooks** システムプロパティーを **standalone.xml** ファイルに追加します。以下に例を示します。

```
<system-properties>
  <property name="org.uberfire.nio.git.hooks" value="_EAP_HOME_/hooks">
  </property>
  ...
</system-properties>
```

- Business Central の実行時に、**-Dorg.uberfire.nio.git.hooks** 環境変数を使用します。以下に例を示します。

```
$ ./standalone.sh -c standalone-full.xml -
Dorg.uberfire.nio.git.hooks=_EAP_HOME_/hooks
```

- Business Central を起動します。  
post-commit の Git フックは、Business Central の内部リポジトリとプロジェクトのリポジトリすべてにコピーされます。

## 関連資料

- [Customizing Git - Git Hooks](#)

## 19.5. リモート GIT リポジトリの統合

以下の例では、post-commit の Git フックと Java コードを使用して、Business Central とリモートの Git リポジトリを統合します。Java コードの例は、[Business Central post-commit Git Hooks Integration](#) を参照してください。この例では、以下の機能を提供します。

- テンプレート **.gitremote** 設定ファイルを自動生成
- 必須のパラメーターに対する **.gitremote** 設定ファイルを検証
- **.gitremote** ファイルの ignore パラメーターで定義するパターンを Git で無視
- ユーザーにメッセージおよび通知を出力
- GitLab および GitHub トークン認証をサポート
- GitLab グループおよびサブグループプロジェクト作成をサポート
- GitHub 組織リポジトリ作成をサポート

## 前提条件

- Red Hat JBoss EAP 7.2 サーバーインスタンスに Red Hat Process Automation Manager がインストールされている。
- Java Development Kit (JDK) 8 がインストールされている。
- Maven がインストールされている。

## 手順

1. 端末ウィンドウで、GitHub リポジトリをシステムにクローンします。

```
$ git clone https://github.com/kiigroup/bc-git-integration-push.git
```

2. クローンしたリポジトリに移動します。

```
$ cd bc-git-integration-push
```

3. Maven の新規インストールを実行します。

```
$ mvn clean install
```

4. **EAP\_HOME** ディレクトリに **/hooks** ディレクトリを作成します。

```
$ mkdir -p _EAP_HOME_/hooks/
```

5. **git-push-2.1-SNAPSHOT.jar** を **EAP\_HOME/hooks/** ディレクトリにコピーします。

```
$ cp bc-git-integration-push/target/git-push-2.1-SNAPSHOT.jar _EAP_HOME_/hooks/
```

6. 必要に応じて、テンプレートの **.gitremote** 設定ファイルを作成するには **git-push-2.1-SNAPSHOT.jar** を実行します。

```
$ java -jar git-push-2.1-SNAPSHOT.jar
```

### テンプレート **.gitremote** 設定ファイルの例

```
#This is an auto generated template empty property file
provider=GIT_HUB
login=
password=
token=
remoteGitUrl=https://api.github.com/
useSSH=false
ignore=.*demo.*, test.*
githubOrg=OrgName
gitlabGroup=Group/subgroup
```

7. **.gitremote** 設定ファイルパラメーターを変更します。

### 表19.1.gitremote パラメーターの例

パラメーター	説明
<b>provider</b>	Git プロバイダー。許容値は GIT_HUB と GIT_LAB の 2 つのみ。必須。
<b>login</b>	Git プロバイダーのユーザー名。必須。
<b>password</b>	プレーンテキストのパスワードです。 <b>token</b> が指定されている場合は必要ありません。
<b>token</b>	<b>username</b> および <b>password</b> ベースの、安全対策がされていない接続の代替りとなる生成トークン。注記: これが設定されていない場合は、セキュアでない接続を使用している旨の警告が表示されます。 <b>password</b> が指定されている場合は必須ではありません。注記: GitLab はトークン認証のみをサポートします。
<b>remoteGitUrl</b>	パブリックのプロバイダー URL またはプロバイダー用にローカルでホストされたエンタープライズ URL。必須。注記: 公開 GitHub URL は API URL である必要があります。たとえば、api.github.com です。
<b>useSSH</b>	SSH プロトコルがリモートリポジトリに変更をプッシュできるようにするブール型。任意。デフォルト値 = false。注記: このパラメーターはローカルの ~/.ssh/ ディレクトリーを使用して SSH 設定を取得します。
<b>ignore</b>	これらの式のいずれかに一致するプロジェクト名を無視するために、コマ区切りの正規表現。任意。
<b>githubOrg</b>	プロバイダーとして GitHub を使用する場合にリポジトリ組織を定義します。任意。
<b>gitlabGroup</b>	プロバイダーとして GitLab を使用する場合にリポジトリグループおよびサブグループを定義します。任意。

