



Red Hat JBoss BPM Suite 6.2

Guide d'installation

Pour les administrateurs Red Hat JBoss

Red Hat JBoss BPM Suite 6.2 Guide d'installation

Pour les administrateurs Red Hat JBoss

Kanchan Desai
Red Hat Customer Content Services
kadesai@redhat.com

Doug Hoffman
Red Hat Customer Content Services
dhoffman@redhat.com

Eva Kopalova
Red Hat Customer Content Services
ekopalova@redhat.com

B Long
Red Hat Customer Content Services
belong@redhat.com

Petr Penicka
Red Hat Customer Content Services
ppenicka@redhat.com

Gemma Sheldon
Red Hat Customer Content Services
gsheldon@redhat.com

Tomas Radej
Red Hat Customer Content Services
tradej@redhat.com

Notice légale

Copyright © 2015 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

Ce guide fournit aux administrateurs les étapes nécessaires pour installer Red Hat JBoss BPM Suite, les plug-ins pour Red Hat JBoss Developer Studio et explique comment mettre en place les projets cités en exemples.

Table des matières

CHAPITRE 1. INTRODUCTION	3
1.1. À PROPOS DE RED HAT JBOSS BPM SUITE	3
1.2. PLATEFORMES PRISES EN CHARGE	3
1.3. CAS D'UTILISATION : SOLUTIONS BASÉES SUR LES PROCESSUS DANS LE SECTEUR DES PRÊTS	3
PARTIE I. INSTALLATION DE RED HAT JBOSS BPM SUITE	5
CHAPITRE 2. OPTIONS D'INSTALLATION	6
2.1. INSTALLATION DE RED HAT JBOSS BPM SUITE INSTALLER	6
2.2. L'INSTALLATION D'UN GROUPEMENT DÉPLOYABLE STANDARD	23
CHAPITRE 3. CONFIGURATIONS SPÉCIALES	33
3.1. INSTALLATION DE LA PERSISTANCE POUR BUSINESS CENTRAL	33
3.2. INSTALLATION DE LA PERSISTANCE POUR DASHBUILDER	34
3.3. INSTALLATION SPÉCIALE DE LA BASE DE DONNÉES IBM DB2	35
CHAPITRE 4. RÔLES ET UTILISATEURS	36
4.1. DÉFINIR LES RÔLES	36
4.2. CRÉATION D'UTILISATEURS	36
CHAPITRE 5. TEST DE L'INSTALLATION	38
5.1. DÉMARRAGE DU SERVEUR	38
5.2. JAVA SECURITY MANAGER ET GESTION DE PERFORMANCE	40
5.3. CONNECTEZ-VOUS À BUSINESS CENTRAL	40
CHAPITRE 6. MISE EN CLUSTER	41
MÉCANISME DE MISE EN CLUSTER DU RÉFÉRENTIEL GIT	41
MISE EN CLUSTER DES RÉFÉRENTIELS MAVEN	42
6.1. MISE EN CLUSTER AVEC JBOSS EAP	43
6.2. MISE EN CLUSTER D'UN ENSEMBLE GÉNÉRIQUE	52
CHAPITRE 7. RÉFÉRENTIELS MAVEN	57
7.1. À PROPOS DE MAVEN	57
7.2. À PROPOS DES RÉFÉRENTIELS MAVEN FOURNIS	57
7.3. CONFIGURER MAVEN POUR UTILISER LES RÉFÉRENTIELS DE SYSTÈME DE FICHIER	58
7.4. CONFIGURER MAVEN POUR UTILISER LES RÉFÉRENTIELS EN LIGNE	61
7.5. GESTION DES DÉPENDANCES	65
CHAPITRE 8. RED HAT JBOSS DEVELOPER STUDIO	66
8.1. RED HAT JBOSS DEVELOPER STUDIO	66
8.2. INSTALLATION DES PLUG-INS D'UN JBOSS DEVELOPER STUDIO	66
8.3. DÉFINISSEZ LE RUNTIME DE DROOLS	66
8.4. CONFIGURATION DU SERVEUR JBOSS BPM SUITE	67
8.5. IMPORTATION DE PROJETS À PARTIR D'UN RÉFÉRENTIEL GIT DANS JBOSS DEVELOPER STUDIO	67
CHAPITRE 9. CORRECTIFS ET MISE À NIVEAU DE RED HAT JBOSS BPM SUITE	72
9.1. CORRECTIFS ET MISES À NIVEAU	72
9.2. APPLIQUER DES CORRECTIFS DANS RED HAT JBOSS BPM SUITE 6.2.	72
9.3. CORRIGER D'AUTRES PLATEFORMES ET APPLICATIONS	74
ANNEXE A. HISTORIQUE DES VERSIONS	76

CHAPITRE 1. INTRODUCTION

1.1. À PROPOS DE RED HAT JBOSS BPM SUITE

Red Hat JBoss BPM Suite est une suite de gestion de processus d'entreprise open source qui combine la gestion des Business Process et la gestion des règles métier. Elle permet aux utilisateurs IT d'entreprises de créer, gérer, valider et déployer des règles et procédés d'entreprise.

Red Hat JBoss BRMS et Red Hat JBoss BPM Suite utilise un répertoire centralisé, où toutes les ressources sont stockées. Cela garantit la cohérence, la transparence et la capacité d'audit au sein de l'entreprise. Les utilisateurs professionnels peuvent modifier la logique métier et les processus d'entreprise sans avoir recours au service informatique.

Pour accommoder le composant Business Rules, Red Hat JBoss BPM Suite inclut Red Hat JBoss BRMS intégré.

Business Resource Planner est inclus dans cette version.

Red Hat JBoss BPM Suite peut être utilisé avec Red Hat Enterprise Linux 7 (RHEL7) .

1.2. PLATEFORMES PRISES EN CHARGE

Red Hat JBoss BPM Suite et Red Hat JBoss BRMS sont pris en charge sur les conteneurs suivants :

- Red Hat JBoss Enterprise Application Platform 6.4
- Red Hat JBoss Web Server 2.1 (Tomcat 7) on JDK 1.7
- IBM WebSphere Application Server 8.5.5.0
- Oracle WebLogic Server 12.1.3 (12c)

1.3. CAS D'UTILISATION : SOLUTIONS BASÉES SUR LES PROCESSUS DANS LE SECTEUR DES PRÊTS

Cette section décrit un cas de déploiement de JBoss BPM Suite pour automatiser les processus métier auprès d'une banque (tels que les processus d'approbation des prêts). Ce cas d'utilisation est un déploiement classique axé sur les processus spécifiques qui pourrait être la première étape dans le cadre d'une plus large adoption de JBoss BPM Suite dans une entreprise. Elle s'appuie à la fois sur les fonctionnalités de processus et de règles métier de JBoss BPM Suite.

Une banque de détail propose plusieurs types de produits de prêt avec des modalités et conditions d'admissibilité qui varient. Les clients ayant besoin d'un prêt doivent déposer une demande de prêt auprès de la banque, qui traite ensuite l'application en plusieurs étapes, en vérifiant l'admissibilité, définissant les termes du contrat, recherchant toute activité frauduleuse et déterminant le produit de prêt le plus approprié. Une fois approuvé, la banque crée et finance un compte de prêt pour le déposant, qui peut alors avoir accès aux fonds. La banque doit s'assurer de se conformer à toutes les réglementations bancaires à chaque étape du processus et elle doit gérer son portefeuille de prêts en vue de maximiser la rentabilité. Des politiques sont en place pour aider à la prise de décisions à chaque étape, et ces politiques sont gérées activement afin d'optimiser les résultats pour la banque.

Les analystes d'affaires de la banque modélisent les processus de demande de prêt en utilisant les outils de création de BPMN2 (concepteur de processus) dans JBoss BPM Suite. Voici le processus :

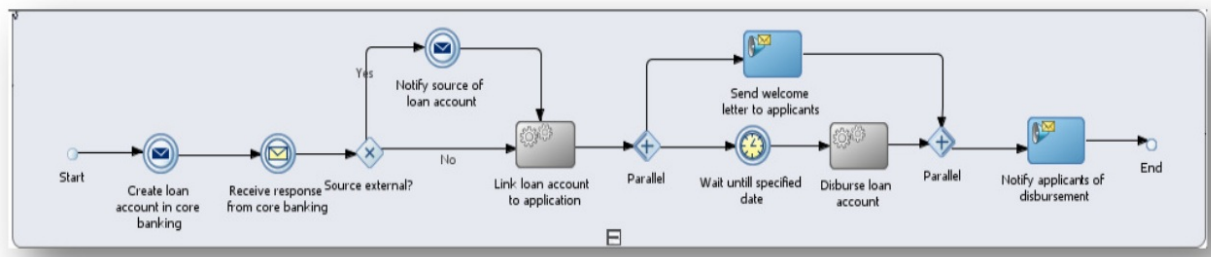


Figure 1.1. Flux de processus d'application de prêt de haut niveau

Des règles de gestion (Business Rules) sont développées avec les outils de création de JBoss BPM Suite afin d'appliquer des politiques et prendre des décisions. Les règles sont liées aux modèles de processus pour faire appliquer les politiques qui conviennent à chaque étape du processus.

L'organisation IT de la banque déploie JBoss BPM Suite pour que le processus d'application de la demande de prêt soit totalement automatisée.

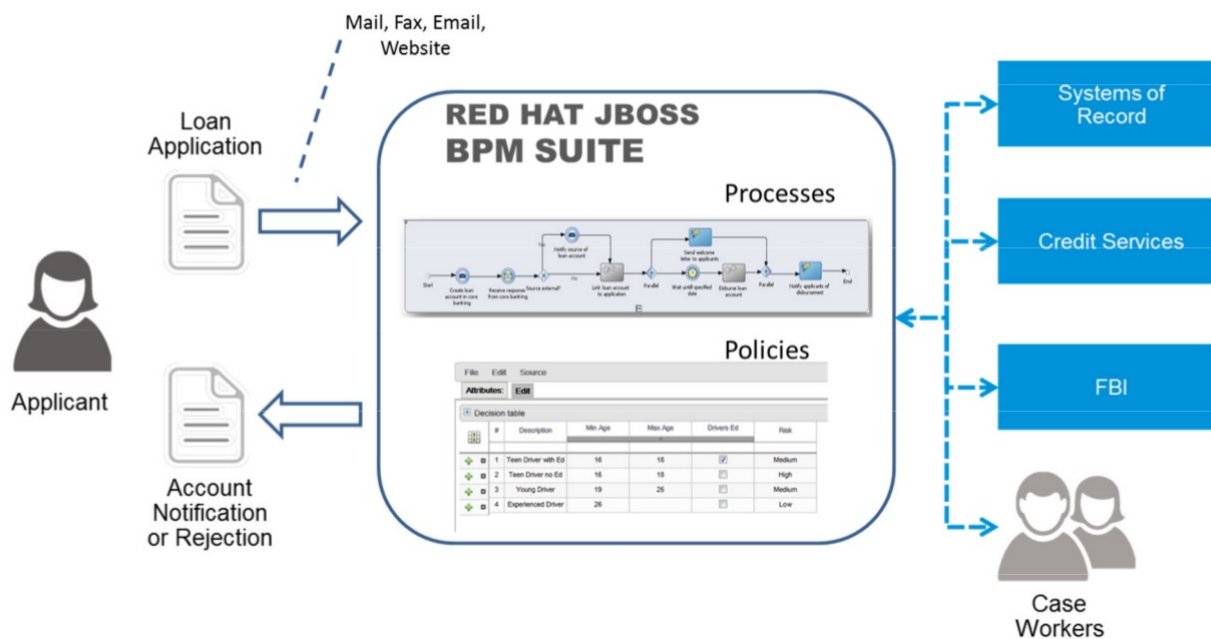


Figure 1.2. Automatisation du processus de demande de prêts

Le processus de prêt en totalité et ses règles sont modifiables à tout moment par les analystes de la banque. La banque est en mesure de s'adapter à tout instant à l'évolution de la réglementation et est capable de rapidement vous présenter des nouveautés en matière de prêt et d'améliorer les politiques de prêt, afin de rester dans la course et générer des profits.

PARTIE I. INSTALLATION DE RED HAT JBOSS BPM SUITE

CHAPITRE 2. OPTIONS D'INSTALLATION

Red Hat JBoss BPM Suite est disponible sous deux versions :

- Programme d'installation jar exécutable pour installation sur Red Hat JBoss Enterprise Application Platform (EAP) 6.4.
- Installation de fichier zip disponible en deux versions :
 - **jboss-bpmsuite-6.2-deployable-eap6.x.zip** : version adaptée pour les déploiements sur Red Hat JBoss Enterprise Application Platform (EAP 6.4).
 - **jboss-bpmsuite-6.2-deployable-generic.zip** : version déployable avec bibliothèques supplémentaires adaptée aux déploiements sur Red Hat JBoss Web Server (EWS), Apache Tomcat 6 et Apache Tomcat 7.

Selon votre environnement, vous pouvez choisir l'option d'installation qui correspond le mieux à vos projets.



NOTE

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**".



IMPORTANT

À partir de JBoss BPM Suite 6.1 et versions ultérieures, vous devez posséder JBoss EAP 6.4 ou version ultérieure déjà installée avant de tenter une installation de JBoss BPM Suite.

2.1. INSTALLATION DE RED HAT JBOSS BPM SUITE INSTALLER

Cette section décrit les étapes requises pour installer Red Hat JBoss BPM Suite à l'aide de la méthode d'installation du programme d'installation du fichier jar. Le fichier jar est un fichier exécutable qui installe JBoss BPM Suite sur une installation JBoss EAP 6 existante.

À partir de JBoss BPM Suite 6.1 et versions suivantes, vous devez avoir JBoss EAP 6.4 ou version ultérieure déjà installée afin d'installer JBoss BPM Suite.



NOTE

IBM JDK ne pouvant pas utiliser de keystores générés sur d'autres JDK, il n'est pas possible d'installer JBoss BPM Suite sur un JBoss EAP existant exécuté sur IBM JDK avec un keystore généré sur un autre JDK.

2.1.1. Installation de Red Hat JBoss BPM Suite à l'aide du programme d'installation

Le programme d'installation pour Red Hat JBoss BPM Suite est un fichier jar Java exécutable. Vous pouvez l'utiliser pour installer JBoss BPM Suite sur une installation EAP 6.4 existante.

**NOTE**

Pour des raisons de sécurité, il est conseillé d'exécuter le programme d'installation **non-root user**.

Pré-requis

Avant d'installer JBoss BPM Suite, assurez-vous d'avoir déjà installé Red Hat JBoss EAP 6.4 ou version ultérieure.

Veillez vous assurer que le binaire **jar** soit présent dans **\$PATH** avant d'exécuter le programme d'installation. Sur Red Hat Enterprise Linux, ce binaire se trouve dans le paquet **java-\$JAVA_VERSION-openjdk-devel**.

1. Naviguez vers le dossier à partir duquel vous avez téléchargé le fichier d'installation et exécutez la commande suivante :

```
java -jar jboss-bpmsuite-6.2.0.GA-installer.jar
```

**NOTE**

Pendant l'exécution du programme d'installation sur Windows, il peut vous être demandé de fournir les informations d'authentification d'administrateur lors de l'installation. Pour éviter cela, ajoutez l'option **izpack.mode=privileged** à la commande d'installation : **java -Dizpack.mode=privileged -jar jboss-bpmsuite-6.2.0.GA-installer.jar**

Le programme d'installation graphique sera exécuté et affichera un écran d'accueil et une page d'accord de licence.

2. Veuillez accepter la licence pour continuer.
3. Sur l'écran suivant, fournissez l'emplacement parent d'un JBoss EAP existant où JBoss BPM Suite doit être installé. La capture d'écran ci-dessous montre un exemple de chemin d'accès de répertoire :

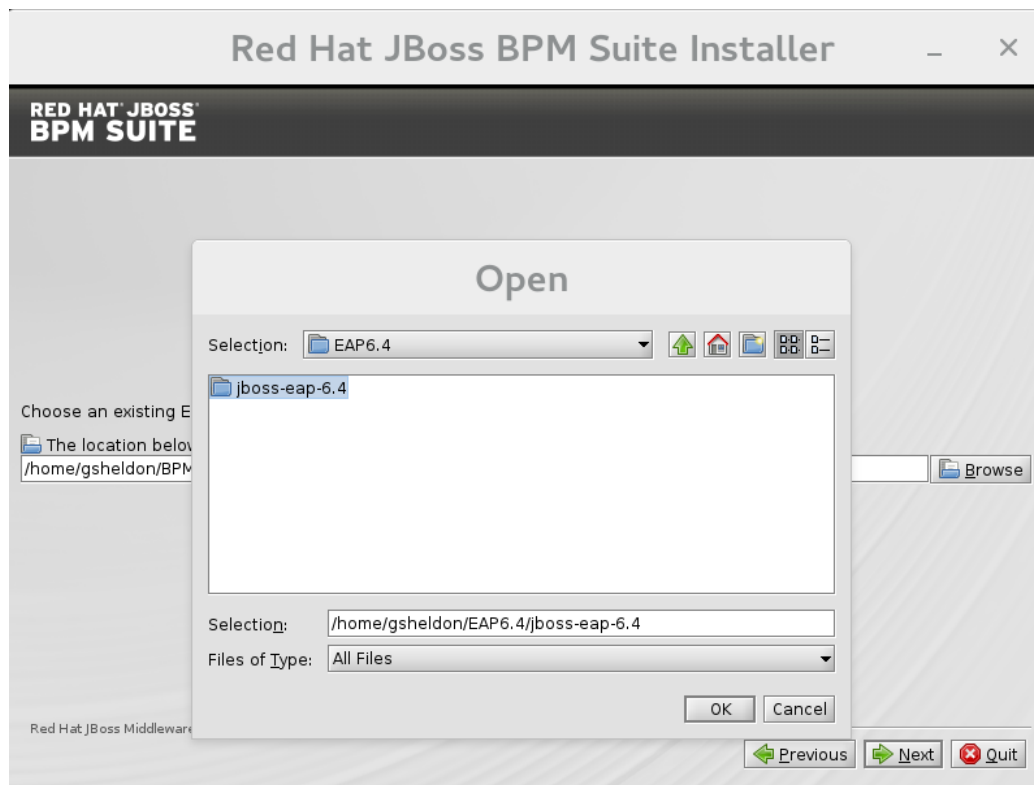


Figure 2.1. Red Hat JBoss BPM Suite pour chemin d'accès de répertoire JBoss EAP

4. Aux deux écrans suivants, créez deux utilisateurs : le premier pour la console de gestion de EAP (ManagementRealm) et le second pour gérer JBoss BPM Suite (ApplicationRealm).

La création du premier utilisateur pour la console de gestion de JBoss EAP est facultative : vous pouvez donc ne pas l'effectuer si celle-ci n'est pas requise.

Veuillez noter ces noms d'utilisateurs et mots de passe : ils vous serviront pour accéder respectivement au serveur JBoss EAP (si vous décidez de le créer) et à l'application JBoss BPM Suite.

