



Red Hat JBoss BPM Suite 6.3

Guide d'administration et de configuration

Le guide d'administration et de configuration de Red Hat JBoss BPM Suite

Red Hat JBoss BPM Suite 6.3 Guide d'administration et de configuration

Le guide d'administration et de configuration de Red Hat JBoss BPM Suite

Red Hat Content Services

Klara Kufova

kkufova@redhat.com

Marek Czernek

mczernek@redhat.com

Tomas Radej

tradej@redhat.com

Vidya Iyengar

viyengar@redhat.com

Notice légale

Copyright © 2016 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Résumé

Guide pour les administrateurs et les utilisateurs avancés qui s'occupent des installations, configurations et utilisations avancées de Red Hat JBoss BPM Suite.

Table des matières

PARTIE I. INTRODUCTION	5
CHAPITRE 1. BPMN (DE L'ANGLAIS BUSINESS PROCESS MODEL AND NOTATION)	6
1.1. COMPOSANTS	6
1.2. PROJET	7
1.3. CRÉATION D'UN PROJET	7
1.4. AJOUTER DES DÉPENDANCES	9
PARTIE II. CONFIGURATION	11
CHAPITRE 2. CONFIGURATION DE BUSINESS CENTRAL	12
2.1. CONTRÔLE DE L'ACCÈS	12
2.2. CONFIGURATION D'UN PROFIL DE BUSINESS CENTRAL	13
2.3. PERSONNALISATION DE L'APPLICATION BUSINESS CENTRAL	14
2.4. DESCRIPTEURS DE DÉPLOIEMENT	17
2.5. GESTION DE LA POLITIQUE DE REMPLACEMENT DE DÉPLOIEMENT	21
2.6. ÉTENDRE BUSINESS CENTRAL	22
2.7. CONFIGURER DES COLONNES SUR UN TABLEAU	29
CHAPITRE 3. HOOKS DE RÉFÉRENTIEL	31
3.1. CONFIGURER LES GIT HOOKS.	31
CHAPITRE 4. CONFIGURATION DE LA LIGNE DE COMMANDE	34
4.1. DÉMARRER L'OUTIL KIE-CONFIG-CLI EN MODE EN LIGNE	34
4.2. DÉMARRER L'OUTIL KIE-CONFIG-CLI EN MODE HORS LIGNE	34
4.3. COMMANDES DISPONIBLES POUR L'OUTIL KIE-CONFIG-CLI	35
CHAPITRE 5. MIGRATION	37
5.1. MIGRATION DES DONNÉES	37
5.2. MIGRATION DE RUNTIME	38
5.3. API ET COMPATIBILITÉ RÉTROACTIVE	39
5.4. MIGRATION DU SERVICE DE TÂCHES	39
CHAPITRE 6. GESTION DES DONNÉES	41
6.1. SAUVEGARDE DES DONNÉES	41
6.2. INSTALLATION DES INDEXES	41
6.3. INSTALLER LA BASE DE DONNÉES	41
6.4. MODIFIER LA BASE DE DONNÉES	42
6.5. SCRIPTS DDL	43
CHAPITRE 7. RÉFÉRENTIEL DE RESSOURCES	44
7.1. CRÉATION D'UNE UNITÉ ORGANISATIONNELLE	44
7.2. CRÉATION D'UN RÉFÉRENTIEL	46
7.3. CLONAGE D'UN RÉFÉRENTIEL	48
7.4. SUPPRIMER UN RÉFÉRENTIEL	50
7.5. GESTION DES RESSOURCES	51
7.6. RÉFÉRENTIEL MAVEN	56
7.7. CONFIGURER UN DÉPLOIEMENT POUR UN RÉFÉRENTIEL NEXUS DISTANT	57
7.8. CONFIGURATION SYSTÈME	58
CHAPITRE 8. EXPORTATION ET IMPORTATION D'UN PROCESSUS	61
8.1. CRÉER UNE DÉFINITION DE PROCESSUS	61
8.2. IMPORTATION D'UNE DÉFINITION DE PROCESSUS	61
8.3. IMPORTER JPDL 3.2 DANS BPMN2	62

8.4. EXPORTATION D'UN PROCESSUS	63
PARTIE III. INTÉGRATION	64
CHAPITRE 9. LE DÉPLOIEMENT D'ARTEFACTS DE RED HAT JBOSS BPM SUITE DANS S-RAMP (DE L'ANGLAIS REPOSITORY ARTIFACT MODEL AND PROTOCOL)	65
9.1. LE DÉPLOIEMENT D'ARTEFACTS DE RED HAT JBOSS BPM SUITE DANS SOA REPOSITORY ARTIFACT MODEL AND PROTOCOL (S-RAMP) EN UTILISANT MAVEN	65
9.2. LE DÉPLOIEMENT D'ARTEFACTS DE RED HAT JBOSS BPM SUITE DANS SOA REPOSITORY ARTIFACT MODEL AND PROTOCOL (S-RAMP) EN UTILISANT L'INTERFACE GRAPHIQUE (GUI).	67
CHAPITRE 10. INTÉGRER RED HAT JBOSS BPM SUITE DANS RED HAT JBOSS FUSE	68
10.1. FONCTIONNALITÉS DE BASE DE JBOSS BPM SUITE ET DE JBOSS BRMS	68
10.2. AUTRES FONCTIONNALITÉS DE L'INTÉGRATION SWITCHYARD ET CAMEL	69
10.3. INSTALLER / METTRE À JOUR LES FONCTIONNALITÉS D'INTÉGRATION PRINCIPALES	70
10.4. INSTALLER DES FONCTIONNALITÉS D'INTÉGRATION SUPPLÉMENTAIRES	71
10.5. CONFIGURATION DES DÉPENDANCES	72
10.6. INSTALLER LES APPLICATIONS QUICKSTART DE JBOSS FUSE INTEGRATION	73
CHAPITRE 11. INSTALLATION DE RED HAT JBOSS BPM SUITE AVEC RED HAT SSO.	76
11.1. AUTHENTIFICATION BUSINESS CENTRAL VIA RH-SSO	77
11.2. AUTHENTIFICATION INTELLIGENT PROCESS SERVER VIA RH-SSO	83
11.3. AUTHENTIFICATION DE TIERCE PARTIE VIA RH-SSO	85
CHAPITRE 12. INTÉGRATION AVEC SPRING	88
12.1. CONFIGURER RED HAT JBOSS BPM SUITE AVEC SPRING	88
CHAPITRE 13. INTÉGRATION CDI	90
13.1. INTÉGRATION CDI	90
CHAPITRE 14. PERSISTANCE	92
14.1. SESSION	92
14.2. INSTANCE DE PROCESSUS	93
14.3. ÉLÉMENT DE TRAVAIL	94
14.4. CONFIGURATION DE LA PERSISTANCE	95
CHAPITRE 15. TRANSACTIONS	99
15.1. TRANSACTIONS	99
15.2. DÉFINIR DES TRANSACTIONS	99
15.3. CMT (CONTAINER MANAGED TRANSACTIONS)	100
CHAPITRE 16. JOURNALISATION	102
16.1. JOURNALISATION DES ÉVÉNEMENTS DANS LA BASE DE DONNÉES	104
16.2. FONCTIONNALITÉ LOGBACK	105
16.3. CONFIGURER LA JOURNALISATION	106
16.4. GÉRER DES FICHIERS JOURNAUX	107
CHAPITRE 17. LOCALISATION ET PERSONNALISATION	112
17.1. LANGUES DISPONIBLES	112
17.2. MODIFIER LES PARAMÈTRES DE LANGUE	112
17.3. EXÉCUTER UNE JVM AVEC LA CODIFICATION UTF-8	113
PARTIE IV. EXÉCUTION	114
CHAPITRE 18. CONFIGURATION DU SERVEUR D'EXÉCUTION DES PROCESSUS	115
18.1. RÈGLES D'ATTRIBUTION	115
18.2. SESSION MAIL	116

CHAPITRE 19. PLUG-IN DE RED HAT JBOSS DEVELOPER STUDIO	118
19.1. PLUG-IN	118
PARTIE V. MONITORING	119
CHAPITRE 20. MONITORING DES PROCESSUS	120
20.1. RÉSEAU JBOSS OPERATIONS NETWORK	120
20.2. INSTALLER LE PLUG-IN JBOSS BRMS DANS JBOSS ON	120
20.3. CONTRÔLE DES KIEBASES ET DES KIESESSIONS	121
CHAPITRE 21. GESTION DE LA SÉCURITÉ DANS RED HAT JBOSS BPM SUITE DASHBUILDER	122
21.1. ACCÉDER À RED HAT JBOSS BPM SUITE DASHBUILDER	122
21.2. GESTION DE LA SÉCURITÉ	122
21.3. PERMISSIONS D'ESPACE DE TRAVAIL	122
21.4. PERMISSIONS DE PAGE	125
21.5. PERMISSIONS DE PANNEAU	126
ANNEXE A. HISTORIQUE DES VERSIONS	128

PARTIE I. INTRODUCTION

CHAPITRE 1. BPMN (DE L'ANGLAIS BUSINESS PROCESS MODEL AND NOTATION)

BPMN (Business Process Model and Notation) est un système d'annotations standards de modélisation des processus métier. BPMS aspire à réduire l'écart entre les analystes et les programmeurs en procurant un langage de flux de travail qui puisse être clairement compris par les deux.

1.1. COMPOSANTS

Red Hat JBoss BPM Suite intègre plusieurs composantes pour soutenir les processus métier tout au long de leur cycle de vie et pour fournir des fonctionnalités de gestion des processus et des outils pour les analystes, les développeurs et les utilisateurs professionnels. Le produit peut être déployé sur plusieurs serveurs JEE compatibles ; l'option recommandée est Red Hat JBoss Enterprise Application Platform 6.

Red Hat JBoss BPM Suite contient les composantes principales suivantes :

- **Execution Engine** - fournit l'environnement d'exécution des processus et des règles d'entreprise. Il comprend une bibliothèque de flux de travail qui peut être incorporée dans une application web de l'utilisateur. Le Runtime Manager est l'objet root et il contient les composantes suivants :
 - **Runtime Engine** - implémente le comportement principal du langage informatique de la machine et il est fourni par le Runtime Manager.
 - **Process Engine** - est l'environnement d'exécution du modèle de processus d'entreprise.
 - **Task Service** - gère les cycles de vie des tâches humaines.
 - **Rule Engine** - peut être utilisé avec le moteur de processus ou tout seul.
 - **Rules Evaluation** - exécute les règles métier sur la base d'un ensemble de faits donnés.
 - **Complex Event Processing** - applique les règles métier sur des flux d'événements entrants.
- **Business Central** - une application basée-web qui accommode des outils de création de ressources, de gestion et de contrôle en procurant un environnement web intégré.
 - **Référentiel Ressources** - c'est l'emplacement central de partage des connaissances (Knowledge Store) des ressources d'entreprise, processus, règles, formulaires, etc.. Sur la page Project Explorer de Business Central via **Création** → **Création Projet**, les utilisateurs peuvent accéder à ce référentiel. Par défaut, le produit initialise un référentiel GIT local comme son référentiel de ressources. Toutefois, d'autres référentiels peuvent être ajoutés ou retirés si nécessaires.
 - **Référentiel d'artefacts** - c'est un référentiel de stockage d'artefacts de jars de projets basé Maven
 - **Serveur d'exécution** - fournit un environnement d'exécution pour les tâches et les instances de processus d'entreprise.

- o **Surveillance Activité** - fournit un affichage personnalisable sur la performance de l'entreprise.



NOTE

Red Hat JBoss BRMS est fourni avec sa propre application Business Central qui représente un sous-ensemble de l'application Business Central de Red Hat JBoss BPM Suite.

1.2. PROJET

Un projet est un conteneur pour les paquets de ressources (processus d'entreprise, règles, définitions de travaux, tables de décision, modèles de faits, modèles de données et des DSL) qui se trouvent dans le référentiel de connaissances. C'est ce conteneur qui définit les propriétés de KIE Base et de KIE Session, qui sont appliquées à son contenu. Dans l'interface graphique, vous pouvez modifier ces entités dans l'éditeur de projet.

Un projet étant un projet Maven, il contient le fichier de Project Object Model (`pom.xml`) avec des informations sur la façon de construire l'artefact sortant. Il contient également le fichier de descripteur de module, `kmodule.xml`, qui contient la configuration de KIE Base et de KIE Session pour les ressources du projet.

1.3. CRÉATION D'UN PROJET

Il est possible de créer un projet dans la perspective **Création de projet** de Business Central, en utilisant les invocations d'API REST.

Créer un projet dans Business Central



IMPORTANT

Notez que seuls les utilisateurs ayant un rôle **admin** dans Business Central peuvent créer des projets.

Procédure 1.1. Utiliser Business Central pour créer un projet

1. Dans Business Central, aller dans **Création** → **Création du projet**.
2. Dans **Project Explorer**, sélectionner l'unité organisationnelle et le référentiel où vous souhaitez créer le projet.
3. Dans le menu perspective, cliquer sur **Nouvel élément** → **Projet**.

La fenêtre **Nouveau projet** apparaît.

New Project

☒ New Project Wizard

Project General Settings

Project Name

MortgageProject

Project Description

Insert a project description for documentation purposes ...

Group artifact version

Group ID ⓘ

org.mycompany.commons

Example: com.myorganization.myprojects

Artifact ID ⓘ

myframework

Example: MyProject

Version ⓘ

2.1.1

Example: 1.0.0

< Previous

Next >

Cancel

✓ Finish

4. Définir les informations **Configuration générale Projet** et **Version Artefact Groupe** du nouveau projet. Ces paramètres se trouvent dans le fichier de configuration `pom.xml` de Maven.

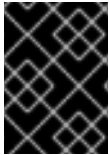
Voir la description détaillée des paramètres :

- **Nom du projet** : nom du projet; (par exemple `MortgageProject`).
- **Description du projet** : description du projet qui peut être utile à des buts de documentation du projet.
- **ID Groupe** : ID de groupe du projet (par exemple `org.mycompany.commons`).
- **ID Artefact** : ID d'artefact unique au groupe (par exemple `myframework`). Évitez d'utiliser un espace ou un autre caractère spécial qui risque de vous mener à un nom non valide.
- **ID Version** : version du projet (par exemple `2.1.1`).

5. Cliquer sur **Terminer**.

L'affichage sur l'écran du projet est mise à niveau avec des nouveaux détails sur le projet, ainsi définis dans le fichier `pom.xml`. Vous pouvez basculer entre les fichiers de descripteur du projet et modifier leur contenu en cliquant sur le bouton **Configuration Projet : Configurations générales du projet** en haut de l'écran.

Créer un projet en utilisant l'API REST



IMPORTANT

Notez que seuls les utilisateurs ayant un rôle **rest-all** ou **rest-project** peuvent créer des projets.

Pour créer un projet dans le référentiel, invoquez un appel API REST **POST**. Les informations sur le projet sont définies par l'entité JSON correspondante.

Le paramètre de sortie de l'appel est une instance **Entity**. L'appel retourne une instance **CreateProjectRequest**.

Exemple 1.1. Créer un projet en utilisant l'utilitaire Curl

Exemple d'entité JSON contenant les détails d'un projet à créer :

```
{
  "name"      : "MortgageProject",
  "description": null,
  "groupId"   : "org.mycompany.common",
  "version"   : "2.1.1"
}
```

Exécuter la commande suivante :

```
curl -X POST 'localhost:8080/business-
central/rest/repositories/REPOSITORY_NAME/projects/' -u
USERNAME:PASSWORD -H 'Accept: application/json' -H 'Content-Type:
application/json' -d
'{"name":"MortgageProject","description":null,"groupId":"org.mycompany.c
ommons","version":"2.1.1"}'
```

Pour obtenir plus d'informations, voir *Guide de développement de Red Hat JBoss BPM Suite* chapitre *API REST Knowledge Store*, section *Appels de référentiels*

1.4. AJOUTER DES DÉPENDANCES

Pour ajouter des dépendances à votre projet, procéder ainsi :

1. Ouvrir le **Project Editor** pour le projet donné :
 - a. Sur la page **Project Explorer** de la perspective **Création Projet**, ouvrir le répertoire du projet.
 - b. Cliquer sur **Open Project Editor** pour afficher le projet.
2. Dans la vue **Écran du projet**, sélectionner dans la boîte de dialogue **Configuration Projet** l'élément **Dépendances**.
3. Sur l'écran **Écran du projet** mis à jour, cliquer sur le bouton **Ajouter** pour ajouter une dépendance Maven ou cliquer sur le bouton **Ajouter à partir du référentiel** pour ajouter une dépendance du Knowledge Store (référentiel d'artefacts) :
 - o Quand il ajoute une dépendance Maven, un utilisateur doit définir les **ID Groupe**, **ID**

Artefact et ID Version dans la boîte de dialogue **Dépendances**.

- Quand on ajoute une dépendance du Knowledge Store, il faut sélectionner la dépendance dans la boîte de dialogue qui s'affiche : la dépendance sera ajoutée au tableau des dépendances.
4. Pour pouvoir appliquer les différents changements, les dépendances doivent être sauvegardées.

De plus, vous pouvez utiliser la **Liste blanche Paquet** quand vous travaillez avec des dépendances. Quand vous ajoutez un référentiel, vous pouvez cliquer sur la roue dentée et sélectionner **Tout ajouter** ou **Ne rien ajouter**, ce qui aboutit à tous ou aucuns des paquets de la dépendance ajoutée.



AVERTISSEMENT

Si vous travaillez avec des artefacts modifiés, ne chargez pas à nouveau des artefacts non liés à des instantanés modifiés car Maven ne reconnaîtra pas ces artefacts mis à jour, et il ne fonctionnera pas si c'est déployé ainsi.

PARTIE II. CONFIGURATION

CHAPITRE 2. CONFIGURATION DE BUSINESS CENTRAL

Business Central est une application web, tous les paramètres de configuration sont chargés à partir du `DEPLOY_DIRECTORY/business-central.war/WEB-INF/web.xml` et des fichiers référencés, et si déployé dans Red Hat JBoss EAP 6, également dans `jboss-web.xml` et `jboss-deployment-structure.xml`.

Notez que l'application totale peut être exécutée dans des profils différents (voir *Red Hat JBoss BPM Suite Installation Guide*).

2.1. CONTRÔLE DE L'ACCÈS

Le mécanisme de contrôle de l'accès inclut l'autorisation et l'authentification. Dans l'environnement unifié de Red Hat JBoss BPM Suite, les utilisateurs peuvent mettre à jour les rôles d'utilisateur situés dans `$JBOSS_HOME/standalone/deployments/business-central.war/WEB-INF/classes/userinfo.properties`.

Pour donner accès à la JBoss BPM Suite à un utilisateur, l'utilisateur devra avoir le rôle suivant assigné :

- **admin** : administre le système JBoss BPM Suite et possède des droits pour effectuer les changements nécessaires, y compris la possibilité d'ajouter ou de supprimer des utilisateurs du système.
- **développeur** : implémente le code requis pour que les processus fonctionnent et il a accès à tout sauf aux tâches administratives.
- **analyste** : crée et conceptualise des processus et des formulaires, instancie les artefacts de processus et de déploiement. Ce rôle est similaire à celui d'un développeur, sans l'accès au référentiel des ressources, ni aux déploiements.
- **utilisateur** : réclame, effectue et fait appel à d'autres actions (par exemple, remontée, rejet, etc.) sur les tâches affectées et n'a pas accès aux fonctions de création (authoring).
- **manager** : contrôle le système et ses statistiques, et a uniquement accès au tableau de bord.
- **utilisateur de tâches d'entreprise** : effectue les tâches d'entreprise requises pour que les processus puissent continuer. Travaille surtout sur la liste de tâches.

Si vous utilisez Red Hat JBoss EAP, pour créer un utilisateur avec des rôles particuliers, exécutez le script `$JBOSS_HOME/add-user.sh` et créez un utilisateur d'application dans `ApplicationRealm` avec les rôles respectifs.

Configuration de workbench

Dans Red Hat JBoss Suite BPM, les utilisateurs peuvent créer des rôles à l'aide de LDAP pour modifier les rôles existants. Les utilisateurs peuvent modifier les rôles dans la configuration workbench pour s'assurer que les rôles uniques basés LDAP soient conformes aux standards de l'entreprise en modifiant le répertoire de déploiement situé dans `$JBOSS_HOME/standalone/deployments/business-central.war/WEB-INF/classes/workbench-policy.properties`.

Si vous authentifiez un utilisateur via LDAP via GIT, les administrateurs doivent définir la propriété système `org.uberfire.domain` au nom du module de connexion qu'ils doivent utiliser pour authentifier les utilisateurs par le service GIT. Cela doit être défini dans le fichier `standalone.xml` dans EAP.

Authentification pour les tâches humaines

Chaque tâche qui doit être exécutée est assignée à un ou plusieurs rôles ou groupes, de façon à ce que chaque rôle donné ou que le groupe donné assigné puisse réclamer l'instance de la tâche et l'exécuter. Les tâches peuvent être assignées directement à un ou à plusieurs utilisateurs. JBoss BPM Suite utilise l'interface `UserGroupCallback` pour assigner des tâches aux utilisateurs.



AVERTISSEMENT

Un groupe de tâche humaine ne doit pas être nommé d'après un utilisateur existant dans le système. Cela pourrait causer des problèmes intermittents.

2.2. CONFIGURATION D'UN PROFIL DE BUSINESS CENTRAL

Le serveur de Red Hat JBoss BPM Suite 6 (ou versions supérieures) est capable de démarrer l'application Business Central de trois façons différentes :

- Full profile - profil par défaut actif sans besoin de configuration supplémentaire (IU et services à distance, par ex. REST).
- Execution server profile - désactive les composants IU de l'application complètement et permet un accès éloigné uniquement, par ex. via l'interface REST.
- UI server profile - désactive les services à distance comme REST et permet uniquement l'accès à l'IU à l'application.

Pour modifier le profil, utiliser les étapes de configuration suivantes.

Procédure 2.1. Configuration des profils de Business Central

1. Sélectionner le `web.xml` désiré dans `$BPMS_HOME/standalone/deployments/business-central.war/WEB-INF/`. Les fichiers suivants sont fournis.
 - `web.xml` (valeur par défaut) pour le profil complet
 - `web-exec-server.xml` pour le profil de serveur d'exécution
 - `web-ui-server.xml` pour le profil de serveur de l'IU
2. Pour activer un profil autre que le profil complet (full profil) par défaut, le fichier `web-<PROFILE>.xml` doit être renommé `web.xml`. Les étapes suivantes montrent une façon d'exécuter le profil du serveur.
 - a. Sauvegarder le fichier `web.xml` du profil complet (full profile).

```
$ mv web.xml web-full.xml
```

- b. Renommer le fichier `web-exec-server.xml` :

```
$ mv web-exec-server.xml web.xml
```

3. Démarrer le serveur d'applications avec une propriété système supplémentaire pour instruire le gestionnaire de profils d'activer le profil donné.
 - `Dorg.kie.active.profile=full` - pour activer le profil complet (full profil) ou éviter la propriété complètement.
 - `Dorg.kie.active.profile=exec-server` - pour activer le profil de serveur d'exécution
 - `Dorg.kie.active.profile=ui-server` - pour activer le serveur de l'IU

2.3. PERSONNALISATION DE L'APPLICATION BUSINESS CENTRAL

L'application web Business Central permet de personnaliser son apparence en vous permettant de substituer certains de ses styles qui existent par défaut. La possibilité de personnaliser Business Central vous permet d'obtenir une apparence cohérente à travers toutes vos applications, ce qui améliore votre expérience utilisateur. C'est également utile lorsque plusieurs équipes utilisent l'application. Chaque équipe peut développer sa propre interface utilisateur personnalisée. Les éléments personnalisables sont construits en utilisant des feuilles de style en cascade (CSS), des images et des fichiers HTML, fournissant ainsi une approche flexible et facile à personnaliser, sans besoin de recompiler le code.

Vous pouvez modifier les éléments suivants dans l'application Business Central pour rester en phase avec l'image de marque de votre entreprise :

- Écran de connexion

Vous pouvez personnaliser les attributs de l'écran de connexion de Business Central suivants :

- L'image d'arrière-plan
- Le logo de la société
- Le logo de l'application

- L'en-tête de l'application

Vous pouvez personnaliser les attributs de l'en-tête de l'application de Business Central suivants :

- L'en-tête de Business Central contient le titre et le logo de la bannière

- Fenêtres contextuelles d'assistance

Vous pouvez personnaliser les attributs suivants des fenêtres contextuelles d'assistance de l'écran de démarrage :

- Les images d'assistance de l'écran de démarrage
- Texte de l'étiquette

2.3.1. Personnalisation de la page de connexion de Business Central

Procédure 2.2. Modifier l'image d'arrière-plan de la page de connexion de Business Central

1. Démarrer le serveur EAP et ouvrir <http://localhost:8080/business-central> dans un navigateur web.
2. Copier la nouvelle image d'arrière-plan dans le répertoire `$EAP_HOME/standalone/deployments/business-central.war/images` de votre installation JBoss BPM Suite.
3. Naviguez dans le répertoire `$EAP_HOME/standalone/deployments/business-central.war/styles` et ouvrir le fichier `login-screen.css` dans l'éditeur de texte.
4. Dans le fichier `login-screen.css`, indiquez l'emplacement de votre nouvelle image d'arrière-plan dans l'attribut ***background-image*** suivant.

```
background-image: url("../images/login-screen-background.jpg");
```

L'attribut ***background-image*** pointe vers l'image `login-screen-background.jpg` par défaut.

En plus de l'image d'arrière-plan, vous pouvez modifier les autres attributs comme la taille de l'image, sa position, la couleur du fond dans le fichier `login-screen.css`.

Réactualiser la page de connexion de Business Central pour apercevoir les changements.

Procédure 2.3. Modifier les logos de la société et du projet dans Business Central

1. Démarrer le serveur EAP et ouvrir <http://localhost:8080/business-central> dans un navigateur web.
2. Naviguez dans le répertoire `$EAP_HOME/standalone/deployments/business-central.war/images` de votre installation JBoss BPM Suite.
3. Remplacer l'image par défaut `login-screen-logo.png` par une nouvelle image. Ce sera le logo de la société, qui apparaîtra dans le coin de droite de votre page de connexion.
4. Remplacer l'image par défaut `RH_JBoss_BPMS_Logo.png` `RH_JBoss_BRMS_Logo.png` par une nouvelle image. Ce sera le logo du projet qui apparaîtra au centre légèrement à gauche de la page de connexion.

Réactualiser la page de connexion de Business Central pour apercevoir les changements.

2.3.2. Personnalisation de l'en-tête de l'application Business Central

Procédure 2.4. Modifier l'en-tête de l'application Business Central (bannière)

1. Démarrer le serveur EAP et ouvrir <http://localhost:8080/business-central> dans un navigateur web.
2. Connectez-vous à l'application Business Central par vos identifiants d'utilisateur.
3. Copier la nouvelle image d'en-tête de votre application dans le répertoire `$EAP_HOME/standalone/deployments/business-central.war/images` de votre installation JBoss BPM Suite.
4. Ouvrir le fichier `$EAP_HOME/standalone/deployments/business-central.war/banner/banner.html` dans un éditeur de texte.

5. Dans le fichier **banner.html**, modifier la balise `<img>` suivante pour donner un nom à votre nouvelle image d'en-tête :

```

```

L'image par défaut est **logo.png**.

Réactualiser la page d'accueil de Business Central pour apercevoir les changements.

2.3.3. Personnalisation des fenêtres d'assistance de l'écran de démarrage de Business Central

Le répertoire `$EAP_HOME/standalone/deployments/business-central.war/plugins` contient les pages d'écran de démarrage et les fichiers html correspondants. Chaque page d'écran de démarrage (splash page) contient le nom du fichier html qui contient des informations sur le ou les image(s) à afficher. Par exemple, la page d'écran de démarrage `authoring_perspective.splash.js` pointe vers le fichier `authoring_perspective.splash.html`. Le fichier `authoring_perspective.splash.html` contient les noms et emplacements de tous les fichiers d'images qui apparaissent sur la page d'assistance de démarrage d'Authoring Perspective, ainsi que leurs légendes.

Procédure 2.5. Modifier les images et légendes des fenêtres contextuelles d'assistance de Business Central

1. Démarrer le serveur EAP et ouvrir <http://localhost:8080/business-central> dans un navigateur web.
2. Connectez-vous à l'application Business Central par vos identifiants d'utilisateur.
3. Copier la ou les nouvelle(s) image(s) d'assistance de la page de démarrage dans le répertoire `$EAP_HOME/standalone/deployments/business-central.war/images` de votre installation JBoss BPM Suite.
4. Ouvrir le fichier html correspondant du répertoire `$EAP_HOME/standalone/deployments/business-central.war/plugins` dans l'éditeur de texte.
5. Modifiez le fichier html pour qu'il pointe vers votre nouvelle image d'aide de l'écran de démarrage. Par exemple, pour modifier la première image qui apparaît dans l'aide de Perspective Authoring, modifiez la balise `<img>` suivante dans le fichier `authoring_perspective.splash.html` pour ajouter votre nouvelle image :

```

```

L'image par défaut est **authoring_perspective1.png**, qui apparaît dans l'assistance de la première page d'Authoring Perspective.

6. Pour modifier la légende de l'image qui apparaît sur la page d'aide de l'écran de démarrage, modifier les contenus des balises `<h4>` et `<p>` sous la balise `<img>` :

```
<h4>Authoring</h4>
<p>Modularized and customizable workbench</p>
```

Réactualiser la page d'accueil de Business Central et accéder aux fenêtres contextuelles d'assistance de la page de démarrage pour apercevoir les changements.

2.4. DESCRIPTEURS DE DÉPLOIEMENT

Les processus et les règles de Red Hat JBoss BPM Suite 6 sont stockés dans un empackage basé Apache Maven, et sont appelés `kjar` (de l'anglais *knowledge archives*). Les règles, les processus, les ressources, etc. font partie d'un *built* de fichier `jar`, et sont gérés par Maven. Un fichier conservé dans le répertoire **META-INF** du `kjar` appelé `kmodule.xml` peut être utilisé pour définir les bases de connaissances et de sessions. Par défaut, ce fichier `kmodule.xml` est vide.

Quand un composant de runtime comme Business Central est sur le point de traiter un `kjar`, il consulte `kmodule.xml` pour construire la représentation de runtime.

Deployment Descriptors, les descripteurs de déploiement font partie d'une nouvelle fonctionnalité introduite dans la branche 6.1 de Red Hat JBoss BPM Suite, qui vous permet un contrôle affiné de votre déploiement et qui complémente le fichier `kmodule.xml`. La présence de ces descripteurs est optionnelle et votre déploiement pourra avoir lieu sans qu'ils soient présents. Les propriétés qui vous pouvez définir en utilisant ces descripteurs sont purement techniques de nature, et incluent des *meta* valeurs comme la stratégie de runtime, l'auditing et la persistance.