8. **EAP\_HOME/hooks** に、 **post-commit** Git フックファイルを作成します。

```
$ touch post-commit
```

9. **post-commit** ファイルのパーミッションを **755** に設定します。

```
$ chmod 755 post-commit
```

10. **#!/bin/bash** と **git-push-2.1-SNAPSHOT.jar** を実行するコードを **post-commit** ファイルに追加します。

```
$ echo "#!/bin/bash\njava -jar $APP_SERVER_HOME/hooks/git-push-2.1-SNAPSHOT.jar" > hooks/post-commit
```

11. **-Dorg.uberfire.nio.git.hooks** 環境変数を設定して、Business Central を起動します。以下に例を示します。

```
$ ./standalone.sh -c standalone-full.xml -Dorg.uberfire.nio.git.hooks=_EAP_HOME_/hooks
```



## 注記

post-commit の Git フックを使用して Java コードを実行するには、以下の Java ライブラリーを使用する必要があります。

- [JGit](#): 内部の Business Central Git リポジトリと対話するのに使用します。
- [GitHub API for Java](#): GitHub との通信に使用します。

post-commit の Git フックと Java コードの例に関する情報は、[Business Central post-commit Git Hooks Integration](#) を参照してください。

## 19.6. GIT フックの終了コード

Git フックの終了時には、整数値が返され、Git フックの実行ステータスを判断します。この整数値は、Git フックの終了コードとして知られています。実行ステータスは、成功 (1)、警告 (2 から 30) または失敗 (31 から 255) です。

## 19.7. GIT フック通知のカスタマイズ

Business Central には、フックの終了コードをもとに、カスタマイズした Git フック通知をユーザーが受信可能なメカニズムがあります。

この通知メカニズムを有効にするには、カスタムのメッセージを含む **\*.properties** ファイルを作成してから、**appformer.git.hooks.bundle** システムプロパティの値として対象のファイルのパスを指定する必要があります。

### 手順

1. **\*.properties** ファイルを作成して、各終了コードに以下の形式の適切なメッセージを 1 行追加してください。

```
<exit_code>=<display_message>
```

**<exit\_code>** は、Git フックの終了コードに、**<display\_message>** はユーザーに表示するカスタムメッセージに置き換えます。

以下に例を示します。

```
0=Success! All working as expected.
1=Warning! Please check the logs and advise your admin.
.
.
31=Error! Please advise your admin immediately.
```



## 注記

考えられる終了コードをすべて \*.properties ファイルに定義する必要はありません。\*.properties ファイルに定義されている終了コードに対する通知のみが表示されます。





## 重要

通知サービスは、プロパティファイルに設定されている **ISO 8859-1 (LATIN 1)** 文字のみサポートします。拡張文字を使用する場合には、Unicode 文字コードのエスケープシーケンスを使用してください。

- Git フック通知を有効化するには、**appformer.git.hooks.bundle** システムプロパティの値としてファイルへのパスを指定します。

**Messages.properties** ファイルを参照する設定が含まれている、以下の **standalone.xml** ファイル例を確認してください。

```
<system-properties>
  <property name="appformer.git.hooks.bundle" value="/opt/jboss-as/git-hooks-
messages/Messages.properties">
  </property>
  ...
</system-properties>
```

### 19.7.1. Business Central の Git フック通知

Business Central で Git フックの通知を確認できます。Git フックの終了コードの通知タイプは 3 種類あります。

表19.2 Git フックの UI 通知タイプ

終了コード	カスタマイズしたメッセージ	UI での通知の色
0	Success!All working as expected. (成功!すべて想定どおりに機能しています。)	Green
1 から 30	Warning!Please check the logs and advise your admin. (警告!ログを確認して管理者に連絡してください。)	Orange
31 から 255	Error!Please advise your admin immediately. (エラー!今すぐ管理者に連絡してください。)	Red



## 重要

UNIX マシンがサポートするエラーコードは 0 (success) から 255 (error) までで、この範囲外の終了コードは別のコードに変換され、誤った通知メッセージが表示される可能性があります。

Windows マシンには、制限がなく、幅広い終了コードをサポートします。

### 19.7.2. Git フック通知の国際化サポート

通知メッセージの国際化には、**appformer.git.hooks.bundle** システムプロパティとして指定した元のプロパティファイルと同じパスに、追加でプロパティファイルを配置してください。

各種ローカライズファイルの名前は、**<filename>\_<lang>.properties** に指定し、**<filename>** は元のファイルと同じファイル名にしてください。たとえば、システムプロパティが **Messages.properties**