Si une configuration avancée n'a pas été effectuée, le mot de passe utilisateur de JBoss BPM Suite sera utilisé comme mot de passe par défaut pour les keystores JMS SSL client et serveur et les keystores d'archivage sécurisé des mots de passe.



NOTE

Le nom d'utilisateur que vous créerez doit être *différent* des autres rôles prédéfinis (veuillez consulter [Section 4.1, « Définir les rôles »](#)).

Vos mots de passe doivent contenir au moins 8 caractères dont un caractère numérique et un caractère non alphanumérique (à l'exception du signe &).



NOTE

Le rôle d'application attribué au second utilisateur créé est le rôle **admin**. Vous pouvez attribuer des rôles supplémentaires à cet utilisateur au cours de cette étape.

5. Configurez ensuite l'environnement de sécurité de votre nouvelle installation JBoss BPM Suite. Choisissez d'activer ou non le gestionnaire Java Security Manager au cours de cette

étape en cochant la case correspondante. Le gestionnaire Java Security Manager offre une meilleure sécurisation de votre système mais ses performances peuvent en être altérées. Veuillez faire un choix en vous basant sur votre environnement.

6. Choisissez d'effectuer ou non une configuration IPv6 pure sur le serveur sur lequel l'installation est effectuée, ce qui vous permettra d'effectuer plus tard des configurations de runtime spécifiques à IPv6.

7. Configurer l'environnement de runtime

Cette étape offre la possibilité d'utiliser une configuration par défaut ou de spécifier une configuration avancée.

o Configuration par défaut

Choisissez la configuration par défaut pour l'environnement de runtime dans l'étape suivante puis cliquez sur **suivant** pour revoir les informations d'installation. Si les informations fournies sont correctes, cliquez sur **suivant** pour démarrer l'installation ou cliquez sur **précédent** pour retourner à l'écran précédent et effectuer des modifications.

o Configuration avancée

Sélectionnez pour activer les options de configuration avancée. Sélectionnez **Perform advanced configuration** (Effectuer une configuration avancée) et choisissez les options de configuration avancée que vous souhaitez activer pour votre environnement en utilisant les cases à cocher.

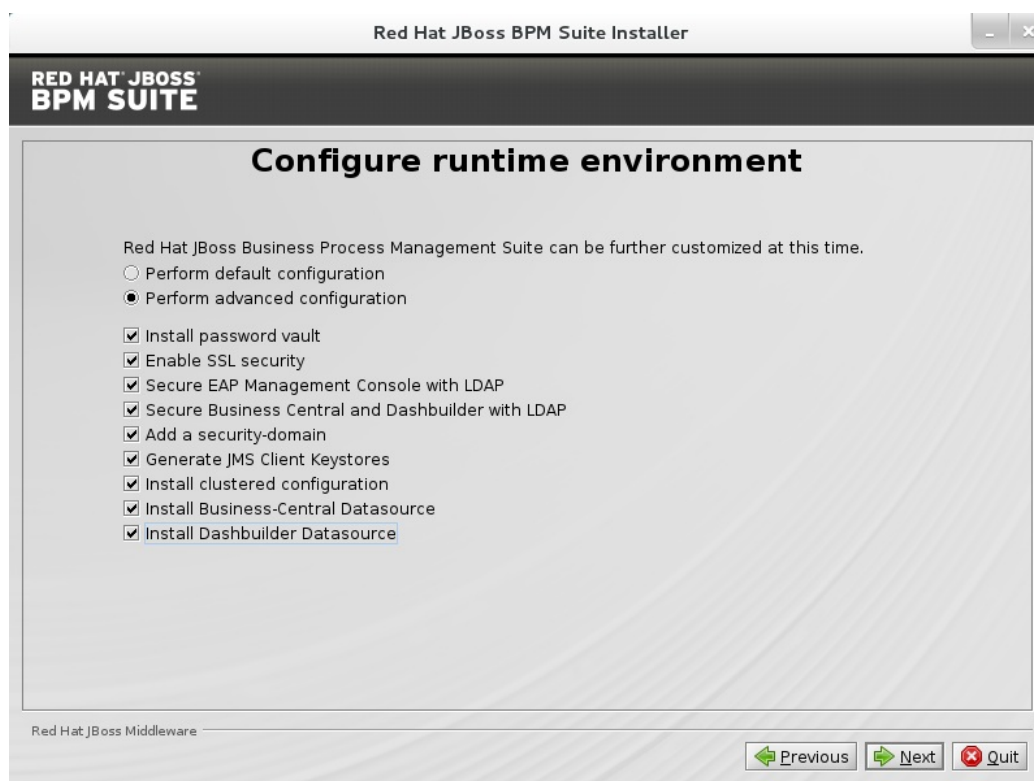


Figure 2.2. Options de configuration avancée

■ Configurer l'archivage sécurisé des mots de passe

L'archivage sécurisé des mots de passe est utilisé pour masquer les mots de passe dans les différents descripteurs de serveur en utilisant une clé secrète java générée lors du processus d'installation ou manuellement à l'aide du keytool. Cela évite que les mots de passe soit stockés en texte brut dans les descripteurs. *iteration count* et *salt* sont tous deux des paramètres du processus de cryptage.

Dans le cas de JBoss BPM Suite, un archivage sécurisé est toujours installé, même si l'utilisateur ne décide pas d'en installer un avec ses propres paramètres. Dans ce cas-là, les valeurs par défaut seront utilisées.

Pour plus d'informations à propos de l'archivage sécurisé des mots de passe, veuillez consulter le *Guide de Sécurité de Red Hat JBoss EAP*

Red Hat JBoss BPM Suite Installer

RED HAT JBOSS BPM SUITE

Configure password vault

A password vault encrypts sensitive strings and stores them in an encrypted keystore. The vault mechanism manages decrypting the strings for use with security domains, security realms, or other verification systems. Note that the keystore must be of type "JCEKS".

Please make note of your entry below in order to mask any subsequent passwords. See JBoss EAP 6 documentation for further details.

The password must have no fewer than 6 characters.

Vault alias:

Salt (8-chars)

Iteration count

Vault keystore password:

Re-enter vault keystore password:

New Keystore location:

Encrypted vault directory:

Red Hat JBoss Middleware

Figure 2.3. Configurer l'archivage sécurisé des mots de passe

■ Sécurité SSL

Cet écran vous permet d'ajouter les éléments `<ssl>` et `<truststore>` au ManagementRealm security-realm à l'aide du keystore fourni.

- L'élément `<ssl>` pousse le serveur à présenter le certificat du keystore comme étant son identité, ce qui permet à l'utilisateur d'appliquer son certificat officiel.
- L'élément `<truststore>` active l'authentification « Client-Cert », ce qui signifie que si un client distant tente de se connecter à une ressource gérée par le ManagementRealm, le client peut présenter un certificat et, si une entrée dans le truststore correspond, sera authentifié sans avoir besoin de fournir un nom d'utilisateur/mot de passe.

Le résultat final est une connexion cryptée sécurisée entre le client et le serveur pour le ManagementRealm.

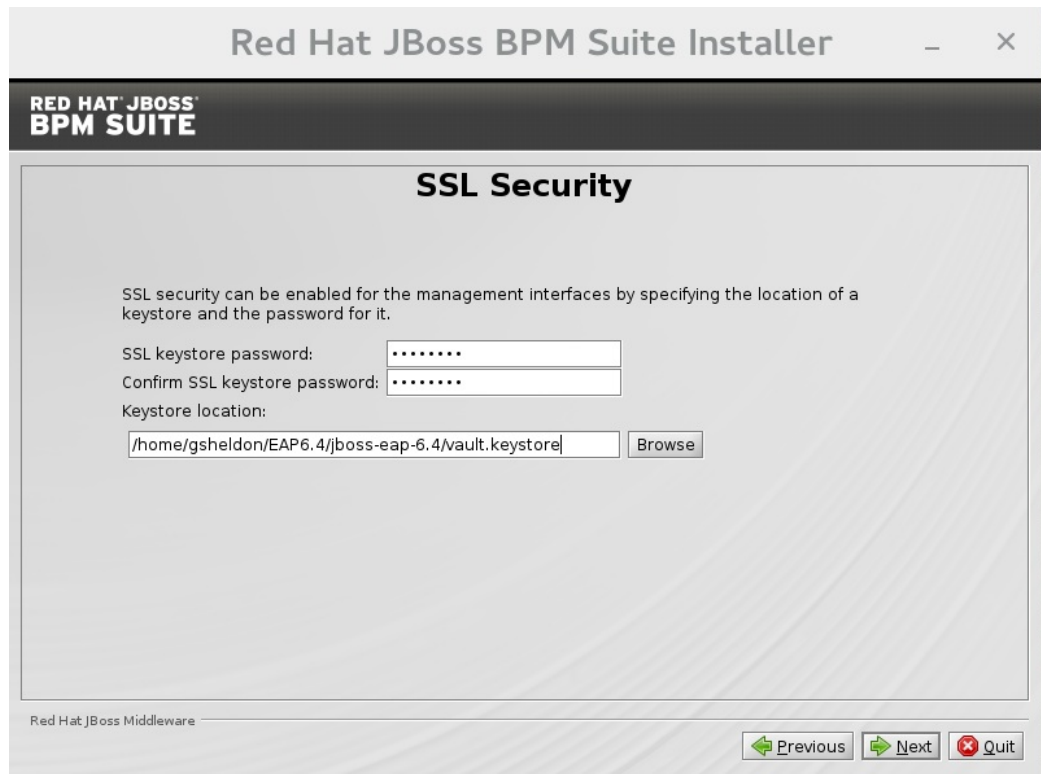


Figure 2.4. Configuration de Sécurité SSL

■ Sécurité LDAP

Cette étape du programme d'installation permet à l'utilisateur de définir un serveur LDAP, qui en retour définit les utilisateurs qui devraient être autorisés à s'authentifier avec ManagementRealm. Cette étape remplace la configuration par défaut.

L'écran **LDAP Connection** permet aux utilisateurs de définir comment se connecter au serveur LDAP.

- Le Nom Distinctif (DN ou Distinguished Name) : l'utilisateur pouvant se connecter au serveur LDAP. Le nom distinctif définit généralement un utilisateur spécial à cet effet.

Red Hat JBoss BPM Suite Installer

RED HAT JBOSS BPM SUITE

LDAP Connection

Red Hat JBoss Enterprise Application Platform can be configured to allow you to use a LDAP server as the authentication and authorization authority, both for applications and the management interface. This is a 2-step process.

Connection Name:

Directory Server:

Distinguished Name (DN):

DN Password:

Confirm DN Password:

Red Hat JBoss Middleware

Figure 2.5. Configuration de connexion LDAP

Sécurité LDAP (console de gestion)

L'écran **Configuration LDAP de console de gestion** vous permet d'installer un domaine de sécurité et permet d'ajouter l'élément `<security-realm>` aux descripteurs et utilise la connexion définie précédemment.

Red Hat JBoss BPM Suite Installer

RED HAT JBOSS BPM SUITE

Management Console LDAP Configuration

A new security realm will be added with the attributes below, and will utilize the LDAP outbound connection defined in the previous step. This new security realm will be associated with the management interfaces. The value for Filter can either be of the form attribute="value", where value is the name of the attribute which holds the username, or of the form filter="value", where value is a query in standard LDAP syntax. If using an LDAP query, escape "&" with "&".

Security Realm Name:

Base DN:

Filter Type:

Username filter:

Recursive directory search:

Red Hat JBoss Middleware

Figure 2.6. Configuration LDAP de console de gestion

- **Nom distinctif de base** : définit généralement une « recherche de base » ou « contexte root » pour commencer à rechercher des utilisateurs.

- **Type de filtre** : indique à JBoss EAP comment trouver l'attribut LDAP qui définit un utilisateur : il peut s'agir d'un simple attribut ou d'un filtre LDAP complexe.
- **Attribut de nom d'utilisateur** : l'attribut LDAP détenant les valeurs de nom d'utilisateur. Un nom d'utilisateur saisi dans ce domaine est utilisé pour les requêtes de recherches comme valeur de l'attribut « uid ». Si un utilisateur choisit « LDAP syntax query » comme type de filtre, cette requête doit être spécifiée dans ce champ.
- **Recherche récursive de répertoire** : si activée, JBoss EAP traversera l'arborescence LDAP de manière récursive, en commençant par Base DN. Sinon, la recherche sera limitée à Base DN.

Sécurité LDAP (Business Central)

La plupart des champs suivants sont semblables à la Base DN. Les contextes sont utilisés pour rechercher des rôles, ce qui permet d'effectuer une autorisation en plus d'une authentification. Sinon, les champs **context** sont analogues à la Base DN précédente et les champs **attribute** sont analogues à l'attribut Nom d'utilisateur. Les filtres permettent un contrôle précis des valeurs d'un attribut donné qui seront acceptées.

Dans JBoss BPM Suite, les fichiers `jbpm.usergroup.callback.properties` et `jbpm.user.info.properties` utilisés par les composants `LDAPUserGroupInfo` et `LDAPUserInfo` de Task Service sont également remplis avec des valeurs saisies sur la page de configuration Business Central LDAP.

Les valeurs de saisie de la page de configuration Business Central LDAP sont utilisées pour configurer un nouveau domaine de sécurité, qui utilise le module de connexion `LdapExtended`. Ce domaine de sécurité est configuré par défaut pour l'application web Business Central. Pour plus d'informations sur les domaines de sécurité et modules de connexion, veuillez consulter le *Guide de sécurité Red Hat JBoss EAP*

Red Hat JBoss BPM Suite Installer

RED HAT JBOSS BPM SUITE

Business Central LDAP Configuration

Enter the details below to configure LDAP authentication on Business Central. The LDAP server URL, bind DN and bind password previously defined will be used. The contexts below define the users and roles that should be allowed access to Business Central. A security domain will be defined using these details.

User context:

User filter:

User roles context:

User roles filter:

Role context:

Role filter:

Role attribute id:

Role name id:

☐ Role attribute is distinguished name

Red Hat JBoss Middleware

Previous Next Quit

Figure 2.7. Configuration Business Central LDAP

■ Domaine de sécurité et JSSE

L'écran de domaine de sécurité vous permet de configurer tous les éléments du sous-système de sécurité *<security-domain>* pour gérer les informations de sécurité, y compris la configuration JSSE. Pour plus d'informations sur la configuration des domaines de sécurité, veuillez consulter le *Guide de sécurité Red Hat JBoss EAP*

Red Hat JBoss BPM Suite Installer

RED HAT JBOSS BPM SUITE

Security Domain

Configure a security-domain using the fields below. The only required field is the security-domain name. The rest of the fields are optional. Most fields have a predefined list of valid values.

Security-Domain Name:

Security-Domain Cache-type:

Add authentication element:

Authentication code: Authentication flag: Authentication options:

Add authorization element:

Authorization code: Authorization flag: Authorization options:

Add mapping element:

Mapping code: Mapping type: Mapping options:

Red Hat JBoss Middleware

Figure 2.8. Domaine de sécurité

Red Hat JBoss BPM Suite Installer

RED HAT JBOSS BPM SUITE

JSSE Configuration

Configure a JSSE element. The JSSE element requires either a keystore or a truststore to be configured.

☒ Add jsse element

Cipher suites used by SSLContext:

Protocols used by SSLContext:

Alias of client-side keystore:

Alias of server-side keystore:

Third party validation token:

☒ Add keystore element

JSSE keystore password:

Confirm JSSE keystore password:

Keystore Provider:

Provider argument:

Keystore type:

Keystore URL:

☒ Add keystore manager element

KeyManagerFactory algorithm:

KeyManagerFactory Provider:

Red Hat JBoss Middleware

Figure 2.9. Configuration JSSE

■ Configurer les keystores JMS SSL

Cet écran permet le cryptage de messages JMS envoyés à Business Central. Les keystores client sont distribués aux systèmes nécessitant une communication avec le serveur afin de faciliter les communications cryptées. Les utilisateurs peuvent utiliser leur keystore pré-existants ou en générer de nouveaux.

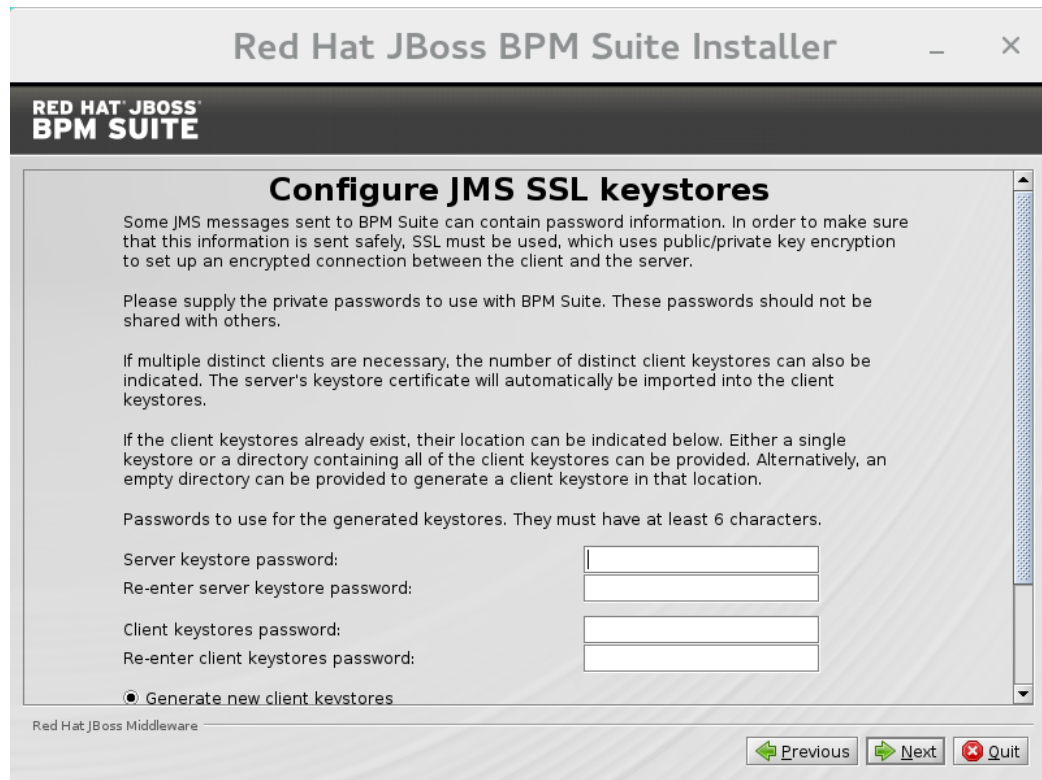


Figure 2.10. Configurer les keystores JMS SSL

■ Configurer la mise en cluster

Sélectionner cette option installe JBoss BPM Suite, qui est prêt à effectuer des opérations clusterisées. Pour obtenir davantage d'informations, veuillez consulter [Section 6.1, « Mise en cluster avec JBoss EAP »](#).