Ces descripteurs vous permettent de configurer le serveur d'exécution sur plusieurs niveaux (niveau serveur par défaut, différents descripteurs de déploiement par `kjar` etc.). Cela vous permet de faire des personnalisations simples à la configuration out-of-the-box du serveur d'exécution (probablement par `kjar`).

Vous pouvez définir ces descripteurs dans un fichier nommé `kie-deployment-descriptor.xml` et mettre ce fichier à côté de votre fichier `kmodule.xml` dans le dossier **META-INF**. Vous pouvez changer cet emplacement par défaut (et le nom du fichier) en le spécifiant comme paramètre système :

```
-Dorg.kie.deployment.desc.location=file:/path/to/file/company-deployment-descriptor.xml
```

2.4.1. Configuration du descripteur de déploiement

Les descripteurs de déploiement permettent à l'utilisateur de configurer le serveur d'exécution sur plusieurs niveaux :

- **niveau serveur** : le niveau principal, s'applique à tous les `kjars` déployés sur le serveur.
- **niveau `kjar`** : vous permet de configurer les descripteurs sur la base d'un `kjar`.
- **niveau temps de déploiement** : descripteurs qui s'appliquent quand un `kjar` est en cours de déploiement.

Les éléments de configuration granulaire spécifiés par les descripteurs de déploiement ont préséance sur ceux niveau serveur, sauf dans le cas d'éléments de configuration basés collection, qui sont fusionnés. La hiérarchie fonctionne ainsi : *configuration du temps de déploiement* > *configuration `kjar`* > *configuration du serveur*.



NOTE

La configuration du temps de déploiement s'applique à des déploiements effectués via l'API REST.

Ainsi, si le mode de persistance (l'un des éléments que vous pouvez configurer) défini au niveau du serveur est **NONE**, mais que le même mode est spécifié comme **JPA** au niveau kjar, le mode réel sera **JPA** pour cette kjar. Si rien n'est spécifié pour le mode de persistance dans le descripteur de déploiement pour ce kjar (ou s'il n'y a aucun descripteur de déploiement), cela renverra à la configuration niveau serveur, qui est dans ce cas **NONE** (aucun) (ou **JPA** s'il n'y a aucun descripteur de déploiement au niveau du serveur).

Pouvez-vous substituer ce comportement de mode de fusionnement par hiérarchie ?

Oui. Dans le mode par défaut, s'il existe des descripteurs de déploiement présents à plusieurs niveaux, les propriétés de configuration seront fusionnées avec les propriétés granulaires se substituant aux valeurs brutes, et avec les éléments de configuration manquants au niveau granulaire fournis par les valeurs présentes aux niveaux supérieurs. Le résultat final est une configuration fusionnée de descripteur de déploiement. Ce mode de fusion par défaut s'appelle le mode **MERGE_COLLECTIONS**. Mais vous pouvez en changer ([Section 2.4.2, « Gestion des descripteurs de déploiement »](#)) si cela ne convient pas à votre environnement :

- **KEEP_ALL**: dans ce mode, toutes les valeurs de niveau supérieur remplacent toutes les valeurs de niveau inférieur (les valeurs de niveau serveur remplacent les valeurs kjar)
- **OVERRIDE_ALL**: dans ce mode, toutes les valeurs de niveau inférieur remplacent toutes les valeurs de niveau supérieur (les valeurs kjar remplacent les valeurs de niveau serveur)
- **OVERRIDE_EMPTY**: dans ce mode, tous les éléments de configuration *non vides* des niveaux inférieurs remplacent ceux des niveaux supérieurs, y compris les éléments qui sont représentés sous forme de collections.
- **MERGE_COLLECTIONS (DEFAULT)**: dans ce mode, tous les éléments de configuration non vides de niveau inférieur remplacent les éléments de niveaux supérieurs (comme **OVERRIDE_EMPTY**), mais les propriétés de collection sont mergées (combinées).

Les descripteurs de déploiement des kjars dépendants sont mis à un niveau inférieur à celui auquel le kjar est déployé, mais ils sont toujours situés à un niveau supérieur par rapport au serveur.

Est-ce que je dois fournir un descripteur de déploiement complet pour tous les kjars ?

Non. Et c'est ici précisément que la fusion entre différents fichiers peut vous être utile. Il est possible et recommandé de fournir des descripteurs de déploiement partiels. Par exemple, si vous souhaitez substituer le mode auditing dans un kjar, vous aurez juste besoin de fournir ceci et le reste des valeurs sera fusionné du niveau serveur aux kjars de niveaux supérieurs.

Il est important de noter que quand on utilise le mode **OVERRIDE_ALL**, tous les éléments de configuration doivent être spécifiés car le kjar pertinent les utilisera toujours et ne mergera avec aucun autre descripteur de déploiement dans la hiérarchie.

Peut-on configurer ?

Les informations de configuration technique de haut niveau peuvent être configurées par les descripteurs de déploiement. Le tableau suivant en donne la liste, ainsi que des valeurs autorisées et par défaut pour chacun.

Tableau 2.1. Descripteurs de déploiement

Configuration	Entrée XML	Valeurs autorisées	Valeur par défaut
Nom d'unité de persistance pour les données de runtime	persistance-unit	N'importe quel nom valide de package de persistance	org.jbpm.domain

Configuration	Entrée XML	Valeurs autorisées	Valeur par défaut
Nom d'unité de persistance pour les données d'auditing	audit-persistence-unit	N'importe quel nom valide de package de persistance	org.jbpm.domain
Mode de persistance	persistence-mode	JPA, NONE	JPA
Mode d'auditing	audit-mode	JPA, JMS ou NONE	JPA
Stratégie de runtime	runtime-strategy	SINGLETON, PER_REQUEST ou PER_PROCESS_INSTANCE	SINGLETON
Liste des listeners d'événements à enregistrer	event-listeners	Noms de classe de listener comme ObjectModel	Pas de valeur par défaut
Liste des listeners d'événements de tâches à enregistrer	task-event-listeners	Noms de classe de listener comme ObjectModel	Pas de valeur par défaut
Liste de Work Item Handlers à enregistrer	work-item-handlers	Classes de Work Item Handlers données comme NamedObjectHandler	Pas de valeur par défaut
Liste des variables globales à enregistrer	globals	Variables globales valides données comme NamedObjectModel	Pas de valeur par défaut
Stratégies de marshallig à enregistrer (pour une persistance variable enfichable)	marshalling-strategies	Classes ObjectModel valides	Pas de valeur par défaut
Rôles exigés pour pouvoir avoir accès aux ressources du kjar	required-roles	Noms de rôles de string	Pas de valeur par défaut
Entrées environnementales supplémentaires pour les sessions de connaissance	environment-entries	NamedObjectModel valide	Pas de valeur par défaut

Configuration	Entrée XML	Valeurs autorisées	Valeur par défaut
Options de configuration supplémentaires pour les sessions de connaissance	configurations	NamedObjectModel valide	Pas de valeur par défaut

Comment procurer des valeurs aux éléments de configuration basés collections ?

Dans le tableau d'éléments de configuration valide vu plus haut, vous aurez sans doute remarqué que les valeurs valides pour les éléments basés collection sont soit **ObjectModel**, soit **NamedObjectModel**. Les deux sont similaires et fournissent une définition de l'objet à construire ou créer pendant le runtime, sauf que les informations sur l'objet **NamedObjectModel** nomment l'objet à observer. Ces deux types sont définis à l'aide d'un identificateur, de paramètres facultatifs et d'un résolveur (qui résout l'objet).

- identifiant - définit toutes les informations sur l'objet, comme le nom de classe complet, l'id du bean Spring ou une expression MVEL.
- paramètres - paramètres optionnels qui doivent être utilisés quand on crée des instances d'objets à partir de ce modèle.
- resolver - identificateur du programme de résolution qui sera utilisé pour créer des instances d'objet à partir du modèle - (reflection, mvel ou Spring).

Ainsi, si vous avez construit une stratégie de marshaling personnalisée et que vous souhaitez que vos déploiements utilisent cette stratégie plutôt que de la valeur par défaut, vous devrez fournir cette stratégie comme **ObjectModel**, avec pour identificateur `com.mycompany.MyStrategy`, le programme de résolution étant reflection (le plus simple et la valeur par défaut) et tous les paramètres requis pour que votre stratégie fonctionne. Reflection servira ensuite pour créer une instance de cette stratégie en utilisant le nom de classe qualifié complet que vous avez fourni comme identificateur.

```
<marshalling-strategy>
  <resolver>reflection</resolver>
  <identifiant>com.myCompany.MyStrategy</identifiant>
  <parameters>
    <parameter xsi:type="xs:string"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      param
    </parameter>
  </parameters>
</marshalling-strategy>
```

Dans le cas où un programme de résolution basé reflection ne suffise pas (comme c'est le cas dans l'exemple précédent), vous pourrez utiliser un programme de résolution basé MVEL comme identificateur du modèle d'objet. Quand vous évaluez les expressions, vous pouvez substituer les paramètres out-of-the-box. Exemple :

```
<marshalling-strategy>
  <resolver>mvel</resolver>
  <identifiant>new com.myCompany.CustomStrategy(runtimeManager)
</identifiant>
</marshalling-strategy>
```


Le programme de résolution Spring vous permet de chercher un bean par son identificateur à partir d'un contexte d'application Spring. Chaque fois que JBoss BPM Suite est utilisée avec Spring, ce programme de résolution aide à déployer les kjar dans l'environnement de runtime. A titre d'exemple (à noter que l'identificateur est, dans ce cas, un bean nommé en contexte Spring) :

```
<marshalling-strategy>
  <resolver>spring</resolver>
  <identifiant>customStrategy</identifiant>
</marshalling-strategy>
```

2.4.2. Gestion des descripteurs de déploiement

Les descripteurs de déploiement peuvent être modifiés via Business Central d'une des deux façons suivantes. Soit graphiquement (en cliquant sur **Création** → **Création de projet**, puis en sélectionnant **Outils** → **Descripteur de déploiement**) ou soit en cliquant sur le menu **Création** → **Administration**, puis en cliquant sur le dossier **META-INF** de File Explorer. Cliquer sur le fichier **kie-deployment-descriptor.xml** pour pouvoir le modifier manuellement.

À chaque fois qu'un projet est créé, un fichier **kie-deployment-descriptor.xml** de stockage est généré avec les valeurs par défaut décrites plus haut.

Remplacer le comportement de mode de fusionnement par hiérarchie

Pour changer le mode par défaut de **MERGE_COLLECTIONS** à **KEEP_ALL**, **OVERRIDE_ALL** ou **OVERRIDE_EMPTY**, vous pouvez utiliser une des méthodes suivantes, selon les besoins.

- Définir la propriété système **org.kie.dd.mergemode** à l'une des valeurs suivantes. Ce mode de fusion deviendra le mode par défaut de tous les kjar déployés dans le système, à moins que vous ne les remplaciez au niveau kjar par la prochaine méthode.
- Quand vous déployez une nouvelle unité de déploiement via Business Central (**Déployer** → **Déploiements**), vous pouvez sélectionner le mode de fusion à utiliser pour ce kjar particulier.
- Quand vous déployez via l'API REST, vous pouvez ajouter le paramètre de recherche **mergemode** à l'un de ces modes pour définir le mode de fusion pour ce déploiement.

Limiter l'accès à Runtime Engine

Un des éléments de configuration discutés plus tôt, **required-roles**, peut être modifié par les descripteurs de déploiement. Cette propriété limite l'accès au moteur d'exécution sur la base d'un kjar ou par niveau de serveur, en veillant à ce que l'accès à certains processus ne soit accordé qu'aux utilisateurs qui appartiennent à des groupes définis par cette propriété.

Le rôle de sécurité peut être utilisé pour limiter l'accès aux définitions de processus ou pour restreindre l'accès pendant le runtime.

Le comportement par défaut consiste à ajouter les rôles requis à cette propriété sur la base des restrictions du référentiel. Vous pouvez, bien sûr, modifier ces propriétés manuellement si nécessaire, comme décrit ci-dessus, en fournissant des rôles qui correspondent aux rôles définis dans le domaine de sécurité.

2.5. GESTION DE LA POLITIQUE DE REMPLACEMENT DE DÉPLOIEMENT

Si un utilisateur essaie de déployer un artefact par un GAV (Group-Id, Artifact-Id et Version) qui existe déjà dans le système, le déploiement échouera et un message d'erreur s'affichera dans le panneau **Messages**.

Cette fonctionnalité empêche l'utilisateur de remplacer un déploiement existant par erreur.

Cette fonctionnalité est activée par défaut, c'est à dire que, par défaut, le système empêchera l'utilisateur de remplacer une installation existante par le même GAV.

Toutefois, il peut exister des cas quand l'utilisateur *souhaite* remplacer des déploiements existants ayant le même GAV. Bien que vous ne puissiez pas activer les remplacements par déploiement, vous pouvez installer cette fonctionnalité pour le système dans son ensemble en utilisant la configuration `org.kie.override.deploy.enabled`. Ce paramètre est `false` par défaut. Modifiez-la à `true` pour activer le remplacement des déploiements ayant le même GAV, que vous procurez au moment du démarrage de votre serveur (`-Dorg.kie.override.deploy.enabled=true`).

2.6. ÉTENDRE BUSINESS CENTRAL

Depuis la version 6.1 de JBoss BPM Suite, Business Central peut être configuré pour ajouter de nouveaux écrans, des menus, des éditeurs, des splashscreens et des perspectives par l'administrateur. Ces éléments peuvent étendre les fonctionnalités de Business Central et sont accessibles via le menu **Extensions** et sont classifiés **plugins de gestion**.

Vous pouvez maintenant définir vos propres plugins basés Javascript ou HTML pour étendre Business Central et les ajouter sans vous soucier de copier les fichiers dans le système de fichiers sous-jacent. Ajouter un nouvel écran au système pour afficher les options de base de cette fonctionnalité.

2.6.1. Gestion des plugins

Vous pouvez accéder à l'écran **Gestion Plugin** en cliquant sur **Extensions** → **Gestion Plugin**. Vous afficherez ainsi l'écran **Plugin Explorer** qui donne la liste de tous les plugins : **Plugin Perspective**, **Plugin Écran**, **Editor Plugin**, **Plugin SplashScreen** et **Menu dynamique**. Ouvrez-en une et vous verrez tous les plugins existants pour chaque catégorie, y compris ceux qui ne sont pas modifiables et qui sont générés par le système.

Nous allons créer un nouveau plugin qui fait écho à « Hello World » lorsque les utilisateurs visitent l'écran de ce plugin. En général, les étapes pour créer un nouveau plugin sont les suivantes :

- Créer un nouvel écran
- Créer une nouvelle perspective (et y ajouter le nouvel écran)
- Créer un nouveau menu (et y ajouter le nouvel écran)
- Apps (en option)

Ajouter un nouvel écran

Cliquer sur le bouton  et sélectionner **Nouvel écran**. On vous invitera à saisir le nom de ce nouvel écran. Saisir "HelloWorldJS" et appuyer sur le bouton **OK**. L'éditeur du plugin d'écran apparaîtra divisé en 4 sections : Modèle, CSS, JavaScript et Media.



NOTE

Tous les éléments créés manuellement entrent dans des catégories respectives, au cas où vous souhaiteriez les modifier plus tard. Dans ce cas, pour ouvrir l'éditeur de plugin d'écran à nouveau si vous le fermez, ouvrez la catégorie **Plugin Écran** et passez les écrans générés par le système pour aller vers le plugin créé manuellement, et cliquer dessus pour ouvrir l'éditeur de plugins d'écran à nouveau.

Le modèle est l'endroit où votre HTML ira, CSS est pour le style, JavaScript pour vos fonctions et Media pour télécharger et charger des images.

Comme nous faisons un simple plugin de Hello World, saisir le code suivant dans la section Modèle : `<div>Mon ecran Helloworld</div>`. Il peut s'agir de n'importe quel code HTML, et vous pouvez utiliser les exemples **Angular** et **Knockout** fournis. Pour les besoins de cet exemple, nous n'utilisons pas un de ces exemples, mais vous pouvez en choisir un en le sélectionnant dans le menu déroulant au bas dans la section Modèle.

Saisir votre code JavaScript dans la section JavaScript. Certaines méthodes et propriétés sont définies pour vous, y compris `main`, `on_close` et `on_open`. Pour cette démo, sélectionnez `on_open` et saisir ce qui suit : `function () { alert('Hello World'); }`

Cliquer sur le bouton **Enregistrer** pour terminer la création de l'écran. Après avoir sauvegardé l'écran, réactualisez Business Central pour le que plugin Screen (écran) soit listé dans le composant écran du plugin Perspective.

Ajouter une nouvelle Perspective

Une fois que l'écran est créé, vous devez créer une perspective sur laquelle cet écran va se trouver. Les perspectives peuvent être créées de la même façon que celle dont un écran est créé en cliquant sur le bouton **Nouvelle**, puis, en sélectionnant **Nouvelle Perspective**. Vous pourrez donner un nom à cette perspective, comme **HelloWorldPerspective**. Cela ouvrira un éditeur de plugin Perspective, un peu comme dans l'éditeur Screen Plugin.

L'éditeur de perspective est comme un générateur de grille que l'on peut glisser / déplacer pour les écrans et composants HTML. Retirer les grilles existantes, puis faites glisser une grille 6 6 du côté droit vers le côté gauche.

Ensuite, ouvrir la catégorie **Composants** et faites glisser un élément d'écran du côté droit vers la gauche (dans une grille). Ceci ouvrira la boîte de dialogue **Modifier Composant** où vous pouvez entrer votre écran créé à l'étape précédente (**HelloWorldJS**). Cliquer sur le bouton **OK**, puis sur le bouton **Enregistrer** pour sauvegarder la perspective. Pour tagger votre perspective, saisir **Accueil** dans le champ de nom de balise et cliquer sur le bouton **Balises**. Enfin, cliquer sur le bouton **OK** et sauvegardez les changements.

Vous pouvez ouvrir cette perspective à nouveau à partir des plugins Perspective figurant sur la liste de gauche.

Ajout d'un nouveau menu

La dernière étape dans la création de notre plugin est d'ajouter un menu dynamique d'où le nouvel écran/ou la nouvelle perspective peuvent être appelés. Pour ce faire, aller dans **Extensions** → **Gestion Plugin** puis cliquer sur le bouton **Nouveau Menu Dynamique**. Nommer ce menu dynamique (**HelloWorldMenu**), puis cliquer sur le bouton **OK**. L'éditeur de menu dynamique s'ouvrira.

Saisir le nom de perspective (**HelloWorldPerspective**) comme **Id d'activité**, ainsi que le nom de la liste déroulante (**HelloWorldMenuDropDown**). Cliquer sur **OK**, puis cliquer sur le bouton **Enregistrer**.

Ce nouveau menu sera ajouté à votre plan de travail la prochaine fois que vous réactualiserez Business Central. Réactualiser maintenant pour voir **HelloWorldMenu** ajouté dans votre menu de niveau supérieur. Cliquer dessus pour que **HelloWorldMenuDropDown** apparaisse, et si vous cliquez dessus, votre écran ou perspective s'ouvrira accompagné de **Hello World**.

Vous venez de créer votre premier plugin !

Travailler avec des applications (optionnel)

Si vous créez plusieurs plugins, vous pourrez utiliser la fonctionnalité de répertoire d'applications pour organiser vos composants et plugins, au lieu de dépendre uniquement des entrées qui se trouvent dans le menu supérieur.

Quand vous sauvegardez une nouvelle perspective, vous pouvez ajouter des étiquettes (balises) et ces étiquettes (balises) sont utilisées pour associer une perspective à un répertoire d'applications. Vous pouvez ouvrir les répertoires d'applications en cliquant sur **Extensions** → **Apps**.

Le répertoire d'applications fournit un autre moyen d'ouvrir votre perspective. Lorsque vous avez créé votre **HelloWorldPerspective**, vous avez entré la balise **Accueil** de page d'accueil. L'annuaire d'applications par défaut contient un répertoire unique appelé **Accueil** auquel vous avez associé votre perspective. C'est là où vous le trouverez quand vous ouvrirez le répertoire d'applications. Vous pouvez cliquer dessus pour exécuter la perspective maintenant.

Vous pouvez créer plusieurs répertoires et associer des perspectives à ces répertoires en fonction des besoins fonctionnels et verticaux. Par exemple, vous pourriez créer un répertoire de ressources humaines et ensuite associer tous les perspectives HR à ce répertoire pour mieux gérer les applications.

Vous pouvez créer un nouveau répertoire en cliquant sur bouton .

2.6.2. API JavaScript (JS) API pour Extensions

L'extensibilité de Business Central est réalisée grâce à une API de JavaScript (JS) sous-jacente qui sera automatiquement chargée s'il est présent dans le dossier **plugins** de l'application web de Business Central (normalement : {INSTALL_DIR}/business-central.war/plugins/) ou il peut être chargé par des appels réguliers de JavaScript.

Cet API est divisée en ensembles multiples suivant sa fonctionnalité.

- **API Register Perspective** : permet la création dynamique de perspectives. L'exemple ci-dessous crée un panneau par la méthode **registerPerspective** :

```
$registerPerspective({
  id: "Home",
  is_default: true,
  panel_type:
"org.uberfire.client.workbench.panels.impl.MultiListWorkbenchPanelPr
esenter",
  view: {
    parts: [
      {
        place: "welcome",
        min_height: 100,
        parameters: {}
      }
    ],
    panels: [
      {
        width: 250,
        min_width: 200,
        position: "west",
        panel_type:
"org.uberfire.client.workbench.panels.impl.MultiListWorkbenchPanelPr
```

```

esenter",
        parts: [
            {
                place: "YouTubeVideos",
                parameters: {}
            }
        ],
    },
    {
        position: "east",
        panel_type:
"org.uberfire.client.workbench.panels.impl.MultiListWorkbenchPanelPr
esenter",
        parts: [
            {
                place: "TodoListScreen",
                parameters: {}
            }
        ],
    },
    {
        height: 400,
        position: "south",
        panel_type:
"org.uberfire.client.workbench.panels.impl.MultiTabWorkbenchPanelPre
senter",
        parts: [
            {
                place: "YouTubeScreen",
                parameters: {}
            }
        ],
    }
]
});

```

- **API Editor** : vous permet de créer des éditeurs de façon dynamique et de les associer à un type de fichier. L'exemple ci-dessous crée un exemple d'éditeur et l'associe au type **nom de fichier**.

```

$registerEditor({
    "id": "sample editor",
    "type": "editor",
    "templateUrl": "editor.html",
    "resourceType":
"org.uberfire.client.workbench.type.AnyResourceType",
    "on_concurrent_update":function(){
        alert('on_concurrent_update callback')
    }

    $vfs_readAllString(document.getElementById('filename').innerHTML,
    function(a) {
        document.getElementById('editor').value= a;
    });
},

```

```

    "on_startup": function (uri) {
        $vfs_readAllString(uri, function(a) {
            alert('sample on_startup callback')
        });
    },
    "on_open":function(uri){
        $vfs_readAllString(uri, function(a) {
            document.getElementById('editor').value=a;
        });
        document.getElementById('filename').innerHTML = uri;
    }
});

```

En plus des méthodes **on_startup** et **on_open** vues dans les exemples précédents, l'API expose les événements de callback (rappels) suivants pour le cycle de vie de l'éditeur :

- on_concurrent_update;
- on_concurrent_delete;
- on_concurrent_rename;
- on_concurrent_copy;
- on_rename;
- on_delete;
- on_copy;
- on_update;
- on_open;
- on_close;
- on_focus;
- on_lost_focus;
- on_may_close;
- on_startup;
- on_shutdown;

Vous pouvez afficher cet éditeur via un modèle html :

```

<div id="sampleEditor">
    <p>Sample JS editor (generated by editor-sample.js)</p>
    <textarea id="editor"></textarea>

    <p>Current file:</p><span id="filename"></span>
    <button id="save" type="button"
onclick="$vfs_write(document.getElementById('filename').innerHTML,
document.getElementById('editor').value,  function(a)
{}));">Save</button>

```

```

<br>

<p>This button change the file content, and uberfire send a
callback to the editor:</p>
<button id="reset" type="button"
onclick="$vfs_write(document.getElementById('filename').innerHTML,
'Something else', function(a) {});">Reset File</button>
</div>

```

- **API PlaceManager** : les méthodes de cet API vous permettent de demander à Business Central d'afficher un composant particulier associé à une cible :
\$goToPlace("componentIdentifier");
- **API Register plugin** : les méthodes de cet API vous permettent de créer des plugins dynamiques de façon dynamique (qui seront transformées dans les écrans de Business Central) via l'API JS.

```

$registerPlugin( {
  id: "my_angular_js",
  type: "angularjs",
  templateUrl: "angular.sample.html",
  title: function () {
    return "angular " + Math.floor(Math.random() * 10);
  },
  on_close: function () {
    alert("this is a pure JS alert!");
  }
});

```

Le plugin se réfère au modèle **angular.sample.html** :

```

<div ng-controller="TodoCtrl">
  <span>{{remaining()}} of {{todos.length}} remaining</span>
  [ <a href="" ng-click="archive()">archive</a> ]
  <ul class="unstyled">
    <li ng-repeat="todo in todos">
      <input type="checkbox" ng-model="todo.done">
      <span class="done-{{todo.done}}">{{todo.text}}</span>
    </li>
  </ul>
  <form ng-submit="addTodo()">
    <input type="text" ng-model="todoText" size="30"
placeholder="add new todo here">
    <input class="btn-primary" type="submit" value="add">
  </form>
  <form ng-submit="goto()">
    <input type="text" ng-model="placeText" size="30"
placeholder="place to go">
    <input class="btn-primary" type="submit" value="goTo">
  </form>
</div>

```

Un plugin peut être ajouté à un événement de Business Central par une série de rappels (callback) JavaScript :

- `on_concurrent_update;`
 - `on_concurrent_delete;`
 - `on_concurrent_rename;`
 - `on_concurrent_copy;`
 - `on_rename;`
 - `on_delete;`
 - `on_copy;`
 - `on_update;`
 - `on_open;`
 - `on_close;`
 - `on_focus;`
 - `on_lost_focus;`
 - `on_may_close;`
 - `on_startup;`
 - `on_shutdown;`
- **API Register Splash Screen** : utilise les méthodes de cette API pour créer des écrans de démarrage.

```
$registerSplashScreen({
  id: "home.splash",
  templateUrl: "home.splash.html",
  body_height: 325,
  title: function () {
    return "Cool Home Splash " + Math.floor(Math.random() * 10);
  },
  display_next_time: true,
  interception_points: ["Home"]
});
```

- **API Virtual File System (VFS) API** : avec cette API, vous pouvez lire ou écrire un fichier sauvegardé dans un système de fichiers par un appel asynchrone.

```
$vfs_readAllString(uri, function(a) {
  //callback logic
});

$vfs_write(uri,content, function(a) {
  //callback logic
})
```

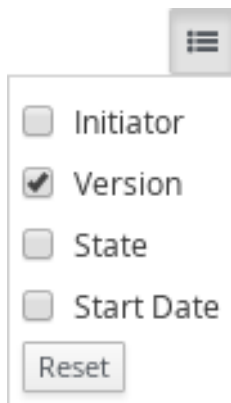

2.7. CONFIGURER DES COLONNES SUR UN TABLEAU

Business Central permet de configurer des vues qui contiennent des listes d'éléments sous forme de tableaux. Vous pouvez redimensionner les colonnes, déplacer des colonnes, ajouter ou supprimer la liste par défaut de colonnes et les trier. Cette fonctionnalité est fournie pour toutes les vues qui contiennent des tableaux.

Une fois que vous aurez fait des changements aux colonnes d'un affichage de colonnes, ces changements seront persistés pour l'utilisateur connecté.

Ajout et suppression de colonnes

Les tableaux qui permettent que les colonnes soient configurées ont le bouton  dans le coin à droite. Si vous cliquez sur ce bouton, vous ouvrirez une liste de colonnes qui peuvent être ajoutées ou supprimées dans le tableau par une case se situant à côté de chaque colonne :



Redimensionner les colonnes

Pour redimensionner les colonnes, placez votre curseur entre les bordures de l'en-tête de la colonne et déplacer-la dans la direction souhaitée :

Id	Name
1	HelloWorld

Déplacer des colonnes

Pour repositionner, glissez-déplacez une colonne dans une position différente, pointer votre souris sur l'endroit le plus à droite de l'en-tête de la colonne :

Description	Version
HelloWorld	1.0

Vous pouvez maintenant saisir la colonne et la déplacer :

Description		Version	
HelloWorld		1.0	

Déposer-le sur l'en-tête de colonne que vous souhaitez déplacer.

Ordonner les colonnes

Pour ordonnancer les colonnes, cliquer sur l'en-tête de colonne que vous souhaitez. Pour revenir en arrière sur un ordonnancement, cliquer à nouveau.

CHAPITRE 3. HOOKS DE RÉFÉRENTIEL

Dans Business Central, il est possible de déclencher une action donnée pour chaque événement ayant lieu. Dans ce but, vous pouvez configurer le référentiel pour qu'il utilise des scripts intitulés hooks.

3.1. CONFIGURER LES GIT HOOKS.

Business Central peut pousser automatiquement les changements vers un autre référentiel en utilisant les Git Hooks. Il existe maintenant un support pour les Git Hooks dans Red Hat JBoss BPM Suite 6.2.0.



NOTE

Veillez noter que seul le crocher **post-commit** est pris en charge actuellement. Les hooks **Post-commit** ne sont déclenchés qu'une fois que le processus de validation totale est terminé.

La procédure suivante montre comment configurer le hook **post-commit** pour pousser automatiquement vos changements dans un référentiel distant.

1. Dans Business Central, aller dans **Création** → **Administration**.
2. Sous le menu principal, cliquez sur **Référentiels** → **Cloner référentiel**
3. Dans la boîte de dialogue **Cloner Référentiel** affichée, définir les informations du référentiel :
 - Nom du référentiel
 - Unité d'organisation
 - Git URL: par exemple `https://github.com/USERNAME/REPOSITORY-NAME.git`



IMPORTANT

Il est important d'utiliser le protocole HTTPS ou Git et non pas un URL SSH style SCP. Business Central ne prend pas en charge l'URL SSH et échoue avec un *Format URL non valide*.

Clone Repository

×

Repository Information

Repository Name *

NewRepository

Organizational Unit *

example

▼

Git URL *

git@github.com:user/ExampleRepository/repo.git

Invalid URL format

User Name

user

Password

.....