を参照している場合、英語は **Messages\_en.properties**、フランス語は **Messages\_fr.properties**、またはイタリア語は **Messages\_it.properties** として作成できます。

通知サービスは、ユーザー言語をもとにプロパティファイルを選択します。対象言語の翻訳がない場合は、元の **Messages.properties** ファイルからのエントリーを使用します。

## 第20章 BUSINESS CENTRAL のブランチでのロールベースアクセス制御

Business Central には、特定のコラボレータータイプのターゲットブランチに対するアクセスを制限するオプションがあります。セキュリティーチェックは、**Security Management** の画面とコントリビューターソースの両方を使用してスペースやプロジェクトに対してパーミッションを与えたり、拒否したりします。たとえば、コントリビュータータイプをもとにして、プロジェクトを更新するパーミッションと、対象のブランチへの書き込みパーミッションがある場合は、新規アセットを作成できます。

### 20.1. ロールベースのブランチアクセスのカスタマイズ

Business Central では、プロジェクトのブランチごとにコントリビューターロールのパーミッションをカスタマイズできます。たとえば、ブランチに割り当てたロールごとに、**Read**、**Write**、**Delete**、および **Deploy** のアクセス権を設定できます。

#### 手順

1. Business Central で、**Menu** → **Design** → **Projects** に移動します。
2. 必要に応じて、新規コントリビューターを追加します。
  - a. プロジェクト名をクリックし、**Contributors** タブをクリックします。
  - b. **Add Contributor** をクリックします。
  - c. テキストフィールドにユーザー名を入力します。
  - d. ドロップダウンリストから **Contributor** のロールタイプを選択します。
  - e. **OK** をクリックします。
3. 関連のコントリビューターのブランチへのアクセス権限をロールベースでカスタマイズします。
  - a. **Settings** → **Branch Management** をクリックします。
  - b. ドロップダウンリストからブランチ名を選択します。
  - c. **Role Access** のセクションで、パーミッションのチェックボックスを選択または選択解除して、利用可能なロールタイプごとにロールベースのブランチアクセスを指定します。
  - d. **Save** をクリックし、再度 **Save** をクリックして変更を確定します。

## 第21章 プロセスインスタンスログの表示

**Logs** タブから、インスタンスのプロセスイベントをすべて表示できます。インスタンスログは、現在と過去のプロセスの状態をすべて表示します。Business Central には、**Business** と **Technical** ログの2種類のプロセスインスタンスログがあります。

### 手順

1. Business Central で、**Menu** → **Manage** → **Process Instances** に移動します。
2. **Manage Process Instances** ページで、表示するログのプロセスインスタンスをクリックします。
3. **Logs** タブを選択します。
  - ビジネスイベントログを表示するには、**Business** をクリックします。
  - テクニカルイベントログを表示するには、**Technical** をクリックします。
  - **Asc** または **Desc** をクリックすると、ログファイルの順序が変わります。