8. Le programme d'installation effectuera les étapes d'installation de JBoss BPM Suite ainsi que les étapes de configuration post-installation lorsque vous cliquez sur **Suivant**. Le programme d'installation peut également lancer le serveur JBoss BPM Suite dans le mode de votre choix (en mode autonome ou de domaine, davantage d'informations se trouvent dans [Section 5.1, « Démarrage du serveur »](#)), puis il s'y connectera pour valider l'installation. Cliquez sur **Suivant** pour arriver au dernier écran, sur lequel vous pourrez générer le script d'installation et le fichier des propriétés. Pour quitter le programme d'installation, veuillez cliquer sur **Terminé**.

Vous venez d'installer Red Hat JBoss BPM Suite à l'aide du programme d'installation.

2.1.2. Installation de Red Hat JBoss BPM Suite à l'aide du programme d'installation en mode CLI

Le programme d'installation pour Red Hat JBoss BPM Suite peut également être exécuté à travers l'interface en ligne de commande CLI (Command-Line Interface). La procédure ci-dessous indique les étapes que vous pourriez rencontrer si vous utilisez cette option pour l'installation de JBoss BPM Suite.

Pré-requis

Avant d'installer JBoss BPM Suite, assurez-vous d'avoir déjà installé Red Hat JBoss EAP 6.4 ou version ultérieure.

Veillez vous assurer que le binaire `jar` soit présent dans `$PATH` avant d'exécuter le programme d'installation. Sur Red Hat Enterprise Linux, ce binaire se trouve dans le paquet `java-$JAVA_VERSION-openjdk-devel`.

1. Naviguez vers le dossier à partir duquel vous avez téléchargé le fichier d'installation et exécutez la commande suivante :

```
java -jar jboss-bpmsuite-6.2.0.GA-installer.jar -console
```

2. Le processus interactif de ligne de commande démarrera et affichera le contrat de licence de l'utilisateur final. Au bas de ce contrat, il vous sera demandé de sélectionner une option :

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

3. Saisissez 1 pour démarrer l'installation et tapez le répertoire parent d'une installation EAP existante.

```
L'emplacement ci-dessous doit indiquer le répertoire JBOSS_HOME  
d'une installation EAP existante.
```

```
[/home/user/BPMSuite-6.2.0/jboss-eap-6.4]
```

Le programme d'installation vérifiera l'emplacement de l'installation EAP JBoss à l'emplacement prévu. Tapez 1 pour confirmer et continuer.

4. Facultatif : créez un utilisateur pour la console de gestion de JBoss EAP (Management Realm) :

```
Create an administrative user  
This user will be added to the host container's management realm for  
administrative purposes. It can be used to access the management  
console, the management CLI or other applications secured in this  
realm.
```

Tapez 1 si vous souhaitez créer un nouvel utilisateur administratif, sinon tapez 0. Si vous décidez d'en créer un, suivez ces étapes :

1. Veuillez saisir un nom d'utilisateur :

```
Admin username: [admin]
```

2. Créer et confirmer un mot de passe pour l'utilisateur de la console de gestion EAP :

```
The password must have at least 8 characters, and contain at  
least one number and one non-alphanumeric symbol (not including  
the character &).
```

```
Admin password: []
```

```
Confirm admin password: [*****]
```

Après la création de l'utilisateur, passez à l'étape suivante.

5. Créez un utilisateur administrateur JBoss BPM Suite.

Create a Business Process Management Suite Admin User
Create a BPM Suite admin user. The user will be added to the ApplicationRealm, and can be used to access the Business Central Console. The User will be assigned the 'admin' application roles. The BPM Suite username cannot be any of the following: 'admin', 'analyst', 'user', 'manager' or 'developer'.

BPM Suite username: [bpmsAdmin]

6. Choisissez un nom d'utilisateur puis créez et confirmez un mot de passe.

The password must have at least 8 characters, and contain at least one number and one non-alphanumeric symbol (not including the character &).

BPM Suite password: []

Confirm BPM Suite password: [*****]

7. Une fois le mot de passe saisi et confirmé, une étape supplémentaire vous permettra de définir d'autres rôles pour cet utilisateur (en plus du rôle admin). Saisissez ces rôles dans une liste séparée par des virgules ou appuyez sur la touche Entrée pour ignorer cette étape.

(Optional) You can add this user to additional roles that will be used for task management. These roles are custom named and used again when building your processes with human tasks. Add your custom named roles in a comma separated list below.

Additional user roles: []

8. Configurez le Java Security Manager en tapant 1 pour le sélectionner ou 0 pour annuler la sélection.

Configure the Java Security Manager
A Java security manager offers JVM level security beyond what is provided by the application container. It enforces access rules at the JVM runtime based on one or more security policies.

This installer will place two security policies in the installation directory with the filenames 'security.policy' and 'kie.policy' regardless of choice. Those policies will be enabled at runtime if the option below is selected.

Please note that a security manager imposes a significant performance overhead when enabled. It is suggested the included policies be applied in production if user requirements call for a stronger measure than what is already provided by the application container's authentication and authorization mechanism.

Please see the JBoss Business Process Management Suite administrative documentation for further details and consideration.

[] Enable the Java security manager
Input 1 to select, 0 to deselect:

9. Après avoir sélectionné le gestionnaire de sécurité Java, choisissez une option à partir de l'invite ci-dessous :

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

10. Précisez si vous utilisez IPv6.

IPv6 configuration

If this computer is using a pure IPv6 configuration, please check the box below. A pure IPv6 setup requires additional configuration at runtime to ensure the proper bindings of the management and http interfaces.

[] Enable pure IPv6 configuration
Input 1 to select, 0 to deselect:

Après avoir sélectionné ou annulé la sélection de la configuration IPv6, sélectionnez l'une des options suivantes :

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

11. Configurer l'environnement runtime en choisissant la configuration par défaut ou les options avancées.

Configure runtime environment

Red Hat JBoss Business Process Management Suite can be further customized at this time.

0 [x] Perform default configuration
1 [] Perform advanced configuration

Si vous sélectionnez 1, « Perform advanced configuration », effectuez les configurations suivantes :

- [] Install password vault
Input 1 to select, 0 to deselect:

- [] Enable SSL security
Input 1 to select, 0 to deselect:

- [] Secure EAP Management Console with LDAP
Input 1 to select, 0 to deselect:

- [] Secure Business Central with LDAP
Input 1 to select, 0 to deselect:

- ☐ Add a security-domain
Input 1 to select, 0 to deselect:
- ☐ Generate JMS Client Keystores
Input 1 to select, 0 to deselect:
- ☐ Install clustered configuration
Input 1 to select, 0 to deselect:

12. Puis sélectionnez une option à partir de l'invite ci-dessous :

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

13. Le fichier .jar débutera sa décompression et configuration.

14. Suite à l'installation, la ligne de commande vous demandera si vous souhaitez générer un script d'installation automatique et un fichier de propriétés.

```
Installation has completed successfully.
Application installed on /home/user/BPMSuite-6.2.0/jboss-eap-6.4
Would you like to generate an automatic intallation script and
properties file?
(y/n) [n]:
```

15. Si vous tapez [y], indiquez un chemin d'accès pour le script d'installation automatique :

```
Select path for the automatic installation script:
[/home/user/BPMSuite-6.2.0/jboss-eap-6.4/<auto script filename>]
```

Ce script généré permettra à l'utilisateur d'exécuter le programme d'installation de la manière suivante pour les installations futures :

```
java -jar jboss-bpmsuite-6.2.0.GA-installer.jar <auto script
filename>
```

NOTE

Cette méthode d'exécution du programme d'installation aura pour résultat une installation identique à l'installation à partir de laquelle le script automatique a été généré. Veuillez noter que les valeurs sensibles, telles que les mots de passe, devront être fournies à partir d'un fichier externe ou lors de l'installation automatique. L'argument facultatif ci-dessous permet à l'utilisateur de fournir ces valeurs automatiquement :

```
-variablefile <variable filename>
```

Les valeurs sensibles peuvent également être fournies à l'aide de l'argument suivant :

```
-variables key1=value1, key2=value2
```

16. La ligne de commande affichera le message suivant suite à la création d'un script automatique et/ou l'installation de la console :

```
XML written successfully.  
[ Console installation done ]  
[BPMS_Installer]$
```

17. Démarrez JBoss EAP comme décrit dans [Section 5.1, « Démarrage du serveur »](#).
18. Naviguer vers <http://localhost:8080/business-central> dans un navigateur web.
19. Connectez-vous avec le nom d'utilisateur et mot de passe attribués à l'utilisateur de JBoss BPM Suite à l'étape Créer et confirmer un mot de passe pour l'utilisateur JBoss BPM Suite.

2.1.3. Télécharger un ZIP déployable pour Red Hat JBoss EAP

Dans le cas où vous souhaiteriez ne pas utiliser le programme d'installation JAR, vous pouvez télécharger le fichier ZIP déployable en effectuant les étapes suivantes :

1. Connectez-vous au [Portail Client Red Hat](#).
2. Cliquez sur **Téléchargements** → **Téléchargements de produits**.
3. À partir de la page **Téléchargements de produits** qui s'ouvre, cliquez sur **Red Hat JBoss BPM Suite**.
4. À partir du menu déroulant **Version**, sélectionnez la version 6.2.
5. Sélectionnez **Red Hat JBoss BPM Suite 6.2 Deployable for EAP 6.4** puis cliquez sur **Télécharger**.
6. Veuillez extraire l'archive. Le répertoire en résultant est nommé **\$EAP_HOME**.
7. Avec la ligne de commande, veuillez vous déplacer sur le répertoire **\$EAP_HOME**, puis lancez le serveur autonome :

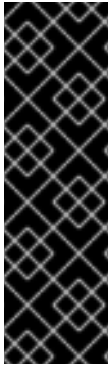
Dans un environnement Unix, exécutez :

```
./standalone.sh
```

Dans un environnement Windows, exécutez :

```
./standalone.bat
```

Configurer les modes de domaine



IMPORTANT

Si vous avez installé JBoss BPM Suite avec le programme d'installation JAR (Section 2.1.1, « Installation de Red Hat JBoss BPM Suite à l'aide du programme d'installation » ou Section 2.1.2, « Installation de Red Hat JBoss BPM Suite à l'aide du programme d'installation en mode CLI »), vous pouvez ignorer cette section – le mode domaine est déjà configuré.

La manière d'exécuter JBoss BPM Suite en mode domaine est décrite dans Section 5.1.2, « Mode Domaine ».

Pour installer le paquet déployable pour JBoss EAP en mode domaine, veuillez reproduire les étapes suivantes :

1. Téléchargez et extrayez le fichier ZIP Red Hat JBoss BPM Suite 6.2.0 Deployable for EAP 6 à partir du [Portail Client Red Hat](#) et copiez les répertoires suivants dans votre installation locale d'EAP 6.4 :

- o `bin`
- o `domain`

Omettez le répertoire `standalone`.

2. Avec la ligne de commande, veuillez vous déplacer sur le répertoire `$EAP_HOME/bin`, puis lancez le domaine :

Dans un environnement Unix, exécutez :

```
./domain.sh
```

Dans un environnement Windows, exécutez :

```
./domain.bat
```

3. Veuillez déployer les archives via des scripts CLI (`$EAP_HOME/bin/jboss-cli.sh` ou `$EAP_HOME/bin/jboss-cli.bat`), ou via l'interface utilisateur web de gestion (<http://localhost:9990/>) :



NOTE

Les applications web `business-central.war` et `dashbuilder.war` fournies dans les binaires déployables EAP sont des répertoires, mais pour un déploiement dans le domaine, vous devrez utiliser des archives WAR. Pour les créer, veuillez compresser le contenu des répertoires `business-central.war` et `dashbuilder.war`.

- a. Pour déployer l'archive via les scripts CLI, veuillez vous déplacer sur `$EAP_HOME/bin` et déployer le fichier WAR, en suivant les invites à l'écran :

Dans un environnement Unix, exécutez :

```
./jboss-cli.sh
```

Dans un environnement Windows, exécutez :

```
./jboss-cli.bat
```

b. Pour déployer l'archive via l'interface utilisateur web de gestion (<http://localhost:9990/>) :

- connectez-vous avec votre compte de gestion EAP
- sélectionnez **Domaine -> Gérer Déploiements -> Référentiel de contenu -> Ajouter**
- sélectionnez l'archive web à partir du système de fichier et téléversez l'archive web
- sélectionnez le déploiement puis cliquez sur le bouton **Attribuer**
- Sélectionnez le groupe de serveurs



NOTE

Afin de pouvoir se connecter à Business Central déployé sur des machines Host Controller (HC), l'utilisateur créé sur la Domain Controller Machine doit être créé sur les machines Host Controller également, en reproduisant les étapes de la section [Section 4.2, « Création d'utilisateurs »](#).

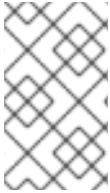
Installation de plusieurs instances JBoss BPM Suite

Un utilisateur aura parfois besoin de regrouper un ensemble de nœuds EAP 6 sur la même machine et de leur donner un nom facilitant sa gestion. Des valeurs uniques doivent être incorporées aux propriétés de système de chaque instance de serveur. Vous trouverez ci-dessous les propriétés courantes pouvant être spécifiées avec un nœud JBoss BPM Suite unique pour modifier la configuration par défaut ; elles doivent cependant être spécifiées pour différents nœuds exécutés sur une machine unique pour que chaque nœud puisse diriger vers un répertoire différent :

- `org.uberfire.nio.git.dir`
- `org.uberfire.metadata.index.dir`
- `org.uberfire.nio.git.ssh.cert.dir`

Lorsque plusieurs nœuds JBoss BPM Suite sont utilisés sur une seule machine, les propriétés suivantes doivent être spécifiées :

- `org.uberfire.nio.git.daemon.host` - peut être laissé sur défaut pour être relié au localhost.
- `org.uberfire.nio.git.daemon.port`
- `org.uberfire.nio.git.ssh.host` - peut être laissé sur défaut pour être relié au localhost.
- `org.uberfire.nio.git.ssh.port`

**NOTE**

org.uberfire.nio.git.daemon.port et **org.uberfire.nio.git.ssh.port** nécessitent tous deux des valeurs de port différentes de manière à éviter les conflits de port.

Incorporez les propriétés précédentes dans le fichier

\$EAP_HOME/domain/configuration/host.xml tel qu'illustré dans les deux nœuds ci-dessous :

Nœud A :

```
<system-properties>
  <property name="org.uberfire.nio.git.dir" value="/valid/path/.." boot-
time="false"/>
  <property name="org.uberfire.metadata.index.dir"
value="/valid/path/.." boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/valid/path/.." boot-time="false"/>

  <property name="org.uberfire.nio.git.daemon.host" value="10.10.10.10"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.port" value="9417" boot-
time="false"/>
  <property name="org.uberfire.nio.git.ssh.host" value="10.10.10.10"
boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.port" value="8002" boot-
time="false"/>
</system-properties>
```

Nœud B :

```
<system-properties>
  <property name="org.uberfire.nio.git.dir" value="/valid/path/.." boot-
time="false"/>
  <property name="org.uberfire.metadata.index.dir"
value="/valid/path/.." boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/valid/path/.." boot-time="false"/>

  <property name="org.uberfire.nio.git.daemon.host" value="10.10.10.10"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.port" value="9418" boot-
time="false"/>
  <property name="org.uberfire.nio.git.ssh.host" value="10.10.10.10"
boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.port" value="8003" boot-
time="false"/>
</system-properties>
```

Les propriétés de système décrites ci-dessus doivent indiquer l'hôte, le port ou l'emplacement des fichiers **.index** ou **.niogit**. Ces fichiers, qui doivent être utilisés par un nœud respectif, seront ensuite groupés dans un domaine particulier.

2.2. L'INSTALLATION D'UN GROUPEMENT DÉPLOYABLE STANDARD

Pour installer Red Hat JBoss BPM Suite sur Red Hat JBoss Server Web (EWS), veuillez utiliser le paquet déployable générique du produit.

Pour l'installation sur EWS, le paquet déployable générique contient le gestionnaire de transactions supplémentaires et les bibliothèques de sécurité qui ne font pas partie de Red Hat JBoss EWS.

Veuillez noter que pour installer le paquet déployable générique, vous avez besoin des fichiers ZIP suivants :

- **jboss-bpmsuite-VERSION-deployable-generic.zip** : ce fichier contient les applications web **business-central.war**, **dashbuilder.war** et **kie-server.war**.
- **jboss-bpmsuite-VERSION-engine.zip** : les bibliothèques d'exécution d'engine prises en charge pour l'intégration de engine dans votre application et autres bibliothèques nécessaires pour un déploiement générique.

2.2.1. Téléchargement du paquet déployable standard

Pour télécharger le paquet Red Hat JBoss BPM Suite dans JBoss Web Server, procéder ainsi :

1. Connectez-vous au [Portail Client Red Hat](#).
2. Cliquez sur **Téléchargements**.
3. À partir de la liste de produits, cliquez sur **Red Hat JBoss BPM Suite**.
4. À partir du menu déroulant **Version**, sélectionnez la version 6.2 (si cela n'est pas déjà fait).
5. Dans la section Téléchargements de logiciels qui s'affiche, allez sur la ligne **Red Hat JBoss BPM Suite 6.2 Deployable for All Supported Containers** puis cliquez sur **Télécharger**.
6. Placez-vous sur la ligne des fichiers **Red Hat JBoss BPM Suite 6.2 Engine**, puis cliquez sur **Télécharger** pour télécharger les fichiers JBoss BPM Suite Engine.

2.2.2. Installation d'un paquet déployable standard

Pour installer le paquet déployable générique, vous devez mettre en place ce qui suit après avoir installé la plateforme sous-jacente (Red Hat JBoss WS) :

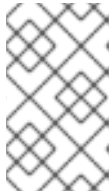
- Configurez le pilote de la base de données et le gestionnaire de transactions (Bitronix) (voir [Section 2.2.2.1, « Installer le gestionnaire de transactions dans Red Hat JBoss Web Server 2.1 \(Tomcat 7\) »](#)).
- Installez l'application Business Central : créer des utilisateurs et des rôles, et créer des persistance (voir [Section 2.2.2.2, « Installation de Business Central dans Red Hat JBoss Web Server 2.0 \(Tomcat 7\) »](#)).
- Installer l'application Dashbuilder : créer des utilisateurs et des rôles, et créer des persistance (voir [Section 2.2.2.3, « Installation de Dashbuilder dans Red Hat JBoss Web Server 2.0 \(Tomcat 7\) »](#)).

2.2.2.1. Installer le gestionnaire de transactions dans Red Hat JBoss Web Server 2.1 (Tomcat 7)

1. Extrayez le paquet zip déployable que vous avez téléchargé à partir du [Portail Client Red Hat](#) vers un emplacement temporaire. Ce paquet zip contient les trois archives d'application web suivantes : `business-central.war`, `dashbuilder.war` et `kie-server.war` dans un format éclaté. Renommez ces dossiers pour supprimer l'extension `.war`.
2. Copiez ces dossiers directement dans le dossier `$TOMCAT_DIR/webapps`.