Cancel

Clone

Figure 3.1. Un URL SSH style SCP non valide

- Nom d'utilisateur : votre nom d'utilisateur Git
- Mot de passe : votre mot de passe Git

4. Rendez-vous dans le référentiel créé :

```
~]$ cd $JBOSS_HOME/bin/.niogit/REPOSITORY-NAME.git
```

5. Changer l'URL distant :

```
~]$ git remote set-url origin git@github.com:USERNAME/REPOSITORY-NAME.git
```

Vérifiez que vous pouvez accéder au référentiel distant par ligne de commande en utilisant SSH. Par exemple, la clé SSH privée du référentiel doit exister dans le répertoire `~/ .ssh/`.

6. Vérifiez que le référentiel distant ait été ajouté :

```
~]$ git remote -v
```

La commande doit énumérer ce qui suit :

```
origin  git@github.com:USERNAME/REPOSITORY-NAME.git (fetch)
origin  git@github.com:USERNAME/REPOSITORY-NAME.git (push)
```

7. Créer un fichier intitulé **post-commit** ayant comme permissions **rwxr--r--** sous **`$JBOSS_HOME/bin/.niogit/REPOSITORY-NAME.git/hooks`** avec le contenu suivant :

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

8. Veillez à ce que la configuration soit réussie en créant une nouvelle règle guidée dans Business Central : allez dans **Création** → **Création de projet**, puis **Nouvel élément** → **Règle** ci-dessous. Remplir les informations demandées dans la boîte de dialogue **Créer nouvelle règle** et cliquer sur **Ok**.

Tous les changements doivent être poussés automatiquement.

Pour obtenir plus d'informations sur les référentiels Git, voir l'article : [Comment configurer un serveur BxMS 6 pour qu'il puisse utiliser un référentiel Git pour stocker les ressources ?](#)

De plus, il est également possible de spécifier la propriété système `org.uberfire.nio.git.hooks`. Sa valeur détermine un répertoire avec des fichiers hook par défaut, qui seront copiés dans les référentiels Git nouvellement créés. Voir le fichier exemple `standalone.xml` avec la configuration ci-dessous :

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

CHAPITRE 4. CONFIGURATION DE LA LIGNE DE COMMANDE

L'outil `kie-config-cli` est un outil de ligne de commande qui fournit des fonctions de gestion de répertoires de systèmes par ligne de commande et qui peut être utilisé en ligne ou hors ligne.

1. **Online mode** (par défaut et conseillé) - au démarrage, l'outil se connecte à un répertoire Git par le serveur Git fourni par `kie-wb`. Toutes les modifications ont lieu localement et sont publiées en amont une fois que la commande `push-changes` a été lancée. Utilisez la commande `exit` pour publier les modifications locales. Pour ignorer les modifications locales à la sortie, utiliser la commande `discard`.
2. **Offline mode** (un style d'installateur) - crée et manipule le référentiel du système directement sur le serveur - il n'y a aucune option de rejet (`discard`).

L'outil se trouve dans le [Portail Client de Red Hat](#). Pour télécharger `kie-config-cli`, procéder ainsi :

1. Rendez-vous dans le [Portail Client de Red Hat](#) et connectez-vous.
2. Cliquez sur **TÉLÉCHARGEMENTS** en haut de la page.
3. À partir de la page **Téléchargements de produits** qui s'ouvre, cliquer sur **Red Hat JBoss BPM Suite**.
4. Dans le menu déroulant **Version**, sélectionner 6.3.0.
5. Dans le tableau qui s'affiche, naviguer sur la ligne **Outils supplémentaires** et cliquer sur **Téléchargement**.

Extraire le package zip pour obtenir des outils supplémentaires que vous pouvez télécharger à partir du [Portail Client de Red Hat](#). Il contient le répertoire `kie-config-cli-6.MINOR_VERSION-redhat-x-dist` avec le fichier `kie-config-cli.sh`.

4.1. DÉMARRER L'OUTIL KIE-CONFIG-CLI EN MODE EN LIGNE

1. Pour démarrer l'outil `kie-config-cli` en ligne, naviguer dans le répertoire `kie-config-cli-6.MINOR_VERSION-redhat-x-dist` où vous avez installé l'outil et exécutez ensuite la commande suivante.
2. Dans un environnement Unix, exécuter :

```
./kie-config-cli.sh
```

Dans un environnement Windows, exécuter :

```
./kie-config-cli.bat
```

Par défaut, l'outil démarre en mode en ligne et demande les informations d'identification de l'utilisateur et une URL de Git pour se connecter (la valeur par défaut est `git://localhost/system`). Pour vous connecter à un serveur distant, remplacer l'hôte et le port par les valeurs appropriées. Exemple : `git://kie-wb-host:9148/system`

4.2. DÉMARRER L'OUTIL KIE-CONFIG-CLI EN MODE HORS LIGNE

Pour opérer en mode hors ligne, ajouter le paramètre hors ligne à la commande ci-dessous.

1. Naviguer dans le répertoire `kie-config-cli-6.MINOR_VERSION-redhat-x-dist` où vous avez installé l'outil.
2. Dans un environnement Unix, exécuter :

```
./kie-config-cli.sh offline
```

Dans un environnement Windows, exécuter :

```
./kie-config-cli.bat offline
```

L'exécution de cette commande change le comportement de l'outil et affiche une requête pour spécifier le dossier où se trouve le référentiel du système (`.niogit`). Si `.niogit` n'existe pas encore, la valeur du dossier peut être laissée vide et une toute nouvelle configuration sera créée.

4.3. COMMANDES DISPONIBLES POUR L'OUTIL `KIE-CONFIG-CLI`

Les commandes suivantes sont disponibles pour gérer le répertoire GIT par l'outil `kie-config-cli` :

- **add-deployment** - ajoute une nouvelle unité de déploiement
- **add-repo-org-unit** - ajoute un référentiel à l'unité organisationnelle
- **add-role-org-unit** - ajoute un ou des rôle(s) à l'unité organisationnelle
- **add-role-project** - ajoute un ou des rôle(s) à un projet
- **add-role-repo** - ajoute un ou des rôle(s) à un référentiel
- **create-org-unit** - crée une nouvelle unité organisationnelle
- **create-repo** - crée un nouveau référentiel git
- **discard** - ne publie aucun changement local, nettoie les répertoires temporaires et ferme l'outil
- **exit** - publie le travail, nettoie les répertoires temporaires et ferme l'outil
- **fetch-changes** - récupère les changements faits en amont dans le référentiel
- **help** - imprime les commandes disponibles avec leurs descriptions
- **list-deployment** - fait la liste des déploiements disponibles
- **list-org-units** - fait la liste des unités organisationnelles
- **list-repo** - fait la liste des référentiels disponibles
- **push-changes** - pousse les modifications dans le référentiel en amont (en mode en ligne uniquement)
- **remove-deployment** - supprime le déploiement existant

- **remove-org-unit** -supprime l'unité organisationnelle existante
- **remove-repo** - supprime un référentiel existant de la config uniquement
- **remove-repo-org-unit** - supprime un référentiel de l'unité organisationnelle
- **remove--role-org-unit** -supprime le ou les rôle(s) d'une unité organisationnelle
- **add-role-project** - supprime un ou des rôle(s) d'un projet
- **remove-role-repo** - supprime un ou des rôle(s) d'un référentiel

CHAPITRE 5. MIGRATION

La migration de vos projets de Red Hat JBoss BPM Suite 5 à Red Hat JBoss BPM Suite 6 exige une planification minutieuse et une évaluation étape par étape des différentes questions. Vous pouvez planifier des migrations, soit manuellement, soit en utilisant des processus automatiques. La plupart des migrations nécessiteront une combinaison des deux.

Comme JBoss BPM Suite 6 utilise GIT pour le stockage des ressources, des artefacts et des référentiels de code comprenant des procédures et des règles, vous devez commencer par créer un projet vide dans JBoss BPM Suite 6 comme base de votre migration avec des fichiers factices comme espaces réservés pour vos différentes ressources et artefacts. En exécutant un clone GIT de ce projet vide dans votre IDE préféré, vous initiez le processus de migration.

Sur la base des fichiers d'espace réservé dans votre projet cloné, vous pouvez commencer à ajouter des ressources dans les emplacements corrects. Le système de JBoss BPM Suite 6 est assez intelligent pour prendre ces changements en compte et les appliquer correctement. Assurez-vous bien que, lorsque vous importez des vieux fichiers de règles, ils soient importés avec la bonne structure de nommage de package.

Comme Maven est utilisé pour créer des projets, les ressources de projets comme les règles, processus et modèles sont accessibles sous forme de simple fichier jar.

Cette section fait la liste des manières de migrer votre projet étape par étape. Cependant, il ne s'agit là que de lignes directrices, et la migration peut varier sérieusement de ce modèle.

En général, vous devez...

1. Commencer par migrer les données : ce sont vos ressources d'entreprise.
2. Puis, migrer vos processus de runtime.
3. Enfin, convertissez vos anciens appels d'API en nouveaux, l'un après l'autre.

Voyons ces étapes en détail dans les prochaines sections.

5.1. MIGRATION DES DONNÉES

Pour migrer des données de Red Hat JBoss BPM Suite, procédez ainsi :

1. Télécharger l'outil de migration en vous connectant sur le [Portail client de Red Hat](#), puis naviguer vers la section de téléchargements des logiciels Red Hat JBoss BPM Suite. Cliquer sur **Red Hat JBoss BPM Suite Migration Tool** pour télécharger l'archive zip.
2. Décompresser l'archive zip téléchargée dans un répertoire de votre choix, et naviguer vers ce répertoire par l'intermédiaire d'une invite de commande. Ce répertoire contient quatre dossiers :
 - o **bin** - contient les scripts de lancement.
 - o **jcr-exporter-libs** - contient les fichiers libs spécifiques à la partie **export-from-JCR** de la migration.
 - o **jcr-importer-libs** - contient les fichiers libs spécifiques à la partie **import-from-Git** de la migration.
 - o **conf** - contient la configuration de l'outil de migration global.

3. Pour les bases de données de production, copier le pilote JDBC pour la base de données utilisée par le référentiel JCR du répertoire **jcr-exporter-libs** de l'outil de migration.
4. Exécuter la commande suivante :

```
./bin/runMigration.sh -i <source-path> -o <destination-path> -r
<repository-name>
```

Où :

- **<source-path>** est un chemin d'accès vers le référentiel JCR source.
- **<destination-path>** est un chemin d'accès vers une destination GIT VFS. Ce dossier ne doit pas être déjà existant.
- **<repository-name>** un nom arbitraire de nouveau référentiel.

Le référentiel est ensuite migré vers l'emplacement spécifié.

En plus de la commande **-i**, vous pouvez également utiliser **-h** pour imprimer un message d'aide et **-f** qui oblige un écrasement du répertoire de sortie, éliminant ainsi le besoin de suppression manuelle de ce répertoire.

Comment importer un référentiel de Business Central

Le référentiel peut être importé dans Business Central en le clonant. Dans la perspective Administration, cliquer sur le menu **Référentiels**, puis cliquer sur le menu **Cloner Référentiel** pour démarrer le processus.



NOTE

Les ressources peuvent être migrées manuellement. Après tout, ce ne sont que des fichiers de texte. La spécification BPMN2 et la syntaxe DRL n'ont pas changé entre les versions.

Importer le référentiel dans JBDS

Pour importer le référentiel dans JBoss Developer Studio, procéder ainsi

1. Démarrer JBoss Developer Studio
2. Démarrer le serveur Red Hat JBoss BPM Suite (s'il n'est pas déjà en cours d'exécution) en sélectionnant le serveur à partir de l'onglet serveur et en cliquant sur l'icône de démarrage.
3. Sélectionner **Fichier** → **Importer...** et naviguer dans le fichier Git. Ouvrir le dossier Git et sélectionner **Projets de Git** puis cliquer Suite.
4. Sélectionner la source du référentiel en tant que **Sortir du référentiel local** et cliquer sur Suite.
5. Sélectionner le référentiel qui doit être configuré à partir de la liste de référentiels disponibles.
6. Importer le projet en tant que projet général dans la prochaine fenêtre et cliquer sur Suite. Nommer ce projet et cliquer sur Terminer.

5.2. MIGRATION DE RUNTIME

Pour exécuter les processus de Red Hat JBoss BPM Suite 5 dans Red Hat JBoss BPM Suite 6, procéder ainsi :

1. Démarrer la propriété système `jbpm.v5.id.strategy` à `true` dans le fichier JBoss BPM Suite `standalone.xml` :

```
<property name="jbpm.v5.id.strategy" value="true"/>
```

2. Charger la `KieSession` comme suit :

```
KieSession ksession =
    JPAKnowledgeService.loadStatefulKnowledgeSession(sessionID, kbase,
    sessionConf, env);
```

3. Continuer l'exécution normale du processus en utilisant les méthodes :

```
ksession.signalEvent("SomeEvent", null);
```

5.3. API ET COMPATIBILITÉ RÉTROACTIVE

Migration vers Version 6.1

Dans la version 6.1, les API 5.x ne sont plus pris en charge.

Red Hat JBoss BPM Suite n'assure plus la compatibilité rétroactive avec les règles, événements et interfaces de programmation d'applications de processus (API) à partir de JBoss BRMS 5. Le contenu du fichier `knowledge-api JAR` n'est plus supporté dans la version 6.1 et est remplacé par des API contenues dans le fichier `kie-api JAR` qui ont été introduites dans JBoss BPM Suite 6.0.

Si vous avez utilisé une ancienne API 5.x (se trouvant dans `knowledge-api.jar`), veuillez migrer (réécrire) les appels d'API dans la nouvelle API de KIE. Notez bien que plusieurs autres API ont changé entre JBoss BRMS 5.x et JBoss BPM Suite 6.x, à savoir l'API de service de tâches et l'API REST.

Migration vers Version 6.0

Le système de 6 JBoss BRMS assure la compatibilité rétroactive avec les interactions de processus, règles, et événements de JBoss BRMS 5. Vous devez éventuellement migrer (réécrire) ces interactions sur le tout nouvel API de base nouvellement conçu car cette compatibilité rétroactive est susceptible d'être obsolète.

Si vous ne pouvez pas migrer votre code pour utiliser la nouvelle API, vous pouvez utiliser l'API fournie par le jar `connaissances-api` spécialement conçu pour le code rétro-compatible. Cette API est l'interface publique qui sert à travailler dans JBoss BPM Suite et JBoss BRMS et elle est rétro-compatible.

Si vous utilisez à la place l'API REST dans JBoss BPM Suite 5, notez que cela a changé également, et qu'il n'existe aucun mécanisme de rétro-compatibilité dedans.

5.4. MIGRATION DU SERVICE DE TÂCHES

JBoss BPM Suite 6 prend en charge un serveur de tâches en cours d'exécution locale uniquement. Cela signifie que vous n'avez pas besoin de configurer de service de messagerie dans votre projet. Cela diffère de JBoss BPM Suite 5 parce qu'il fournissait un serveur de tâches relié au moteur de base, en utilisant, le plus souvent, du système de messagerie fourni par HornetQ

Pour vous aider à combler l'écart jusqu'à ce que vous migriez ceci dans votre architecture actuelle, il y

a une méthode d'utilitaire ou assistant, **LocalHTWorkItemHandler**.

Comme l'API TaskService fait partie de l'API publique, vous allez maintenant devoir refactoriser vos importations à cause des changements de packages ou bien, vous allez devoir refactoriser vos méthodes à cause des changements au niveau des API elles-mêmes.

CHAPITRE 6. GESTION DES DONNÉES

6.1. SAUVEGARDE DES DONNÉES

Quand vous appliquez un mécanisme de sauvegarde à Red Hat JBoss BPM Suite, assurez-vous bien de sauvegarder les ressources suivantes :

- tout descripteur de déploiement personnalisé (tel que, `web.xml`, `jboss-web.xml`, `jboss.xml`)
- tous les fichiers de propriétés personnalisés



NOTE

Considérez la sauvegarde du fichier `business-central.war` total et des fichiers `dashbuilder.war`.

6.2. INSTALLATION DES INDEXES

Installation des indexes de clés étrangères

Certaines bases de données, comme Oracle et Postgres, ne créent pas automatiquement un index pour chaque clé étrangère. Cela peut entraîner les blocages. Pour éviter cette situation, il faut créer un index sur toutes les clés étrangères, en particulier dans la base de données Oracle.

Installation d'indexes pour le tableau de tâches et de processus

Le tableau de bord de processus et de tâches de 6.1 a été refactorisé pour faire face à un volume élevé d'instances de processus et de tâches. Afin d'obtenir des temps de réponse satisfaisants lors de l'interrogation de la base de données, les suites BPM de JBoss doivent être indexées : `processinstancetype` et `bamtasksummary`.

Notez que *TOUTES* les colonnes de ces deux tableaux ont besoin d'être indexées, pas seulement les clés primaires ou étrangères.

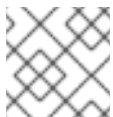
6.3. INSTALLER LA BASE DE DONNÉES

L'application dashbuilder requiert une base de données existante, qui aura été créée avant d'exécuter l'application. Pour créer une base de données, vous pouvez utiliser n'importe quel outil client de base de données et exécuter les commandes suivantes :

Postgres

L'énoncé sql suivant est utilisé pour créer une base de données Postgres :

```
CREATE DATABASE dashbuilder
WITH ENCODING='UTF8'
OWNER=dashbuilder
CONNECTION LIMIT=-1
```



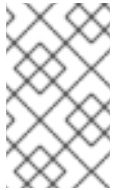
NOTE

La codification de la base de données doit être UTF8

DB2

La base de données DB2 peut être créée en utilisant l'énoncé sql suivant :

```
CREATE DATABASE dashb PAGESIZE 16384
```

**NOTE**

La taille de page des systèmes DB2 par défaut est de 4k, ce qui n'est pas suffisant pour la taille des colonnes du tableau dashbuilder. La taille d'une page doit être forcée à 16384 comme le montre l'énoncé suivant.

Une fois que la base de données est créée, la source de données du serveur d'applications doit être configurée. Vous devez modifier le fichier de configuration de JBoss EAP et configurer le sous-système de la source de données comme dans l'un des exemples suivants :

```
<datasource jndi-name="java:jboss/datasources/jbpmDS" enabled="true" use-
java-context="true" pool-name="postgresqlDS">
  <connection-url>jdbc:postgresql://localhost/test</connection-url>
  <driver>postgresql</driver>
  <pool></pool>
  <security>
    <user-name>sa</user-name>
    <password>sa</password>
  </security>
</datasource>
<drivers>
  <driver name="postgresql" module="org.postgresql">
    <xa-datasource-class>org.postgresql.xa.PGXDataSource</xa-
datasource-class>
  </driver>
</drivers>
```

6.4. MODIFIER LA BASE DE DONNÉES

Dashbuilder exige que JBoss BPM Suite ait des tableaux de bases de données de journalisation de l'historique. Il est impératif de déployer la console de tâches humaines (ou un sur-ensemble, c'est à dire : kie-wb) pour commencer. Sinon, le tableau de bord n'est pas initialisé correctement et il ne sera pas possible d'afficher ses indicateurs de performance clés.

Par défaut, l'application est configurée de façon à utiliser une source de données ayant le nom JNDI suivant :

```
java:jboss/datasources/ExampleDS
```

C'est spécifié dans le fichier de configuration de JBoss EAP; par exemple, `standalone.xml`.

**NOTE**

Cette source de données est à but de développement/démo ; elle est présente par défaut dans les installations JBoss.

Si vous souhaitez déployer dans une base de données différente d'H2 comme Oracle, MySQL, Postgres ou MS SQL Server, veuillez procéder aux étapes suivantes :

Procédure 6.1. Changer de base de données

1. Installer le pilote de base de données sur JBoss (voir la documentation du pilote de JBoss).
2. Créer une base de données vide et une source de données de JBoss qui se connecte au pilote de la base de données.
3. Modifier le fichier `dashbuilder.war/WEB-INF/jboss-web.xml` :

```
<jboss-web>
  <context-root>/dashbuilder</context-root>
  <resource-ref>
    <res-ref-name>jdbc/dashbuilder</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <jndi-name>java:jboss/datasources/myDataSource</jndi-name>
  </resource-ref>
  ...
```

4. Remplacer la valeur du paramètre `jndi-name` par le nom du chemin JNDI de la source de données JBoss que vous venez de créer.
5. Modifier le fichier `dashbuilder.war/WEB-INF/jboss-deployment-structure.xml`
6. Ajouter l'extrait de configuration suivant à l'intérieur de la balise `deployment`, ou `jdbcDriverModuleName` correspond au nom du module de pilote de JBoss JDBC :

```
<dependencies>
  <module name="jdbcDriverModuleName" />
</dependencies>
```

6.5. SCRIPTS DDL

Les scripts DDL de tableaux de bases de données pour JBoss BRMS et BPM Suite sont disponibles par téléchargement via le portail client. Ces scripts vous permettent d'étudier les tableaux et de les utiliser pour créer les tableaux et les index manuellement ou dans les bases de données qui ne sont pas directement prises en charge.

Pour télécharger ces scripts, connectez-vous à [Customer Portal](#) et cliquer sur Red Hat JBoss BPM Suite. Sélectionner la version du produit que vous souhaitez, puis cliquer sur **Téléchargement** sur la ligne **Red Hat JBoss BPM Suite 6.x.x Supplementary Tools** pour télécharger les outils supplémentaires.

Décompresser le fichier dans votre machine. Les scripts DDL se situent dans le dossier `ddl-scripts`. Les scripts de bases de données sont fournis pour DB2, H2, MySQL5, Oracle, PostgreSQL et SQLServer.

Le diagramme Entity Relationship complet se trouve à l'adresse suivante [Red Hat Solution](#).

CHAPITRE 7. RÉFÉRENTIEL DE RESSOURCES

Les règles métier, les fichiers de définitions de processus et les autres ressources créées dans Business Central sont stockées dans le référentiel de ressources (Asset), connu également sous le nom de Référentiel de connaissance ou Knowledge Store.

Le Knowledge Store est un référentiel centralisé de connaissances commerciales. Il est connecté au référentiel GIT, ce qui vous permet de stocker différentes ressources de connaissances et d'artefacts en un seul et même endroit. Business Central vous offre une base web frontale qui vous permet de visualiser et de mettre à jour le contenu stocké. Vous pouvez y accéder via **Project Explorer** à partir de l'environnement unifié de Red Hat BPM Suite.

Toutes les ressources sont stockées dans ces référentiels. Ces référentiels sont alors sauvegardés dans des répertoires nommés unités organisationnelles. Par défaut, le référentiel Artefact ne contient pas d'unités organisationnelles. De ce fait, pour créer vos propres ressources commerciales, vous devez créer une unité organisationnelle et un répertoire pour commencer.

7.1. CRÉATION D'UNE UNITÉ ORGANISATIONNELLE

Il est possible de créer une unité organisationnelle dans la perspective **Administration** de Business Central, en utilisant l'outil `kie-config-cli` ou les invocations d'API REST.

Créer une unité organisationnelle dans Business Central



IMPORTANT

Notez que seuls les utilisateurs ayant un rôle `admin` dans Business Central peuvent créer des unités organisationnelles.

Procédure 7.1. Utiliser Business Central pour créer une unité organisationnelle

1. Dans Business Central, aller dans **Création** → **Administration**.
2. Dans le menu **Perspective**, cliquer sur **Unités organisationnelles** → **Gérer Unités organisationnelles**.
3. Dans la vue **Gestionnaire d'unités organisationnelles**, cliquer sur **Ajouter**.

La boîte de dialogue **Ajouter Nouvelle Unité organisationnelle** va s'ouvrir.

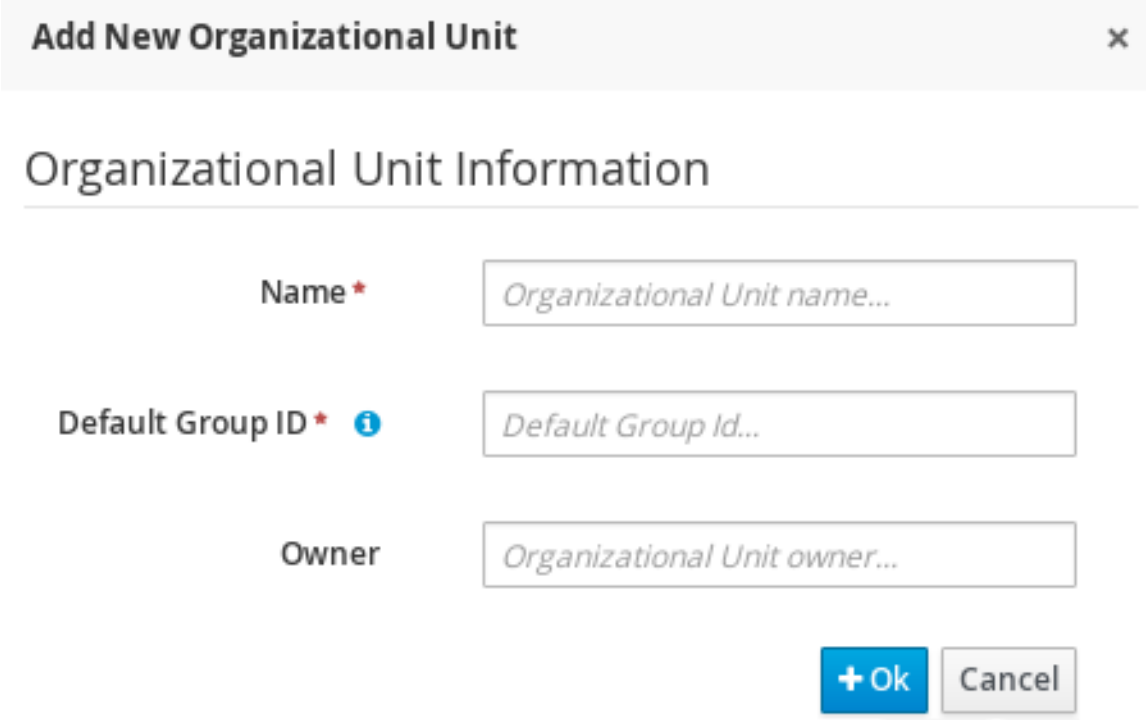


Figure 7.1. Boîte de dialogue Ajouter Nouvelle Unité organisationnelle

4. Saisir les deux paramètres obligatoires (nom et ID Groupe par défaut) et cliquer sur **Ok**.

Créer une unité organisationnelle en utilisant l'outil `kie-config-cli`

Pour créer des unités organisationnelles à l'aide de l'outil `kie-config-cli`, exécutez la commande `create-repo`. L'outil vous guidera le long du processus de création d'unité organisationnelle en vous demandant les autres paramètres requis. Saisir `help` pour obtenir une liste de toutes les commandes.

Pour plus d'informations concernant l'outil `kie-config-cli`, voir [Chapitre 4, Configuration de la ligne de commande](#).

Créer une unité organisationnelle en utilisant l'API REST



IMPORTANT

Notez que seuls les utilisateurs ayant un rôle `rest-all` peuvent créer des unités organisationnelles.

Pour créer une unité organisationnelle dans le Knowledge Store, invoquez un appel API REST **POST**. Les informations sur l'unité organisationnelle sont définies par l'entité JSON.

Le paramètre de sortie de l'appel est une instance `OrganizationalUnit`. L'appel retourne une instance `CreateOrganizationalUnitRequest`.

Exemple 7.1. Créer une unité organisationnelle en utilisant l'utilitaire Curl

Exemple d'entité JSON contenant les détails d'une unité organisationnelle à créer :

```
{
  "name"      : "helloWorldUnit",
  "owner"     : "tester",
```

```
"description" : null,
"repositories" : []
}
```

Exécuter la commande suivante :

```
curl -X POST 'localhost:8080/business-central/rest/organizationalunits/'
-u USERNAME:PASSWORD -H 'Accept: application/json' -H 'Content-Type:
application/json' -d
'{"name":"helloWorldUnit","owner":"tester","description":null,"repositor
ies":[]}'
```

Pour obtenir plus d'informations, voir *Guide de développement de Red Hat JBoss BPM Suite* chapitre *API REST Knowledge Store*, section *Invocations d'unités organisationnelles*.

7.2. CRÉATION D'UN RÉFÉRENTIEL

Il y a trois façons de créer un référentiel : en utilisant la perspective **Administration** de Business Central, l'outil `kie-config-cli` ou les appels d'API REST.

Créer un référentiel dans Business Central



IMPORTANT

Notez que seuls les utilisateurs ayant un rôle **admin** dans Business Central peuvent créer des référentiels.

Procédure 7.2. Utiliser Business Central pour créer un référentiel

1. Dans Business Central, aller dans **Création** → **Administration**.
2. Dans le menu perspective, cliquer sur **Référentiels** → **Nouveau référentiel**.

La fenêtre **Nouveau référentiel** s'ouvrira.

Figure 7.2. Fenêtre de dialogue **Nouveau référentiel**

3. Spécifier les deux paramètres obligatoires :
 - o nom du référentiel

**NOTE**

Veillez à ce que le nom du référentiel soit un nom de fichier valide. Évitez d'utiliser un espace ou un caractère spécial qui puisse mener à un nom non valide.

- unité organisationnelle : indique l'emplacement du référentiel nouvellement créé.

4. Cliquer sur Terminer.

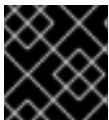
Vous pourrez apercevoir de référentiel nouvellement créé dans **File Explorer** ou **Project Explorer**.

Créer un référentiel en utilisant l'outil `kie-config-cli`

Pour créer un nouveau référentiel Git à l'aide de l'outil `kie-config-cli`, exécutez la commande `create-repo`. L'outil vous guidera le long du processus de création de référentiel en vous demandant les autres paramètres requis. Saisir `help` pour obtenir une liste de toutes les commandes.

Pour plus d'informations concernant l'outil `kie-config-cli`, voir [Chapitre 4, Configuration de la ligne de commande](#).

Créer un référentiel par l'API REST

**IMPORTANT**

Notez que seuls les utilisateurs ayant un rôle `rest-all` peuvent créer des référentiels.

Pour créer un référentiel dans le Knowledge Store, lancer l'appel d'API REST **POST**. Les informations du référentiel sont définies par l'entité JSON. Veillez bien à établir une session HTTP authentifiée avant d'invoquer cet appel.

Le paramètre de sortie de l'appel est une instance `Repository Request`. L'appel retourne une instance `CreateOrCloneRepositoryRequest`.