## 第22章 BUSINESS CENTRAL システムプロパティー

このセクションに記載の Business Central のシステムプロパティーは **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ーを使用することができます。

```
java -jar rhpam-7.6.0-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ーに、**<value>** をそのプロパティーに割り当てる値に置き換えます。

### Git ディレクトリー

以下のプロパティーを使用して、Business Central Git ディレクトリーの場所と名前を設定します。

- **org.uberfire.nio.git.dir**: Business Central の Git ディレクトリーの場所。
- **org.uberfire.nio.git.dirname**: Business Central の Git ディレクトリーの名前。デフォルト値は **.niogit** です。
- **org.uberfire.nio.git.ketch**: Git ketch を有効化または無効化。
- **org.uberfire.nio.git.hooks**: Business Central の Git ディレクトリーの場所。

### HTTP 経由の Git

次のプロパティーを使用して、HTTP 経由で Git リポジトリーにアクセスできるように設定します。

- **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。デフォルト値は **false** です。
- **http.proxyHost**: HTTP プロキシのホスト名を定義します。デフォルト値は **null** です。
- **http.proxyPort**: HTTP プロキシのポート (整数値) を定義します。デフォルト値は **null** です。
- **http.proxyUser**: HTTP プロキシ名を定義します。
- **http.proxyPassword**: HTTP プロキシのユーザーパスワードを定義します。
- **org.uberfire.nio.git.http.enabled**: HTTP デモンを有効または無効にします。デフォルト値は **true** です。
- **org.uberfire.nio.git.http.host**: このデーモンは、HTTP デモンが有効な場合にホストの識別子としてこのプロパティーを使用します。これは、HTTP 経由で Git リポジトリー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。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.http.hostname**: HTTP デ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ーを使用します。これは、HTTP 経由で Git リポジトリー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。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.http.port**: このデーモンは、HTTP デモンが有効な場合にポート番号としてこのプロパティーを使用します。これは、HTTP 経由で Git リポジトリー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。デフォルト値は **8080** です。

## HTTPS 経由の Git

次のプロパティを使用して、HTTPS 経由で Git リポジトリにアクセスできるように設定します。

- **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。デフォルト値は **false** です。
- **https.proxyHost:** HTTPS プロキシのホスト名。デフォルト値は **null** です。
- **https.proxyPort:** HTTPS プロキシのホストポート (整数値)。デフォルト値は **null** です。
- **https.proxyUser:** HTTPS プロキシ名を定義します。
- **https.proxyPassword:** HTTPS プロキシのユーザーパスワードを定義します。
- **user.dir:** ユーザーディレクトリーの場所。
- **org.uberfire.nio.git.https.enabled:** HTTPS デーモンを有効または無効にします。デフォルト値は **false** です。
- **org.uberfire.nio.git.https.host:** このデーモンは、HTTPS デーモンが有効な場合に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。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.https.hostname:** このデーモンは、HTTPS デーモンが有効な場合に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。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.https.port:** このデーモンは、HTTPS デーモンが有効な場合に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。デフォルト値は **8080** です。

## JGit

- **org.uberfire.nio.jgit.cache.instances:** JGit キャッシュサイズを定義します。
- **org.uberfire.nio.jgit.cache.overflow.cleanup.size:** JGit キャッシュオーバーフローのクリーンアップサイズを定義します。
- **org.uberfire.nio.jgit.remove.eldest.iterations:** 最も古い JGit の反復を削除するかどうかを定義します。
- **org.uberfire.nio.jgit.cache.evict.threshold.duration:** JGit 退避のしきい値の期間を定義します。
- **org.uberfire.nio.jgit.cache.evict.threshold.time.unit:** JGit 退避のしきい値の時間単位を定義します。

## Git デーモン

次のプロパティを使用して、Git デーモンを有効にして設定します。

- **org.uberfire.nio.git.daemon.enabled:** Git デーモンを有効または無効にします。デフォルト値は **true** です。

- **org.uberfire.nio.git.daemon.host**: Git デーモンが有効な場合は、このプロパティーをローカルホストの識別子として使用します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.daemon.hostname**: Git デーモンが有効な場合は、このプロパティーをローカルホスト名の識別子として使用します。デフォルト値は **localhost** です。
- **org.uberfire.nio.git.daemon.port**: Git デーモンが有効な場合は、このプロパティーをポート番号として使用します。デフォルト値は **9418** です。
- **org.uberfire.nio.git.http.sslVerify**: Git リポジトリを確認する SSL 証明書を有効または無効にします。デフォルト値は **true** です。



### 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。

## Git SSH

次のプロパティーを使用して、Git SSH デーモンを有効にして設定します。

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



### 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。デフォルトは作業ディレクトリーです。
- **org.uberfire.nio.git.ssh.idle.timeout**: SSH のアイドルタイムアウトを設定します。
- **org.uberfire.nio.git.ssh.passphrase**: SCP スタイルの URL を持つ Git リポジトリ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。たとえば、**git@github.com:user/repository.git** です。
- **org.uberfire.nio.git.ssh.algorithm**: SSH で使用されるアルゴリズム。デフォルト値は **RSA** です。
- **org.uberfire.nio.git.gc.limit**: GC の制限を設定します。

- **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。

## Process Server ノードおよび Process Automation Manager コントローラー

以下のプロパティを使用して、Process Automation Manager コントローラーから Process Server ノードとの接続を設定します。