Vous devriez obtenir trois dossiers dans un format éclaté :

`$TOMCAT_DIR/webapps/business-central`, `$TOMCAT_DIR/webapps/dashbuilder` et `$TOMCAT_DIR/webapps/kie-server`.



NOTE

`$TOMCAT_DIR` correspond au répertoire d'accueil où se situe le serveur web. Remplacez-le par le chemin actuel qui mène au répertoire d'accueil de votre serveur web, ainsi : `/home/john/jboss-ews-2.1/tomcat7/`

3. Extraire le contenu de l'archive des fichiers JBoss BPM Suite Engine vers un emplacement temporaire à partir duquel vous pourrez copier les bibliothèques dont vous avez besoin. Ce dossier contient maintenant toutes les bibliothèques JBoss BPM Suite principales dans le dossier extrait et le dossier `lib`.
4. Installer le gestionnaire de transactions.



AVERTISSEMENT

Veuillez noter que la section suivante décrit l'installation d'un gestionnaire de transaction, Bitronix, qui n'est pas officiellement pris en charge par Red Hat.

Copiez les bibliothèques jar du gestionnaire de transactions suivantes à partir du dossier `lib` vers le répertoire `$TOMCAT_DIR/lib/` :

- `btm-VERSION.jar`
- `btm-tomcat55-lifecycle-VERSION.jar`
- `jta-VERSION.jar`
- `slf4j-api-VERSION.jar`
- `slf4j-jdk14-VERSION.jar`

De plus, téléchargez la bibliothèque suivante et copiez-la dans le dossier `$TOMCAT_DIR/lib/` également :

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

5. Installez le pilote sur votre base de données

Copiez le fichier jar et le pilote de base de donnée qui convient dans `$TOMCAT_DIR/lib/`.

**NOTE**

Si vous utilisez la base de données H2 intégrée, le pilote sera disponible dans **business-central/WEB-INF/lib/**.

6. Créer les fichiers de configuration du gestionnaire de transactions dans **\$TOMCAT_DIR/conf/** :

- o btm-config.properties

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

- o resources.properties (le resource.ds1.uniqueName définit le nom de la source de données utilisée dans la définition de ressource tomcat par la suite - et notez cette valeur).

Veillez à bien changer les valeurs dans les définitions suivantes pour qu'elles correspondent bien à votre environnement.

Exemple 2.1. Définition de source de données H2

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

Exemple 2.2. Définition de source de données MySQL 5.5

```
resource.ds1.className=com.mysql.jdbc.jdbc2.optional.MysqlXADat
aSource
resource.ds1.uniqueName=jdbc/jbpm
resource.ds1.minPoolSize=0
resource.ds1.maxPoolSize=10
resource.ds1.driverProperties.URL=jdbc:mysql://localhost:3306/s
ampled
resource.ds1.driverProperties.user=dbuser
resource.ds1.driverProperties.password=dbpassword
resource.ds1.allowLocalTransactions=true
```

Exemple 2.3. Définition de source de données DB2 de Type 4

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

Exemple 2.4. Définition de source de données Oracle

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

Exemple 2.5. Définition de source de données Microsoft SQL Server

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

7. Installez le listener du gestionnaire de transaction dans `$TOMCAT_DIR/conf/server.xml` pour démarrer et arrêter le Bitronix sur arrêt et démarrage du conteneur :

Ajouter l'élément suivant comme dernier élément `<Listener>` dans l'élément `<Server>` :

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

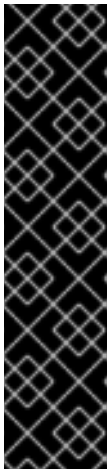
8. Définissez la propriété système `btm.root` et l'emplacement où le fichier de config bitronix se trouve :

Dans `$TOMCAT_DIR/bin/`, créez un fichier `setenv.sh` accessible en lecture avec le contenu suivant :

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

La propriété `org.jbpm.designer.perspective` est définie sur `RuleFlow` pour permettre à la perspective par défaut du designer d'être `RuleFlow` plutôt que `Full`. Accordez les permissions d'exécution du fichier le cas échéant.

La propriété `java.security.auth.login.config` doit être définie pour que le clone ssh du référentiel git fonctionne.



IMPORTANT

Dans les systèmes Microsoft Windows, remplacez la valeur `$CATALINA_HOME` du contenu du fichier par le nom de la variable d'environnement équivalente ou utilisez le chemin complet et ajoutez les valeurs dans le fichier `setenv.bat` comme dans l'exemple suivant :

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

2.2.2.2. Installation de Business Central dans Red Hat JBoss Web Server 2.0 (Tomcat 7)

Pour installer Business Central, procédez ainsi :

1. Installez une **Valve** pour que l'application web de Web Central puisse télécharger les utilisateurs créés dans Tomcat :
 - a. Définissez les utilisateurs et les rôles dans `$TOMCAT_DIR/conf/tomcat-users.xml`. Notez que Business Central nécessite que les utilisateurs aient les rôles `admin` et/ou `analyst` (pour obtenir plus d'informations sur les définitions de rôle ou d'utilisateur, veuillez consulter la documentation Tomcat 7).

La liste de programme ci-dessous est un exemple qui montre comment ces deux rôles peuvent être ajoutés et comment affecter ces rôles à un utilisateur nommé `bpmsadmin`.



NOTE

Assurez-vous que les utilisateurs et les rôles ne portent pas le même nom. Par exemple, ne donnez pas le nom `admin` à un utilisateur.

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

- b. Déplacez (ne copiez pas) **kie-tomcat-integration-*VERSION*.jar** de ***\$TOMCAT_DIR/webapps/business-central/WEB-INF/lib/*** dans ***\$TOMCAT_DIR/lib/***.
- c. Copiez **jboss-jaxb-api-*VERSION*.jar** de ***\$TOMCAT_DIR/webapps/business-central/WEB-INF/lib/*** dans ***\$TOMCAT_DIR/lib/***.
- d. Dans ***\$TOMCAT_DIR/conf/server.xml***, ajoutez la déclaration de Valve Tomcat dans l'élément **<hôte>** qui convient :

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

2. Si vous utilisez une source de données autre que la valeur par défaut fournie par la base de données sous-jacente de H2, vous aurez besoin d'installer une persistance. Si vous utilisez la base de données de H2 par défaut, vous pouvez alors ignorer le reste des étapes de cette procédure.

Dans cette procédure, vous configurez une source de données avec le nom JNDI **jdbc/myDatasource** défini dans **uniqueName=jdbc/jbpm** du fichier **bitronix/resources.properties** précédent (pour l'option MySQL).

- a. Dans **business-central/META-INF/context.xml**, remplacez le nom JNDI de la source de données de l'élément **<Resource>**. L'attribut **uniqueName** se rapporte à la propriété **resource.ds1.uniqueName** définie dans **resources.properties** :

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

- b. Dans **business-central/WEB-INF/web.xml**, remplacez le nom JNDI de la source de données de l'élément **>res-ref-name<** par le nom de votre source de données :

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

- c. Modifiez **business-central/WEB-INF/classes/META-INF/persistence.xml**.

Dans ce fichier, modifiez le nom du dialecte hibernate utilisé dans votre base de données, si vous utilisez une autre base de données que H2. Le code ci-dessous montre les informations de la base de données d'origine de **persistence.xml** :

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

Cette information peut être mise à jour de la manière suivante (comme démontré dans la base de données MySQL ci-dessous) :

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

**NOTE**

Le dialecte de DB2 est `org.hibernate.dialect.DB2Dialect`, pour DB2 dans AS/400, c'est `org.hibernate.dialect.DB2400Dialect`, Oracle, c'est `org.hibernate.dialect.Oracle10gDialect` et pour le Microsoft SQL Server, c'est `org.hibernate.dialect.SQLServerDialect`.

- d. Modifiez le fichier `business-central/WEB-INF/classes/META-INF/persistence.xml` de façon à ce que le moteur de process JBoss BPM Suite puisse utiliser la nouvelle base de données.

Le code ci-dessous démontre les informations de la source de données d'origine pour `persistence.xml` :

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

Modifiez cette valeur pour qu'elle corresponde à la valeur de la source de données définie plus tôt :

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

3. Vous pouvez démarrer dès maintenant le JBoss Web Server pour vous connecter à Business Central.

- a. Exécutez `startup.sh` dans le répertoire `$TOMCAT_HOME/bin`.

```
./startup.sh
```

- b. Naviguer vers <http://localhost:8080/business-central> dans un navigateur web.
- c. Connectez-vous avec le nom d'utilisateur et le mot de passe définis précédemment dans le fichier `tomcat-users.xml`.

2.2.2.3. Installation de Dashbuilder dans Red Hat JBoss Web Server 2.0 (Tomcat 7)

**NOTE**

Avant d'installer Dashbuilder sur Red Hat JBoss Web Server, vous devez vous assurer que vous avez installé et démarré correctement Business Central tel que décrit dans [Section 2.2.2.2, « Installation de Business Central dans Red Hat JBoss Web Server 2.0 \(Tomcat 7\) »](#). Dashbuilder requiert la présence des tableaux de base de données de journalisation, lesquels sont uniquement fournis par Business Central. Si ces tableaux ne sont pas présents dans la base de données avant de reproduire les étapes ci-dessous, vous pourriez rencontrer des erreurs d'initialisation.

Pour installer Dashbuilder dans Red Hat JBoss Web Server, procédez ainsi :

1. Définissez les utilisateurs et les rôles dans `$TOMCAT_DIR/conf/tomcat-users.xml`. Notez que Dashbuilder nécessite que les utilisateurs aient les rôles `admin` et/ou `analyst`. Si vous avez déjà défini ces utilisateurs plus tôt dans Business-Central, vous n'avez pas besoin de les définir à nouveau.
2. Activez un seul sign-on entre Dashbuilder et Business Central en annulant le commentaire des lignes suivantes dans le fichier `$TOMCAT_DIR/conf/server.xml` :

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

3. Comme avec l'installation Business Central, si vous utilisez une base de données autre que la base de données par défaut et H2 intégrée, vous devrez installer une persistance.

Dans cette procédure, vous configurez une source de données avec le nom JNDI `jdbc/dashbuilderDS` défini dans `uniqueName=jdbc/jbpm` dans le fichier `resources.properties` :

- a. Dans `dashbuilder/META-INF/context.xml`, remplacer le nom JNDI de la source de données de l'élément `<Resource>`. L'attribut `uniqueName` se rapporte à la propriété `resource.ds1.uniqueName` définie dans `resources.properties` :

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



NOTE

Suivant votre base de données, vous souhaitez peut-être définir d'autres propriétés. Par exemple, dans un environnement Oracle, cette entrée peut ressembler au listing suivant.

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

- b. Dans `dashbuilder/WEB-INF/web.xml`, ajoutez le nom JNDI de la source de données de l'élément `>res-ref-name<` par le nom de votre source de données :

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

- c. Dans `dashbuilder/META-INF/context.xml`, définissez la fabrique de transactions :

■

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

- d. Mettez à jour le nom JNDI de la base de données dans **dashbuilder/WEB-INF/etc/hibernate.cfg.xml** de l'élément **<session-factory>** :

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

4. Redémarrez le serveur web pour appliquer les changements. Une fois redémarré, vous pourrez vous rendre dans Dashbuilder à partir de Business Central ou directement via : **<http://localhost:8080/dashbuilder>**.

CHAPITRE 3. CONFIGURATIONS SPÉCIALES

3.1. INSTALLATION DE LA PERSISTANCE POUR BUSINESS CENTRAL

Red Hat JBoss BPM Suite est configuré pour utiliser un exemple de source de données avec un nom JNDI (Java Naming and Directory Interface) `java:jboss/datasources/ExampleDS`. Pour Business Central, ce exemple de source de données est situé dans le fichier `business-central.war/WEB-INF/classes/META-INF/persistence.xml`.

Si vous souhaitez configurer JBoss BPM Suite pour utiliser une base de données externe, effectuez les modifications suivantes. Pour le fichier `business-central.war/WEB-INF/classes/META-INF/persistence.xml`:

1. Installez le pilote JDBC (Java Database Connectivity) à l'aide d'une approche modulaire pour une configuration facilitée (veuillez consulter [EAP 6 documentation](#)).
2. Créez une nouvelle source de données suivant l'exemple dans la [documentation EAP 6, section 6.7.1. Exemple de Source de données PostgreSQL](#). Voici la configuration de source de données spécifique à la base de données H2 utilisée par défaut :

```
<subsystem xmlns="urn:jboss:domain:datasources:1.1">
  <datasources>
    <datasource jndi-name="java:jboss/datasources/ExampleDS"
pool-name="ExampleDS" enabled="true" use-java-context="true">
      <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-
1</connection-url>
      <driver>h2</driver>
      <security>
        <user-name>sa</user-name>
        <password>sa</password>
      </security>
    </datasource>
  </drivers>
  <driver name="h2" module="com.h2database.h2">
    <xa-datasource-
class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
  </driver>
</drivers>
</datasources>
```

3. Utilisez le nom JNDI de la source de données pour mettre à jour l'entrée suivante à l'intérieur du fichier `persistence.xml`, qui est par défaut défini sur cette entrée.

```
<jta-data-source>java:jboss/datasources/ExampleDS</jta-data-source>
```



IMPORTANT

Lors de la configuration de votre source de données, assurez-vous d'activer JTA (en ajoutant `jta="true"` à la balise `datasource`).

4. Remplacez le texte suivant par le nom de dialecte hibernate spécifique à la base de données.

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

Par exemple, pour un Oracle Database Express Edition 11g :

```
<property name="hibernate.dialect"
value="org.hibernate.dialect.Oracle10gDialect" />
```

NOTE

Lorsque JBoss BPM Suite utilise Oracle 11 comme source de données, plusieurs messages d'avertissement (**WARN**) sont produits dans les journaux, sans activité Business Central correspondante effectuée. Cela est normal. Pour désactiver ces messages, définissez le niveau de la catégorie `org.hibernate.loader` du journal sur **ERROR** dans le fichier `standalone.xml`.

```
<logger category="org.hibernate.loader">
  <level name="ERROR"/>
</logger>
```

3.2. INSTALLATION DE LA PERSISTANCE POUR DASHBUILDER

Étant donné que Dashbuilder dépend de la configuration de Business Central, assurez-vous que Business Central est configuré selon [Section 3.1, « Installation de la Persistance pour Business Central »](#). Red Hat JBoss BPM Suite 6 est configuré pour utiliser une source de données avec un nom JNDI (Java Naming and Directory Interface) `java:jboss/datasources/ExampleDS`. Si vous souhaitez faire fonctionner l'application avec une base de données différente de H2, par exemple Oracle, MySQL, Postgres ou MS SQL Server, suivez ces étapes.

NOTE

Si vous souhaitez utiliser UTF 8 pour afficher des caractères spéciaux, vérifiez la documentaiton concernant votre base de données et définissez le codage au niveau de la base de données afin que Dashbuilder puisse fonctionner correctement. Par exemple dans MySQL, ajoutez ce qui suit au fichier de configuration du serveur :

```
[mysqld]
character-set-server=utf8
collation-server=utf8_general_ci
```

IMPORTANT

Sur les systèmes de type Unix, changez la valeur par défaut de MySQL `lower_case_table_names` de `0` (sensible à la casse) en `1` (non sensible à la casse). Les requêtes JBoss BPM Suite KPI sont écrites en minuscule mais les noms de tableau sont écrits en casse mixte. En modifiant la propriété `lower_case_table_names`, vous éviterez que des exceptions se produisent ultérieurement.

1. Installez le pilote de base de données et créez une nouvelle source de données suivant l'exemple dans la [documentation EAP 6, section 6.7.1. Exemple de Source de données PostgreSQL](#). Installez le pilote JDBC de manière modulaire pour faciliter la configuration.

2. Créer une base de données vide.
3. Modifiez le fichier **dashbuilder.war/WEB-INF/jboss-web.xml** dont l'entrée par défaut est :

```
<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/ExampleDS</jndi-name>
  </resource-ref>
```

4. Modifiez également les fichiers **WEB-INF/jboss-deployment-structure.xml** à des applications **business-central.war** et **dashbuilder.war**, et ajoutez une dépendance sous la section des dépendances sur le module de pilote JDBC créé précédemment lors de la création de source de données. Le snippet suivant montre un exemple de configuration où **jdbcDriverModuleName** représente le nom du module pilote JBoss EAP 6 JDBC.

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

3.3. INSTALLATION SPÉCIALE DE LA BASE DE DONNÉES IBM DB2

Si vous souhaitez utiliser une base de données IBM DB2 comme source de données sous-jacente pour Business Central, vous devrez augmenter la taille de la page de la base de données. La taille de la page par défaut de 4 ko n'est pas suffisante pour la taille des colonnes du tableau Dashbuilder.

Lors de la création de la base de données, ajustez la taille de la page à 16384 comme dans l'exemple ci-dessous :

Exemple 3.1. Ajuster la taille de la page

```
CREATE DATABASE dashb PAGESIZE 16384
```

Cette augmentation de la taille de la page pour la base de données sous-jacente doit être effectuée avant la première exécution de BPM Suite.

CHAPITRE 4. RÔLES ET UTILISATEURS

4.1. DÉFINIR LES RÔLES

Avant de démarrer le serveur et de vous connecter au Business Central, vous devrez créer des comptes utilisateur. Cette section décrit les différents rôles qui sont utilisés dans Red Hat JBoss BPM Suite :

- **admin** : les utilisateurs ayant un rôle admin sont les administrateurs de l'application. Les administrateurs peuvent gérer les utilisateurs, les référentiels (les créer et les cloner) et avoir un accès total pour effectuer des changements dans l'application. Les admins ont accès à toutes les régions du système.
- **developer** : un développeur a accès à presque toutes les fonctionnalités et peut gérer des règles, des modèles, des flux de processus, des formulaires et tableaux de bord. Ils peuvent gérer le référentiel des ressources, ils peuvent créer, construire et déployer des projets et ils peuvent même utiliser Red Hat JBoss Developer Studio pour afficher les processus. Seules certaines fonctions administratives comme la création et le clonage d'un nouveau référentiel sont masquées au rôle de développeur.
- **analyst** : un rôle d'analyste a accès à toutes les fonctions de haut niveau pour modéliser et exécuter des projets. Cependant, l'accès **Authoring** → **Administration** n'est pas disponible aux utilisateurs ayant un rôle d'analyste. Certaines fonctions de bas niveau qui ciblent les développeurs, comme l'affichage **Deployment** → **Artifact Repository** sont cachées à ce rôle. Cependant, le bouton **Build & Deploy** est accessible au rôle d'analyste quand celui-ci utilise l'éditeur de projet.
- **utilisateur** : l'utilisateur travaille sur les listes de tâches d'entreprise qui sont utilisées pour l'exploitation d'un certain processus. Un utilisateur avec ce rôle peut accéder au tableau de bord et gérer les processus.
- **manager** : un manager est une personne qui voit un système et s'intéresse aux statistiques sur les processus d'entreprise et la performance, les indicateurs clés de l'entreprise et autres rapports du système. Un utilisateur avec ce rôle a accès à la BAM uniquement.