Exemple 7.2. Créer un référentiel en utilisant l'utilitaire Curl

Exemple d'entité JSON contenant les détails d'un référentiel à créer :

```
{
  "name"                : "newRepository",
  "description"         : null,
  "gitURL"              : null,
  "requestType"         : "new",
  "organizationalUnitName" : "helloWorldUnit"
}
```

Exécuter la commande suivante :

```
curl -X POST 'localhost:8080/business-central/rest/repositories/' -u
USERNAME:PASSWORD -H 'Accept: application/json' -H 'Content-Type:
application/json' -d
'{"name":"newRepository","description":null,"requestType":"new","gitURL"
:null,"organizationalUnitName":"helloWorldUnit"}'
```

Pour obtenir plus d'informations, voir *Guide de développement de Red Hat JBoss BPM Suite* chapitre *API REST Knowledge Store*, section *Appels de référentiels*

7.3. CLONAGE D'UN RÉFÉRENTIEL

Il est possible de cloner un référentiel dans Business Central ou par les appels d'API REST. L'outil `kie-config-cli` ne peut pas être utilisé pour cloner des référentiels de façon arbitraire - exécuter `git clone` ou bien, utiliser une des options suivantes à la place.

Cloner un référentiel dans Business Central



IMPORTANT

Notez que seuls les utilisateurs ayant un rôle `admin` dans Business Central peuvent cloner des référentiels.

Procédure 7.3. Utiliser Business Central pour cloner un référentiel

1. Dans Business Central, aller dans **Création** → **Administration**.
2. À partir du menu en perspective, sélectionner **Référentiels** → **Cloner référentiel**

La fenêtre **Clonage d'un référentiel** s'ouvre.

Clone Repository [X]

Repository Information

Repository Name *

Organizational Unit *

Git URL *

User Name

Password

Figure 7.3. Fenêtre de dialogue Clonage d'un référentiel

3. Dans la boîte de dialogue **Clonage d'un référentiel**, saisir les informations sur le référentiel :
 - a. Saisir le **Nom de référentiel** à utiliser comme identifiant pour le référentiel de ressources Asset et sélectionner l'**Unité organisationnelle** dans laquelle il devra être ajouté.
 - b. Saisir l'URL du référentiel GIT :
 - pour un référentiel local, utiliser `file:///PATH_TO_REPOSITORY/REPOSITORY_NAME;`
 - pour un référentiel distant ou déjà existant, utiliser `https://github.com/USERNAME/REPOSITORY_NAME.git` ou `git://HOST_NAME/REPOSITORY_NAME`.



IMPORTANT

Il est important d'utiliser le protocole HTTPS ou Git et non pas un URL SSH style SCP. Business Central ne prend pas en charge l'URL SSH et échoue avec un *Format URL non valide*.



NOTE

Le protocole de fichier n'est pris en charge que pour les opérations READ. Les opérations WRITE ne sont *pas* prises en charge.

- c. Si nécessaire, saisir les **Nom d'utilisateur** et **Mot de passe** de votre compte Git à utiliser pour l'authentification.
4. Cliquer sur **Cloner**.

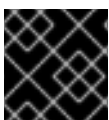
Vous apercevrez une invite contenant une notification pour vous indiquer que le référentiel a été créé.

5. Cliquez sur **Ok**.

Le référentiel est maintenant en cours d'indexation. Certaines fonctionnalités de Workbench risquent de ne pas être disponibles tant que l'indexation n'est pas terminée.

Vous pourrez apercevoir de référentiel cloné dans **File Explorer** ou **Project Explorer**.

Créer un référentiel par l'API REST



IMPORTANT

Notez que seuls les utilisateurs ayant un rôle **rest-all** peuvent cloner des référentiels.

Pour cloner un référentiel, invoquez l'API REST **POST**. Cet appel crée ou clone (selon la valeur du paramètre **requestType**) le référentiel défini par l'entité JSON.

Le paramètre de sortie de l'appel est une instance **Repository Request**. L'appel retourne une instance **CreateOrCloneRepositoryRequest**.

Exemple 7.3. Cloner un référentiel par l'utilitaire Curl

Exemple d'entité JSON contenant les détails d'un référentiel à cloner :

```
{
  "name"           : "clonedRepository",
  "description"    : null,
  "requestType"    : "clone",
  "gitURL"         : "git://localhost:9418/newRepository",
  "organizationalUnitName" : "helloWorldUnit"
}
```

Exécuter la commande suivante :

```
curl -X POST 'localhost:8080/business-central/rest/repositories/' -u
USERNAME:PASSWORD -H 'Accept: application/json' -H 'Content-Type:
application/json' -d
'{"name":"clonedRepository","description":null,"requestType":"clone","gi
tURL":"git://localhost:9418/newRepository","organizationalUnitName":"hel
loWorldUnit"}'
```

Pour obtenir plus d'informations, voir *Guide de développement de Red Hat JBoss BPM Suite* chapitre *API REST Knowledge Store*, section *Appels de référentiels*

7.4. SUPPRIMER UN RÉFÉRENTIEL

On peut supprimer les référentiels par l'une des procédures suivantes :

Comment supprimer un référentiel dans Business Central

La manière la plus simple de supprimer un référentiel est d'utiliser le **RepositoryEditor** de Business Central.

Procédure 7.4. Utiliser Business Central pour supprimer un référentiel

1. Dans Business Central, aller dans **Création** → **Administration**.
2. Sélectionner les **Référentiels** du menu arborescent sur la gauche.
3. Dans le **RepositoryEditor** qui se trouve à droite de la page, chercher le référentiel à supprimer dans la liste des référentiels disponibles.
4. Dans le menu déroulant, sélectionner **master** et cliquer sur **Supprimer**.

Le message suivant apparaîtra :

```
Êtes-vous certain de vouloir supprimer le référentiel
"REPOSITORY_NAME" ? Certains éditeurs de texte sont soudain
inopérables si on ne peut plus accéder à leur contenu.
```

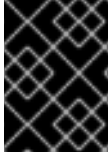
5. Cliquer sur **Ok** pour supprimer le référentiel.

Supprimer un référentiel en utilisant l'outil `kie-config-cli`

Les référentiels peuvent être supprimés grâce à l'outil `kie-config-cli` également. Pour cela, exécutez la commande `remove-repo`.

Pour plus d'informations concernant l'outil `kie-config-cli`, voir [Chapitre 4, Configuration de la ligne de commande](#).

Supprimer un référentiel par l'API REST



IMPORTANT

Notez que seuls les utilisateurs ayant un rôle `rest-all` peuvent supprimer des référentiels.

Pour créer un référentiel à partir du Knowledge Store, lancer l'appel d'API REST **DELETE**. Veillez bien à établir une session HTTP authentifiée avant d'invoquer cet appel.

L'appel retourne une instance `RemoveRepositoryRequest`.

Exemple 7.4. Supprimer un référentiel avec l'utilitaire curl

Exécuter la commande suivante :

```
curl -X DELETE 'localhost:8080/business-
central/rest/repositories/REPOSITORY_NAME' -u USERNAME:PASSWORD -H
'Accept: application/json' -H 'Content-Type: application/json'
```

Pour obtenir plus d'informations, voir *Guide de développement de Red Hat JBoss BPM Suite* chapitre *API REST Knowledge Store*, section *Appels de référentiels*

7.5. GESTION DES RESSOURCES



NOTE

Le contenu de cette section est classifié aperçu technologique de la version Red Hat JBoss BPM Suite. Il est fourni tel quel et aucun support n'est offert.

Pour activer et utiliser les fonctionnalités décrites ici, vous devrez vous connecter à Business Central par un utilisateur à qui on a donné le rôle spécial de `kiemgmt`.

Pour faciliter la gestion des projets, Red Hat JBoss BPM Suite offre maintenant une façon de gérer plusieurs projets sur la base de standards établis. Cela permet de créer des structures de référentiels conformes aux meilleures pratiques de l'industrie pour les questions de maintenance, de contrôle de version et de distribution de vos projets.

Pour commencer, les référentiels peuvent maintenant être gérés ou non.

Référentiels gérés ou non-gérés

Les référentiels non gérés correspondent aux structures de référentiels auxquelles vous êtes habitués. Ils peuvent contenir plusieurs projets sans aucun lien entre eux.

Les référentiels gérés, d'autre part, offrent un contrôle de version au niveau du projet et des branches

de projets pour gérer le cycle de sortie de projet. De plus, les référentiels de projets peuvent se limiter à un seul projet ou réunir plusieurs projets. Quand vous créez un référentiel géré, le processus de configuration de la gestion des ressources est lancé automatiquement afin de créer des branches de référentiel. La structure de projet correspondante est également créée.

Procédure 7.5. Création d'un référentiel non géré

1. Dans Business Central, aller dans **Création** → **Administration**.
2. Cliquer sur **Référentiels** → **Nouveau référentiel**.

La fenêtre **Nouveau référentiel** s'ouvre.

3. Saisir le nom du référentiel et sélectionner une unité organisationnelle à laquelle le référentiel appartient.
4. Cliquer sur **Terminer**.

Procédure 7.6. Création d'un référentiel géré

1. Dans Business Central, aller dans **Création** → **Administration**.
2. Cliquer sur **Référentiels** → **Nouveau référentiel**.

La fenêtre **Nouveau référentiel** s'ouvre.

3. Saisir le nom du référentiel et sélectionner une unité organisationnelle à laquelle le référentiel appartient.
4. Sélectionner la case **Référentiel géré** et cliquer sur **Suivant** pour saisir des informations supplémentaires sur le référentiel géré.

5. Sélectionner soit l'option **Référentiel Projet Unique** ou l'option **Référentiel Multi Projets**.

Si le projet que vous créez est simple et autonome, sélectionner l'option **Référentiel Projet Unique**. Notez que vous ne pourrez pas ajouter de projets à ce référentiel par la suite.

Pour les projets plus complexes, où il y a sûrement un projet parent qui inclut des sous-projets, sélectionner l'option **Référentiel Multi Projets**.

New Repository

Basic Settings

Managed Repository Settings

Repository Type:

☐ Single-project Repository
 ☒ Multi-project Repository

Project Branches:

☒ Automatically Configure Branches (master/dev/release)

Project Settings:

Name *

test

Description

enter project description

Group *

demo

Artifact *

test

Version *

1.0.0-SNAPSHOT

< Previous

Next >

Cancel

✓ Finish

- Saisir les détails du projet géré ainsi que les informations GAV.

Notez que tous les projets créés dans un **Référentiel Multi Projets** seront gérés ensemble, avec leurs numéros de version incrémentés ensemble également. Les détails du projet parent seront hérités par tous les futurs projets que vous créerez dans ce référentiel géré.

- Cliquer sur **Terminer**.

Branches gérées

Les référentiels gérés ont l'avantage de venir avec des branches gérées. Comme dans GIT, vous pouvez choisir de travailler sur des branches différentes de votre projet (par exemple : master, dev ou sorties de version). Ce processus de branching peut être fait pour vous automatiquement, si vous sélectionnez la case quand vous créez un nouveau référentiel géré (à la fois pour les projets simples ou multiples).

Vous pouvez passer d'une branche à l'autre en sélectionnant la branche voulue quand vous travaillez dans Project Explorer.

Structure du référentiel

Si vous ne sélectionnez pas le branching automatique lors de la création d'un référentiel, vous pourrez créer des branches manuellement par la suite. Pour les référentiels gérés, vous pouvez le faire en utilisant le bouton **Configurer**. Ce bouton, ainsi que des boutons **Version** et **Diffuser**, se trouvent sur la page **Structure du référentiel**. Vous pouvez accéder à cette page, en cliquant sur **Référentiel → Structure du référentiel** dans le menu de perspective de Project Explorer.

Cliquer sur le bouton **Configurer** vous permet de créer des branches et de modifier celles qui sont créées automatiquement.

Configure Repository
✕

Repository

test2

Source Branch

master

Dev Branch *

dev

The branch will be called (dev)-1.0.0-SNAPSHOT

Release Branch *

release

The branch will be called (release)-1.0.0-SNAPSHOT

Version *

1.0.0-SNAPSHOT

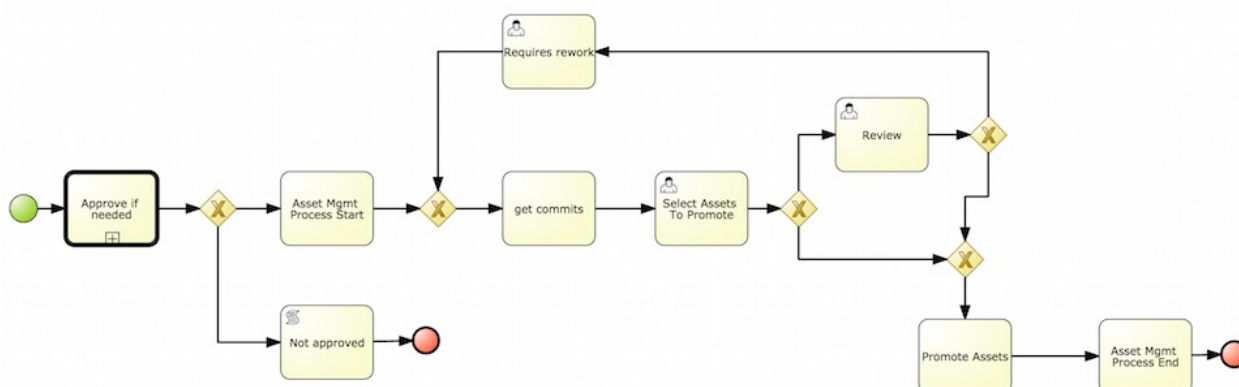
The current repository version is: 1.0.0-SNAPSHOT

+ Ok

Cancel

Vous pouvez promouvoir des actifs de la branche master à d'autres branches en utilisant le bouton **Promouvoir**. De même, vous pouvez libérer les branches et les déployer sur le serveur en utilisant le bouton **Diffuser**.

Ces deux fonctions sont contrôlées en interne par l'utilisation de processus prédéfinis qui sont déployés sur votre instance. Par exemple, lorsque vous cliquez sur le bouton **Promouvoir** après avoir effectué des travaux sur votre branche de développement, un processus de promotion des changements démarre en arrière-plan. Un utilisateur, avec le rôle **kiemgmt** verra une tâche utilisateur apparaître dans cette liste de tâches, lui indiquant de revoir les ressources à promouvoir. Cet utilisateur peut alors réclamer cette tâche, et décider de promouvoir tous, certains, ou aucun des éléments de ressources. Le processus sous-jacent va choisir les commits sélectionnés par l'utilisateur dans une branche de sortie de version. Cet utilisateur peut également demander un nouvel examen de ces ressources, et ce processus peut être répété plusieurs fois jusqu'à ce que toutes les ressources soient prêtes à sortir dans une nouvelle version. Le flux de ce processus est illustré ci-dessous :



De même, lorsque vous cliquez sur le bouton **Diffuser**, un flux de processus de sortie de version est initié. Ce flux de processus génère le projet, met à jour tous les artefacts Maven pour la prochaine version et déploie le projet à l'exécution, si les informations de déploiement sont fournies.



AVERTISSEMENT

Les branches de projets à faire sortir dans les nouvelles versions doivent commencer par le mot clé **release** (version).

Release Configuration

Repository

ManagedRepo2

Source Branch

release-1.0.0

* Release Version

1.0.0

The current repository version is: 1.0.0-SNAPSHOT

* Deploy To Runtime

☒

* User Name

vikrambpms

* Password

●●●●●●●●

* Server URL

http://localhost:8080/business-ce

 Ok

Cancel

7.6. RÉFÉRENTIEL MAVEN

Maven est un outil de gestion de projets de logiciels qui utilise un fichier POM (Project Object Model) pour gérer :

- Les builds

- La documentation
- Les rapports
- Les dépendances
- Les versions
- Les SCM
- La distribution

Un référentiel Maven est utilisé pour contenir et stocker des artefacts de builds et des dépendances de projets. Il est, en général, de deux types :

- Local : se réfère à un référentiel local où toutes les dépendances de projet sont stockées et se trouvent dans l'installation en cours comme « m2 » dans le dossier par défaut. C'est un cache des téléchargements à distance qui contient également les artefacts de builds temporaires non encore sortis.
- Remote (distant) : se réfère à n'importe quel autre type de référentiel qui puisse être accessible par une variété de protocoles tels que file:// ou http://. Ces référentiels peuvent être dans un lieu éloigné, mis en place par un tiers pour le téléchargement des artefacts ou un référentiel interne mis en place sur un fichier ou un serveur HTTP, utilisé pour partager des objets privés entre les équipes de développement pour la gestion des versions internes.

7.7. CONFIGURER UN DÉPLOIEMENT POUR UN RÉFÉRENTIEL NEXUS DISTANT

Nexus est un gestionnaire de référentiels souvent utilisé dans les organisations pour centraliser le stockage et la gestion des artefacts de développement de logiciels. Il est possible de configurer votre projet pour que les artefacts produits par chaque build soient automatiquement déployés dans un référentiel sur un serveur Nexus distant.

Pour configurer votre projet afin qu'il déploie des artefacts dans un référentiel Nexus distant, ajouter un élément `distributionManagement` à votre fichier `pom.xml` de projet, comme l'illustre l'exemple ci-dessous.

```
<distributionManagement>
  <repository>
    <id>deployment</id>
    <name>Internal Releases</name>

    <url>http://your_nexus_host:8081/nexus/content/repositories/releases</url>
  </repository>
  <snapshotRepository>
    <id>deployment</id>
    <name>Internal Releases</name>

    <url>http://your_nexus_host:8081/nexus/content/repositories/snapshots/</url>
  </snapshotRepository>
</distributionManagement>
```

Remplacer les URL de l'exemple par les URL réels de vos référentiels Nexus. Le référentiel spécifié

dans l'élément `snapshotRepository` est utilisé quand le qualificateur `-SNAPSHOT` est ajouté au numéro de version actuel du projet. Dans les autres cas, le référentiel spécifié dans l'élément du `repository` (référentiel) sera utilisé.

Si votre serveur Nexus requiert une authentification, vous devrez également modifier vos paramètres de projet Maven pour ajouter vos informations d'identification dans le fichier `settings-security.xml` à l'aide d'un mot de passe maître. Par défaut, ce fichier est dans le dossier `$M2_HOME/conf`, sauf si vous avez changé son emplacement en modifiant la propriété système `kie.maven.settings.custom`. Il est également possible de créer des nouveaux fichiers `settings.xml` et `settings-security.xml` spécifiques à un utilisateur dans le dossier `~/.m2`. Dans ce cas, ces fichiers remplaceront les fichiers d'origine.

Voir l'exemple suivant du `settings-security.xml`:

```
<servers>
  <server>
    <id>deployment</id>
    <username>admin</username>
    <password>{COQLCE6DU6GtcS5P=}</password>
  </server>
</servers>
```

IMPORTANT

Notez que conserver vos identifiants de serveur (comme les mots de passe) en texte brut dans le fichier `settings.xml` n'est *pas* conseillé. Toutes les informations devraient être hachées avec un mot de passe maître dans le fichier `settings-security.xml`.

Pour obtenir des informations supplémentaires sur l'encodage des mots de passe ou sur la création d'un mot de passe maître, voir la documentation Apache Maven™ article [Encodage de mot de passe](#).

Avec cette configuration en place, en cliquant sur le bouton **Construire et Déployer** de Business Central, vous exécutez une build Maven et déployez les artefacts du build dans le référentiel local et dans l'un des référentiels Nexus indiqués dans le fichier `pom.xml`.

7.8. CONFIGURATION SYSTÈME

Dans JBoss EAP, pour changer une propriété de Business Central, comme la configuration de SSH, procédez ainsi :

Procédure 7.7. Modifier les propriétés système

1. Modifier le fichier `$JBOSS_HOME/domain/configuration/host.xml`
2. Chercher les éléments XML qui appartiennent au `main-server-group` et ajouter la propriété système. Par exemple :

```
<system-properties>
  <property name="org.uberfire.nio.git.dir" value="..." boot-
time="false"/>
```

```
...
</system-properties>
```

Voici une liste de toutes les propriétés système disponibles :

- **org.uberfire.nio.git.dir** : emplacement du répertoire .niogit. Par défaut : le répertoire de travail
- **org.uberfire.nio.git.daemon.enabled** : active/désactive le démon GIT. Par défaut: true
- **org.uberfire.nio.git.daemon.host** : si le démon GIT est activé, utiliser cette propriété comme identifiant d'hôte local. Valeur par défaut : localhost
- **org.uberfire.nio.git.daemon.port** : si le démon GIT est activé, utiliser cette propriété comme numéro de port. Valeur par défaut : 9418
- **org.uberfire.nio.git.ssh.enabled** : active/désactive le démon SSH. Par défaut: true
- **org.uberfire.nio.git.ssh.host** : si le démon SSH est activé, utiliser cette propriété comme identifiant d'hôte local. Valeur par défaut : localhost
- **org.uberfire.nio.git.ssh.port** : si le démon SSH est activé, utiliser cette propriété comme numéro de port. Valeur par défaut : 8001
- **org.uberfire.nio.git.ssh.cert.dir** : emplacement du répertoire .security où les certificats locaux seront stockés. Valeur par défaut : répertoire de travail.
- **org.uberfire.metadata.index.dir** : emplacement du dossier .index de Lucene. Par défaut : le répertoire de travail
- **org.uberfire.cluster.id** : nom du cluster Helix, par exemple : kie-cluster
- **org.uberfire.cluster.zk** : string de connexion à Zookeeper. Se présente sous la forme : host1:port1,host2:port2,host3:port3. Par exemple : localhost:2188.
- **org.uberfire.cluster.local.id** : id unique du noeud de cluster Helix. Notez que ':' est remplacé par '_'. Par exemple : node1_12345.
- **org.uberfire.cluster.vfs.lock** : nom de la ressource définie sur le cluster Helix, par exemple : kie-vfs
- **org.uberfire.cluster.autostart** : cette valeur retarde la mise en cluster VFS jusqu'à ce que l'application soit entièrement initialisée pour éviter les conflits lorsque tous les membres de cluster créent des clones locaux.
- **org.uberfire.sys.repo.monitor.disabled** : désactive le moniteur de configuration (ne pas désactiver à moins de savoir ce que vous faites). Valeur par défaut : false
- **org.uberfire.secure.key** : mot de passe utilisé par le cryptage de mots de passe. Valeur par défaut : org.uberfire.admin
- **org.uberfire.secure.alg** : algorithme Crypto utilisé par le cryptage de mots de passe. Valeur par défaut : PBESWithMD5AndDES

- **org.guvnor.m2repo.dir** : endroit où le dossier de référentiel Maven est stocké. Valeur par défaut : `working-directory/repositories/kie`
- **org.kie.example.repositories** : dossiers à partir desquels les référentiels de démo seront clonés. Les référentiels de démo doivent avoir été obtenus et doivent être placés dans ces dossiers. Cette propriété système prévaut sur les propriétés `org.kie.demo` et `org.kie.example`. Par défaut : `Not used` (non utilisé).
- **org.kie.demo** : active le clone externe d'une application démo de GitHub. Ce système a priorité sur `org.kie.example`. Valeur par défaut : `true`
- **org.kie.example** : active la structure de l'exemple composé par un référentiel, une unité organisationnelle ou un projet. Valeur par défaut : `false`.
- **org.kie.build.disable-project-explorer** : désactive la génération automatique de projets quand on navigue dans Project Explorer. Valeur par défaut : `false`

CHAPITRE 8. EXPORTATION ET IMPORTATION D'UN PROCESSUS

8.1. CRÉER UNE DÉFINITION DE PROCESSUS

Assurez-vous de vous être bien connecté à JBoss BPM Suite ou si vous êtes dans JBoss Developer Studio au référentiel connecté.

Pour créer un processus, procéder ainsi :

1. Ouvrir la perspective Création de projet (**Création** → **Création Projet**).
2. Dans **Project Explorer** (**Création Projet** → **Project Explorer**), naviguer jusqu'au projet dans lequel vous souhaitez créer la définition de processus (dans la vue **Projet**, sélectionner le projet et le référentiel qui conviennent dans les listes déroulantes ; et dans la vue **Référentiel**, naviguer jusqu'au répertoire **REPOSITORY/PROJECT/src/main/resources/**).



NOTE

Il est conseillé de créer vos ressources, y compris les définitions de processus, dans un package de projet pour permettre l'importation de ressources et leur référencement. Pour créer un package, procéder ainsi :

- Dans la vue **Référentiel** de Project Explorer, naviguer dans le répertoire **REPOSITORY/PROJECT/src/main/resources/**.
- Aller dans **Nouvel élément** → **Package**.
- Dans la boîte de dialogue **Nouvelle ressource**, définir le nom du package et vérifier l'emplacement du package dans le référentiel.

3. Dans le menu perspective, cliquer sur **Nouvel élément** → **Business Process**.
4. Dans la boîte de dialogues **Nouveaux Processus**, saisir le nom du processus et cliquer sur le bouton **OK**. Attendez que l'éditeur de processus apparaisse avec le diagramme de processus.

8.2. IMPORTATION D'UNE DÉFINITION DE PROCESSUS

Pour importer une définition BPMN2 ou JSON existante, procéder ainsi :

1. Dans **Project Explorer**, sélectionner un projet et un package respectif dans lequel vous souhaitez importer la définition du processus.
2. Créer un nouveau processus métier en allant dans **Nouvel Élément** → **Business Process**.
3. Dans la barre d'outils de Process Designer, cliquer sur l'icône **importation**  dans la barre d'outils de l'éditeur et choisissez le format de la définition de processus importé. Notez que vous devez choisir de remplacer la définition de processus existante pour importer.
4. À partir de la fenêtre **Import**, cherchez le fichier de processus (Process) et cliquer sur **Import**.

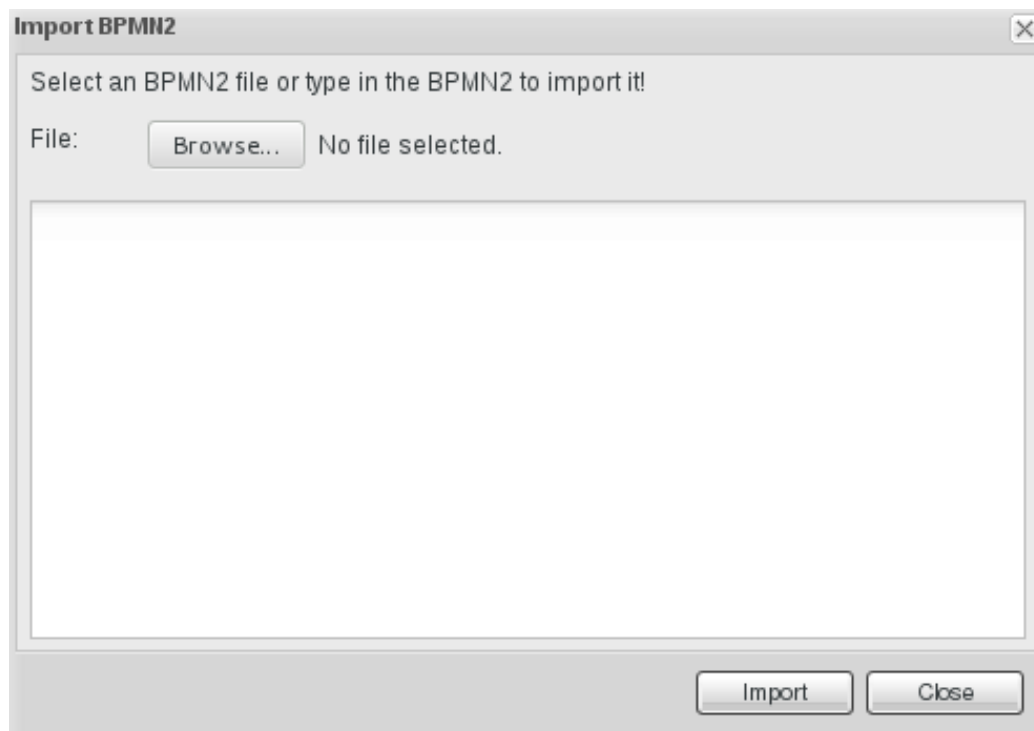


Figure 8.1. F  tre d'importation

Quand une d  finition de processus est import  e, la d  finition import  e existante est remplac  e. Veuillez    ne pas remplacer une d  finition de processus que vous avez modifi  e de fa  on    ne perdre aucun changement.

Un processus peut aussi   tre import   dans le r  f  rentiel git du syst  me de fichiers en clonant le r  f  rentiel, en ajoutant les fichiers de processus et en poussant les changements dans le git. De plus, pour les autres m  thodes d'importation, vous pouvez copier et coller un processus ou simplement ouvrir un fichier dans une bo  te de dialogue d'importation.

Quand on importe des processus, le Process Designer fournit un support visuel aux   l  ments de processus et a donc besoin d'informations sur les positions de l'  l  ment sur le canvas. Si l'information n'est pas fournie dans le processus d'importation, vous devrez l'ajouter manuellement.

8.3. IMPORTER JPDL 3.2 DANS BPMN2

Pour migrer et importer une d  finition jPDL dans BPMN2, dans le Process Designer, cliquer sur le bouton d'importation, puis naviguer vers le bas et s  lectionner **Migrer jPDL 3.2 dans BPMN2**.

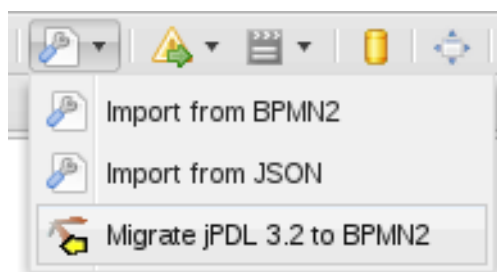


Figure 8.2. Migrer jPDL 3.2 dans BPMN2

Dans la bo  te de dialogue **Migrer dans BPMN2**, s  lectionner le fichier de d  finition de processus et le nom du fichier **gpd**. Confirmer en cliquant sur le bouton **Migrer**.