- **org.kie.server.controller**: この URL は Process Automation Manager コントローラーへの接続に使用されます。たとえば、**ws://localhost:8080/business-central/websocket/controller** などです。
- **org.kie.server.user**: Process Automation Manager コントローラーから Process 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 Process 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。

## Maven など

以下のプロパティを使用して、Maven などの機能を設定します。

- **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。デフォルト値は **false** です。



### 注記

このプロパティは、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。デフォルト値は **true** です。
- **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が表示されないようにします。デフォルト値は **FULL** です。
- **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。このプロパティは、Business Central が起動する前に設定してください。デフォルト値は、内部の **m2** リポジトリへのファイルパスです。

- **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**: このプロパティは、デフォルトの SSH キーストアを使用する場合にユーザーの SSH 公開鍵の保存フォルダーを定義します。このプロパティを指定しないと、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。デフォルト値は **false** です。
- **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.uberfire.metadata.index.dir**: Lucene の **.index** ディレクトリーが保存される場所。デフォルトは作業ディレクトリーです。
- **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。プリンシパルの値とロール名が一致する場合は、アプリケーションのロール名が変数ロール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。デフォルト値は **false** です。
- **org.uberfire.secure.key**: パスワードの暗号化で使用するパスワード。デフォルト値は **org.uberfire.admin** です。
- **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。デフォルト値は **PBEWithMD5AndDES** です。
- **org.uberfire.domain**: uberfire が使用するセキュリティドメイン名。デフォルト値は **ApplicationRealm** です。
- **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。デフォルト値は **<working-directory>/repositories/kie** です。
- **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。デフォルト値は **false** です。
- **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。デフォルト値は **false** です。
- **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。デフォルト値は **false** です。

### Process Automation Manager コントローラー

以下のプロパティを使用して、Process Automation Manager コントローラーへの接続方法を設定します。

- **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。デフォルト値は **kieserver** です。
- **org.kie.workbench.controller.pwd**: Process Automation Manager コントローラーのパスワード。デフォルト値は **kieserver1!** です。
- **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

### Java Cryptography Extension KeyStore (JCEKS)

JCEKS を設定するには、以下のプロパティを使用します。

- **kie.keystore.keyStoreURL**: Java Cryptography Extension KeyStore (JCEKS) の読み込みに使用する URL。たとえば、**file:///home/kie/keystores/keystore.jceks** です。
- **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

### レンダリング

以下のプロパティを使用して、Business Central と Process Server のレンダリングフォームを切り替えます。

- **org.jbpm.wb.forms.renderer.ext**: Business Central と Process 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。デフォルト値は **false** です。
- **org.jbpm.wb.forms.renderer.name**: Business Central と Process 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。デフォルト値は **workbench** です。

## 第23章 BUSINESS CENTRAL 使用時のパフォーマンスチューニングに関する考慮点

以下の主要な概念または推奨のプラクティスを使用すると、Business Central の設定および Red Hat Process Automation Manager のパフォーマンス最適化に役立ちます。本セクションではこの概念についてまとめており、随時、他のドキュメントを相互参照して詳細を説明します。本セクションは、Red Hat Process Automation Manager の新しいリリースで、必要に応じて拡張または変更します。

### 開発時には、必ず開発モードを有効にする

Process Server または特定のプロジェクトを Business Central に設定して、**production** モードまたは **development** モードを使用できます。デフォルトでは、Process Server と Business Central の新規プロジェクトはすべて、開発モードになっています。このモードには、プロジェクトの開発ポリシーに柔軟性をもたせるなど、開発が容易にすすむ機能や、重複した GAV の検出を無効化するなど、開発中の Process Server のパフォーマンスを最適化する機能が含まれます。Red Hat Process Automation Manager 環境が確立し、実稼働モードを実行できる準備が完全に整うまで、開発モードを使用してください。

環境モードの設定または重複する GAV の検出の詳細は、以下の資料を参照してください。

- [18章 Process Server および Business Central での環境モードの設定](#)
- 『[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](#)』

### 複雑なガイド付きデシジョンテーブルの検証および妥当性確認の無効化

Business Central のデシジョンテーブルの検証および妥当性確認機能は、デフォルトで有効になっています。この機能を使用すると、ガイド付きデシジョンテーブルの検証が容易になりますが、複雑なガイド付きデシジョンテーブルの場合は、この機能が原因でデシジョンエンジンのパフォーマンスが低下してしまう可能性があります。**org.kie.verification.disable-dtable-realtime-verification** のシステムプロパティを **true** に設定して、この機能を無効にできます。

ガイド付きデシジョンテーブルの詳細は、『[ガイド付きデシジョンテーブルを使用したデシジョンサービスの作成](#)』を参照してください。

### 大規模なプロジェクトが多数ある場合は、自動ビルドを無効にする

Business Central では、**Project Explorer** のサイドパネルでプロジェクト間を移動すると、選択したプロジェクトが自動でビルドされて、**Alerts** ウィンドウが更新され、プロジェクトのビルドエラーが表示されます。大規模なプロジェクトがある場合や、頻繁に、開発が実際に行われている多数のプロジェクト間を頻繁に切り替える場合は、この機能を使用すると、Business Central とデシジョンエンジンのパフォーマンスに悪影響を与える可能性があります。

プロジェクトの自動ビルドを無効にするには、**org.kie.build.disable-project-explorer** システムプロパティを **true** に設定します。

## 付録A バージョン情報

本書の最終更新日：2021年11月15日（月）