NOTE

Saisissez les rôles mentionnés ci-dessus lors du processus de création d'un utilisateur.

4.2. CRÉATION D'UTILISATEURS

Pour commencer à ajouter de nouveaux utilisateurs, vous devrez exécuter le script `add-user.sh` sur un système Unix ou le fichier `add-user.bat` sur un système Windows à partir du répertoire `bin` EAP.

1. Exécutez `./add-user.sh` sur un système Unix ou `add-user.bat` sur un système Windows à partir du répertoire `bin`.
2. Saisissez `b` pour sélectionner un Utilisateur d'application à l'invite type d'utilisateur, et appuyez sur `Enter`.
3. Acceptez le domaine (realm) par défaut (`ApplicationRealm`) en appuyant sur `Enter`.
4. À l'invite du nom utilisateur, saisissez un nom utilisateur et le confirmer. Par exemple : `helloworlduser`.

**NOTE**

Assurez-vous que les utilisateurs et les groupes ne portent pas le même nom. Par exemple, si un groupe porte le nom `admin` ne donnez pas le nom `admin` à un utilisateur.

5. Créez le mot de passe utilisateur à l'invite du mot de passe et saisissez à nouveau le mot de passe. Par exemple : `HelloWorld@123`.

**NOTE**

Le mot de passe doit comporter au moins 8 caractères et contenir des caractères alphabétiques majuscules et minuscules (par exemple A-Z, a-z), au moins un caractère numérique (par exemple 0-9) et au moins un caractère spécial (p. ex. `~!@#$%^*()-_+=`).

6. Saisissez une liste de rôles séparée par des virgules dont l'utilisateur aura besoin sur invite des rôles (veuillez consulter [Section 4.1, « Définir les rôles »](#)).

Les utilisateurs de Business Central doivent au moins posséder le rôle `analyst`, et les utilisateurs Dashbuilder le rôle `admin`. Les rôles doivent être saisis sous la forme d'une liste séparée par des virgules.

7. Confirmer que vous souhaitez ajouter l'utilisateur.
8. Saisir oui à la prochaine invite (pour activer le clustering dans le future si possible)

CHAPITRE 5. TEST DE L'INSTALLATION

5.1. DÉMARRAGE DU SERVEUR



NOTE

Si vous avez installé JBoss BPM Suite en utilisant la versions déployable générique sur Red Hat Java Web Server, [Section 2.2, « L'installation d'un groupement déployable standard »](#) contient les instructions pour lancer le serveur. Vous pouvez ignorer la discussion suivante.

Une fois le serveur JBoss BPM Suite installé sur JBoss EAP 6, vous pouvez l'exécuter en mode *autonome* ou en mode *domaine*.

5.1.1. Mode autonome

Si vous avez utilisé le programme d'installation JAR pour JBoss EAP 6, auquel il est fait référence dans la [Section 2.1, « Installation de Red Hat JBoss BPM Suite Installer »](#), vous pourrez exécuter le serveur dans l'un des modes autonomes suivants : *standalone* (autonome) ou *standalone-secure* (autonome-sécurisé).



NOTE

Si vous choisissez le fichier ZIP déployable pour JBoss EAP, les étapes de configuration sont décrites dans [Section 2.1.3, « Télécharger un ZIP déployable pour Red Hat JBoss EAP »](#).

Le script de démarrage par défaut, `standalone.sh` qui fait partie de Red Hat JBoss BPM Suite est optimisé dans un souci de performance. Pour exécuter votre serveur en mode de performance, suivez ces instructions :

1. Avec la ligne de commande, déplacez-vous sur le répertoire `$EAP_HOME/bin/`.
2. Dans un environnement Unix, veuillez exécuter :

```
./standalone.sh
```

Dans un environnement Windows, veuillez exécuter :

```
./standalone.bat
```

Red Hat JBoss BPM Suite est aussi fourni avec un script séparé, `standalone-secure.sh` optimisé pour la sécurité. Ce script applique une politique de sécurité par défaut qui protège contre une vulnérabilité de sécurité connue.



NOTE

Il est conseillé que l'environnement de production utilise le script `standalone-secure.sh`.



AVERTISSEMENT

L'utilisation d'un gestionnaire de sécurité impose une baisse sensible de la performances dont vous devez être conscient. Le compromis entre sécurité et performance s'impose en tenant compte des circonstances individuelles. Voir [Section 5.2, « Java Security Manager et Gestion de performance »](#).

Pour exécuter le serveur en mode sécurisé avec ce script, vous devrez faire ce qui suit :

1. Avec la ligne de commande, déplacez-vous sur le répertoire `$EAP_HOME/bin/`.
2. Dans un environnement Unix, veuillez exécuter :

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

Dans un environnement Windows, veuillez exécuter :

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



NOTE

Si vous avez installé JBoss BPM Suite à l'aide du programme d'installation, une option pour appliquer la politique de sécurité vous sera donnée au moment de l'installation. Le programme d'installation ne fournit pas un script `standalone-secure.sh` séparé.

5.1.2. Mode Domaine

Si vous avez utilisé le programme d'installation JAR décrit dans la [Section 2.1, « Installation de Red Hat JBoss BPM Suite Installer »](#), JBoss BPM Suite est déjà configuré pour être exécuté en mode domaine.



NOTE

Si vous choisissez le fichier ZIP déployable pour JBoss EAP, les étapes de configuration du mode domaine sont décrites dans [Section 2.1.3, « Télécharger un ZIP déployable pour Red Hat JBoss EAP »](#).

Pour lancer JBoss BPM Suite en mode domaine, veuillez reproduire les étapes suivantes :

1. Avec la ligne de commande, déplacez-vous sur le répertoire `$EAP_HOME/bin/`.
2. Dans un environnement Unix, veuillez exécuter :

```
./domain.sh
```

Dans un environnement Windows, veuillez exécuter :

```
./domain.bat
```

5.2. JAVA SECURITY MANAGER ET GESTION DE PERFORMANCE

Tel que mentionné précédemment, en activant JSM, le gestionnaire de sécurité Java, à sandbox, l'évaluation de scripts MVEL dans Red Hat JBoss BPM Suite ajoute une charge à la performance dans des environnements à forte charge. On doit conserver à l'esprit les environnements et les marqueurs de performance lorsqu'on déploie une application JBoss BPM Suite. Utilisez les instructions suivantes pour déployer des applications JBoss BPM Suite sécurisées et performantes.

- Dans des environnements de charge élevée où la performance est critique, il est recommandé de seulement déployer des applications qui ont été développées sur d'autres systèmes et adéquatement évaluées. Il est également recommandé de ne pas créer des utilisateurs avec le rôle d'analyste sur de tels systèmes. Si ces précautions sont respectées, il est plus sûr de laisser JSM désactivé sur ces systèmes afin d'éviter toute dégradation de performance.
- Dans les environnements de test et de développement sans charges élevées ou dans des environnements où la règle et le processus de création sont exposés à des réseaux externes, il est recommandé d'avoir JSM activé afin d'obtenir une meilleure sécurité d'évaluation correctement isolée (sandboxed) de MVEL.

Autoriser les utilisateurs ayant un rôle d'analyste à se connecter dans la console Business Central avec JMS désactivé n'est pas sécurisé, ni recommandé.

5.3. CONNECTEZ-VOUS À BUSINESS CENTRAL

Connectez-vous à Business Central une fois que le serveur aura démarré.

1. Naviguer dans <http://localhost:8080/business-central> 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/business-central>.
2. Connectez-vous à l'aide des informations d'identification qui ont été créées lors de l'installation. Par exemple : Utilisateur = `helloworlduser` et mot de passe = `HelloWorld@123`.

CHAPITRE 6. MISE EN CLUSTER

Lors de la mise en cluster de Red Hat JBoss BPM Suite, demandez-vous quels sont les composants à mettre en cluster. Vous pouvez mettre en cluster :

- le référentiel GIT : le référentiel VFS (virtual-file-system) détenant les ressources d'entreprise afin que tous les nœuds de cluster utilisent le même référentiel
- Serveur d'exécution et applications Web : le serveur runtime présent dans le conteneur (tel que Red Hat JBoss EAP) ainsi que les applications web BRMS et BPM Suite pour que les nœuds se partagent les mêmes données de runtime.

Pour savoir comment mettre l'application en cluster, veuillez consulter la documentation relative à la mise en cluster du conteneur.

- Base de données d'arrière-plan : base de données avec les données de statut, telles que, entre autre, les instances de processus, sessions KIE et journal d'historique pour les basculements.

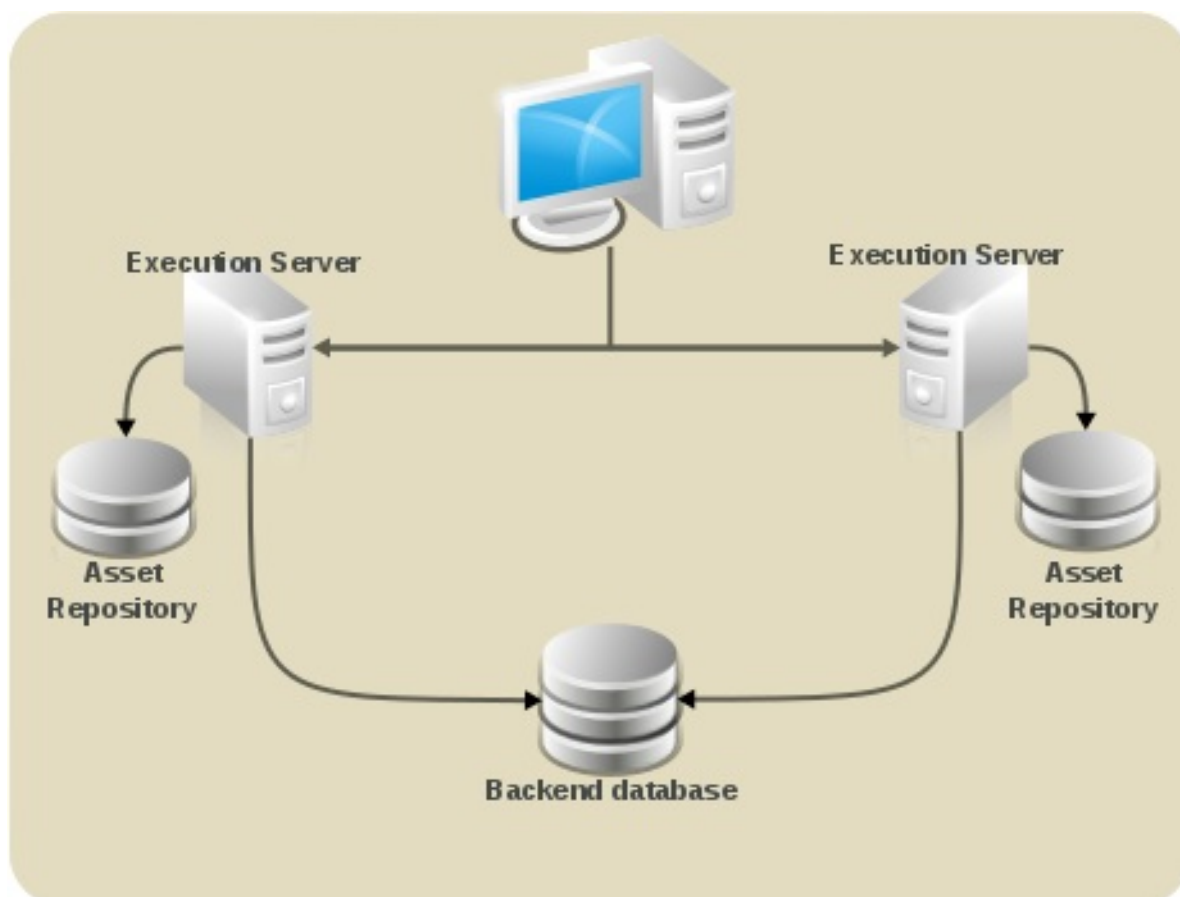


Figure 6.1. Schéma du système Red Hat JBoss BPM Suite avec composants de système individuel

MÉCANISME DE MISE EN CLUSTER DU RÉFÉRENTIEL GIT

Pour mettre le référentiel GIT en cluster, les produits suivants sont utilisés :

- Apache Zookeeper regroupe toutes les parties.
- Apache Helix est le composant de gestion de cluster qui enregistre tous les détails de cluster (le cluster, les nœuds, les ressources).

L'environnement de runtime, c'est-à-dire le serveur d'Exécution, utilise ce qui suit pour fournir les fonctionnalités de mise en cluster :

- uberfire framework, qui fournit le pilier des applications web

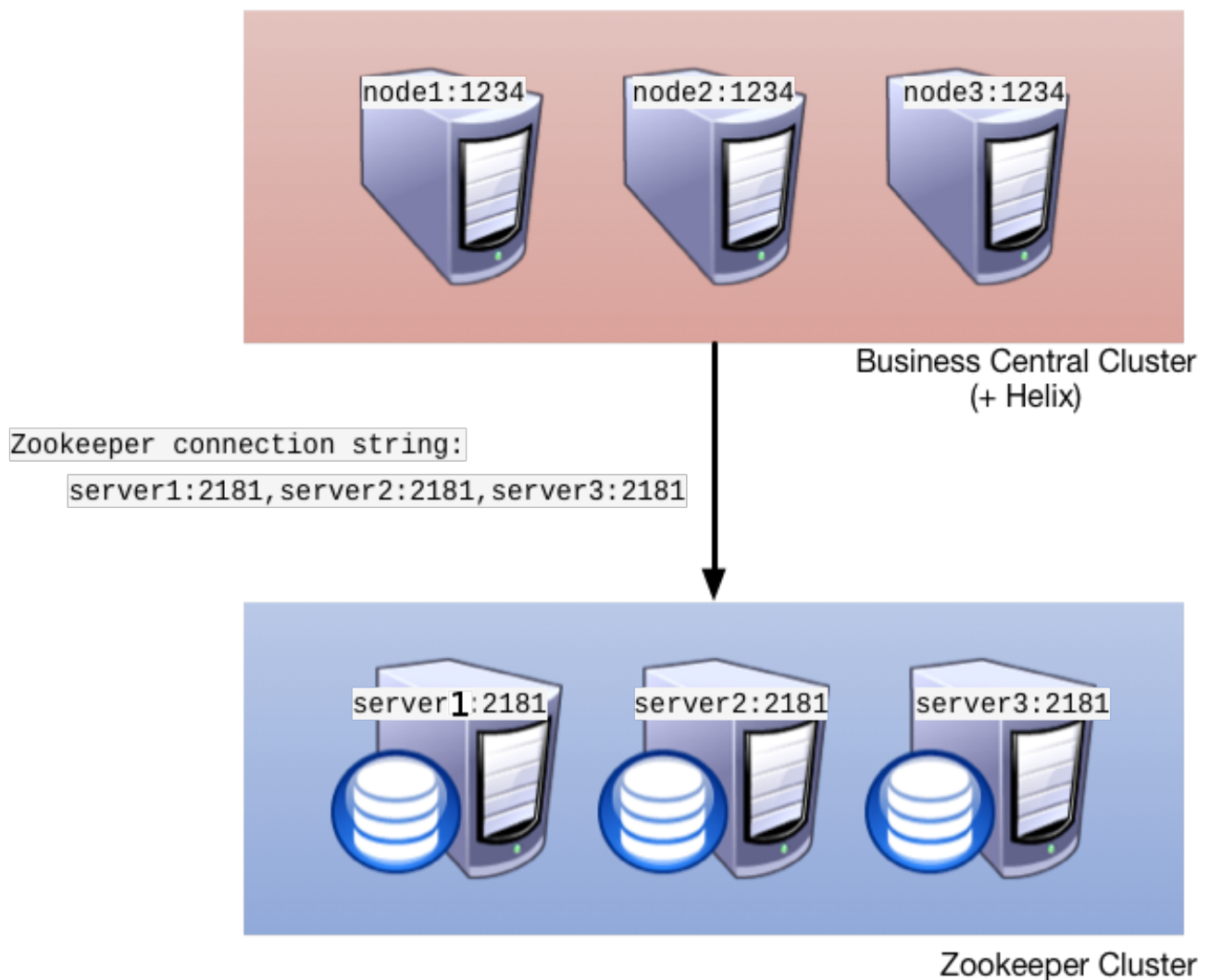


Figure 6.2. Schéma de mise en cluster avec Helix et Zookeeper

Une configuration de mise en cluster typique implique ce qui suit :

- Installation du cluster avec Zookeeper et Helix
- Installation de la base de données d'arrière-plan avec une configuration et des tableaux Quartz
- Configuration de la mise en cluster sur votre conteneur (cette documentation ne fournit que des instructions relatives à la mise en cluster pour Red Hat JBoss EAP 6)

MISE EN CLUSTER DES RÉFÉRENTIELS MAVEN

Différentes opérations dans Business Central publient des JAR vers le référentiel Maven interne de Business Central.

Ce référentiel existe sur le système de fichier du serveur d'application comme fichiers réguliers et n'est pas compatible avec les clusters. Ce dossier n'est pas synchronisé entre les différents nœuds du cluster. Il doit être synchronisé à l'aide d'outils externes comme `rsync`.

À la place d'utiliser un outil de synchronisation externe, il est possible d'installer la propriété de système `org.guvnor.m2repo.dir` sur chaque nœud de cluster pour diriger vers un SAN ou NAS. Dans ce cas, il n'est pas nécessaire de mettre le référentiel Maven en cluster.

6.1. MISE EN CLUSTER AVEC JBOSS EAP

Pour installer JBoss BPM Suite en mode clusterisé, nous vous recommandons d'utiliser le programme d'installation JAR, qui offre un échantillon d'installation prêt à l'emploi. Vous pouvez également installer la mise en cluster avec le fichier ZIP déployable pour EAP.

6.1.1. Mise en cluster à l'aide du programme d'installation JAR



NOTE

Le programme d'installation JAR fournit uniquement un échantillon de l'installation, un ajustement de la configuration sera nécessaire pour mieux correspondre aux besoins de votre projet.

Utiliser le programme d'installation JAR décrit dans la [Section 2.1, « Installation de Red Hat JBoss BPM Suite Installer »](#), vous pouvez paramétrer une configuration de cluster de base de JBoss BPM Suite.

La configuration automatique crée trois instances ZooKeeper, un cluster Helix qui utilise ces instances, et deux magasins de données Quartz (un géré et un non-géré). Cette installation JBoss BPM Suite consiste de deux nœuds EAP qui partagent un référentiel Maven, utilisent Quartz pour coordonner les tâches minutées, et dont `business-central.war`, `dashbuilder.war`, et `kie-server.war` sont déployés. Pour personnaliser l'installation afin qu'elle corresponde à votre scénario, ou pour utiliser la mise en cluster avec le fichier ZIP déployable, veuillez consulter [Section 6.1.2, « Configuration personnalisée \(ZIP déployable\) »](#). Vous pouvez également obtenir davantage d'informations dans la *documentation JBoss EAP*.