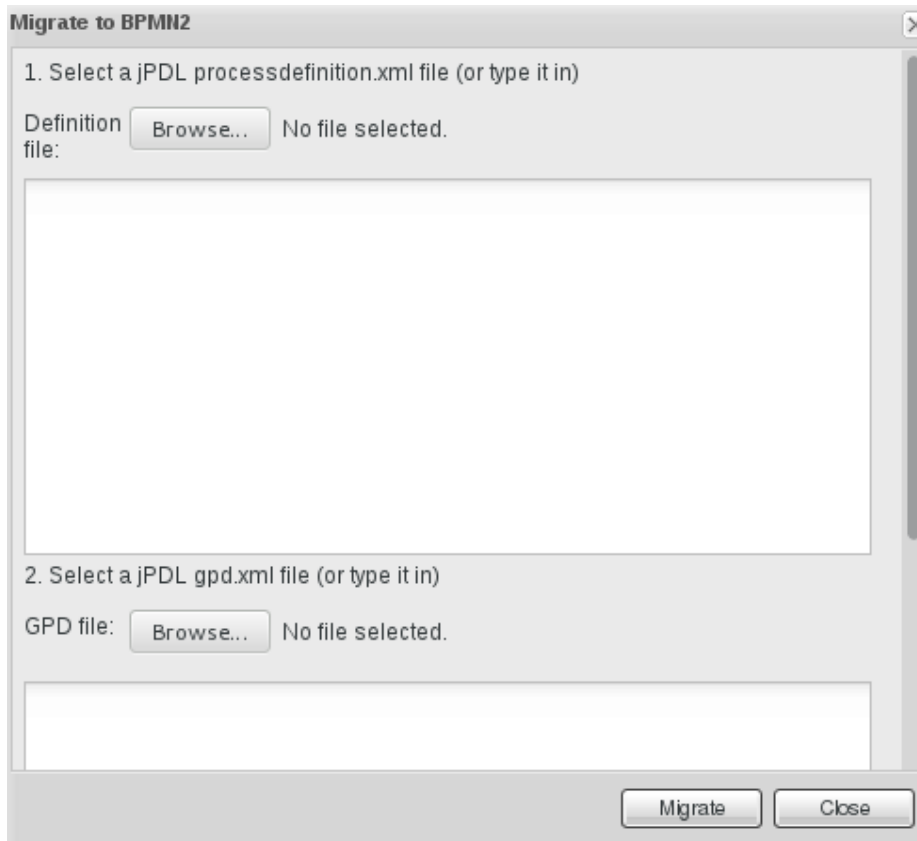


Figure 8.3. Migrer dans la boîte de dialogue BPMN2



IMPORTANT

L'outil de migration de jPDL 3.2 à BPMN2 est une fonctionnalité en aperçu technique, et n'est donc pas prise en charge par le Red Hat JBoss BPM Suite.

8.4. EXPORTATION D'UN PROCESSUS

Procédure 8.1. Exportation d'un processus métier

Pour exporter un processus métier, procéder ainsi :

1. Ouvrir la perspective **Création Projet** : à partir du menu principal, cliquer sur **Création** → **Création Projet**.
2. Sélectionner le processus métier à exporter pour le voir dans le Process Designer.
3. Cliquer sur le bouton () de la barre de conception de processus et sélectionner **Afficher les sources des processus** parmi les options de menu déroulant.
4. La fenêtre **Sources des processus** apparaît
5. Cliquer sur le bouton **Télécharger BPMN2** et sauvegarder le processus métier à l'emplacement désiré.

PARTIE III. INTÉGRATION

CHAPITRE 9. LE DÉPLOIEMENT D'ARTEFACTS DE RED HAT JBOSS BPM SUITE DANS S-RAMP (DE L'ANGLAIS REPOSITORY ARTIFACT MODEL AND PROTOCOL)

Alors que Red Hat JBoss BPM Suite et S-RAMP sont deux produits indépendants, il est possible de déplacer des artefacts entre les deux. Vous pouvez déplacer des artefacts de JBoss BPM Suite à S-RAMP en utilisant Maven ou via une interface utilisateur.

Cette section donne des informations sur ces deux processus.

9.1. LE DÉPLOIEMENT D'ARTEFACTS DE RED HAT JBOSS BPM SUITE DANS SOA REPOSITORY ARTIFACT MODEL AND PROTOCOL (S-RAMP) EN UTILISANT MAVEN

Avant que vous puissiez déployer des artefacts de Red Hat JBoss BPM Suite dans S-RAMP, vous devrez activer S-RAMP Maven Wagon. Maven Wagon est une fonctionnalité clé qui prend en charge le protocole API REST basé S-RAMP Atom. En activant S-RAMP Maven Wagon, les utilisateurs seront en mesure d'accéder aux artefacts du référentiel S-RAMP de dépendances dans un projet Maven.

Activer S-RAMP Maven Wagon en faisant une modification dans le fichier `pom.xml`, comme dans l'exemple ci-dessous :

```
<build>
  <extensions>
    <extension>
      <groupId>org.overlord.sramp</groupId>
      <artifactId>s-ramp-wagon</artifactId>
      <version>${s-ramp-wagon.version}</version>
    </extension>
  </extensions>
</build>
```

Une fois que S-RAMP Maven Wagon est activé, vous pouvez déployer les artefacts de JBoss BPM Suite dans le référentiel S-RAMP. Pour ce faire, procédez aux étapes suivantes :

1. Cloner le référentiel git où vous avez sauvegardé le projet BPM Suite en exécutant cette commande :

```
git clone http://localhost:8001/REPOSITORY_NAME
```

2. Par la ligne de commande, déplacez-vous dans le dossier qui contient le projet.
3. Suivez les instructions qui se trouvent dans *Red Hat JBoss Fuse Service Works 6 Development Guide, Volume 3: Governance*, section Déploiement dans S-RAMP. Utiliser l'URL de l'exemple ci-dessous :

```
<distributionManagement>
  <repository>
    <id>local-sramp-repo</id>
    <name>S-RAMP Releases Repository</name>
    <url>sramp://S-RAMP_SERVER_URL/s-ramp-server/</url>
  </repository>
  <snapshotRepository>
```

```

<id>local-sramp-repo-snapshots</id>
<name>S-RAMP Snapshots Repository</name>
<url>sramp://S-RAMP_SERVER_URL/s-ramp-server/</url>
</snapshotRepository>
</distributionManagement>

```

Avec ces paramètres de configuration, les déploiements de Maven sont envoyés directement dans le référentiel S-RAMP en utilisant l'API S-RAMP. Notez que les artefacts sont ajoutés au référentiel S-RAMP avec un type d'artefact basé sur le type Maven du projet. Vous pouvez substituer ce comportement en ajoutant un paramètre de recherche dans l'URL du référentiel dans le fichier `pom.xml` file. Par exemple :

```

<distributionManagement>
  <repository>
    <id>local-sramp-repo</id>
    <name>S-RAMP Releases Repository</name>
    <url>sramp://S-RAMP_SERVER_URL/s-ramp-server/?
artifactType=KieJarArchive</url>
  </repository>
</distributionManagement>

```

L'exemple ci-dessus entraîne l'artefact Maven à être téléchargé avec un type d'artefact S-RAMP de `KieJarArchive`.

4. Modifier le plug-in de Maven dans le fichier `pom.xml` et ajouter y une dépendance comme suit au cas où le projet ne contienne pas de tables de décisions :

```

<plugins>
  <plugin>
    <groupId>org.kie</groupId>
    <artifactId>kie-maven-plugin</artifactId>
    <version>6.4.0.Final-redhat-3</version>
    <extensions>true</extensions>
    <dependencies>
      <dependency>
        <groupId>org.jbpm</groupId>
        <artifactId>jbpm-bpmn2</artifactId>
        <version>6.4.0.Final-redhat-3</version>
      </dependency>
    </dependencies>
  </plugin>
</plugins>

```

Si le projet contient des tables de décisions, utiliser cette dépendance pour `kie-maven-plugin` à la place :

```

<plugins>
  <plugin>
    <groupId>org.kie</groupId>
    <artifactId>kie-maven-plugin</artifactId>
    <version>6.4.0.Final-redhat-3</version>
    <extensions>true</extensions>
    <dependencies>
      <dependency>
        <groupId>org.drools</groupId>

```

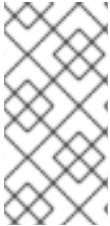
```

        <artifactId>drools-decisiontables</artifactId>
        <version>6.4.0.Final-redhat-3</version>
    </dependency>
</dependencies>
</plugin>
</plugins>

```

5. Exécuter un déploiement propre par la commande suivante :

```
mvn -s sramp-settings.xml deploy
```



NOTE

Pour le déploiement Maven dans le référentiel de S-RAMP, il vous faut disposer d'informations d'identification définies dans le fichier `settings.xml`. Pour plus de détails sur les informations d'identification, consulter la documentation *Red Hat JBoss Fuse Service Works (FSW)* sur l'authentification.

9.2. LE DÉPLOIEMENT D'ARTEFACTS DE RED HAT JBOSS BPM SUITE DANS SOA REPOSITORY ARTIFACT MODEL AND PROTOCOL (S-RAMP) EN UTILISANT L'INTERFACE GRAPHIQUE (GUI).

Pour déployer les artefacts de Red Hat JBoss BPM Suite dans un référentiel S-RAMP utilisant l'interface utilisateur, procéder ainsi :

1. Naviguer dans <http://localhost:8080/s-ramp-ui/> via navigateur web. Si l'interface utilisateur est configurée pour exécuter à partir d'un nom de domaine, substituer `localhost` par le nom de domaine. Par exemple <http://www.example.com:8080/s-ramp-ui/>.
2. Cliquer sur **Artefacts**.
3. Dans la section **Gestion des artefacts**, sélectionner le bouton **Importer**.
4. Situer l'archive kie que vous souhaitez déployer. Dans la boîte de dialogue qui s'ouvre, remplir **KieJarArchive** comme type, et sélectionner **Importer**.
5. Le déploiement crée alors ces entrées dans un référentiel S-RAMP :

KieJarArchive, à partir duquel on a :

- **KieXmlDocument** (si l'archive contient `kmodule.xml`)
- **BpmnDocument** (si l'archive contient les définitions `bpmn`)
- **DroolsDocument** (si l'archive contient les définitions `drl`)

CHAPITRE 10. INTÉGRER RED HAT JBOSS BPM SUITE DANS RED HAT JBOSS FUSE

L'intégration de Red Hat JBoss Fuse permet aux utilisateurs de JBoss Fuse de compléter leur solution d'intégration avec les fonctionnalités supplémentaires fournies par JBoss BPM et JBoss BRMS.

Red Hat JBoss BPMS Integration est fournie par deux fichiers `features.xml` :

- `drools-karaf-features-<version>-features.xml`

Ce fichier fournit les principales fonctionnalités de JBoss BPM Suite et JBoss BRMS, et définit les fonctionnalités de OSGi qui peuvent être déployées dans JBoss Fuse. Ce fichier fait partie du produit JBoss BPM Suite et JBoss BRMS. Les utilisateurs OSGi peuvent installer des fonctionnalités à partir de ce fichier pour installer JBoss BRMS Engine ou JBoss BPM Suite Engine dans Fuse et les utiliser dans leurs applications.

- `karaf-features-<version>-features.xml`

Ce fichier fournit des fonctionnalités supplémentaires utilisées pour l'intégration de BPM JBoss Suite et JBoss BRMS avec Camel, principalement dans Fuse. Ce fichier fait partie du Pack d'intégration, et il définit les fonctionnalités OSGi qui permettent l'intégration avec Camel et SwitchYard. En plus du `xml karaf-features`, le Pack d'integration contient également un fichier `features.xml` pour des démarrages rapides.

10.1. FONCTIONNALITÉS DE BASE DE JBOSS BPM SUITE ET DE JBOSS BRMS

Les fonctionnalités JBoss BPM Suite et JBoss BRMS sont fournies par le fichier `drools-karaf-features-<version>-features.xml` présent dans votre référentiel de produits Maven ou le fichier `jboss-brms-bpmsuite<version>-redhat<version>fuse-features.zip`. Il fournit les fonctionnalités suivantes :

- `drools-common`
- `drools-module`
- `drools-templates`
- `drools-decisiontable`
- `drools-jpa`
- `kie`
- `kie-ci`
- `kie-spring`
- `kie-aries-blueprint`
- `jbpm-commons`
- `jbpm-human-task`

- jbpm
- droolsjbpm-hibernate
- h2

Le tableau suivant donne un exemple de cas d'utilisation de certaines fonctionnalités listées ci-dessus.

Tableau 10.1. Exemples de fonctionnalités et de cas d'utilisation

Fonctionnalité	Cas d'utilisation
drools-module	Utiliser le moteur JBoss BRMS pour l'évaluation de règles, sans avoir besoin de tables de décisions, processus ou persistances.
drools-jpa	Utiliser le moteur de JBoss BRMS pour l'évaluation de règles de persistance ou de transactions, sans exiger de tables de décision ou de processus. La fonctionnalité drools-jpa contient déjà drools-module , mais vous devrez peut-être également installer la fonctionnalité droolsjbpm-hibernate , ou vous assurer qu'il y ait un package Hibernate compatible installé.
drools-decisiontable	Utiliser le moteur JBoss BRMS avec des tables de décision.
jbpm	Utiliser JBoss BPM Suite (ou le moteur JBoss BRMS avec les processus). La fonctionnalité jbpm contient déjà drools-module , et drools-jpa . Vous devrez peut-être également installer la fonctionnalité droolsjbpm-hibernate , ou vous assurer qu'il y ait un package Hibernate compatible installé.
jbpm et jbpm-human-task	Utiliser JBoss BPM Suite (ou le moteur JBoss BRMS avec des processus) avec Human Task.
Jars de moteurs de base et kie-ci .	Utiliser JBoss BRMS ou JBoss BPM Suite avec KieScanner (KIE-CI) pour télécharger les kJARs d'un référentiel Maven.
kie-spring	Utiliser l'intégration KIE-Spring.
kie-spring et kie-aries-blueprint .	Utiliser l'intégration KIE-Aries-Blueprint.

10.2. AUTRES FONCTIONNALITÉS DE L'INTÉGRATION SWITCHYARD ET CAMEL

Les fonctionnalités supplémentaires suivantes pour l'intégration de SwitchYard et de Camel dans JBoss Fuse sont fournies par le package d'intégration :

- fuse-bxms-switchyard-common-knowledge
- fuse-bxms-switchyard-rules
- fuse-bxms-switchyard-bpm
- kie-camel
- jbpmm-workitems-camel

Les fonctionnalités du Pack d'intégration sont définies dans le fichier `karaf-features-<version>-features.xml`. Ce fichier et les référentiels de support sont situés dans <http://repository.jboss.org/nexus/content/repositories/public>, qui est déjà configuré pour une utilisation dans JBoss Fuse prêt à l'emploi dans `installDir/etc/org.ops4j.pax.url.mvn.cfg`.

Le fichier peut également être téléchargé de la page produits de JBoss Fuse ou JBoss BPM Suite à partir du portail clients de Red Hat.

10.3. INSTALLER / METTRE À JOUR LES FONCTIONNALITÉS D'INTÉGRATION PRINCIPALES



NOTE

Cette section fait référence aux fonctionnalités qui se trouvent dans le fichier `drools-karaf-features-<version>-features.xml`. Pour obtenir des fonctionnalités d'intégration supplémentaires, consulter [Section 10.4, « Installer des fonctionnalités d'intégration supplémentaires »](#).

Si vous avez déjà installé une ancienne version des fonctionnalités de base de JBoss BPM Suite et de JBoss BRMS (par exemple, `drools-karaf-features-6.2.0.Final-redhat-6-features.xml`), vous devrez les supprimer, ainsi que tous les fichiers associés avant d'installer le fichier `features.xml` plus récent.

Procédure 10.1. Supprimer une installation `drools-karaf-features` existante

1. Démarrer la console Fuse par :

```
$ ./installDir/bin/fuse
```

2. Retirer l'installation d'anciennes fonctionnalités/apps qui utilisaient l'ancien fichier `features.xml`. Par exemple :

```
JBossFuse:karaf@root> features:uninstall drools-module
JBossFuse:karaf@root> features:uninstall jbpmm
JBossFuse:karaf@root> features:uninstall kie-ci
```

3. Chercher des références de lots en utilisant `drools/kie/jbpmm` et les supprimer :

```
karaf@root> list -t 0 -s | grep drools
karaf@root> list -t 0 -s | grep kie
karaf@root> list -t 0 -s | grep jbpmm
```

Pour supprimer les lots :

```
karaf@root> osgi:uninstall <BUNDLE_ID>
```

4. Supprimer l'ancien url drools-karaf-features :

```
karaf@root> features:removeurl mvn:org.drools/drools-karaf-features/6.2.0.Final-redhat-<version>/xml/features
```

5. Redémarrer Fuse.

Pour installer **drools-karaf-features** :

Procédure 10.2. Installer les fonctionnalités de base de JBoss BPM Suite et de JBoss BRMS

1. Configurer les référentiels demandés

- a. Modifier le fichier *installDir/etc/org.ops4j.pax.url.mvn.cfg* dans votre installation JBoss Fuse et ajouter la variable *org.ops4j.pax.url.mvn.repositories*, et noter que les entrées sont séparées par *' , \'* :

- <https://maven.repository.redhat.com/ga/>

2. Démarrer JBoss Fuse :

```
$ ./installDir/bin/fuse
```

3. Ajouter un référence au fichier de fonctionnalités de base en exécutant la commande de console suivante :

```
JBossFuse:karaf@root> features:addurl mvn:org.drools/drools-karaf-features/<version>/xml/features
```

Par exemple :

```
features:addurl mvn:org.drools/drools-karaf-features/6.4.0.Final-redhat-3/xml/features
```

4. Vous pouvez installer les fonctionnalités fournies dans ce fichier en exécutant, par exemple, la commande de console :

```
JBossFuse:karaf@root> features:install jbpm
```

10.4. INSTALLER DES FONCTIONNALITÉS D'INTÉGRATION SUPPLÉMENTAIRES

Utiliser la procédure suivante pour une intégration supplémentaire avec SwitchYard et Camel.

Procédure 10.3. Intégration SwitchYard et Camel

1. Télécharger le package *fuse-integration* qui est aligné à votre version de JBoss Fuse.

**NOTE**

Ainsi, si vous souhaitez utiliser la version 6.2.0.redhat-117 de JBoss Fuse, vous devrez installer les fonctionnalités d'intégration de JBoss Fuse **fuse-6.2.0.redhat-117**

2. Ajouter le référentiel Maven distant qui contient les dépendances de Fuse dans votre instance de karaf :

- Modifier *Fuse_home/etc/org.ops4j.pax.url.mvn.cfg*

3. Mettez à jour l'URL de fonctionnalités Drools :

```
JBossFuse:karaf@root> features:addurl
mvn:org.switchyard.karaf/mvn:org.switchyard.karaf/switchyard/<SWITCH
YARD_VERSION>/xml/core-features
JBossFuse:karaf@root> features:addurl mvn:org.drools/drools-karaf-
features/<DROOLS_VERSION>/xml/features
JBossFuse:karaf@root> features:addurl
mvn:org.jboss.integration.fuse/karaf-features/1.0.0.redhat-
<version>/xml/features
```

4. Vous pouvez installer les fonctionnalités fournies pour l'intégration SwitchYard and Camel en exécutant, par exemple, la commande de console suivante :

```
JBossFuse:karaf@root> features:install fuse-bxms-switchyard-rules
JBossFuse:karaf@root> features:install kie-camel
JBossFuse:karaf@root> features:install jbpm-workitems-camel
```

10.5. CONFIGURATION DES DÉPENDANCES

Quand vous configurez KIE, JBoss BRMS, ou Jboss BPM Suite dans votre application, vous pouvez suivre une des approches suivantes pour créer des groupements d'applications OSGi :

- Grouper les dépendances requises dans votre groupement d'applications. Avec cette approche, vous déclarez tous les artefacts requis en tant que dépendances de runtime dans votre `pom.xml`. Donc, vous ne devez pas importer les paquets qui fournissent ces artefacts que vous avez déjà ajoutés en tant que dépendances.
- Importer des dépendances requises dans le groupement d'applications. Il s'agit d'une approche privilégiée pour créer des groupements OSGi, car elle adhère aux principes de framework OSGi. Avec cette approche, vous ne déclarez que les jars d'API (par exemple `org.kie:kie-api`) en tant que dépendances dans votre groupement d'applications. Vous aurez besoin d'installer les groupements MRB et BPM Suite nécessaires, puis vous devrez les importer dans votre application.



AVERTISSEMENT

La classe `MVELUserGroupCallback` ne peut pas s'initialiser dans un environnement OSGi. Ne pas utiliser ou inclure `MVELUserGroupCallback` car ce n'est pas conçu à but de production.

10.6. INSTALLER LES APPLICATIONS QUICKSTART DE JBOSS FUSE INTEGRATION

Les fonctionnalités suivantes des applications quickstart de JBoss Fuse Integration sont fournies par `org/jboss/integration/fuse/quickstarts/karaf-features/1.0.0.redhat-<version>/karaf-features-1.0.0.redhat-<version>-features.xml` :

- `fuse-bxms-switchyard-quickstart-bpm-service`
- `fuse-bxms-switchyard-quickstart-rules-camel-cbr`
- `fuse-bxms-switchyard-quickstart-rules-interview`
- `fuse-bxms-switchyard-quickstart-rules-interview-container`
- `fuse-bxms-switchyard-quickstart-rules-interview-dtable`
- `fuse-bxms-switchyard-demo-library`
- `fuse-bxms-switchyard-demo-helpdesk`
- `fuse-bxms-camel-blueprint-drools-decision-table`
- `fuse-bxms-camel-spring-drools-decision-table`
- `fuse-bxms-jbpm-workitems-camel-quickstart`
- `fuse-bxms-spring-jbpm-osgi-example`

Ce fichier et les référentiels dépendants se situent dans <http://repository.jboss.org/nexus/content/repositories/public>, qui est déjà configuré pour être utilisé dans JBoss Fuse prêt à l'emploi dans `installDir/etc/org.ops4j.pax.url.mvn.cfg`.

Procédure 10.4. Installer l'application Quickstart

1. Ajouter une référence au fichier de fonctionnalités en exécutant la commande de console suivante :

```
JBossFuse:karaf@root> features:addurl
mvn:org.jboss.integration.fuse.quickstarts/karaf-
features/1.0.0.redhat-<version>/xml/features
```

2. Vous pouvez maintenant installer les applications quickstart fournies par ce fichier de fonctionnalités en exécutant, par exemple, la commande de console suivante :

```
JBossFuse:karaf@root> features:install fuse-bxms-switchyard-quickstart-bpm-service
```

Procédure 10.5. Télécharger et installer les fichiers ZIP de Quickstart

1. Télécharger le fichier ZIP d'installation Quickstart.
2. Extraire les éléments de contenu du répertoire quickstart dans votre répertoire *installDir/quickstart* existant.
3. Extraire les éléments de contenu du répertoire de système dans votre répertoire *installDir/system* existant.

10.6.1. Tester votre première application Quickstart

Procédure 10.6. Tester l'application Quickstart

1. Démarrer JBoss Fuse :

```
$ ./installDir/bin/fuse
```

2. Installer et démarrer **switchyard-bpm-service** en exécutant la commande de console suivante :

```
JBossFuse:karaf@root> features:install fuse-bxms-switchyard-quickstart-bpm-service
```



NOTE

Toutes les fonctionnalités dépendantes spécifiées dans le fichier des fonctionnalités de l'application seront installées automatiquement.

3. Soumettre une requête web pour invoquer la passerelle SOAP.
 - a. Ouvrir une fenêtre de terminal, et naviguez dans le répertoire quickstart associé extrait du fichier ZIP de l'application quickstart (dans ce cas, **switchyard-bpm-service**).
 - b. Exécuter la commande suivante :

```
$ mvn clean install
```



NOTE

Vous aurez besoin des référentiels suivants configurés dans votre fichier **settings.xml** :

- <http://maven.repository.redhat.com/ga/>
- <http://repository.jboss.org/nexus/content/repositories/public/>

- c. Exécuter la commande suivante :

```
$ mvn exec:java -Pkaraf
```

4. Vous recevrez la réponse suivante :

```
SOAP Reply:
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-
ENV:Header xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" /><soap:Body><ns2:sub
mitOrderResponse xmlns:ns2="urn:switchyard-quickstart:bpm-
service:1.0">
  <orderId>test1</orderId>
  <accepted>true</accepted>
  <status>Thanks for your order, it has been shipped!</status>
</ns2:submitOrderResponse></soap:Body></soap:Envelope>
```

CHAPITRE 11. INSTALLATION DE RED HAT JBOSS BPM SUITE AVEC RED HAT SSO.



NOTE

L'intégration de la BPM Suite avec le nouveau service Red Hat Single Sign-On (RH-SSO) à venir est offert gracieusement aux clients qui veulent essayer les dernières fonctionnalités de dernières sorties de version majeures, et qui peuvent avoir besoin d'un support si des problèmes surgissent.

Les composants d'intégration de la BPM Suite seront totalement pris en charge, ce qui inclut l'utilisation en production et les correctifs de bogues, mais le support du composant Red Hat SSO sous-jacent sera limité à la fonctionnalité Gestion Groupe et Utilisateur jusqu'à la sortie de la version RH-SSO GA. Le support limité de RH-SSO devrait inclure les cas de support technique de niveau de sévérité 4, l'accès à la base de données et les solutions de contournement, si disponibles. Tous les correctifs de bogue seront revus et considérés pour les versions à venir.

Quand RH-SSO sera en GA, les clients pourront utiliser tout son potentiel, en plus des scénarios intégrés de la Red Hat BPM Suite.

Red Hat Single Sign-On (RH-SSO) est une solution SSO que vous pouvez utiliser pour sécuriser vos applications de navigateur et vos services web REST. Ce chapitre décrit comment intégrer RH-SSO dans Red Hat JBoss BPM Suite et tirer partie de ses fonctionnalités.

L'intégration avec RH-SSO nous apporte un environnement SSO et IDM (Identity Management) intégrés pour Red Hat JBoss BPM Suite. De plus, cela vous permet d'utiliser vos comptes de connexion sociaux pour accéder à JBoss BPM Suite. La fonctionnalité de gestion de session de RH-SSO vous permet d'utiliser des environnements JBoss BPM Suite sur le web en authentifiant une seule fois.

Pour obtenir plus d'informations sur RH-SSO, consulter [documentation RH-SSO](#).

Points d'intégration de Red Hat SSO

Comme Red Hat JBoss BPM Suite a plusieurs Intelligent Process Servers autonomes, vous pouvez intégrer RH-SSO en utilisant les points d'intégration suivants :

- Authentification Business Central via serveur RH-SSO

Authentifier JBoss BPM Suite Business Central via RH-SSO implique de sécuriser le client web de Business Central et les services à distance via RH-SSO. Cette intégration vous permet de vous connecter à Business Central par une interface web ou par un consommateur de services à distance via RH-SSO.

- Authentification Intelligent Process Server via serveur RH-SSO

Authentifier JBoss BPM Suite Intelligent Process Server via RH-SSO implique de sécuriser les services à distance fournis par Intelligent Process Server car cela ne procure pas d'interface web pour l'authentification du serveur. Cela permet à un consommateur de service (utilisateur ou service) BPM Suite de s'authentifier via RH-SSO.

- Authentification de tierce partie via serveur RH-SSO

Authentifier un client de tierce partie par l'intermédiaire d'un serveur RH-SSO implique que les clients de tierce partie s'authentifient eux-mêmes par RH-SSO, afin de consommer les points de terminaison du service à distance fourni par Business Central et Intelligent Process Server.

Les sections suivantes décrivent comment réaliser une intégration RH-SSO par ces points d'intégration.

11.1. AUTHENTIFICATION BUSINESS CENTRAL VIA RH-SSO

Pour authentifier Business Central via RH-SSO :

1. Installer et exécuter un serveur RH-SSO avec un domaine client dans Business Central.
2. Installer et configurez l'adaptateur client RH-SSO dans EAP.
3. Sécuriser Business Central Remote Service via RH-SSO.
4. Créer un client RH-SSO et configurer l'adaptateur client RH-SSO dans Dashbuilder (BAM).

11.1.1. Configurer RH-SSO avec un domaine client dans Business Central

Les domaines de sécurité sont utilisés pour restreindre l'accès à différentes ressources d'applications. Si vous intégrez Red Hat JBoss BPM Suite dans une instance de RH-SSO déjà déployée, il est inutile d'ajouter un domaine. Si vous intégrez avec une instance de RH-SSO qui est partagée avec d'autres installations de produits pour pouvoir faire un SSO avec ces applications, toutes ces applications doivent utiliser le même domaine.

Voici comment vous pouvez installer un serveur RH-SSO et créer un domaine de sécurité dans Business Central :

Procédure 11.1. Configurer RH-SSO avec un domaine client

1. Installer et configurer un serveur autonome RH-SSO. Pour cela, suivre les instructions du [Guide d'installation RH-SSO](#).



NOTE

Si vous souhaitez exécuter à la fois les serveurs RH-SSO et Red Hat JBoss BPM Suite sur la même machine, veillez à éviter les conflits de ports. Pour cela, procédez ainsi :

- o Mettez à jour le fichier `$RHSSO_HOME/standalone/configuration/standalone.xml` et définissez le décalage de port à 100. Exemple :

```
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="${jboss.socket.binding.port-offset:100}">
```

- o Utiliser une variable d'environnement pour exécuter le serveur :

```
bin/standalone.sh -Djboss.socket.binding.port-offset=100
```

- Démarrez le serveur RH-SSO en utilisant la commande suivante dans `$RHSSO_HOME/bin`:

```
./standalone.sh
```

Une fois que le serveur RH-SSO a démarré, ouvrir <http://localhost:8180/auth/admin> dans un navigateur web, et connectez-vous à l'aide de vos identifiants admin que vous avez créés lors de l'installation de RH-SSO.



NOTE

Si vous n'avez pas créé d'utilisateur RH-SSO admin pendant l'installation, vous pourrez en créer un en naviguant dans `$RHSSO_HOME/bin` par la commande suivante :

```
./add-user.sh -r master -u admin -p <ADMIN_PASSWORD>
```

- Dans la console admin RH-SSO, cliquer sur l'onglet **Configurations de domaine**.
- Dans la page **Configurations de domaine**, cliquer sur **Ajouter Domaine**.
La page **Ajouter Domaine** s'ouvrira.
- Dans la page **Ajouter Domaine**, donner un nom au domaine et cliquer sur **Créer**.
- Cliquer sur l'onglet **Client** dans le menu principal d'administration de console, et cliquer sur **Créer**.

La page **Ajouter Client** s'ouvrira.

- Dans la page **Ajouter Client**, fournir les informations requises pour créer un nouveau client pour votre domaine. Exemple :
 - ID Client** : kie
 - Protocole Client** : openid-connect
 - Type d'accès** : confidentiel
 - URL Root** : `http://localhost:8080`
 - URL de base** : `/business-central`
 - URI de redirection valides** : `/business-central/*`



NOTE

Les valeurs des **URI de redirection valides** et l'**URL de base** doivent correspondre à votre nom de contexte d'application Red Hat JBoss BPM Suite et la valeur de l'**URL Root** doit correspondre à l'URL de base HTTP de votre application.

- Cliquez **Enregistrer** pour enregistrer vos modifications.

À ce stade, le serveur RH-SSO est configuré avec un domaine et un client pour des applications de Red Hat JBoss Suite BPM (Business Centre, dans cet exemple) et en cours d'exécution, à l'écoute des connexions HTTP à **localhost : 8180**. Ce domaine fournit différents utilisateurs, rôles et sessions pour les applications de Red Hat JBoss Suite BPM.

11.1.2. Configurer l'adaptateur client RH-SSO dans EAP

Pour configurer l'adaptateur client RH-SSO dans EAP :

1. Installer l'adaptateur RH-SSO dans EAP.
2. Configurez l'application Red Hat JBoss BPM Suite et l'adaptateur client RH-SSO.

Procédure 11.2. Installer l'adaptateur de client RH-SSO dans EAP

1. Installer EAP 6.4.x. Consulter [Chapitre 2. Instructions d'installation](#) du Guide d'installation de Red Hat JBoss Enterprise Application Platform.
2. Télécharger l'adaptateur EAP de la [la page de téléchargements de BPM Suite](#).
3. Décompressez et installez l'adaptateur. Pour obtenir des instructions sur l'installation, voir [Installer des adaptateurs pour Red Hat SSO](#) dans le Guide d'installation RH-SSO.

Procédure 11.3. Configurez l'adaptateur RH-SSO

1. Naviguez dans `$JBOSS_EAP/standalone/configuration` dans votre installation EAP et modifier `standalone.xml` pour pouvoir ajouter la configuration du sous-système RH-SSO. Exemple :

```
<subsystem xmlns="urn:jboss:domain:keycloak:1.1">
  <secure-deployment name="business-central.war">
    <realm>demo</realm>
    <realm-public-
key>MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCrVrCuTtArbgaZzL1hvh0xtL5m
c7o0NqPVnYXkLvgcwIC3BjLGW1tGEGoJaXDuSaRllobm53JBhjx33UNv+5z/UMG4kytB
WxheNVKnL6GgqLNabMaFfPLPCF8kAgKnsi79NMo+n6KnSY8YeUmec/p2vj02NjsSAVcW
EQMVhJ31LwIDAQAB</realm-public-key>
    <auth-server-url>http://localhost:8180/auth</auth-server-url>
    <ssl-required>external</ssl-required>
    <enable-basic-auth>true</enable-basic-auth>
    <resource>kie</resource>
    <credential name="secret">password</credential>
    <principal-attribute>preferred_username</principal-attribute>
  </secure-deployment>
</subsystem>
```

Voici ici,

- `secure-deployment name`: nom du fichier WAR de votre application.
- `realm`: nom du domaine que vous avez créé à utiliser pour vos applications.
- `realm-public-key`: clé publique du domaine que vous avez créé. Vous pourrez la trouver dans l'onglet **Clés** qui se trouve dans la page de domaine **Configurations Domaine** que vous avez créée dans la console d'administration RH-SSO. Si vous ne donnez aucune

valeur à `realm-public-key`, le serveur en extraira une automatiquement.

- `auth-server-url`: URL du serveur d'authentification RH-SSO.
- `enable-basic-auth`: paramètre pour activer le mécanisme d'authentification de base, afin que les clients puissent utiliser les deux méthodes d'authentification: basée jetons ou de base pour exécuter les requêtes.
- `resource`: nom du client que vous avez créé.
- `credential name`: clé secrète du client que vous avez créé. Vous pourrez la trouver dans l'onglet **Informations d'identification** dans la page **Clients** de la console d'administration RH-SSO.
- `principal-attribute`: le nom de connexion de l'utilisateur. Si vous ne procurez pas cette information, votre ID Utilisateur s'affichera à la place de votre nom d'utilisateur dans l'application.



NOTE

Le serveur RH-SSO convertit en minuscules les noms d'utilisateurs. Donc, après l'intégration avec RH-SSO, votre nom d'utilisateur apparaîtra en minuscules dans Business Central. Si vous avez des noms d'utilisateur en lettres majuscules codées en dur dans les processus métier, l'application n'est peut-être pas en mesure d'identifier les majuscules dans l'utilisateur.

2. Naviguez dans `$EAP_Home/bin/` et démarrez le serveur EAP par la commande suivante :

```
./standalone.sh
```

Vous pouvez maintenant vous connecter à votre application Red Hat BPM Suite (pour cet exemple, Business Central) une fois que le serveur exécute avec les identifiants admin RH-SSO.



NOTE

Vous pouvez également configurer l'adaptateur RH-SSO dans EAP en mettant à jour votre fichier WAR d'applications pour utiliser le sous-système de sécurité RH-SSO. Cependant, l'approche recommandée consiste à configurer l'adaptateur dans le sous-système de RH-SSO. Cela signifie que vous mettez à jour configuration EAP au lieu d'appliquer la configuration sur chaque fichier WAR.

11.1.3. Sécuriser Business Central Remote Service via RH-SSO.

Business Central fournit des points de terminaison de services à distance différents pouvant être consommés par des clients tiers à l'aide d'une API distante. Pour authentifier ces services par l'entremise de RH-SSO, vous devez désactiver un filtre de sécurité nommé *BasicAuthSecurityFilter*. Pour ce faire, procédez comme suit :

Procédure 11.4. Désactiver BasicAuthSecurityFilter

1. Ouvrir votre fichier de descripteur de déploiement d'applications (`WEB-INF/web.xml`) et appliquez-y les changements suivants :

- Supprimer les lignes suivantes pour supprimer le filtre du servlet et ses mappages pour la classe `org.uberfire.ext.security.server.BasicAuthSecurityFilter` :

```
<filter>
  <filter-name>HTTP Basic Auth Filter</filter-name>
  <filter-
class>org.uberfire.ext.security.server.BasicAuthSecurityFilter</f
ilter-class>
  <init-param>
    <param-name>realmName</param-name>
    <param-value>KIE Workbench Realm</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>HTTP Basic Auth Filter</filter-name>
  <url-pattern>/rest/*</url-pattern>
  <url-pattern>/maven2/*</url-pattern>
  <url-pattern>/ws/*</url-pattern>
</filter-mapping>
```

- Ajouter les lignes suivantes pour ajouter la *security-constraint* aux url-patterns que vous avez supprimés du mappage de filtre :

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>remote-services</web-resource-name>
    <url-pattern>/rest/*</url-pattern>
    <url-pattern>/maven2/*</url-pattern>
    <url-pattern>/ws/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>rest-all</role-name>
  </auth-constraint>
</security-constraint>
```

2. Enregistrez vos modifications.

11.1.4. Créez et configurez l'adaptateur client RH-SSO dans BAM

Une fois que vous aurez créé un client de domaine dans Business Central et configuré votre adaptateur client RH-SSO dans EAP, vous pourrez répéter les mêmes étapes pour intégrer BAM avec RH-SSO. Après l'intégration des RH-SSO dans BAM, vous pouvez naviguer dans Business Central et BAM sans avoir à vous re-connecter.

Procédure 11.5. Intégrer RH-SSO dans BAM

1. Dans la console d'administration RH-SSO, ouvrir le domaine de sécurité que vous avez créé.
2. Cliquer sur l'onglet **Client** dans le menu principal d'administration de console, et cliquer sur **Créer**.

La page **Ajouter Client** s'ouvrira.

3. Dans la page **Ajouter Client**, fournir les informations requises pour créer un nouveau client BAM pour votre domaine. Exemple :

- **Nom** : dashbuilder
- **URL Root** : http://localhost:8080
- **Protocole Client** : openid-connect
- **Type d'accès** : confidentiel
- **URI de redirection valides** : /dashbuilder/*
- **URL de base** : /dashbuilder



NOTE

Le chemin de contexte de l'application Dashbuilder est *dashbuilder* par défaut.

4. Configurer l'adaptateur client RH-SSO. Pour ce faire, naviguez dans **\$JBOSS_EAP/standalone/configuration** dans votre installation EAP et modifier **standalone.xml** pour pouvoir ajouter la configuration du sous-système RH-SSO. Exemple :

```
<secure-deployment name="dashbuilder.war">
  <realm>demo</realm>
  <realm-public-
key>MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCrVrCuTtArbgaZzL1hvh0xtL5m
c7o0NqPVnYXkLvGcwIC3BjLGw1tGEGoJaXDuSaRllobm53JBhjx33UNv+5z/UMG4kytB
WxheNVKnL6GgqINabMaFfPLPCF8kAgKnsi79NMo+n6KnSY8YeUmec/p2vj02NjsSAVcW
EQMVhJ31LwIDAQAB</realm-public-key>
  <auth-server-url>http://localhost:8180/auth</auth-server-url>
  <ssl-required>external</ssl-required>
  <resource>dashbuilder</resource>
  <enable-basic-auth>true</enable-basic-auth>
  <credential name="secret">e92ec68d-6177-4239-be05-
28ef2f3460ff</credential>
  <principal-attribute>preferred_username</principal-attribute>
</secure-deployment>
```

Voici ici,

- **secure-deployment name**: nom du fichier BAM en cours de déploiement.
- **realm-public-key**: clé publique du domaine que vous avez créé.
- **resource**: nom du client que vous avez créé pour BAM (dashbuilder).
- **enable-basic-auth**: paramètre pour activer le mécanisme d'authentification de base, afin que les clients puissent utiliser les deux méthodes d'authentification: basée jetons ou de base pour exécuter les requêtes.
- **credential name**: clé secrète du client que vous avez créé pour BAM. Vous pourrez la trouver dans l'onglet **Informations d'identification** dans la page **Clients** de la console d'administration RH-SSO.

- **principal-attribute**: le nom de connexion de l'utilisateur. Si vous ne procurez pas cette information, votre ID Utilisateur s'affichera à la place de votre nom d'utilisateur dans l'application.
5. Redémarrez le serveur EAP et ouvrez <http://localhost:8080/dashbuilder> dans un navigateur web pour accéder BAM. Connectez-vous à BAM par le biais de vos identifiants admin RH-SSO.

Vous pouvez maintenant accéder aux deux applications (Business Central et BAM) sans avoir à vous re-connecter.

11.2. AUTHENTIFICATION INTELLIGENT PROCESS SERVER VIA RH-SSO

Red Hat JBoss BPM Suite Intelligent Process Server fournit une API REST pour les clients de tierce partie. Vous pouvez intégrer Intelligent Process Server avec RH-SSO pour déléguer la gestion de l'identité des clients au serveur RH-SSO.

Une fois que vous aurez créé un client de domaine dans Business Central et configuré l'adaptateur client RH-SSO dans EAP, vous pourrez répéter les mêmes étapes pour intégrer Intelligent Process Server avec RH-SSO.

11.2.1. Créer un client pour Intelligent Process Server dans RH-SSO

Pour créer un nouveau client sur votre domaine déjà créé dans une console admin RH-SSO :

Procédure 11.6. Créer un client pour Intelligent Process Server

1. Dans la console d'administration RH-SSO, ouvrez le domaine de sécurité que vous avez créé.
2. Cliquez sur l'onglet **Client** dans le menu principal, et cliquez sur **Créer**.

La page **Ajouter Client** s'ouvrira.

3. Dans la page **Ajouter Client**, fournissez les informations requises pour créer un nouveau client pour votre domaine. Exemple :

- **Nom**: kie-execution-server
- **URL Root**: <http://localhost:8080/>
- **Protocole Client** : openid-connect
- **Type d'accès** : confidentiel
- **URI de redirection valides**: /kie-server/*
- **URL de base**: /kie-server

4. Naviguez dans l'onglet **Informations d'identification** et copiez la clé secrète, puis collez-la dans l'écran de configuration du client kie-execution-server.
5. Cliquez **Enregistrer** pour enregistrer vos modifications.

11.2.2. Installer et configurer Intelligent Process Server avec un adaptateur client.

Pour consommer les points de terminaison du service à distance Intelligent Process Server, vous devez tout d'abord créer et assigner le rôle *kie-server* dans la console admin RH-SSO admin.

Procédure 11.7. Créer Intelligent Process Server

1. Naviguez dans `$JBoss_EAP/standalone/configuration` dans votre installation EAP et modifiez `standalone.xml` pour pouvoir ajouter la configuration du sous-système RH-SSO. Exemple :

```
<subsystem xmlns="urn:jboss:domain:keycloak:1.1">
  <secure-deployment name="kie-server.war">
    <realm>demo</realm>
    <realm-public-key>MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCrVrCuTtArbgaZzL1hvh0xtL5m
c7o0NqPVnYXkLvgcwic3BjLGW1tGEGoJaXDuSaRllobm53JBhjx33UNv+5z/UMG4kytB
WxheNVKnL6GgqINabMaFfPLPCF8kAgKnsi79NMo+n6KnSY8YeUmec/p2vj02NjsSAVcW
EQMVhJ31LwIDAQAB</realm-public-key>
    <auth-server-url>http://localhost:8180/auth</auth-server-url>
    <ssl-required>external</ssl-required>
    <resource>kie-execution-server</resource>
    <enable-basic-auth>true</enable-basic-auth>
    <credential name="secret">03c2b267-7f64-4647-8566-
572be673f5fa</credential>
    <principal-attribute>preferred_username</principal-attribute>
  </secure-deployment>
</subsystem>

<system-properties>
<property name="org.kie.server.sync.deploy" value="false"/>
</system-properties>
```

Voici ici,

- `secure-deployment name`: nom du fichier WAR de votre application.
- `realm`: nom du domaine que vous avez créé à utiliser pour vos applications.
- `realm-public-key`: clé publique du domaine que vous avez créé. Vous pourrez la trouver dans l'onglet **Clés** qui se trouve dans la page de domaine **Configurations de domaine** que vous avez créée dans la console admin RH-SSO. Si vous ne donnez aucune valeur à cette clé publique, le serveur en extraira une automatiquement.
- `auth-server-url`: URL du serveur d'authentification RH-SSO.
- `resource`: nom du serveur client que vous avez créé.
- `enable-basic-auth`: paramètre pour activer le mécanisme d'authentification de base, afin que les clients puissent utiliser les deux méthodes d'authentification: basée jetons ou de base pour exécuter les requêtes.
- `credential name`: clé secrète du serveur client que vous avez créé. Vous pourrez la trouver dans l'onglet **Informations d'identification** dans la page **Clients** de la console admin RH-SSO.

- o `principal-attribute`: le nom de connexion de l'utilisateur. Si vous ne procurez pas cette information, votre ID Utilisateur s'affichera à la place de votre nom d'utilisateur dans l'application.
2. Enregistrez vos changements dans `standalone.xml`.
 3. Utilisez la commande suivante pour redémarrer le serveur EAP et exécuter Intelligent Process Server.

```
$EXEC_SERVER_HOME/bin/standalone.sh -Dorg.kie.server.id=<ID> -
Dorg.kie.server.user=<USER> -Dorg.kie.server.pwd=<PWD> -
Dorg.kie.server.location=<LOCATION_URL> -Dorg.kie.server.controller=
<CONTROLLER_URL> -Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PASSWORD>
```

Voici un exemple :

```
$EXEC_SERVER_HOME/bin/standalone.sh -Dorg.kie.server.id=kieserver1 -
Dorg.kie.server.user=kieserver -Dorg.kie.server.pwd=password -
Dorg.kie.server.location=http://localhost:8080/kie-
server/services/rest/server -
Dorg.kie.server.controller=http://localhost:8080/business-
central/rest/controller -
Dorg.kie.server.controller.user=kiecontroller -
Dorg.kie.server.controller.pwd=password
```

4. Une fois que Intelligent Process Server est en cours d'exécution, vous pourrez vérifier le statut du serveur à l'aide de la commande suivante :

```
curl http://admin:password@localhost:8080/kie-
server/services/rest/server/
```



NOTE

L'autorisation basée jetons est également prise en charge pour la communication entre Business Central et Intelligent Process Server. De plus, vous pouvez utiliser le jeton complet comme propriété système (au lieu du nom d'utilisateur et du mot de passe) pour vos applications. Toutefois, vous devez vérifier que le jeton n'expire pas pendant la période d'interaction entre les applications, car il n'est pas automatiquement réactualisé.

11.3. AUTHENTIFICATION DE TIERCE PARTIE VIA RH-SSO

Pour utiliser les services à distance différents fournis par Red Hat JBoss Suite BPM ou par Intelligent Process Server, votre client doit être authentifié sur le serveur de RH-SSO, et avoir un jeton valide pour exécuter les requêtes. Pour utiliser les services à distance, l'utilisateur authentifié doit avoir les rôles suivants :

- ***rest-all***: pour utiliser les services à distance de Business Central

**NOTE**

Le rôle `rest-all` doit contenir l'utilisateur dans la propriété `org.kie.server.controller.user`.

- **kie-server**: pour utiliser les services à distance Intelligent Process Server

Utiliser la console d'administration RH-SSO pour créer ces rôles et les assigner aux utilisateurs qui vont consommer les services distants.

Pour réussir une authentification de client de tierce partie via RH-SSO, vous pourrez choisir l'une des options suivantes :

- Authentification de base (si le client de l'application le supporte)
- Authentification basée jetons

11.3.1. Authentification de base

Si vous avez activé l'authentification de base dans la configuration de l'adaptateur du client RH-SSO pour Business Central et Intelligent Process Server à la fois, vous pourrez éviter les appels de demandes/réactualisation de jetons et appeler les services comme expliqué dans l'exemple ci-dessous.

- Pour les points de terminaison distants basés web :

```
curl http://admin:password@localhost:8080/business-
central/rest/repositories
```

- Pour Intelligent Process Server :

```
curl http://admin:password@localhost:8080/kie-
server/services/rest/server/
```

11.3.2. Authentification basée jetons

Si vous souhaitez opter pour une option plus sécurisée de l'authentification, vous pouvez consommer les services à distance de Central Business et Intelligent Process Server en utilisant un jeton obtenu par un nouveau client RH-SSO.

Procédure 11.8. Obtenir et utiliser un jeton pour autoriser les appels distants

1. Cliquer sur l'onglet **Client** dans le menu principal d'administration de console, et cliquer sur **Créer** pour créer un nouveau client.

La page **Ajouter Client** s'ouvrira.

2. Dans la page **Ajouter Client**, fournir les informations requises pour créer un nouveau client pour votre domaine. Exemple :

- **ID Client** : kie-remote
- **Protocole Client** : openid-connect

- **Type d'accès :** public
- **URI de redirection valides :** http://localhost/

3. Cliquez **Enregistrer** pour enregistrer vos modifications.

4. Obtenir un jeton à partir de **Configurations de domaine**.

1. Dans la console admin RH-SSO, cliquer sur l'onglet **Configurations de domaine**.
2. Cliquer sur l'onglet **Jetons**.
3. Changer la valeur de **Durée de vie des jetons d'accès** à **15 minutes**.

Cela vous donne suffisamment de temps pour obtenir un jeton et invoquer le service avant qu'il n'expire.

4. Cliquez **Enregistrer** pour enregistrer vos modifications.

5. Une fois qu'un client public est créé pour vos clients distants, vous pourrez obtenir le jeton en faisant une requête HTTP au point de terminaison de jeton du serveur RH-SSO, en utilisant :

```
RESULT=`curl --data "grant_type=password&client_id=kie-remote&username=admin&password=password"
http://localhost:8180/auth/realms/demo/protocol/openid-connect/token`
```

6. Pour apercevoir le jeton obtenu du serveur RH-SSO, utiliser la commande suivante :

```
TOKEN=`echo $RESULT | sed 's/.*access_token":"//g' | sed 's/".*//g'`
```

Vous pouvez maintenant utiliser ce jeton pour autoriser les appels distants. Par exemple, si vous souhaitez vérifier les référentiels internes BPM Suite, utiliser le jeton comme expliqué ci-dessous :

```
curl -H "Authorization: bearer $TOKEN" http://localhost:8080/business-central/rest/repositories
```

CHAPITRE 12. INTÉGRATION AVEC SPRING

12.1. CONFIGURER RED HAT JBOSS BPM SUITE AVEC SPRING

Consulter le Guide d'installation de Red Hat JBoss BPM Suite pour télécharger le module Spring. Vous aurez besoin de télécharger la version déployable de JBoss BPM Suite.

Le module de Spring se trouve dans le fichier `jboss-bpms-engine.zip` et s'appelle `kie-spring-VERSION-redhat-MINORVERSION.jar`.

Ce que vous souhaitez faire avec les modules Spring affecte la façon dont vous aurez besoin de les configurer.

En tant que Moteur de processus auto gérés

C'est la façon habituelle de commencer à utiliser Suite BPM JBoss dans votre application Spring. Vous la pré-configurez une seule fois et vous l'exécutez dans le cadre de l'application. Grâce à l'API `RuntimeManager`, une synchronisation parfaite entre le service de tâches et le moteur de processus est gérée en interne et l'utilisateur final n'a pas à traiter le code interne pour les faire fonctionner ensemble.

En tant que service de tâches partagées

Lorsque vous utilisez une instance unique d'un `TaskService`, vous avez plus de flexibilité dans la configuration de l'instance de service de tâches car elle est indépendante du `RuntimeManager`. Une fois configurée, elle est ensuite utilisée par le `RuntimeManager` à la demande.

Pour créer un `RuntimeEnvironment` partir de votre application Spring, vous pouvez utiliser la classe `org.kie.spring.factorybeans.RuntimeEnvironmentFactoryBean`. Cette classe de fabrique est chargée de produire des instances de `RuntimeEnvironment` qui sont consommées par le `RuntimeManager` lors de la création. Illustré ci-dessous, vous verrez un `RuntimeEnvironment` configuré avec le gestionnaire d'entités, le gestionnaire de transactions et les ressources pour la classe `org.kie.spring.factorybeans.RuntimeEnvironmentFactoryBean`:

```
<bean id="runtimeEnvironment"
class="org.kie.spring.factorybeans.RuntimeEnvironmentFactoryBean">
  <property name="type" value="DEFAULT"/>
  <property name="entityManagerFactory" ref="jbpmEMF"/>
  <property name="transactionManager" ref="jbpmTxManager"/>
  <property name="assets">
    <map>
      <entry key-ref="process"><util:constant static-
field="org.kie.api.io.ResourceType.BPMN2"/></entry>
    </map>
  </property>
</bean>
```

L'environnement de runtime suivant peut être créé ou configuré :

- **DEFAULT** - configuration par défaut (le plus commun) du `RuntimeManager`
- **EMPTY** - environnement complètement vide à remplir manuellement
- **DEFAULT_IN_MEMORY** - la même chose que **DEFAULT** mais sans persistance du moteur de runtime

- **DEFAULT_KJAR** - la même chose que **DEFAULT** mais les ressources de connaissance sont prises dans le KJAR identifié par **releaseld** ou le **GAV**
- **DEFAULT_KJAR_CL** - généré directement à partir d'un chemin de classe qui consiste en un descripteur **kmodule.xml**

Suivant le type sélectionné, plusieurs propriétés obligatoires sont requises. Cependant, une des propriétés de connaissance suivantes doit être fournie :

- **knowledgeBase**
- **assets**
- **releaseld**
- **groupId**, **artifactId**, **version**

Finalement, pour les types **DEFAULT**, **DEFAULT_KJAR**, **DEFAULT_KJAR_CL**, la persistance doit être configurée sous la forme de valeurs **entity manager factory** ou **transaction manager**. Voici ci-dessous un exemple de **RuntimeManager** pour **org.kie.spring.factorybeans.RuntimeManagerFactoryBean** :

```
<bean id="runtimeManager"
class="org.kie.spring.factorybeans.RuntimeManagerFactoryBean" destroy-
method="close">
  <property name="identifiant" value="spring-rm"/>
  <property name="runtimeEnvironment" ref="runtimeEnvironment"/>
</bean>
```

CHAPITRE 13. INTÉGRATION CDI

13.1. INTÉGRATION CDI

Pour utiliser jbpm-kie-services dans votre système, vous devrez fournir certains mbeans qui puissent satisfaire toutes les dépendances des services. Il y a plusieurs mbeans qui dépendent de certains scénarios.

- Gestionnaire d'entités et Usine de gestionnaires d'entités
- Callback de groupes d'utilisateurs pour des tâches humaines
- Fournisseur d'identité pour passer les informations d'authentification de l'utilisateur aux services.

Quand vous exécutez un environnement JEE, comme JBoss Application Server, le mbean doit remplir toutes les exigences de jbpm-kie-services

```
public class EnvironmentProducer {

    @PersistenceUnit(unitName = "org.jbpm.domain")
    private EntityManagerFactory emf;

    @Inject
    @Selectable
    private UserGroupCallback userGroupCallback;

    @Produces
    public EntityManagerFactory getEntityManagerFactory() {
        return this.emf;
    }

    @Produces
    @RequestScoped
    public EntityManager getEntityManager() {
        EntityManager em = emf.createEntityManager();
        return em;
    }

    public void close(@Disposes EntityManager em) {
        em.close();
    }

    @Produces
    public UserGroupCallback produceSelectedUserGroupCallback() {
        return userGroupCallback;
    }

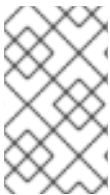
    @Produces
    public IdentityProvider produceIdentityProvider {
        return new IdentityProvider() {
            // implement IdentityProvider
        };
    }
}
```

Alors le fichier `deployments/business-central.war/WEB-INF/beans.xml` peut être configuré pour pouvoir modifier les paramètres de configuration actuels de l'implémentation de `usergroupcallback`.

```
<beans xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://docs.jboss.org/cdi/beans_1_0.xsd">

<alternatives>
  <class>org.jbpm.services.task.identity.JAASUserGroupCallbackImpl</class>
</alternatives>

</beans>
```



NOTE

`org.jbpm.services.task.identity.JAASUserGroupCallbackImpl` est juste un exemple pour démontrer les paramètres de configuration du serveur d'applications quel qu'il fût (LDAP, DB, etc).

CHAPITRE 14. PERSISTANCE

Les données de runtime du Process Engine peuvent être persistées dans les stores de données. Le mécanisme de persistance sauvegarde les données par le marshallng : les données de runtime sont converties en jeu de données et le jeu de données est sauvegardé dans le stockage de données.

Notez que la persistance n'est pas configurée par défaut et le moteur exécute sans persistance.



NOTE

Les données de runtime sont sauvegardées en utilisant le marshallng (persistance binaire). Le mécanisme de marshallng est un mécanisme de sérialisation personnalisée.

Red Hat JBoss BPM Suite persistera ce qui suit quand la persistance sera configurée :

- État de session : cela inclut l'ID de session, la date de la dernière modification, les données de session dont les règles métier peuvent avoir besoin pour l'évaluation, l'état des tâches de minuterie.
- État d'instance de processus : cela inclut l'ID d'instance de processus, l'ID de processus, la date de la dernière modification, la date du dernier accès en lecture, la date de démarrage de l'instance de processus, les données de runtime (le statut d'exécution comprenant le noeud à exécuter, les valeurs des variables, etc.) et les types d'événements.
- État de runtime d'élément de travail : inclut l'ID d'élément de travail, la date de création, le nom, l'ID d'instance de processus, et l'état de l'élément de travail lui-même.

Sur la base des données persistées, il est possible de restaurer l'état d'exécution de toutes les instances exécutant en cas d'échec ou pour supprimer les instances en cours d'exécution de façon temporaire en les restaurant plus tard. Par défaut, aucune persistance n'est configurée.

Pour autoriser la persistance, vous devez ajouter les fichiers jar de jbpmpersistance au CLASSPATH de votre application et configurer le moteur pour pouvoir utiliser la persistance. Le moteur enregistre automatiquement l'état d'exécution dans le stockage lorsque le moteur atteint un point sûr. Les points sécurisés sont des points où l'instance de processus est suspendue. Lorsqu'une invocation d'instance de processus atteint un point sécurisé dans le moteur, le moteur stocke toutes les modifications à l'instance de processus comme un instantané des données de runtime du processus. Toutefois, lorsqu'une instance de processus est terminée, l'instantané persisté des données de runtime de l'instance de processus est automatiquement supprimé.

Si une défaillance a lieu et que vous avez besoin de restaurer le runtime du moteur à partir du stockage, les instances de processus seront automatiquement restaurées et leur exécution continuera, donc vous n'aurez pas besoin de recharger ou de déclencher les instances de processus manuellement.

Les données de persistance de runtime doivent être considérées comme étant internes au moteur. Vous ne devez pas accéder à des données de runtime persistées ou les modifier directement car cela pourrait avoir des effets inattendus.

Pour obtenir des informations sur l'état actuel d'exécution, voir le journal de l'historique. Interroger la base de données de données de temps d'exécution que si c'est absolument nécessaire.

14.1. SESSION

Les sessions sont persistées comme entités **SessionInfo**. Elles persistent l'état de la session KIE de runtime, et stocke les données suivantes :

Tableau 14.1.

Champ	Description	Nullable
id	Clé primaire	false
lastmodificationdate	dernière sauvegarde dans le store de données	N/A
rulesbytearray	jeu de données binaire avec état de session (blob binaire)	false
startdate	démarrage de session	
optlock	numéro de version utilisé pour verrouiller la valeur d'un verrouillage optimiste	

14.2. INSTANCE DE PROCESSUS

Les instances de processus sont persistées en tant qu'entités **ProcessInstanceInfo**, qui persistent l'état d'une instance de processus en cours d'exécution et qui stockent les données suivantes :

Tableau 14.2.

Champ	Description	Nullable
instanceid	Clé primaire	false
lastmodificationdate	dernière sauvegarde dans le store de données	N/A
lastreaddate	dernière lecture dans le store de données	N/A
processid	ID du processus sur lequel l'instance est basée	false
processinstancebytearray	jeu de données binaires avec état d'instance de processus (blob binaire)	false
startdate	Date de démarrage de l'instance de processus	
optlock	numéro de version utilisée comme valeur de verrouillage pour le verrouillage optimiste	
état	État d'instance de processus	false

ProcessInstanceInfo possède une relation 1:N avec l'entité **EventTypes**.

L'entité **EventTypes** contient les données suivantes :

Tableau 14.3.

Champ	Description	Nullable
instanceid	référence à l'instance de processus (clé étrangère de processinstanceinfo)	false
élément	champ de texte lié à un événement subi par une instance de processus	

Support du verrouillage pessimiste

Le mécanisme de verrouillage de persistance des processus par défaut est *optimiste*. Avec une haute simultanéité multi-thread pour une même instance de processus, cette stratégie de verrouillage peut entraîner de mauvaises performances.

Avec la sortie de la version 6.1 de Red Hat JBoss BPM Suite, cela peut être modifié pendant l'exécution pour permettre à l'utilisateur de définir le verrouillage sur la base d'un processus à la fois, et lui permettre d'être *pessimiste* (la modification peut être faite au niveau session KIE ou Runtime Manager et pas seulement au niveau du processus).

Pour configurer un processus qui puisse utiliser le verrouillage pessimiste, procédez ainsi dans un environnement de runtime :

```
import org.kie.api.runtime.Environment;
import org.kie.api.runtime.EnvironmentName;
import org.kie.api.runtime.manager.RuntimeManager;
import org.kie.api.runtime.manager.RuntimeManagerFactory;

...