Suivez le processus d'installation décrit dans [Section 2.1.1, « Installation de Red Hat JBoss BPM Suite à l'aide du programme d'installation »](#) et sélectionnez *Installer une configuration clusterisée* dans *Configuration avancées du runtime*. Après avoir cliqué sur **Suivant**, vous serez guidé à travers les étapes suivantes :



NOTE

Les étapes répertoriées ici décrivent l'installation de l'interface utilisateur graphique. Les étapes pour l'installation de la console sont analogues.

1. Sélectionner le fournisseur JDBC

Sur cet écran, veuillez sélectionner le fournisseur JDBC dans `list`. Vous devrez fournir le ou les JAR correspondants de l'une des manières suivantes :

- Sélectionnez un ou plusieurs fichiers sur le système de fichiers
- Veuillez fournir un ou plusieurs URL. Le programme d'installation télécharge les fichiers automatiquement.

Le programme d'installation copie ensuite le ou les JAR sur l'emplacement approprié sous le répertoire `$EAP_HOME/modules`, où un fichier `module.xml` correspondant est également créé automatiquement.

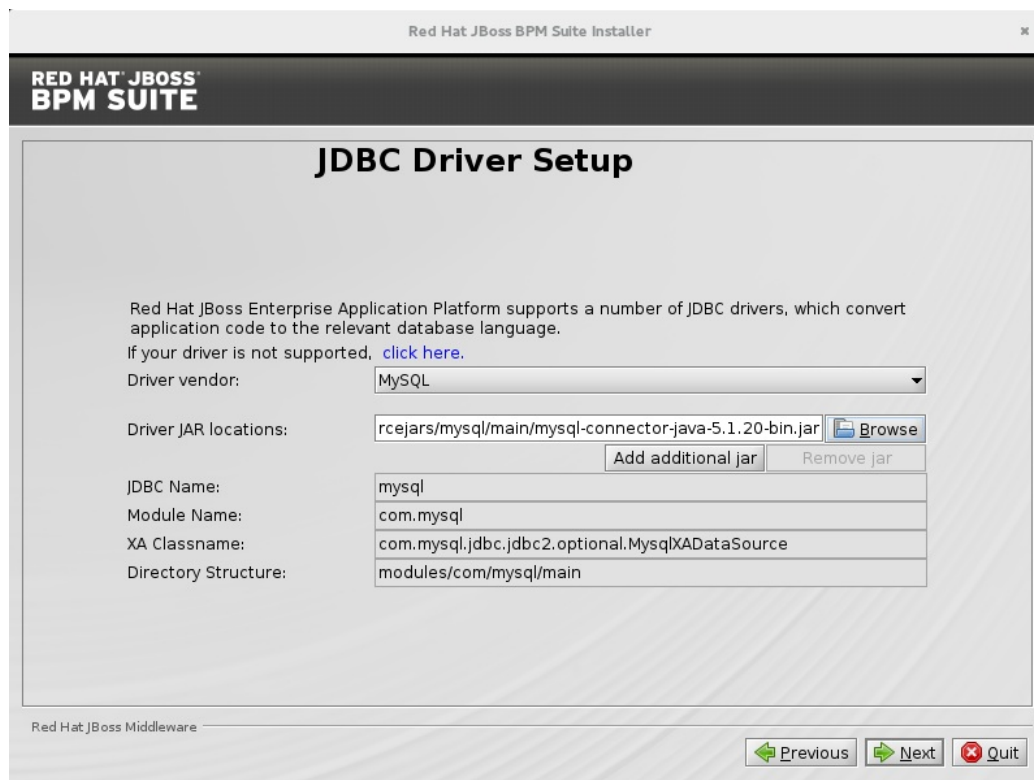


Figure 6.3. Installation du pilote JDBC

2. Configurer la connexion Quartz

Sur l'écran suivant, veuillez fournir les données à la base de données pour Quartz. Le programme d'installation crée automatiquement le fichier de définition Quartz (`$EAP_HOME/domain/configuration/quartz-definition.properties`) et deux sources de données Quartz dans le fichier de configuration du domaine `$EAP_HOME/domain/domain.xml`. Vous pouvez modifier les fichiers après avoir terminé l'installation.



NOTE

Pendant l'installation, les scripts DDL Quartz seront exécutés sur la base de données sélectionnée dans cette étape. Ces scripts rendent des changements nécessaires pour permettre à Quartz d'opérer (d'ajouter des tables, etc...), et se trouvent dans `$EAP_HOME/jboss-brms-bpmsuite-6.2-supplementary-tools/ddl-scripts` comme référence (Il n'est pas nécessaire de les modifier de quelque manière que ce soit).

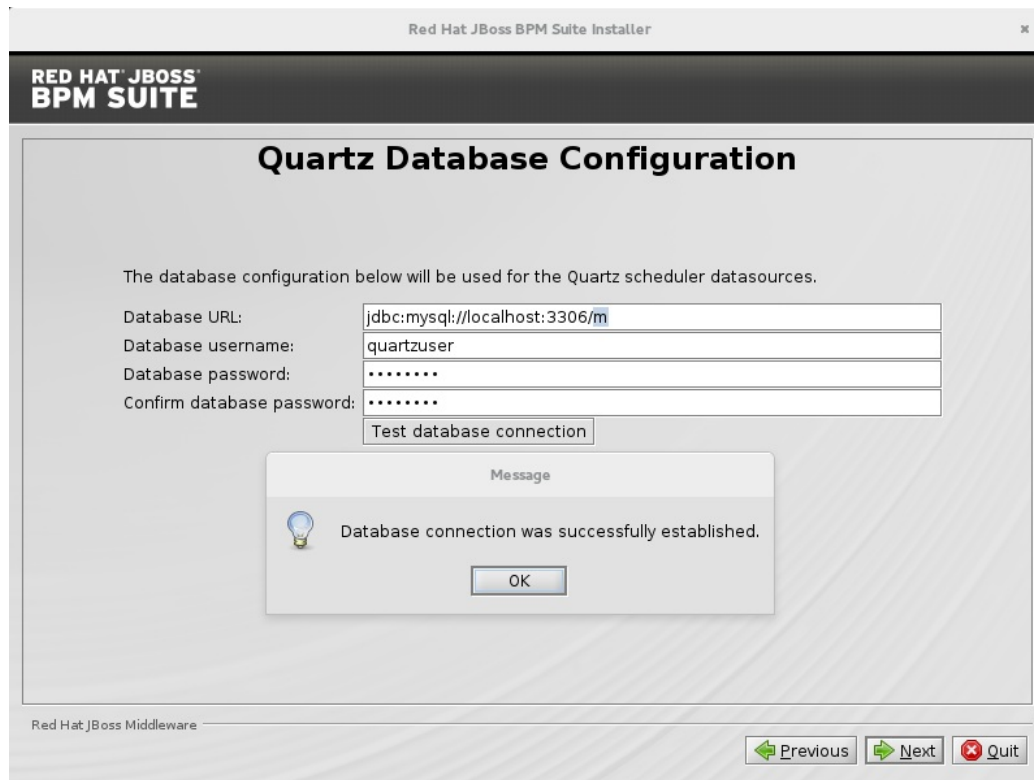
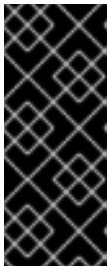


Figure 6.4. Installation de la base de données Quartz

3. Cliquez sur **Suivant** pour lancer l'installation.



IMPORTANT

Lors de l'utilisation du programme d'installation JAR, les archives war sont automatiquement créées à partir des applications résidant dans **\$EAP_HOME/standalone/deployments/**. Cela signifie que de l'espace supplémentaire est nécessaire car les applications existent à la fois dans un état compressé et non-compressé dans le stockage pendant l'installation.

Trois instances de ZooKeeper sont automatiquement créées dans **\$EAP_HOME/jboss-brms-bpmsuite-6.2-supplementary-tools/** (noms des répertoires **zookeeper-one**, **zookeeper-two**, et **zookeeper-three**).

Dans le répertoire **\$EAP_HOME/jboss-brms-bpmsuite-6.2-supplementary-tools/helix-core**, vous pourrez trouver la configuration Helix par défaut et les scripts pour lancer le cluster—**startCluster.sh** pour UNIX et **startCluster.bat** pour Windows.

Une fois l'installation terminée, veuillez **NE PAS** sélectionner d'exécuter le serveur immédiatement. Vous devrez d'abord lancer le cluster en déplaçant dans le répertoire **\$EAP_HOME/jboss-brms-bpmsuite-6.2-supplementary-tools/helix-core** le script de lancement mentionné ci-dessus et en l'exécutant :

Sur les systèmes Unix :

```
./startCluster.sh
```

Sur Windows :

```
./startCluster.bat
```

Ce script lance le cluster Helix et les instances ZooKeeper. Uniquement après cela, lancez le serveur EAP en mode domaine en déplaçant sur le répertoire `$EAP_HOME/bin` et en exécutant :

Sur les systèmes Unix :

```
./domain.sh
```

Sur Windows :

```
./domain.bat
```

Vous avez désormais un cluster JBoss BPM Suite fonctionnant correctement.

6.1.2. Configuration personnalisée (ZIP déployable)

Lors de la mise en cluster JBoss EAP, un contrôleur de domaine JBoss EAP unique existe avec d'autres esclaves JBoss EAP s'y connectant en tant qu'utilisateur de gestion. Les déploiements de Business Central et dashbuilder peuvent être effectués en tant qu'utilisateur de gestion sur un contrôleur de domaine et les déploiements WAR seront distribués à d'autres membres du cluster JBoss EAP.

Pour configurer la mise en cluster sur Red Hat JBoss EAP 6, veuillez effectuer ce qui suit :

1. Configurer ZooKeeper et Helix selon [Section 6.2.1, « Paramétrer un cluster »](#).
2. Configurer Quartz selon [Section 6.2.2, « Installation de Quartz »](#).
3. Installez votre pilote JDBC comme module principal : copiez le jar du pilote sur `$EAP_HOME/modules/system/layers/base/` et créez un fichier `module.xml` dans le répertoire.
4. Veuillez modifier le fichier `module.xml` d'après le module XSD correspondant.

Exemple 6.1. Le contenu du fichier module.xml pour une source de données PostgreSQL

```
<module xmlns="urn:jboss:module:1.0" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-jdbc.jar"/>
  </resources>

  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

5. Configurez la source de données pour le serveur : ouvrez le fichier `host.xml` ou `standalone.xml` pour modification, selon le PROFIL utilisé situé dans `$EAP_HOME/PROFILE/`, localisez le profil `full` et effectuez ce qui suit :

- a. Ajoutez la définition de la source de données principale utilisée par JBoss BPM Suite.

Exemple 6.2. La source de données PostgreSQL définie comme la source de données JBoss BPM Suite principale

```
<datasource jndi-name="java:jboss/datasources/psbpmsDS"
    pool-name="postgresDS" enabled="true" use-java-
context="true">
    <connection-
url>jdbc:postgresql://localhost:5432/jbpm</connection-url>
    <driver>postgres</driver>
    <security>
    <user-name>bpms</user-name>
    <password>bpms</password>
    </security>
</datasource>
```

- b. Ajoutez la définition de la source de données pour le service Quartz.

Exemple 6.3. La source de données PostgreSQL définie comme la source de données Quartz

```
<datasource jta="false" jndi-
name="java:jboss/datasources/quartzNotManagedDS"
    pool-name="quartzNotManagedDS" enabled="true" use-java-
context="true">
    <connection-
url>jdbc:postgresql://localhost:5432/jbpm</connection-url>
    <driver>postgres</driver>
    <security>
    <user-name>bpms</user-name>
    <password>bpms</password>
    </security>
</datasource>
```

- c. Définissez le pilote de la source de données.

Exemple 6.4. La définition du pilote PostgreSQL

```
<driver name="postgres" module="org.postgresql">
    <xa-datasource-class>org.postgresql.xa.PGXADatasource</xa-
datasource-class>
</driver>
```

6. Configurer des nœuds de serveurs individuels dans l'élément **main-server-group** du fichier **\$EAP_HOME/domain/configuration/host.xml** dont les propriétés sont définies dans [Tableau 6.1, « Propriétés de nœud Cluster »](#) :

Veuillez noter que lors de la configuration d'un cluster JBoss EAP avec Zookeeper, il est possible que le nombre de nœuds JBoss EAP soit différent de celui des nœuds de Zookeeper

(Zookeeper doit avoir un nombre impair de nœuds). Il est conseillé que Zookeeper et JBoss EAP ait le même nombre de nœuds.

Tableau 6.1. Propriétés de nœud Cluster

Nom de propriété	Valeur	Description
<code>jboss.node.name</code>	<code>nodeOne</code>	Nom de nœuds unique dans le cluster
<code>org.quartz.properties</code>	<code>/bpms/quartz-definition.properties</code>	chemin d'accès absolu vers le fichier de configuration Quartz
<code>org.uberfire.cluster.autostart</code>	<code>true</code>	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.
<code>org.uberfire.cluster.id</code>	<code>bpms-cluster</code>	Nom de cluster de Helix
<code>org.uberfire.cluster.local.id</code>	<code>nodeOne_12345</code>	ID unique du nœud de cluster Helix Veuillez noter que : est remplacé par __.
<code>org.uberfire.cluster.vfs.lock</code>	<code>vfs-repo</code>	nom de la ressource défini sur le cluster Helix
<code>org.uberfire.cluster.zk</code>	<code>server1:2181</code>	Emplacement Zookeeper
<code>org.uberfire.metadata.index.dir</code>	<code>/home/jbpm/node[N]/index</code>	emplacement où l'index de recherche doit être créé (pris en charge par Apache Lucene)
<code>org.uberfire.nio.git.daemon.host</code>	<code>nodeOne</code>	le nom de la machine d'hôte démon dans un cluster physique.
<code>org.uberfire.nio.git.daemon.port</code>	<code>9418</code>	port utilisé par le référentiel VFS pour accepter les connexions client Le port doit être unique pour chaque membre de cluster.
<code>org.uberfire.nio.git.dir</code>	<code>/home/jbpm/node[N]/repo</code>	Emplacement du référentiel GIT (VFS) sur le nœud [N]

Nom de propriété	Valeur	Description
org.uberfire.nio.git.ssh.host	nodeOne	le nom de la machine d'hôte SSH dans un cluster physique.
org.uberfire.nio.git.ssh.port	8003	Le numéro de port unique pour l'accès SSH au référentiel GIT pour un cluster exécuté sur des machines physiques.
org.uberfire.nio.git.ssh.hostport and org.uberfire.nio.git.daemon.hostport	8003 and 9418	Dans un environnement virtualisé, le port externe à utiliser

Exemple 6.5. Configuration du Cluster nodeOne

```
<system-properties>

  <property name="org.uberfire.nio.git.dir"
value="/tmp/bpms/nodeone" boot-time="false"/>
  <property name="jboss.node.name" value="nodeOne" boot-
time="false"/>

  <property name="org.uberfire.cluster.id" value="bpms-cluster"
boot-time="false"/>
  <property name="org.uberfire.cluster.zk"
value="server1:2181,server2:2181,server3:2181" boot-time="false"/>
  <property name="org.uberfire.cluster.local.id"
value="nodeOne_12345" boot-time="false"/>
  <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo"
boot-time="false"/>
  <property name="org.uberfire.cluster.autostart" value="true"
boot-time="true"/>

  <property name="org.uberfire.nio.git.daemon.host"
value="nodeOne" />
  <property name="org.uberfire.nio.git.daemon.port" value="9418"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.hostport"
value="9418" boot-time="false"/>

  <property name="org.uberfire.nio.git.ssh.port" value="8003"
boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.hostport" value="8003"
boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.host" value="nodeOne"
/>

  <property name="org.uberfire.metadata.index.dir"
value="/tmp/jbpm/nodeone" boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/tmp/jbpm/nodeone" boot-time="false"/>
```

```

    <property name="org.quartz.properties"
value="/tmp/jbpm/quartz/quartz-db-postgres.properties" boot-
time="false"/>

</system-properties>

```

Exemple 6.6. Configuration du Cluster nodeTwo

```

<system-properties>

    <property name="org.uberfire.nio.git.dir"
value="/tmp/bpms/nodetwo" boot-time="false"/>
    <property name="jboss.node.name" value="nodeTwo" boot-
time="false"/>

    <property name="org.uberfire.cluster.id" value="bpms-cluster"
boot-time="false"/>
    <property name="org.uberfire.cluster.zk"
value="server1:2181,server2:2181,server3:2181" boot-time="false"/>
    <property name="org.uberfire.cluster.local.id"
value="nodeTwo_12346" boot-time="false"/>
    <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo"
boot-time="false"/>
    <property name="org.uberfire.cluster.autostart" value="true"
boot-time="true"/>

    <property name="org.uberfire.nio.git.daemon.host"
value="nodeTwo" />
    <property name="org.uberfire.nio.git.daemon.port" value="9418"
boot-time="false"/>
    <property name="org.uberfire.nio.git.daemon.hostport"
value="9418" boot-time="false"/>

    <property name="org.uberfire.nio.git.ssh.port" value="8003"
boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.hostport" value="8003"
boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.host" value="nodeTwo"
/>

    <property name="org.uberfire.metadata.index.dir"
value="/tmp/jbpm/nodetwo" boot-time="false"/>
    <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/tmp/jbpm/nodetwo" boot-time="false"/>

    <property name="org.quartz.properties"
value="/tmp/jbpm/quartz/quartz-db-postgres.properties" boot-
time="false"/>

</system-properties>

```

Exemple 6.7. Configuration du Cluster nodeThree

```

<system-properties>

  <property name="org.uberfire.nio.git.dir"
value="/tmp/bpms/nodethree" boot-time="false"/>
  <property name="jboss.node.name" value="nodeThree" boot-
time="false"/>

  <property name="org.uberfire.cluster.id" value="bpms-cluster"
boot-time="false"/>
  <property name="org.uberfire.cluster.zk"
value="server1:2181,server2:2181,server3:2181" boot-time="false"/>
  <property name="org.uberfire.cluster.local.id"
value="nodeThree_12347" boot-time="false"/>
  <property name="org.uberfire.cluster.vfs.lock" value="vfs-repo"
boot-time="false"/>
  <property name="org.uberfire.cluster.autostart" value="true"
boot-time="true"/>

  <property name="org.uberfire.nio.git.daemon.host"
value="nodeThree" />
  <property name="org.uberfire.nio.git.daemon.port" value="9418"
boot-time="false"/>
  <property name="org.uberfire.nio.git.daemon.hostport"
value="9418" boot-time="false"/>

  <property name="org.uberfire.nio.git.ssh.port" value="8003"
boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.hostport" value="8003"
boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.host"
value="nodeThree" />