// here env is an instance of org.kie.api.runtime.Environment
env.set(EnvironmentName.USE_PESSIMISTIC_LOCKING, true);

// now create your Runtime Manager using this environment
RuntimeManager manager =
RuntimeManagerFactory.Factory.get().newPerRequestRuntimeManager(environment);
```

14.3. ÉLÉMENT DE TRAVAIL

Les éléments de travail sont persistés comme entités **workiteminfo**, qui persistent l'état d'une instance d'élément de travail en particulier en cours d'exécution et qui stocke les données suivantes :

Tableau 14.4.

Champ	Description	Nullable
workitemid	Clé primaire	false
nom	nom de l'élément de travail	
processinstanceid	id d'instance de processus parent	false
état	entier relatif représentant un état d'élément de travail	false
optlock	numéro de version utilisée comme valeur de verrouillage pour le verrouillage optimiste	
workitembytearray	ensemble de données binaires avec état d'élément de travail (blob binaire)	false
creationDate	horodate de création de l'élément de travail	false

14.4. CONFIGURATION DE LA PERSISTANCE

14.4.1. Configuration de la persistance

Bien que la persistance ne soit pas utilisée par défaut, les dépendances nécessaires sont disponibles dans le répertoire d'exécution sous forme de fichiers jar .

La persistance est définie par session, et vous pouvez la définir en utilisant la classe **JBPMHelper** quand vous créez une session ou en utilisant **JPAKnowledgeService** pour créer votre session. La dernière option offre plus de flexibilité, tandis que **JBPMHelper** a une méthode pour créer une session et utilise un fichier de configuration pour configurer cette session.

14.4.2. Configurer une persistance en utilisant JBPMHelper

Pour configurer une persistance de votre session en utilisant **JBPMHelper**, procéder ainsi :

1. Définir votre application pour qu'elle puisse utiliser le constructeur de session **JBPMHelper** :
 - `KieSession ksession = JBPMHelper.newKieSession(kbase);`
 - `KieSession ksession = JBPMHelper.loadKieSession(kbase, sessionId);`
2. Configurer la persistance dans le fichier `jBPM.properties` .

Exemple 14.1. Extrait de fichier `jBPM.properties` avec une persistance pour la base de données H2 en-mémoire

```
# for creating a datasource
persistence.datasource.name=jdbc/jbpm-ds
persistence.datasource.user=sa
```

```

persistence.datasource.password=
persistence.datasource.url=jdbc:h2:tcp://localhost/~/jbpm-db
persistence.datasource.driverClassName=org.h2.Driver

# for configuring persistence of the session
persistence.enabled=true
persistence.persistenceunit.name=org.jbpm.persistence.jpa
persistence.persistenceunit.dialect=org.hibernate.dialect.H2Dialect

# for configuring the human task service
taskservice.enabled=true
taskservice.datasource.name=org.jbpm.task
taskservice.transport=mina
taskservice.usergroupcallback=org.jbpm.task.service.DefaultUserGroupCallbackImpl

```

Toutes les invocations sur la session vont déclencher le processus de persistance.

Veillez à ce que la source de données fonctionne bien au démarrage du moteur. Si vous exécutez la base de données H2 en-mémoire, vous pouvez démarrer la base de données avec votre application par la méthode `JBPMHelper.startH2Server()`; et l'enregistrer dans engine par l'appel de méthode `JBPMHelper.setupDataSource()`;

14.4.3. Configurer une persistance en utilisant JPAKnowledgeService

Pour créer votre session de connaissance et configurer sa persistance avec JPAKnowledgeService, procéder ainsi :

1. Définir votre application pour qu'elle utilise la session de connaissance créée par JPAKnowledgeService :
 - o Définir la session sur la base d'une base de connaissances, d'une configuration de session de connaissance, et d'un environnement. L'environnement doit contenir une référence à l'usine de gestionnaires d'entités :

```

// create the entity manager factory and register it in the
// environment
EntityManagerFactory emf =
Persistence.createEntityManagerFactory(
"org.jbpm.persistence.jpa" );
Environment env = KnowledgeBaseFactory.newEnvironment();
env.set( EnvironmentName.ENTITY_MANAGER_FACTORY, emf );

// create a new knowledge session that uses JPA to store the
// runtime state
KieSession ksession = JPAKnowledgeService.newKieSession( kbase,
null, env );
int sessionId = ksession.getId();

// invoke methods on your method here
ksession.startProcess( "MyProcess" );
ksession.dispose();

```

- o Définir la session sur la base d'un id de session spécifique

```
// recreate the session from database using the sessionId
ksession = JPAKnowledgeService.loadKieSession(sessionId, kbase,
null, env );
```

2. Configurer la persistance dans le fichier **META-INF/persistence.xml** : configurer JPA pour qu'il utilise Hibernate et la base de données respective.

Les informations sur la façon de configurer la source de données sur votre serveur d'applications doivent être dans la documentation livrée avec le serveur d'applications. Pour cette information pour JBoss Enterprise Application Platform, consulter le *Guide de Configuration et d'Administration* pour ce produit.

Exemple 14.2. Extrait de fichier persistence.xml avec une persistance pour une source de données H2 jdbc/jbpm-ds

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<persistence
  version="1.0"
  xsi:schemaLocation=
    "http://java.sun.com/xml/ns/persistence
    http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd
    http://java.sun.com/xml/ns/persistence/orm
    http://java.sun.com/xml/ns/persistence/orm_1_0.xsd"
  xmlns:orm="http://java.sun.com/xml/ns/persistence/orm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/persistence">
  <persistence-unit name="org.jbpm.persistence.jpa" transaction-
type="JTA">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <jta-data-source>jdbc/jbpm-ds</jta-data-source>
    <mapping-file>META-INF/JBPMorm.xml</mapping-file>
    <class>org.drools.persistence.info.SessionInfo</class>

    <class>org.jbpm.persistence.processinstance.ProcessInstanceInfo</c
lass>
    <class>org.drools.persistence.info.WorkItemInfo</class>
    <properties>
      <property name="hibernate.dialect"
value="org.hibernate.dialect.H2Dialect"/>
      <property name="hibernate.max_fetch_depth" value="3"/>
      <property name="hibernate.hbm2ddl.auto" value="update"/>
      <property name="hibernate.show_sql" value="true"/>
      <property name="hibernate.transaction.manager_lookup_class"

value="org.hibernate.transaction.BTMTransactionManagerLookup"/>
    </properties>
  </persistence-unit>
</persistence>
```

Toutes les invocations sur la session vont déclencher le processus de persistance.

Veillez à ce que la source de données fonctionne bien au démarrage du moteur. Si vous exécutez la

base de données H2 en-mémoire, vous pouvez démarrer la base de données avec votre application par la méthode `JBPMHelper.startH2Server()`; et l'enregistrer dans engine par l'appel de méthode `JBPMHelper.setupDataSource()`;



NOTE

Si vous exécutez JBoss BPM Suite dans un environnement Java simple, votre configuration de source de données ressemblera à ce qui suit :

```
PoolingDataSource ds = new PoolingDataSource();
ds.setUniqueName("jdbc/jbpm-ds");
ds.setClassName("bitronix.tm.resource.jdbc.lrc.LrcXADataSource"
);
ds.setMaxPoolSize(3);
ds.setAllowLocalTransactions(true);
ds.getDriverProperties().put("user", "sa");
ds.getDriverProperties().put("password", "sasa");
ds.getDriverProperties().put("URL",
"jdbc:h2:tcp://localhost/~jbpm-db");
ds.getDriverProperties().put("driverClassName",
"org.h2.Driver");
ds.init();
```

CHAPITRE 15. TRANSACTIONS

15.1. TRANSACTIONS

Le moteur de processus supporte les transactions JTA : les transactions locales ne sont supportées que quand les transactions locales Spring Pure ne le sont pas.

Par défaut, chaque invocation de méthode est considérée comme une transaction. Pour changer ce comportement, par exemple, pour combiner plusieurs commandes en une transaction, vous devrez spécifier les limites de la transaction.

15.2. DÉFINIR DES TRANSACTIONS

Pour définir une transaction, procédez ainsi :

1. Enregistrez le gestionnaire de transactions dans votre environnement.

Exemple 15.1. Codifiez avec l'enregistrement du gestionnaire de transactions

```
// create the entity manager factory
EntityManagerFactory emf =
EntityManagerFactoryManager.get().getOrCreate("org.jbpm.persistence
e.jpa");
TransactionManager tm =
TransactionManagerServices.getTransactionManager();
Environment env = EnvironmentFactory.newEnvironment();
env.set(EnvironmentName.ENTITY_MANAGER_FACTORY, emf);
env.set(EnvironmentName.TRANSACTION_MANAGER, tm);

// setup the runtime environment
RuntimeEnvironment environment =
RuntimeEnvironmentBuilder.Factory.get()
.newDefaultBuilder()
.addAsset(ResourceFactory.newClassPathResource("MyProcessDefinitio
n.bpmn2"), ResourceType.BPMN2)
.addEnvironmentEntry(EnvironmentName.TRANSACTION_MANAGER, tm)

.addEnvironmentEntry(EnvironmentName.PERSISTENCE_CONTEXT_MANAGER,
new JpaProcessPersistenceContextManager(env))

.addEnvironmentEntry(EnvironmentName.TASK_PERSISTENCE_CONTEXT_MANA
GER, new JPATaskPersistenceContextManager(env))
.get();
```

2. Initialisez la KieSession :

```
// get the KieSession
RuntimeManager manager =
RuntimeManagerFactory.Factory.get().newPerProcessInstanceRuntimeMana
ger(environment);
```

```
RuntimeEngine runtime =
manager.getRuntimeEngine(ProcessInstanceIdContext.get());
KieSession ksession = runtime.getKieSession();
```

3. Définir le gestionnaire de transactions dans `jndi.properties`.

Exemple 15.2. Definition du gestionnaire de transactions Bitronix dans `jndi.properties`

```
java.naming.factory.initial=bitronix.tm.jndi.BitronixInitialContextFactory
```

NOTE

Pour utiliser un gestionnaire de transactions différent, modifier `hibernate.transaction.manager_lookup_class`, la propriété du gestionnaire de transactions, dans le fichier `persistence.xml` pour charger votre gestionnaire de transactions.

Exemple 15.3. Le gestionnaire de transactions de JBoss comme gestionnaire de transactions

```
<property
name="hibernate.transaction.manager_lookup_class"
value="org.hibernate.transaction.JBossTransactionManagerLookup"/>
```

4. Définir le début et la fin d'une transaction.

```
// start the transaction
UserTransaction ut =
InitialContext.doLookup("java:comp/UserTransaction");
ut.begin();

// perform multiple commands inside one transaction
ksession.insert( new Person( "John Doe" ) );
ksession.startProcess("MyProcess");

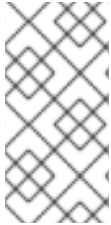
// commit the transaction
ut.commit();
```

15.3. CMT (CONTAINER MANAGED TRANSACTIONS)

Dans les cas où la JBoss BPM Suite est incorporée dans une application qui est dans un conteneur qui permet de gérer les transactions lui-même (Container Managed Transactions - CMT), un gestionnaire de transactions spécialement dédié est fourni grâce à la classe `org.jbpm.persistence.jta.ContainerManagerTransactionManager`. C'est parce que l'implémentation par défaut du gestionnaire de transactions de JBoss BPM Suite est basée sur la

classe `UserTransaction` qui obtient le statut de la transaction. Cependant, certains serveurs d'applications en mode CMT ne permettent pas d'accéder à l'instance de `UserTransaction` à partir du JNDI.

Les opérations exécutées sur ce gestionnaire sont toutes non opérationnelles (NOOP) parce qu'elles ne peuvent pas affecter la CMT sous-jacente. La classe `ContainerManagedTransactionManager` s'attend à ce que la transaction soit toujours active (renvoyant `ACTIVE` à la méthode `getStatus()`).



NOTE

Même si le conteneur gère des transactions, le conteneur doit être informé de toutes les exceptions qui se produisent pendant l'exécution d'instances de processus. Les exceptions levées par le moteur doivent être propagées jusqu'au conteneur pour pouvoir restaurer correctement les transactions.

Configurer le gestionnaire de transactions

Pour pouvoir utiliser le `ContainerManagedTransactionManager`, il doit être inséré dans l'environnement avant que vous puissiez créer ou charger une session :

```
Environment env = EnvironmentFactory.newEnvironment();
env.set(EnvironmentName.ENTITY_MANAGER_FACTORY, emf);
env.set(EnvironmentName.TRANSACTION_MANAGER, new
ContainerManagedTransactionManager());
env.set(EnvironmentName.PERSISTENCE_CONTEXT_MANAGER, new
JpaProcessPersistenceContextManager(env));
```

Ensuite, installer votre fournisseur de service JPA dans votre fichier `persistence.xml`. Par exemple, si vous utilisez WebSphere d'IBM :

```
<property name="hibernate.transaction.factory_class"
value="org.hibernate.transaction.CMTTransactionFactory"/>
<property name="hibernate.transaction.manager_lookup_class"
value="org.hibernate.transaction.WebSphereExtendedJTATransactionLookup"/>
```

Disposer d'une KSession dans une CMT

Dans une CMT, vous ne devriez pas disposer d'une ksession directement (en utilisant la méthode `dispose()`). Ce faisant provoquerait des exceptions à la fin de la transaction car le moteur de processus doit nettoyer l'état quand l'invocation est terminée

Utilisez plutôt la méthode `execute()` de la classe spécialisée `org.jbpm.persistence.jta.ContainerManagedTransactionDisposeCommand`. Cette commande vérifie que la ksession sera supprimée lorsque la transaction sera effectivement terminée.

Cette méthode vérifie si la transaction est active. Si c'est le cas, elle délègue la phase `afterDisposal` de la transaction au lieu de l'exécuter directement. S'il n'y a aucune transaction active, la ksession sera supprimée immédiatement.

CHAPITRE 16. JOURNALISATION

Le mécanisme de journalisation vous permet de stocker des informations sur l'exécution d'une instance de processus. Il est fourni par un listener spécial qui écoute le moteur de processus pour être à l'affut de tout événement pertinent à journaliser, de façon à ce que l'information puisse être stockée séparément de l'information non journalisée dans une base de données intégrée (h2) ou dans une source de données connectée qui utilise JPA ou Hibernate.

Le module jbpm-audit fournit le listener de l'événement et vous permet également de stocker des informations liées aux processus directement dans une base de données en utilisant JPA ou Hibernate. Les données des entités suivantes sont stockées comme suit :

- Instance de processus **processinstance**log
- Instance d'élément **nodeinstance**log
- Instance de variable **variableinstance**log

Tableau 16.1. Champs de la table de ProcessInstanceLog

Champ	Description	Nullable
id	La clé primaire de l'entité de journalisation	Non
end_date	La date de fin de l'instance du processus	Oui
processid	Le nom (id) du processus sous-jacent	Oui
processinstanceid	L'id de l'instance du processus	Non
start_date	La date de démarrage de l'instance du processus	Oui
status	Le statut de l'instance du processus	Oui
parentProcessInstanceid	L'id d'instance de processus de l'instance de processus parent le cas échéant	Oui
outcome	Le résultat de l'instance de processus (comme des information de fin de processus, comme un code d'erreur)	Oui

Tableau 16.2. Champs de la table de NodeInstanceLog

Champ	Description	Nullable
id	La clé primaire de l'entité de journalisation	Non

Champ	Description	Nullable
log_date	La date de l'événement	Oui
nodeid	L'id du noeud de l'élément de processus sous-jacent	Oui
nodeinstanceid	L'id de l'instance de noeud	Oui
nodename	Le nom du noeud sous-jacent	Oui
processid	L'id du processus sous-jacent	Oui
processinstanceid	L'id de l'instance de processus parente	Non
type	Le type d'événement (0 = enter event, 1 = exit event)	Non

Tableau 16.3. Les champs de la table de VariableInstanceLog

Champ	Description	Nullable
id	La clé primaire de l'entité de journalisation	Non
log_date	La date de l'événement	Oui
processid	Le nom (id) du processus sous-jacent	Oui
processinstanceid	L'id de l'instance du processus	Non
value	La valeur de la variable au moment de la journalisation	Oui
variableid	La variable est définie dans la définition du processus	Oui
variableinstanceid	L'id de l'instance de la variable	Oui
outcome	Le résultat de l'instance de processus (comme des information de fin de processus, comme un code d'erreur)	Oui

Si nécessaire, définir votre propre modèle d'informations personnalisé et utiliser les listeners d'événement de processus pour extraire l'information.

16.1. JOURNALISATION DES ÉVÉNEMENTS DANS LA BASE DE DONNÉES

Pour journaliser un événement qui a lieu pendant le runtime d'une instance de processus, une instance d'élément, ou une instance de variable, vous devez procéder ainsi :

1. Mapper les classes de journalisation dans la source de données, de façon à ce que la source de données présentée accepte les entrées de journalisation. Dans Red Hat JBoss EAP, éditez les propriétés de sources de données dans le fichier `persistence.xml`.

Exemple 16.1. Les classes `ProcessInstanceLog`, `NodeInstanceLog` et `VariableInstanceLog` activées pour `processInstanceDS`

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<persistence version="1.0" xsi:schemaLocation=
    "http://java.sun.com/xml/ns/persistence
    http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd
    http://java.sun.com/xml/ns/persistence/orm
    http://java.sun.com/xml/ns/persistence/orm_1_0.xsd"
    xmlns:orm="http://java.sun.com/xml/ns/persistence/orm"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/persistence">

    <persistence-unit name="org.jbpm.persistence.jpa">
        <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <jta-data-source>jdbc/processInstanceDS</jta-data-source>
        <class>org.drools.persistence.info.SessionInfo</class>

        <class>org.jbpm.persistence.processinstance.ProcessInstanceInfo</c
        lass>
        <class>org.drools.persistence.info.WorkItemInfo</class>
        <class>org.jbpm.process.audit.ProcessInstanceLog</class>
        <class>org.jbpm.process.audit.NodeInstanceLog</class>
        <class>org.jbpm.process.audit.VariableInstanceLog</class>

        <properties>
            <property name="hibernate.dialect"
value="org.hibernate.dialect.H2Dialect"/>
            <property name="hibernate.max_fetch_depth" value="3"/>
            <property name="hibernate.hbm2ddl.auto" value="update"/>
            <property name="hibernate.show_sql" value="true"/>
            <property name="hibernate.transaction.manager_lookup_class"
value="org.hibernate.transaction.BTMTransactionManagerLookup"/>
        </properties>
    </persistence-unit>
</persistence>
```

2. Enregistrer un logger dans votre session Kie.

Exemple 16.2. Importer les loggers

```
import org.jbpm.process.audit.AuditLogService;
```

```
import org.jbpm.process.audit.AuditLoggerFactory;
import org.jbpm.process.audit.AuditLoggerFactory.Type;
import org.jbpm.process.audit.JPAAuditLogService;
...
```

Exemple 16.3. Enregistrer un logger dans une session Kie

```
@PersistenceUnit(unitName = PERSISTENCE_UNIT_NAME)
private EntityManagerFactory emf;

private AuditLogService auditLogService;
@PostConstruct
public void configure() {

    auditLogService = new JPAAuditLogService(emf);
    ((JPAAuditLogService)
auditLogService).setPersistenceUnitName(PERSISTENCE_UNIT_NAME);

    if( emf == null ) {
        ((JPAAuditLogService)
auditLogService).setPersistenceUnitName(PERSISTENCE_UNIT_NAME);
    }

    RuntimeEngine runtime =
singletonManager.getRuntimeEngine(EmptyContext.get());
    KieSession ksession = runtime.getKieSession();
    AuditLoggerFactory.newInstance(Type.JPA, ksession, null);

}
```

3. En option, appeler la méthode **addFilter** sur le logger pour filtrer les informations inutiles. Seules les informations acceptées par tous les filtres apparaissent dans la base de données.
4. Les classes de logger peuvent être visualisées dans la vue d'auditing :

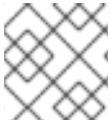
```
<dependency>
  <groupId>org.jbpm</groupId>
  <artifactId>jbpm-audit</artifactId>
  <version>6.0.1.Final</version>
</dependency>
```

16.2. FONCTIONNALITÉ LOGBACK

Red Hat JBoss BPM Suite fournit la fonctionnalité **logback** pour la configuration de la journalisation.

En conséquence, tout ce qui est configuré est consigné dans *Simple Logging Facade for Java* [SLF4J](#), qui délègue à n'importe quel journal Logback, Apache Commons Logging, Log4j ou java.util.logging. Ajouter une dépendance à l'adaptateur de journalisation pour votre infrastructure de journalisation de choix. Si vous n'utilisez pas de framework de journalisation pour l'instant, vous pouvez utiliser Logback en ajoutant cette dépendance Maven :

```
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>1.x</version>
</dependency>
```

**NOTE**

`slf4j-nop` and `slf4j-simple` se prêtent bien à un environnement léger.

16.3. CONFIGURER LA JOURNALISATION

Pour configurer le niveau de configuration des packages, créer un fichier `logback.xml` dans `business-central.war/WEB-INF/classes/logback.xml`. Pour définir le niveau de journalisation du package `org.drools` à "debug" pour une journalisation verbeuse, vous devez ajouter la ligne suivante au fichier :

```
<configuration>

  <logger name="org.drools" level="debug"/>

  ...

</configuration>
```

De même, vous pouvez configurer la journalisation pour des packages comme ceux qui suivent :

- `org.guvnor`
- `org.jbpm`
- `org.kie`
- `org.slf4j`
- `org.dashbuilder`
- `org.uberfire`
- `org.errai`
- etc...

Si vous configurez avec `log4j`, le `log4j.xml` peut être situé dans le fichier `business-central.war/WEB-INF/classes/log4j.xml` et peut être configuré de la manière suivante :

```
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <category name="org.drools">
    <priority value="debug" />
  </category>
```

...

`</log4j:configuration>`**NOTE**

De plus, la journalisation peut être configurée dans le conteneur individuel. Pour configurer la journalisation dans JBoss Enterprise Application Platform, veuillez vous référer au Guide de configuration et d'administration de Red Hat JBoss EAP.

16.4. GÉRER DES FICHIERS JOURNAUX

Red Hat JBoss BPM Suite gère l'ensemble de la maintenance requise. Les données nettoyées automatiquement sont les suivantes :

- Les données d'instances de processus qui sont retirées quand les instances de processus sont terminées.
- Les éléments de travail qui sont retirés quand les éléments de travail sont terminés.
- Les données d'instances de tâches qui sont retirées quand un processus auquel une tâche donnée appartient, se termine.

Données de runtime qui ne sont pas nettoyées automatiquement, comme les données d'informations de sessions. Cela dépend de la stratégie de runtime sélectionnée :

- La stratégie singleton veille à ce que les données de runtime des informations de session ne soient pas retirées automatiquement.
- La stratégie Par requête permet un retrait automatique quand une demande particulière se termine.
- Les instances Par Processus seront retirées automatiquement quand une instance de processus correspondant à une session données se termine ou est abandonnée.

Red Hat JBoss BPM Suite ne supprime pas les requêtes d'exécuteur ou les informations d'erreurs.

Afin de ne pas perdre la trace des instances de processus, Red Hat JBoss BPM Suite propose des tableaux de données d'audit. Ils sont utilisés par défaut et ils gardent la trace de l'environnement de BPM Suite. JBoss BPM Suite a deux façons de gérer et de maintenir les tableaux de données d'audit :

- Le nettoyage automatique
- Le nettoyage manuel

16.4.1. Le nettoyage automatique

Le nettoyage automatique utilise la commande d'exécuteur **LogCleanupCommand**, qui se compose de la logique de nettoyage de toutes ou de certaines données automatiquement. Un avantage de la méthode de nettoyage automatique est la possibilité de planifier des nettoyages répétés à l'aide de la fonctionnalité de tâches récurrentes de l'exécuteur de JBoss BPM Suite. Cela signifie que lorsqu'une tâche est terminée, il fournit des informations à l'exécuteur de JBoss BPM Suite lorsque l'instance suivante doit être exécutée. Par défaut, **LogCleanupCommand** exécute une fois par jour, mais peut être modifiée pour fonctionner sur des intervalles différents.

Il y a plusieurs options de configuration importantes qui peuvent être utilisées avec la commande **LogCleanupCommand** :

Tableau 16.4. Tableau de paramètres de LogCleanupCommand

Nom	Description	Is Exclusive
SkipProcessLog	Indique si le nettoyage d'instances de processus et le nettoyage d'instances de noeuds ou de journalisation de variables doivent être mois (valeur par défaut : false)	Non, peut être utilisé avec d'autres paramètres
SkipTaskLog	Indique si le nettoyage des audits de tâches et des journalisations d'événements de tâches doit être omis (valeur par défaut : false)	Non, peut être utilisé avec d'autres paramètres
SkipExecutorLog	Indique si le nettoyage des entrées d'exécuteur de JBoss BPM Suite doit être omis (valeur par défaut : false)	Non, peut être utilisé avec d'autres paramètres
SingleRun	Indique si la routine de la tâche doit exécuter une fois seulement (valeur par défaut : false)	Non, peut être utilisé avec d'autres paramètres
NextRun	Définit une date pour la prochaine exécution. Par exemple, 12h pour les tâches à exécuter toutes les 12 heures. Si l'option n'est pas donnée, la prochaine tâche exécutera 24 heures après la tâche en cours.	Oui, ne peut pas être utilisé quand le paramètre OlderThanPeriod est utilisé
OlderThan	Supprime les journaux antérieurs à la date donnée. Le format de la date est YYYY-MM-DD. Normalement, ce paramètre est utilisé pour les tâches d'exécution uniques.	En effet, ne peut pas être utilisé quand le paramètre NextRun est utilisé
OlderThanPeriod	Supprime les journaux antérieurs à l'expression donnée sur le minuteur. Par exemple, 30d pour retirer les journaux antérieurs à 30 jours par rapport à la date d'aujourd'hui.	Non, peut être utilisé avec d'autres paramètres
ForProcess	Indique l'ID de définition de processus avec lequel les journaux doivent être retirés	Non, peut être utilisé avec d'autres paramètres
ForDeployment	Indique l'ID de déploiement avec lequel les journaux doivent être retirés	Non, peut être utilisé avec d'autres paramètres
EmfName	Nom d'unité persistant qui doit être utilisé pour une opération de suppression	N/A



NOTE

LogCleanupCommand ne retire pas les instances actives, comme les instances de processus en cours, les instances de tâches ou les tâches d'exécuteur.



AVERTISSEMENT

Bien que les tableaux d'audit ont tous une date et une heure, certains peuvent avoir quelques paramètres manquants, comme l'id de processus, ou d'id de déploiement. De ce fait, il est conseillé d'utiliser le paramètre de date quand on gère la tâche routinière de nettoyage.

16.4.2. Définir une tâche de nettoyage automatique

Pour définir une tâche de nettoyage automatique, procédez ainsi :

1. Ouvrir Business Central dans votre navigateur web (en cas d'exécution locale `http://localhost:8080/business-central`) et connectez-vous en tant qu'utilisateur avec le rôle **admin**.
2. Aller dans **Déploiement** → **Tâches**.
3. Cliquer sur  dans le coin supérieur à droite de la page.
4. Saisir un nom, une date et une heure d'échéance. Saisir ce qui suit dans le champ **Type** :

```
org.jbpm.executor.commands.LogCleanupCommand
```
5. Cliquez sur **Ajouter un paramètre** si vous souhaitez utiliser les paramètres listés ci-dessus. Dans la section clé, saisir un nom de paramètre. Dans la section valeur, saisir `true` ou `false`, selon le résultat escompté.
6. Cliquer sur **Créer** pour en terminer avec l'assistant de création de tâche. Vous aurez maintenant créé une tâche de nettoyage automatique.

16.4.3. Le nettoyage manuel

Vous pouvez utiliser l'API d'audit pour nettoyer manuellement avec plus de contrôle sur les paramètres, et donc, plus de contrôle sur ce qui doit être retiré. L'API d'audit est divisé en sections suivantes :

- Audit de processus, pour le nettoyage des processus, neuds et journaux de variables, accessible à partir du module **jbpm-audit**
- Audit de tâches, pour le nettoyage des tâches et événements de tâches, accessible à partir du module **jbpm-human-task-audit**
- Tâches d'exécuteur, pour le nettoyage des erreurs et des tâches d'exécuteur, accessible à partir du module **jbpm-executor**

Les modules sont ordonnancés de façon hiérarchique, et sont accessibles ainsi :

- ```
org.jbpm.process.audit.JPAAuditLogService
```

- `org.jbpm.services.task.audit.service.TaskJPAAuditService`
- `org.jbpm.executor.impl.jpa.ExecutorJPAAuditService`

Voici plusieurs exemples de nettoyage de manuels :

#### Exemple 16.4. Retrait de journaux d'instances de processus terminés

```
import org.jbpm.process.audit.JPAAuditLogService;
import
org.kie.internal.runtime.manager.audit.query.ProcessInstanceLogDeleteBui
lder;
import org.kie.api.runtime.process.ProcessInstance;

JPAAuditLogService auditService = new JPAAuditLogService(emf);
ProcessInstanceLogDeleteBuilder updateBuilder =
auditService.processInstanceLogDelete().status(ProcessInstance.STATE_COM
PLETED);
int result = updateBuilder.build().execute();
```

#### Exemple 16.5. Retrait de journaux d'audit de tâches pour le déploiement org.jbpm:HR:1.0

```
import org.jbpm.services.task.audit.service.TaskJPAAuditService;
import org.kie.internal.task.query.AuditTaskDeleteBuilder;

TaskJPAAuditService auditService = new TaskJPAAuditService(emf);
AuditTaskDeleteBuilder updateBuilder =
auditService.auditTaskDelete().deploymentId("org.jbpm:HR:1.0");
int result = updateBuilder.build().execute();
```

#### Exemple 16.6. Retrait d'erreurs de requêtes d'exécuteur

```
import org.jbpm.executor.impl.jpa.ExecutorJPAAuditService;
import
org.kie.internal.runtime.manager.audit.query.ErrorInfoDeleteBuilder;
import
org.kie.internal.runtime.manager.audit.query.RequestInfoLogDeleteBuilder
;

ExecutorJPAAuditService auditService = new ExecutorJPAAuditService(emf);
ErrorInfoDeleteBuilder updateBuilder =
auditService.errorInfoLogDeleteBuilder().dateRangeEnd(new Date());
int result = updateBuilder.build().execute();

RequestInfoLogDeleteBuilder updateBuilder2 =
auditService.requestInfoLogDeleteBuilder().dateRangeEnd(new Date());
result = updateBuilder.build().execute();
```

**NOTE**

Lorsque vous retirez des entrées d'exécuteur, veillez à ce que les informations d'erreurs soient retirées avant les informations de requêtes, à cause du dispositif de contraintes défini dans la base de données.

**AVERTISSEMENT**

Parties du code qui utilise l'API interne. Malgré que ceci n'ait aucun impact direct sur la fonctionnalité de notre produit, l'API interne est susceptible de changer et Red Hat ne garantit pas la compatibilité rétroactive.

## CHAPITRE 17. LOCALISATION ET PERSONNALISATION

### 17.1. LANGUES DISPONIBLES

L'interface utilisateur web Red Hat JBoss BPM Suite peut être vue en plusieurs langues :

- Anglais États-Unis (en\_US)
- Espagnol (es\_ES)
- Japonais (ja\_JP)
- Chinois (zh\_CN)
- Portuguais (pt\_BR)
- Français (fr\_CA)
- Allemand (de\_DE)



#### NOTE

Si aucune langue n'est indiquée, l'anglais US est utilisé par défaut.

### 17.2. MODIFIER LES PARAMÈTRES DE LANGUE

#### Modifier la langue dans l'interface utilisateur de Business Central

Par défaut, Business Central utilise le paramètre régional du système. Si vous souhaitez le changer, ajouter le code correspondant au paramètre régional dans l'URL de Business Central. Par exemple, l'URL suivant indiquera le paramètre régional portugais (pt\_BR).

`http://localhost:8080/business-central/?locale=pt_BR`

#### Modifier la langue dans l'interface utilisateur de Dashbuilder

Pour modifier la langue dans l'interface utilisateur de Dashbuilder, procéder ainsi :

1. Connecter-vous à Dashbuilder une fois que le serveur a démarré en naviguant dans <http://localhost:8080/dashbuilder> par un navigateur web.
2. Sélectionner la langue de votre choix en cliquant sur les paramètres régionaux possibles en haut du centre de l'interface utilisateur de Dashbuilder pour modifier la langue.

#### Définir une langue d'interface utilisateur par défaut dans Dashbuilder

Voici un exemple pour définir la langue d'interface utilisateur par défaut dans Dashbuilder :

##### Procédure 17.1. Définir la langue par défaut Français

1. Naviguer dans `jboss-eap-6.4/standalone/configuration` et définir ce qui suit dans le fichier `standalone.xml`.

```
<system-properties>
 <property
 name="org.jboss.dashboard.LocaleManager.installedLocaleIds"
```

```
value="en, es, de, fr, ja, pt, zh"/>
 <property
name="org.jboss.dashboard.LocaleManager.defaultLocaleId"
value="fr"/>
</system-properties>
```

2. La langue d'interface utilisateur par défaut de Dashbuilder est maintenant le français.

## Définir les paramètres régionaux installés dans Dashbuilder

Voici un exemple sur la façon de définir les paramètres régionaux dans Dashbuilder :

### Procédure 17.2. Définir le paramètre régional installé

- Naviguer dans `jboss-eap-6.4/standalone/configuration` et définir ce qui suit dans le fichier `standalone.xml`.

```
<system-properties>
 <property
name="org.jboss.dashboard.LocaleManager.installedLocaleIds"
value="en, es, de, fr, ja, pt"/>
 <property
name="org.jboss.dashboard.LocaleManager.defaultLocaleId"
value="fr"/>
</system-properties>
```

Dans cet exemple, la langue chinoise (zh) a été supprimée de la liste des paramètres régionaux installés pour les utilisateurs, donc les utilisateurs ne seront pas en mesure de passer au chinois dans Dashbuilder. Dashbuilder affichera le contenu en Français, qui est le paramètre régional par défaut. Les utilisateurs seront en mesure de sélectionner d'autres langues définies (fr, es, de, ja, pt) dans ce fichier.



#### NOTE

Dans Business Central, le serveur d'applications n'a pas besoin d'être démarré à nouveau après le changement de paramètre régional si vous avez déjà ajouté ce paramètre à l'URL de Business Central. Cependant, dans Dashbuilder, le serveur d'applications doit être démarré à nouveau après que les fichiers de configuration ont été modifiés.

## 17.3. EXÉCUTER UNE JVM AVEC LA CODIFICATION UTF-8

Red Hat JBoss BPM Suite a été conçu pour fonctionner avec un codage UTF-8. Si un système de codage différent est utilisé par une JVM sous-jacente, des erreurs inattendues peuvent se produire.

Pour vous assurer que UTF-8 est utilisé par la JVM, utilisez la propriété de système suivante "-Dfile.encoding=UTF-8"

## PARTIE IV. EXÉCUTION

## CHAPITRE 18. CONFIGURATION DU SERVEUR D'EXÉCUTION DES PROCESSUS

### 18.1. RÈGLES D'ATTRIBUTION

Les règles d'attribution sont des règles exécutées automatiquement lorsqu'une tâche humaine est créée ou terminée. Ce mécanisme peut être utilisé, par exemple, pour affecter une tâche humaine automatiquement à un utilisateur particulier appartenant à un groupe ou pour empêcher un utilisateur d'effectuer une tâche si des données sont manquantes.

#### 18.1.1. Définir les règles d'attribution

Pour définir les règles d'attribution, procéder ainsi :

1. Créer un fichier qui contienne la définition de la règle sur le chemin d'accès de Business Central (l'emplacement recommandé est `$DEPLOY_DIR/standalone/deployments/business-central.war/WEB-INF/classes/`):
  - o `default-add-task.drl` avec les règles à vérifier quand la tâche humaine est créée.
  - o `default-complete-task.drl` avec les règles à vérifier quand la tâche humaine est terminée.
2. Pour définir les règles dans le fichier.

#### Exemple 18.1. Le contenu de `default-add-task.drl`

```
package defaultPackage

import org.kie.api.task.model.Task;
import org.kie.api.task.model.User;
import org.kie.api.task.model.Status;
import org.kie.api.task.model.PeopleAssignments;
import org.jbpm.services.task.rule.TaskServiceRequest;
import org.jbpm.services.task.exception.PermissionDeniedException;
import org.jbpm.services.task.impl.model.*;
import java.util.HashMap;
import java.util.List;

global TaskServiceRequest request;

rule "Don't allow Mary to complete task when rejected"
when
 $task : Task()
 $actualOwner : User(id == 'mary') from
 $task.getTaskData().getActualOwner()
 $params : HashMap(this["approved"] == false)
then
 request.setAllowed(false);
 request.setExceptionClass(PermissionDeniedException.class);
```

```

 request.addReason("Mary is not allowed to complete task with
approved false");
 end

```

Si les propriétaires potentiels d'une tâche humaine contiennent l'utilisateur **Mary**, la tâche sera assignée automatiquement à l'utilisateur **mary**.

### Exemple 18.2. Le contenu de `default-complete-task.drl`

```

package defaultPackage

import org.kie.api.task.model.Task;
import org.kie.api.task.model.User;
import org.kie.api.task.model.Status;
import org.kie.api.task.model.PeopleAssignments;
import org.jbpm.services.task.rule.TaskServiceRequest;
import org.jbpm.services.task.exception.PermissionDeniedException;
import org.jbpm.services.task.impl.model.*;
import java.util.HashMap;
import java.util.List;

global TaskServiceRequest request;

rule "Don't allow Mary to complete task when rejected"
 when
 $task : Task()
 $actualOwner : User(id == 'mary') from
 $task.getTaskData().getActualOwner()
 $params : HashMap(this["approved"] == false)
 then
 request.setAllowed(false);
 request.setExceptionClass(PermissionDeniedException.class);
 request.addReason("Mary is not allowed to complete task without
approval.");
 end

```

Si les propriétaires potentiels d'une tâche humaine contiennent l'utilisateur **Mary**, la tâche sera assignée automatiquement à l'utilisateur **mary**.

## 18.2. SESSION MAIL

Une session mail définit les propriétés de serveur mail utilisées pour envoyer des emails quand l'application l'exige, comme les mécanismes de notification ou d'escalation (voir le *Red Hat JBoss BPM Suite User Guide*).

### 18.2.1. Mise en place d'une session mail

Pour mettre en place une session mail pour votre moteur d'exécution, procédez ainsi :

1. Ouvrir le fichier de configuration du profil respectif (`standalone.xml` ou `host.xml`) pour effectuer les modifications.



## 2. Ajouter la session mail au sous-système `urn:jboss:domain:mail:1.1`.

### Exemple 18.3. Nouvelle session mail sur le localhost

```
<subsystem xmlns="urn:jboss:domain:mail:1.1">
 <!-- omitted code -->

 <mail-session jndi-name="mail/jbpmMailSession" debug="true"
from="bpms@company.com">
 <smtp-server outbound-socket-binding-ref="bpmsMail"/>
 </mail-session>
</subsystem>
```

## 3. Définir un socket sortant dans le fichier de configuration du profil.

### Exemple 18.4. Définition du socket sortant

```
<outbound-socket-binding name="bpmsMail">
 <remote-destination host="localhost" port="12345"/>
</outbound-socket-binding>
```

## **CHAPITRE 19. PLUG-IN DE RED HAT JBOSS DEVELOPER STUDIO**

### **19.1. PLUG-IN**

## PARTIE V. MONITORING

## CHAPITRE 20. MONITORING DES PROCESSUS

### 20.1. RÉSEAU JBOSS OPERATIONS NETWORK

Un plug-in de réseau d'exploitation JBoss Operations Network peut être utilisé pour surveiller les sessions de règles pour Red Hat JBoss. Le plug-in utilise Java Management Extensions (JMX) pour surveiller les sessions de règles.

En raison d'une limitation dans la possibilité de passer des arguments de monitoring JVM par la ligne de commande Maven, tous les paramètres `com.sun.management.jmxremote.*` doivent être passés à l'application JBoss par le fichier de configuration `pom.xml`.

Veuillez consulter **JBoss Operations Network Installation Guide** pour obtenir des instructions d'installation pour le serveur JBoss ON.

### 20.2. INSTALLER LE PLUG-IN JBOSS BRMS DANS JBOSS ON

Le plug-in Red Hat JBoss BRMS de JBoss Operations Network peut être installé, soit en copiant les fichiers JAR du plug-in dans le répertoire du plug-in JBoss Operations Network, soit par le GUI de JBoss Operations Network.

La procédure suivante guide un utilisateur à copier les fichiers JAR du plug-in dans le répertoire du plug-in JBoss Operations Network

#### Procédure 20.1. Copier les fichiers JAR du plug-in JBoss BRMS

1. Extraire l'archive du pack de plug-in JBoss BRMS dans un lieu temporaire. Cela crée un sous-répertoire nommé `jon-plugin-pack-brms-bpms-3.3.0.GA`. Exemple :

```
[root@server rhq-agent]# unzip jon-plugin-pack-brms-bpms-3.3.0.GA.zip -d /tmp
```

2. Copier les fichiers JAR du plug-in de JBoss BRMS du répertoire `jon-plugin-pack-brms-bpms-3.3.2.GA/` dans le répertoire du plug-in du serveur JBoss ON. Exemple :

```
[root@server rhq-agent]# cp /tmp/jon-plugin-pack-brms-bpms-3.3.0.GA/*.jar /opt/jon/jon-server-3.3.0.GA1/plugins
```

3. Démarrer le serveur JBoss Operations Network pour mettre à jour le plug-in de JBoss BRMS.

Pour télécharger le plug-in de JBoss BRMS dans le GUI de JBoss Operations Network, suivre la procédure suivante :

#### Procédure 20.2. Télécharger le plug-in de JBoss BRMS par le GUI

1. Démarrer le serveur de JBoss Operations Network et connectez-vous pour accéder au GUI.
2. Dans la navigation supérieure du GUI, ouvrez le menu **Administration**.
3. Dans la zone **Configuration** sur la gauche, sélectionner le lien **Server Plugins**.
4. Au bas de la liste des plug-ins de serveur téléchargés, cliquer sur le bouton **Télécharger un plugin** et choisissez le plug-in BRMS.

5. Le plug-in JBoss BRMS de JBoss Operations Network est maintenant téléchargé.

## 20.3. CONTRÔLE DES KIEBASES ET DES KIESESSIONS

Pour que JBoss Operations Network puisse contrôler les KieBases et les KieSessions, les MBeans doivent être activés.

Les MBeans peuvent être activés si on leur passe le paramètre :

```
-kie.mbeans = enabled
```

Ou via l' API :

```
KieBaseConfiguration kbconf =
KieServices.Factory.get().newKieBaseConfiguration();
kbconf.setOption(MBeansOption.ENABLED);
```



### NOTE

Les **Kie Services** ont été créés pour JBoss BRMS 6; pour JBoss BRMS 5, **Services Drool** était la convention de nommage utilisée et elle a des mensurations différentes suivant les sessions. Par exemple, le renommage de **activation** → **correspondance** est apparu dans la version mise à jour.

Veuillez consulter le guide *JBoss Operations Network Resource Monitoring and Operations Reference* pour obtenir des informations sur la façon d'importer des Sessions Kie dans la vue Inventaire à des fins de monitoring.

## CHAPITRE 21. GESTION DE LA SÉCURITÉ DANS RED HAT JBOSS BPM SUITE DASHBUILDER

### 21.1. ACCÉDER À RED HAT JBOSS BPM SUITE DASHBUILDER

Dashbuilder est l'interface utilisateur basée web de Red Hat JBoss BPM Suite pour le Business Activity Monitoring. Pour accéder au Dashbuilder à partir de Business Central, aller dans **Tableaux de bord** → **Tableaux de processus et de tâches**.

Le tableau de bord qui s'affiche fournit des statistiques sur les données de runtime sélectionnées à gauche. Vous pouvez créer votre propre tableau de bord dans le Dashbuilder. Pour cela, afficher le Dashbuilder en cliquant sur **Tableaux de bord** → **tableaux de processus**.

### 21.2. GESTION DE LA SÉCURITÉ

Pour gérer la sécurité, vous devez définir des polices d'autorisation personnalisées pour pouvoir octroyer ou refuser un accès à un espace de travail, une page, ou des instances de panneau par rôle.

Vous trouverez ci-dessous une liste des rôles disponibles de Dashbuilder :

- **admin** - administre le système Red Hat JBoss BPM Suite. A tous les droits de procéder aux changements nécessaires. A également la possibilité d'ajouter ou de supprimer des utilisateurs du système.
- **développeur** - implémente le code requis pour faire fonctionner les processus. Utilise surtout la connexion JBDS pour voir les processus, mais peut utiliser l'outil web de temps en temps :
- **analyste** - est responsable de créer ou de concevoir des processus dans le système. Crée des flux de processus et est chargé des demandes de changement de processus. Doit tester les processus qu'ils créent. Crée également des formulaires et des tableaux de bord.
- **utilisateur** - utilisateur quotidien du système qui réalise les tâches d'entreprise requises pour que les processus puissent avancer. Travaille principalement à partir d'une liste de tâches.
- **manager** - observateur du système intéressé par les statistiques autour des processus métier et leur performance, les indicateurs d'entreprise, les autres rapports sur le système, et les personnes en rapport avec le système.

Grâce au système d'autorisations, vous pouvez créer une structure d'espace de travail de plusieurs pages, menus et panneaux, et définir quelles pages et panneaux d'une page seront visibles pour chaque rôle. Vous pouvez également définir des types spéciaux d'utilisateurs et leur donner un accès restreint à certaines fonctionnalités, ou même un accès restreint à un sous-ensemble de page.

### 21.3. PERMISSIONS D'ESPACE DE TRAVAIL

#### Procédure 21.1. Accéder à des permissions d'espace de travail

1. Connectez-vous aux Business Dashboards de Business Central (comme décrit dans le sujet **Accéder à Red Hat JBoss BPM Suite Dashbuilder**)
2. Sélectionner le dashboard qui convient à partir du menu déroulant de l'espace de travail.

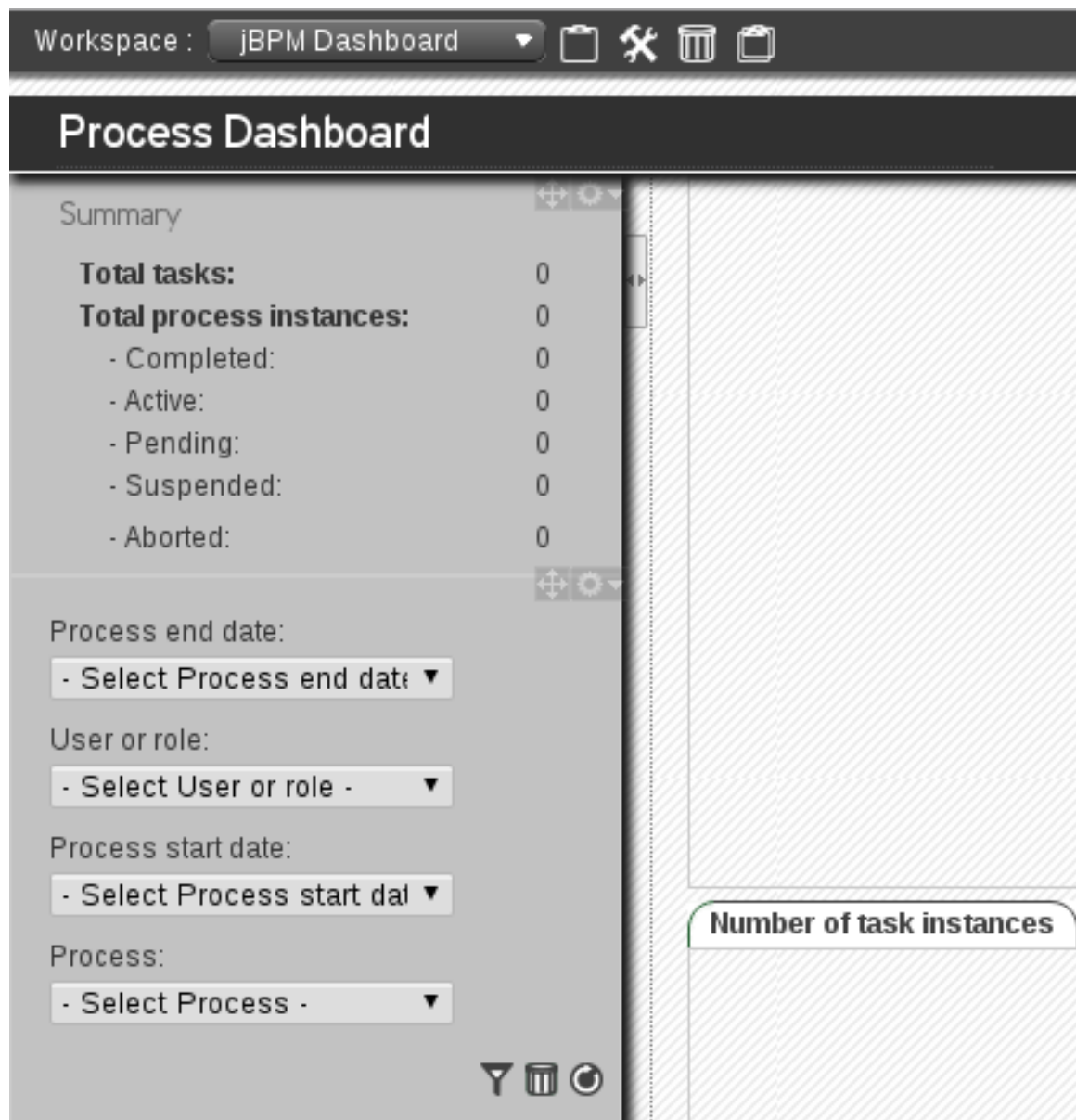


Figure 21.1. Espace de travail de Dashbuilder

3. Cliquer sur le bouton **Modifier les propriétés de l'espace de travail** sélectionné  pour accéder au Tableau de bord d'espace de travail.
4. Cliquer sur l'étiquette **Permissions** pour voir l'écran de gestion des permissions.

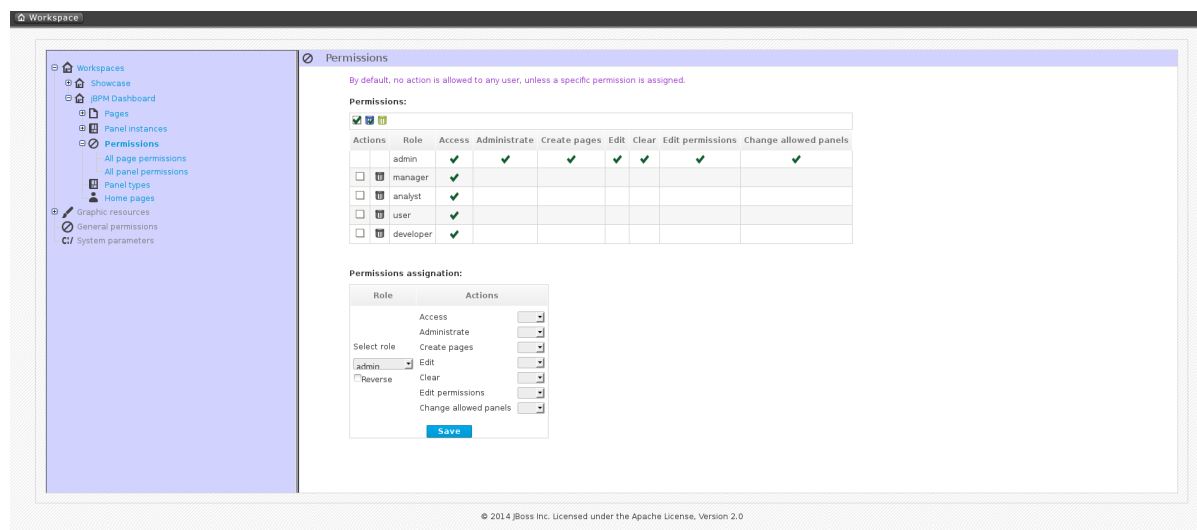


Figure 21.2. Écran des permissions

Dans la section **Allocation de permissions**, vous trouverez une liste des actions autorisées s'appliquant au rôle sélectionné :

- **Accès:** permission de vous connecter à l'application
- **Administration:** permission d'accéder à la barre d'outils et aux fonctionnalités de configuration du système.
- **Créer pages:** possibilité de créer des nouvelles pages de projet.
- **Modifier :** permission de modifier des propriétés d'espace de travail.
- **Supprimer :** possibilité de supprimer l'espace de travail.
- **Modifier permissions :** possibilité d'octroyer/refuser des permissions.
- **Modifier panneaux autorisés :** permission de limiter le type de panneaux qui peuvent être utilisés dans cet espace de travail.

Pour assigner une permission, vous devez sélectionner le rôle cible et la liste des actions autorisées pour une ressource donnée.



**Permissions assignation:**

Role	Actions
Select role  user  <input type="checkbox"/> Reverse	Access  Yes
	Administrate  No
	Create pages  Yes
	Edit  Yes
	Clear  No
	Edit permissions  Yes
	Change allowed panels  Yes
Save	

**Figure 21.3. Attribution des permissions**

- *Rôles Cible (qui)* : quel utilisateur recevra/ne recevra pas les permissions définies.
- *Actions autorisées* : suivant le type de ressource, nous pouvons activer/désactiver ce que l'utilisateur peut faire avec cette ressource.
- *Reverse (optionnel)* : quand on a un groupe de rôles et que l'on veut donner/refuser une permission à tous les rôles sauf un.

**NOTE**

Par défaut, l'ensemble des permissions vont au rôle *admin*. Cela rend les choses plus faciles pour créer un utilisateur qui peut tout faire, tant que le rôle *admin* est attribué.

**21.4. PERMISSIONS DE PAGE**

1. Pour accéder aux **Permissions de page**, chercher le menu déroulant **Pages** qui se trouve sous le jBPM Dashboard (ou un autre tableau de bord que vous aurez sélectionné).
2. Étendre le menu **Pages**, puis l'option **Tableau de bord des processus**.
3. Sélectionner l'option **Permissions de page**.

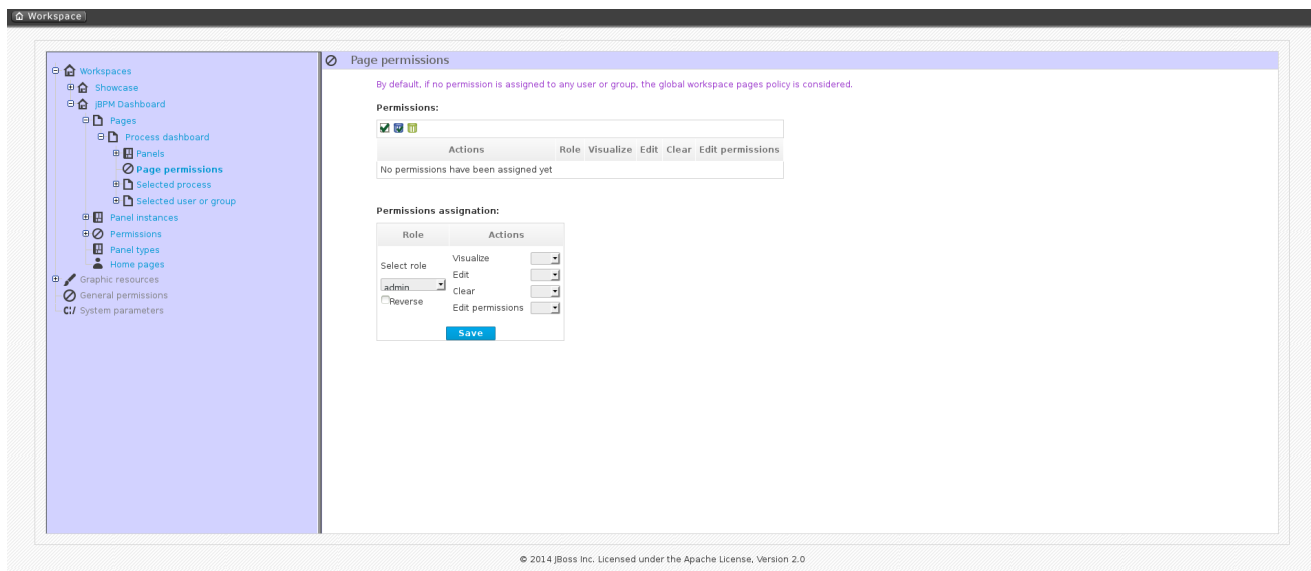


Figure 21.4. Permissions de page

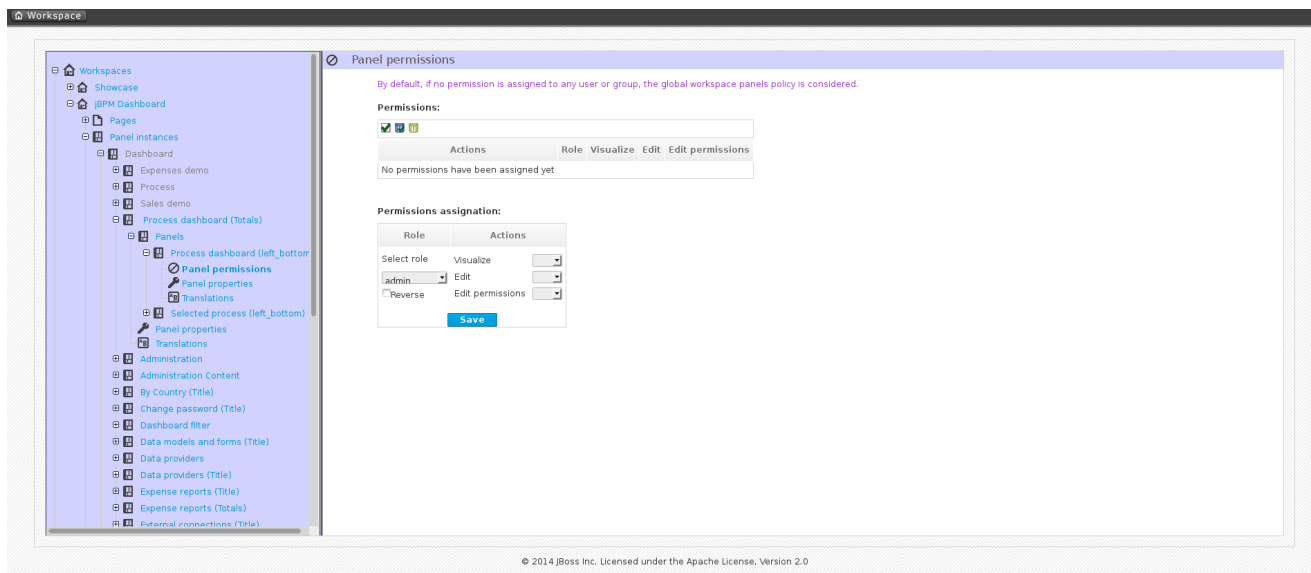
Dans la section **Allocation de permissions**, vous trouverez une liste des actions autorisées s'appliquant au rôle sélectionné :

- **Visualiser** : permission pour rendre la page visible.
- **Modifier** : possibilité de modifier les propriétés d'une page.
- **Supprimer** : possibilité de supprimer la page.
- **Modifier permissions** : possibilité d'octroyer/refuser des permissions pour une page.

## 21.5. PERMISSIONS DE PANNEAU

1. Pour accéder à la page **Permissions de panneau**, étendre l'option **Instances de panneau** sous le **jBPM Dashboard** (ou un autre tableau de bord que vous utilisez).
2. Étendre l'option **Tableau de bord**, puis étendre **Tableau de bord de processus**.
3. Étendre le choix **Panneaux** et sélectionner le processus qui convient.
4. Ouvrir la page **Permissions de panneau**.

Vous trouverez, ci-dessous, un écran de gestion des permissions pour un panneau donné (dans cet exemple, le panneau de processus) :



**Figure 21.5. Écran de configuration des permissions de panneau**

Les actions autorisées sont les suivantes :

- **Visualiser** : pour rendre la page visible.
- **Modifier** : pour modifier des propriétés d'espace de travail.
- **Modifier permissions** : octroyer/refuser des permissions pour une page.

## ANNEXE A. HISTORIQUE DES VERSIONS

<b>Version 6.3.0-6.2</b> Revision 1	<b>Tue Aug 9 2016</b>	<b>Corina Galicher Roe</b>
<b>Version 6.3.0-6.1</b> Fichiers de traduction synchronisés avec les sources XML 6.3.0-6	<b>Fri May 27 2016</b>	<b>Red Hat Localization Services</b>
<b>Version 6.3.0-6</b> Erreurs dans les liens corrigées.	<b>Thu May 5 2016</b>	<b>Tomas Radej</b>
<b>Version 6.3.0-5</b> Créé avec des liens live.	<b>Thu May 5 2016</b>	<b>Tomas Radej</b>
<b>Version 6.3.0-4</b> Liens d'adresses de démarrage	<b>Thu May 5 2016</b>	<b>Tomas Radej</b>
<b>Version 6.3.0-3</b> Tous les livres ont été créés à nouveau.	<b>Thu Apr 28 2016</b>	<b>Tomas Radej</b>
<b>Version 6.3.0-2</b> Tous les livres ont été créés à nouveau.	<b>Thu Apr 28 2016</b>	<b>Tomas Radej</b>
<b>Version 6.3.0-1</b> Version initiale créée pour 6.3.0 JBoss BPM Suite.	<b>Thu Apr 28 2016</b>	<b>Tomas Radej</b>