  <property name="org.uberfire.metadata.index.dir"
value="/tmp/jbpm/nodethree" boot-time="false"/>
  <property name="org.uberfire.nio.git.ssh.cert.dir"
value="/tmp/jbpm/nodethree" boot-time="false"/>

  <property name="org.quartz.properties"
value="/tmp/jbpm/quartz/quartz-db-postgres.properties" boot-
time="false"/>

</system-properties>

```

7. Ajoutez les utilisateurs de gestion tel qu'indiqué dans le *Guide d'Administration et de Configuration* pour Red Hat JBoss EAP et les utilisateurs d'application tel qu'indiqué dans le *Guide d'Administration et de Configuration Red Hat JBoss BPM Suite*
8. Déplacez-vous sur le répertoire **\$EAP_HOME/bin** et lancez le serveur d'applications en mode domaine :

Sur les systèmes Unix :

```
./domain.sh
```

Sur Windows :

```
./domain.bat
```

9. Vérifiez que les nœuds sont disponibles.

Déployez l'application Business Central vers vos serveurs :

1. Remplacez la persistance prédéfinie de l'application par la base de données requise (PostgreSQL) : appliquez les modifications suivantes dans `persistence.xml` :
 - a. nom `jta-data-source` de la source défini sur le serveur d'application (`java:jboss/datasources/psbpmsDS`)
 - b. dialecte hibernate doit correspondre au dialecte de la source de données (`org.hibernate.dialect.PostgreSQLDialect`)
2. Connectez-vous en tant qu'utilisateur de gestion à la console Administration du serveur de votre domaine et ajoutez les nouveaux déploiements à l'aide de l'affichage Runtime de la console. Une fois le déploiement ajouté au domaine, attribuez-le au groupe de serveur approprié (`main-server-group`).



NOTE

Il est important que les utilisateurs vérifient explicitement la disponibilité de l'unité de déploiement de chaque membre de cluster.

Lorsqu'une unité de déploiement est créée sur un nœud de cluster, sa distribution auprès de tous les membres cluster peut prendre un certain temps. Le statut de déploiement peut être vérifié par UI et REST, cependant, si la requête est dirigée vers le nœud où le déploiement était initialement émis, la réponse est **deployed**. Toute requête ciblant cette unité de déploiement envoyé à un membre de cluster échoue avec `DeploymentNotFoundException`.

6.2. MISE EN CLUSTER D'UN ENSEMBLE GÉNÉRIQUE

6.2.1. Paramétrer un cluster

Pour mettre votre référentiel GIT (VFS) en cluster dans Business Central, veuillez effectuer ce qui suit (si vous n'utilisez pas Business Central, vous pouvez ignorer cette section) :

1. Téléchargez `jboss-bpmsuite-brms-VERSION-supplementary-tools.zip`, qui contient Apache Zookeeper, Apache Helix et les scripts Quartz DDL. Après le téléchargement, décompressez l'archive : les répertoires `Zookeeper` (`$ZOOKEEPER_HOME`) et `Helix` (`$HELIX_HOME`) sont créés.
2. Veuillez maintenant configurer ZooKeeper :
 - a. Dans le répertoire ZooKeeper, allez dans le répertoire `conf` et effectuez ce qui suit :

```
cp zoo_sample.cfg zoo.cfg
```

- b. Ouvrez `zoo.cfg` pour effectuer des modifications et ajustez les paramètres dont :

```
# the directory where the snapshot is stored.
dataDir=$ZOOKEEPER_HOME/data/
# the port at which the clients connects
clientPort=2181
server.1=server1:2888:3888
server.2=server2:2888:3888
server.3=server3:2888:3888
```

Assurez-vous de l'existence et de l'accessibilité de l'emplacement `dataDir`.

- c. Attribuez un ID de nœud à chaque membre qui exécutera ZooKeeper. Par exemple, utilisez 1, 2 et 3 pour le nœud 1, 2 et 3 respectivement. ZooKeeper doit avoir un nombre impair d'instances, au moins trois, pour pouvoir récupérer d'un échec.

L'ID de nœud est spécifié dans un champ appelé `myid` sous le répertoire de données de ZooKeeper sur chaque nœud. Par exemple, sur le nœud 1, exécutez : `$ echo "1" > /zookeeper/data/myid`

3. Installez ZooKeeper afin de l'utiliser lors de la création du cluster avec Helix :

- a. Allez sur le répertoire `$ZOOKEEPER_HOME/bin/` et démarrez ZooKeeper :

```
./zkServer.sh start
```

Vous pouvez vérifier le journal de ZooKeeper dans le fichier `$ZOOKEEPER_HOME/bin/zookeeper.out`. Vérifiez ce journal pour vous assurer que « l'ensemble » (cluster) a bien été créé. Un des nœuds doit être élu comme nœud principal, suivi par les deux autres.

4. Une fois l'ensemble ZooKeeper démarré, l'étape suivante consiste à configurer et démarrer Helix. Helix ne doit être configuré qu'une fois, à partir d'un nœud unique. La configuration est ensuite stockée par l'ensemble ZooKeeper et partagée si besoin.

Installez le cluster avec le serveur ZooKeeper comme maître de la configuration :

- a. Créez le cluster en fournissant l'hôte ZooKeeper et importez en tant que liste séparée par des virgules :

```
$HELIX_HOME/bin/helix-admin.sh --zkSvr
ZOOKEEPER_HOST:ZOOKEEPER_PORT --addCluster CLUSTER_NAME
```

- b. Ajoutez vos nœuds au cluster :

```
$HELIX_HOME/bin/helix-admin.sh --zkSvr
ZOOKEEPER_HOST:ZOOKEEPER_PORT --addNode CLUSTER_NAME
NODE_NAMEUNIQUE_ID
```

Exemple 6.8. Ajouter trois nœuds de cluster

```
./helix-admin.sh --zkSvr server1:2181,server2:2181,server3:2181
--addNode bpms-cluster nodeOne:12345
./helix-admin.sh --zkSvr
```

```
server1:2181,server2:2181,server3:2181 --addNode bpms-cluster
nodeTwo:12346
./helix-admin.sh --zkSvr
server1:2181,server2:2181,server3:2181 --addNode bpms-cluster
nodeThree:12347
```

5. Ajouter des ressources au cluster

Exemple 6.9. Ajouter vfs-repo comme ressource

```
./helix-admin.sh --zkSvr server1:2181,server2:2181,server3:2181 --
addResource bpms-cluster vfs-repo 1 LeaderStandby AUTO_REBALANCE
```

6. Rééquilibrez le cluster avec les trois nœuds.

Exemple 6.10. Rééquilibrer le bpms-cluster

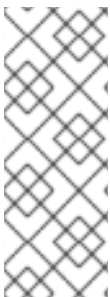
```
./helix-admin.sh --zkSvr server1:2181,server2:2181,server3:2181 --
rebalance bpms-cluster vfs-repo 3
```

Dans la commande ci-dessus, 3 fait référence à trois nœuds Zookeeper.

7. Démarrez le contrôleur Helix dans tous les nœuds du cluster.

Exemple 6.11. Démarrez le contrôleur Helix

```
./run-helix-controller.sh --zkSvr
server1:2181,server2:2181,server3:2181 --cluster bpms-cluster 2>&1
> /tmp/controller.log &
```



NOTE

ZooKeeper doit contenir un nombre impair d'instances, au moins 3 pour récupérer d'un échec. Suite à un échec, le nombre de nœuds restants doit toujours pouvoir former une majorité. Par exemple, un cluster de cinq nœuds ZooKeeper peut supporter la perte de deux nœuds afin de récupérer complètement. Une instance ZooKeeper est toujours possible, la replication fonctionnera, mais aucune possibilité de récupération n'est disponible si elle échoue.

Arrêt de Helix et ZooKeeper

Pour arrêter les processus de Helix et le serveur ZooKeeper, utilisez la procédure suivante.

Procédure 6.1. Arrêt de Helix et ZooKeeper

1. Arrêtez les processus du serveur JBoss EAP

2. Arrêtez le processus Helix créé par `run-helix-controller.sh`, par exemple `kill -15 <pid of HelixControllerMain>`.
3. Arrêtez le serveur ZooKeeper à l'aide de la commande `zkServer.sh stop`

6.2.2. Installation de Quartz



NOTE

Si vous n'utilisez pas (les minuteurs) Quartz dans vos processus métiers, ou si vous n'utilisez pas du tout le serveur d'exécution, vous pouvez ignorer cette section. Veuillez noter que si vous souhaitez répliquer des minuteurs dans vos processus métiers, vous devrez utiliser le composant Quartz.

Avant de pouvoir configurer la base de données de votre serveur d'application, vous devez préparer la base de données pour que Quartz crée des tableaux Quartz, qui détiendra les données de minuterie et le fichier de définition Quartz.

Pour cela, veuillez effectuer ce qui suit :

1. Installez la base de données. Assurez-vous d'utiliser l'une des sources de données non JTA prises en charge. Veuillez noter que Quartz nécessite une source de données non JTA et que vous ne pouvez donc pas utiliser la source de données Business Central. Dans le code en exemple, PostgreSQL est utilisé avec l'utilisateur `bpms` et le mot de passe `bpms`. Cette base de données devra être connectée à votre serveur d'application, assurez-vous donc de noter les informations et détails d'authentification de la base de données.
2. Créez des tableaux Quartz sur votre base de données pour permettre une synchronisation d'événements de minuteur. Pour cela, veuillez utiliser le script DDL pour votre base de données, disponible dans l'archive zip supplémentaire disponible sur `$QUARTZ_HOME/docs/dbTables`.
3. Créez le fichier de configuration Quartz `quartz-definition.properties` dans le répertoire `$JBoss_HOME/PROFILE/configuration/` et définissez les propriétés Quartz.

Exemple 6.12. Fichier de configuration Quartz pour une base de données PostgreSQL

```
#=====
=====
# Configure Main Scheduler Properties
#=====
=====

org.quartz.scheduler.instanceName = jBPMClusteredScheduler
org.quartz.scheduler.instanceId = AUTO

#=====
=====
# Configure ThreadPool
#=====
=====

org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 5
```

```

org.quartz.threadPool.threadPriority = 5

#=====
=====
# Configure JobStore
#=====
=====

org.quartz.jobStore.misfireThreshold = 60000

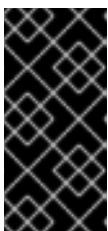
org.quartz.jobStore.class=org.quartz.impl.jdbcjobstore.JobStoreCM
T
org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobst
ore.PostgreSQLDelegate
org.quartz.jobStore.useProperties=false
org.quartz.jobStore.dataSource=managedDS
org.quartz.jobStore.nonManagedTXDataSource=notManagedDS
org.quartz.jobStore.tablePrefix=QRTZ_
org.quartz.jobStore.isClustered=true
org.quartz.jobStore.clusterCheckinInterval = 20000

#=====
=====
# Configure Datasources
#=====
=====

org.quartz.dataSource.managedDS.jndiURL=jboss/datasources/psbpmsD
S
org.quartz.dataSource.notManagedDS.jndiURL=jboss/datasources/quar
tzNotManagedDS

```

Veuillez noter les sources de données configurées qui recevront les deux programmes Quartz à la fin du fichier.



IMPORTANT

L'intervalle recommandé pour la détection de cluster est de 20 secondes. Il est défini dans la propriété `org.quartz.jobStore.clusterCheckinInterval` du fichier **quartz-definition.properties**. Selon votre configuration, prenez en compte l'impact sur la performance et modifiez les paramètres en conséquence.

Veuillez également noter la propriété `org.quartz.jobStore.driverDelegateClass` qui définit le dialecte DB à utiliser lors de la communication avec la base de données configurée (dans cet exemple, `org.quartz.impl.jdbcjobstore.PostgreSQLDelegate`. Si vous utilisez Oracle, utilisez `org.quartz.impl.jdbcjobstore.oracle.OracleDelegate`).

CHAPITRE 7. RÉFÉRENTIELS MAVEN

7.1. À PROPOS DE MAVEN

Apache Maven est un outil d'automatisation de builds distribuées, utilisé en développement Java pour créer et gérer des projets de logiciels. Maven utilise des fichiers de configuration XML nommés Project Object Model, ou POM, pour définir des propriétés de projets et gérer le processus de construction. Les fichiers POM décrivent les dépendances de composant et de module du projet, l'ordre de construction et les cibles pour la sortie et la mise en paquet du projet obtenu. Cela garantit que les projets soient construits de façon correcte et uniforme.

Maven utilise les référentiels pour stocker les bibliothèques, plug-ins et autres artefacts de construction Java. Les référentiels peuvent être locaux ou distants. Un référentiel local est un téléchargement d'artefacts à partir d'un référentiel distant mis en cache sur une machine locale. Un référentiel distant est tout autre référentiel dont l'accès s'effectue à l'aide de protocoles courants tels que `http://` s'il est situé sur un serveur HTTP, ou `file://` s'il est situé sur un serveur de fichiers. Le référentiel par défaut est le [Maven 2 Central Repository](#) distant public.

La configuration de Maven s'effectue en modifiant le fichier `settings.xml`. Vous pouvez soit configurer les paramètres généraux de Maven dans le fichier `M2_HOME/conf/settings.xml`, ou les paramètres de niveau utilisateur dans le fichier `USER_HOME/.m2/settings.xml`.

Pour davantage d'informations sur Maven, veuillez consulter [Welcome to Apache Maven](#).

Pour davantage d'informations sur les référentiels Maven, veuillez consulter [Apache Maven Project - Introduction to Repositories](#).

Pour plus d'informations concernant les fichiers POM Maven, veuillez consulter [Apache Maven Project POM Reference](#).



NOTE

Votre produit Red Hat JBoss a été construit avec Maven 3.0.x. Cette version Maven est par conséquent recommandée pour la construction de vos applications SwitchYard.

7.2. À PROPOS DES RÉFÉRENTIELS MAVEN FOURNIS

Cette version offre un ensemble de référentiels contenant des artefacts nécessaires pour construire des applications basées sur Red Hat JBoss BPM Suite. Maven doit être configuré pour utiliser ces référentiels et le Maven Central Repository (Référentiel Central Maven) de manière à fournir des fonctionnalités de construction correctes.

Deux ensembles de référentiels interchangeables garantissant les mêmes fonctionnalités. Le premier ensemble peut être téléchargé et stocké sur un système de fichiers local. Le second ensemble est hébergé en ligne pour être utilisé en tant que référentiels distants. Si vous avez fourni l'emplacement du fichier `settings.xml` de Maven lors de l'installation, Maven sera déjà configuré pour utiliser les référentiels en ligne.



IMPORTANT

L'ensemble des référentiels distants en ligne est une source d'aperçu technologique de composants n'ayant pas pour objectif d'apporter des corrections. Il est uniquement pris en charge pour une utilisation dans un environnement de développement. L'utilisation de cet ensemble de référentiels en ligne dans un environnement de production peut provoquer des vulnérabilités en matière de sécurité : ce cas d'utilisation ne sera donc pas pris en charge. Pour plus d'informations, veuillez consulter <https://access.redhat.com/site/maven-repository>.

7.3. CONFIGURER MAVEN POUR UTILISER LES RÉFÉRENTIELS DE SYSTÈME DE FICHIER

Vue d'ensemble

Dans les situations où une utilisation de référentiels en ligne est impossible, vous devrez télécharger et configurer les référentiels requis localement.

Procédure 7.1.

1. Télécharger les archives ZIP suivantes contenant les référentiels requis :
 - o <https://access.redhat.com/jbossnetwork/restricted/listSoftware.html?downloadType=distributions&product=bpm.suite&version=6.2.0>
2. Décompressez les fichiers ZIP téléchargés vers un emplacement arbitraire dans un système de fichiers local.
3. Ajoutez les entrées des référentiels décompressés vers le fichier `settings.xml` de Maven. L'exemple de code suivant contient un profil avec les référentiels, une configuration d'authentification pour accéder aux référentiels et une entrée d'activation pour le profil :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <localRepository/>
  <profiles>
    <!-- Profile with local repositories required by JBoss
    BRMS/JBoss BPM Suite -->
    <profile>
      <id>brms-bpms-local-profile</id>
      <repositories>
        <repository>
          <id>jboss-brms-bpmsuite-repository</id>
          <name>BRMS/BPMS 6.2.0 GA Repository</name>
          <url>file://<!-- path to the repository -->/jboss-brms-
bpmsuite-6.2.0.GA-redhat-5-maven-repository/maven-repository</url>
          <layout>default</layout>
          <releases>
            <enabled>>true</enabled>
            <updatePolicy>never</updatePolicy>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
```

```

        <updatePolicy>never</updatePolicy>
      </snapshots>
    </repository>
  </repositories>
  <pluginRepositories>
    <pluginRepository>
      <id>jboss-brms-bpmsuite-repository</id>
      <name>BRMS/BPMS 6.2.0 GA Repository</name>
      <url>file://<!-- path to the repository -->/jboss-brms-
bpmsuite-6.2.0.GA-redhat-5-maven-repository/maven-repository</url>
      <layout>default</layout>
      <releases>
        <enabled>true</enabled>
        <updatePolicy>never</updatePolicy>
      </releases>
      <snapshots>
        <enabled>false</enabled>
        <updatePolicy>never</updatePolicy>
      </snapshots>
    </pluginRepository>
  </pluginRepositories>
</profile>
</profiles>

<!-- Configuring pre-emptive authentication for the repository
server -->
<server>
  <id>brms-bpms-m2-repo</id>
  <username>admin</username>
  <password>admin</password>
  <configuration>
    <wagonProvider>httpClient</wagonProvider>
    <httpConfiguration>
      <all>
        <usePreemptive>true</usePreemptive>
      </all>
    </httpConfiguration>
  </configuration>
</server>

<!-- Alternative to enabling pre-emptive authentication -
configuring
the Authorization HTTP header with Base64-encoded credentials
<server>
  <id>brms-bpms-m2-repo</id>
  <configuration>
    <httpHeaders>
      <property>
        <name>Authorization</name>
        <value>Basic YWRtaW46YWRtaW4= </value>
      </property>
    </httpHeaders>
  </configuration>
</server>
-->

```

```

<activeProfiles>
  <!-- Activation of the JBoss BRMS/JBoss BPM Suite profile -->
  <activeProfile>brms-bpms-local-profile</activeProfile>
</activeProfiles>
</settings>

```

Résultat

Les référentiels Maven sont téléchargés, décompressés dans un système de fichiers local, enregistrés dans le fichier `settings.xml` de Maven et prêt à être utilisés lors des générations Maven.

Résolution de problèmes

Q : Pourquoi obtiens-je des messages d'erreur lors de la génération ou du déploiement de mes applications ?

R : Question

Lorsque vous générez ou déployez un projet, celui-ci échoue avec un ou les deux messages d'erreur suivants :

```

[ERROR] Failed to execute goal on project PROJECT_NAME
       : Could not find artifact ARTIFACT_NAME

```

Cause

Your cached local Maven repository might contain outdated artifacts. (Il se peut que votre référentiel Maven local mis en cache contienne des artefacts obsolètes)

Résolution

Pour résoudre ce problème, supprimez le référentiel local mis en cache : le répertoire `~/.m2/repository/` sur Linux ou le répertoire `%SystemDrive%\Users\USERNAME\.m2\repository\` sur Windows, puis exécutez `mvn clean install -U`. Cela obligera Maven à télécharger les versions correctes des artefacts nécessaires lors de l'exécution de la génération suivante.

Q : Pourquoi JBoss Developer Studio utilise-t-il mon ancienne configuration Maven ?

R : Question

Vous avez mis votre configuration Maven à jour, mais cette configuration n'est pas visible dans JBoss Developer Studio.

Cause

Si votre JBoss Developer Studio est en marche lorsque vous modifiez votre fichier `settings.xml` Maven, cette configuration ne sera pas visible dans JBoss Developer Studio.

Résolution

Actualisez les paramètres Maven dans l'IDE. À partir du menu, sélectionnez **Window** → **Preferences** (Fenêtre - Préférence). Dans la fenêtre **Preferences**, agrandissez **Maven** et sélectionnez **User Settings** (Paramètres utilisateur). Cliquez sur le bouton **Update Settings** (Mettre à jour les paramètres) pour réactualiser les paramètres de l'utilisateur Maven dans JBoss Developer Studio.

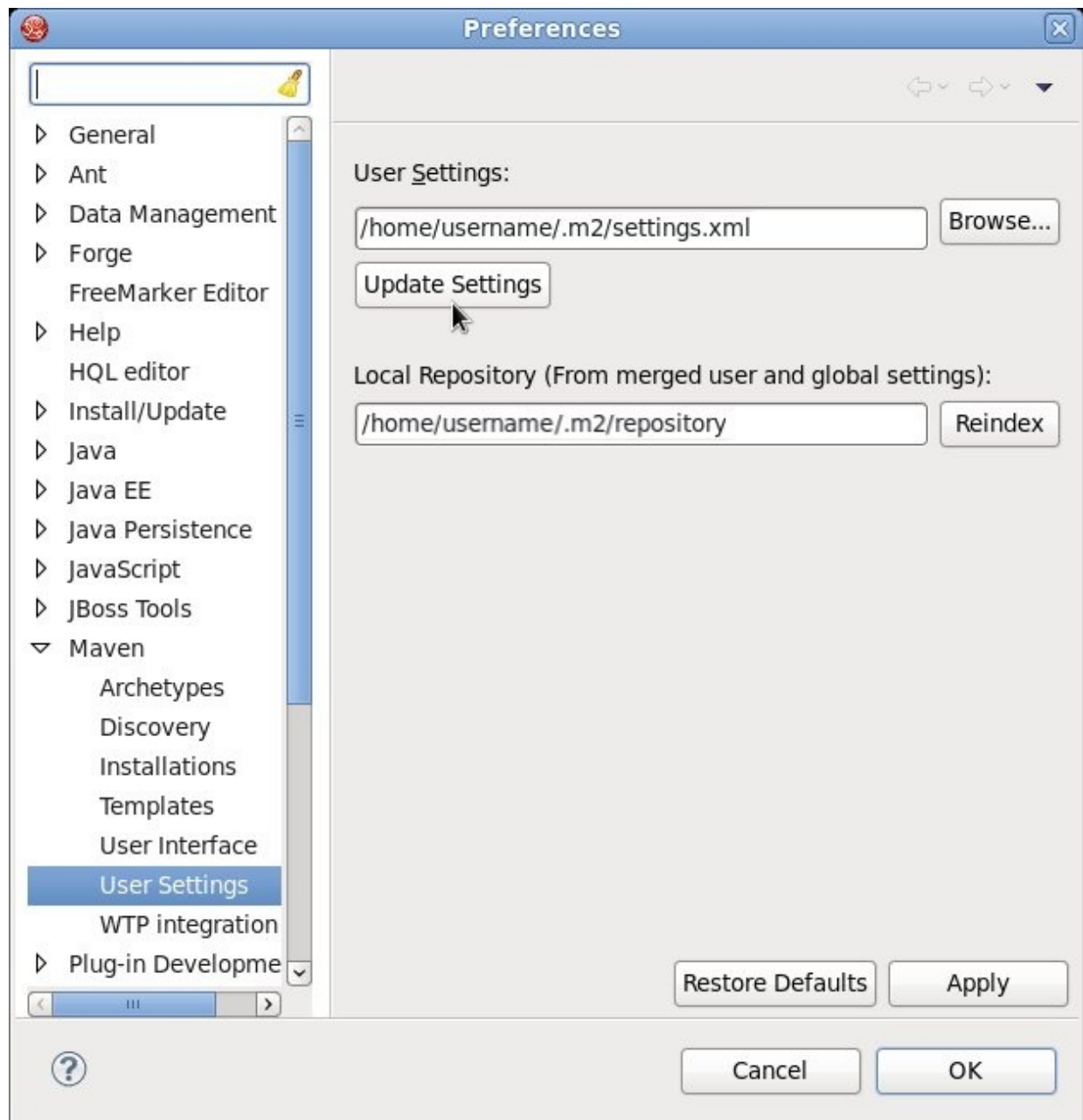


Figure 7.1. Mise à jour des paramètres de configuration de l'utilisateur Maven

7.4. CONFIGURER MAVEN POUR UTILISER LES RÉFÉRENTIELS EN LIGNE

Les référentiels en ligne nécessaires pour les applications Red Hat JBoss BPM Suite sont situés sur <http://maven.repository.redhat.com/techpreview/all/>.

Si vous n'avez pas configuré le référentiel Maven lors de l'installation, vous pouvez le configurer en utilisant la procédure suivante. (Il est également possible de le faire en utilisant le fichier POM du projet, mais cela n'est pas recommandé.)

Procédure 7.2. Configurer Maven pour utiliser les référentiels en ligne

1. Ajoutez des entrées aux référentiels en ligne et la configuration de l'authentification pour y accéder au fichier `settings.xml` Maven tel que dans l'exemple de code ci-dessous :

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">

  <profiles>
    <!-- Profile with online repositories required by BRMS/BPMS -->
    <profile>
      <id>brms-bpms-online-profile</id>
      <repositories>
        <repository>
          <id>jboss-ga-repository</id>

<url>http://maven.repository.redhat.com/techpreview/all</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>jboss-ga-plugin-repository</id>

<url>http://maven.repository.redhat.com/techpreview/all</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>

  <!-- Configuring pre-emptive authentication for the repository
server -->
  <server>
    <id>brms-bpms-m2-repo</id>
    <username>admin</username>
    <password>admin</password>
    <configuration>
      <wagonProvider>httpclient</wagonProvider>
      <httpConfiguration>
        <all>
          <usePreemptive>true</usePreemptive>
        </all>
      </httpConfiguration>
    </configuration>
  </server>
```

```

    <!-- Alternative to enabling pre-emptive authentication -
    configuring
        the Authorization HTTP header with Base64-encoded credentials
    <server>
        <id>brms-bpms-m2-repo</id>
        <configuration>
            <httpHeaders>
                <property>
                    <name>Authorization</name>
                    <value>Basic YWRtaW46YWRtaW4= </value>
                </property>
            </httpHeaders>
        </configuration>
    </server>
    -->

    <activeProfiles>
        <!-- Activation of the BRMS/BPMS profile -->
        <activeProfile>brms-bpms-online-profile</activeProfile>
    </activeProfiles>

</settings>

```

2. Si vous avez modifié le fichier `settings.xml` alors que JBoss Developer Studio était en cours d'exécution, vous devrez réactualiser les paramètres Maven dans l'IDE. À partir du menu, sélectionnez **Window** → **Preferences**. Dans la fenêtre **Preferences**, agrandissez **Maven** et sélectionnez **User Settings**. Cliquez sur le bouton **Update Settings** pour réactualiser les paramètres de l'utilisateur Maven dans JBoss Developer Studio.

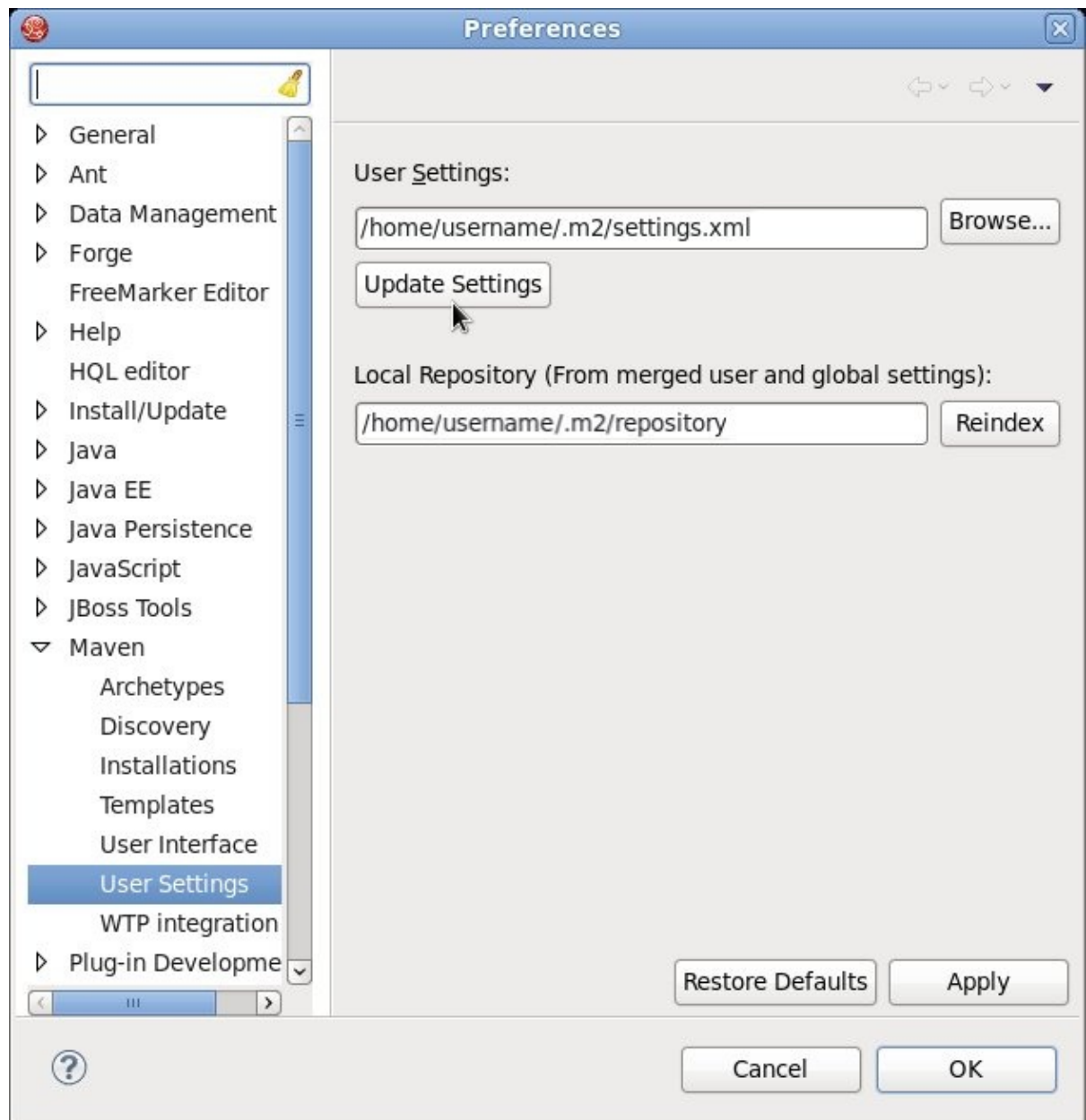


Figure 7.2. Mise à jour des paramètres de configuration de l'utilisateur Maven

Résultat

Maven a été configuré pour utiliser les référentiels en ligne fournis pour votre produit Red Hat JBoss.

IMPORTANT

Si votre référentiel Maven local mis en cache contient des artefacts obsolètes, vous risquez de rencontrer un des messages d'erreur Maven suivants quand vous créez ou déployez votre projet :

- Missing artifact *ARTIFACT_NAME*
- [ERROR] Failed to execute goal on project *PROJECT_NAME*; Could not resolve dependencies for *PROJECT_NAME*

Pour résoudre ce problème, supprimez le référentiel local mis en cache : le répertoire `~/.m2/repository/` sur Linux ou le répertoire `%SystemDrive%\Users\USERNAME\.m2\repository\` sur Windows. Cela obligera Maven à télécharger les versions correctes des artefacts nécessaires lors de la génération suivante.

7.5. GESTION DES DÉPENDANCES

Afin d'utiliser les bonnes dépendances Maven dans votre projet Red Hat JBoss BPM Suite, vous devez ajouter les fichiers BOM (Nomenclature des matériaux et produits) au fichier `pom.xml` du projet. Ces fichiers BOM garantissent que les bonnes versions des dépendances transitives des référentiels Maven fournis sont inclus dans le projet.

Le référentiel Maven dans la version 6.1.0 et les versions supérieures est conçu pour être utilisé uniquement avec Maven Central. Aucun autre référentiel n'est requis.

Selon les besoins de votre projet, déclarez les dépendances dans votre fichier POM dans la section dépendance :

- `org.jboss.bom.brms:jboss-brms-bpmsuite-bom:VERSION` : l'indicateur BOM de base sans support Java EE6.
- `org.jboss.bom.brms:jboss-javaee-6.0-with-brms-bpmsuite:VERSION` : offre un support pour Java EE6.

CHAPITRE 8. RED HAT JBOSS DEVELOPER STUDIO

8.1. RED HAT JBOSS DEVELOPER STUDIO

Red Hat JBoss Developer Studio est l'environnement de développement intégré de JBoss (IDE) basé sur Eclipse. Obtenez le tout dernier JBoss Developer Studio à partir du portail de support client de Red Hat à <https://access.redhat.com>. JBoss Developer Studio fournit des plug-ins avec des outils et des interfaces pour Red Hat JBoss BRMS et Red Hat JBoss BPM Suite. Ces plug-ins sont basés sur la version communautaire de ces produits. Ainsi, le plugin de JBoss BRMS s'appelle le plugin Drools et le plugin JBoss BPM Suite s'appelle le plugin jBPM.

Voir la documentation *Red Hat JBoss Developer Studio* pour obtenir les instructions d'installation et de configuration.



AVERTISSEMENT

La gestion des noms de règles multi-octet rencontrant quelques problèmes, veuillez vous assurer que l'instance de JBoss Developer Studio démarre avec le code de fichier défini sur UTF-8. Vous pouvez pour cela modifier le fichier `$JBDS_HOME/studio/jbdevstudio.ini` et ajouter la propriété suivante : `"-Dfile.encoding=UTF-8"`

8.2. INSTALLATION DES PLUG-INS D'UN JBOSS DEVELOPER STUDIO

Les plug-ins Drools et jBPM de JBoss Developer Studio sont accessibles via le site de mise à jour.

Procédure 8.1. Installer le plugin Drools et jBPM JBoss Developer Studio

1. Démarrer JBoss Developer Studio.
2. Sélectionnez **Help** → **Install New Software (Installer Nouveau Logiciel)**.
3. Cliquez sur **Add** (ajouter) pour saisir le menu **Add Repository** (ajouter référentiel).
4. Donnez au site du logiciel un nom à côté du champ **Name** et ajoutez l'url suivant **Location** : <https://devstudio.jboss.com/updates/8.0/integration-stack/>
5. Cliquez sur **OK**.
6. Sélectionnez la fonctionnalité **JBoss Business Process and Rule Development** parmi les options disponibles puis cliquez sur **Next** (suivant) et sur **Next** à nouveau.
7. Lisez les termes de la licence et acceptez-la en sélectionnant le bouton radio qui convient, puis cliquez sur **Finish** (Terminer).
8. Après l'installation des plug-ins, démarrez JBoss Developer Studio à nouveau.

8.3. DÉFINISSEZ LE RUNTIME DE DROOLS

Pour utiliser le plug-in Red Hat JBoss BRMS dans Red Hat JBoss Developer Studio, il est nécessaire de définir le runtime.

Un runtime est une collection de fichiers jar qui représentent une version spécifique du logiciel et fournit les bibliothèques nécessaires à la compilation et à l'exécution des ressources d'entreprise.

Procédure 8.2. Configurer le Runtime de JBoss BRMS

1. Extrayez les fichiers jar de runtime situés dans l'archive **jboss-brms-VERSION-engine.zip** que vous pouvez télécharger à partir du [Portail Client Red Hat](#).
2. À partir du menu JBoss Developer Studio, sélectionnez **Window** et cliquez sur **Preferences**.
3. Sélectionnez **Drools** → **Installed Drools Runtimes**.
4. Cliquez sur **Add . . .** (Ajouter) ; choisissez un nom pour le nouveau runtime puis cliquez sur **Browse** (Parcourir) pour naviguer vers le répertoire dans lequel vous avez extrait les fichiers de runtime à l'étape 1. Cliquez sur **OK** pour enregistrer le runtime sélectionné dans JBDS.
5. Indiquez que le runtime que vous avez créé est le runtime Drools par défaut en cliquant sur la case adjacente.
6. Cliquez sur **OK**. Si vous avez des projets existants, une boîte de dialogue vous indiquera de démarrer à nouveau JBoss Developer Studio pour mettre le Runtime à jour.

8.4. CONFIGURATION DU SERVEUR JBOSS BPM SUITE

JBoss Developer Studio peut être configuré pour exécuter le serveur Red Hat JBoss BPM Suite.

Procédure 8.3. Configurer le serveur

1. Ouvrez l'affichage jBPM en sélectionnant **Window** → **Open Perspective** → **Other** (Fenêtre - Ouvrir Perspective - Autre) et sélectionnez **jBPM**, puis cliquez sur **OK**.
2. Ajoutez l'affichage du serveur en sélectionnant **Window** → **Show View** → **Other...** et en sélectionnant **Server** → **Servers**.
3. Ouvrez le Menu de serveur en effectuant un clic droit sur le panneau de serveurs et en sélectionnant **New** → **Server**.
4. Définissez le serveur en sélectionnant **JBoss Enterprise Middleware** → **JBoss Enterprise Application Platform 6.4+** et en cliquant sur **Next**.
5. Configurez le répertoire d'accueil en cliquant sur le bouton **Browse** (Parcourir). Recherchez et sélectionnez le répertoire d'installation de JBoss EAP 6.4 avec JBoss BPM Suite installé.
6. Donnez un nom au serveur dans le champ **Name**, assurez-vous que le fichier de configuration est défini et cliquez sur **Finish**.

8.5. IMPORTATION DE PROJETS À PARTIR D'UN RÉFÉRENTIEL GIT DANS JBOSS DEVELOPER STUDIO

Vous pouvez configurer JBoss Developer Studio pour vous connecter à un référentiel central de ressources Git. Le référentiel stocke les versions de règles, les modèles, les fonctions et les processus

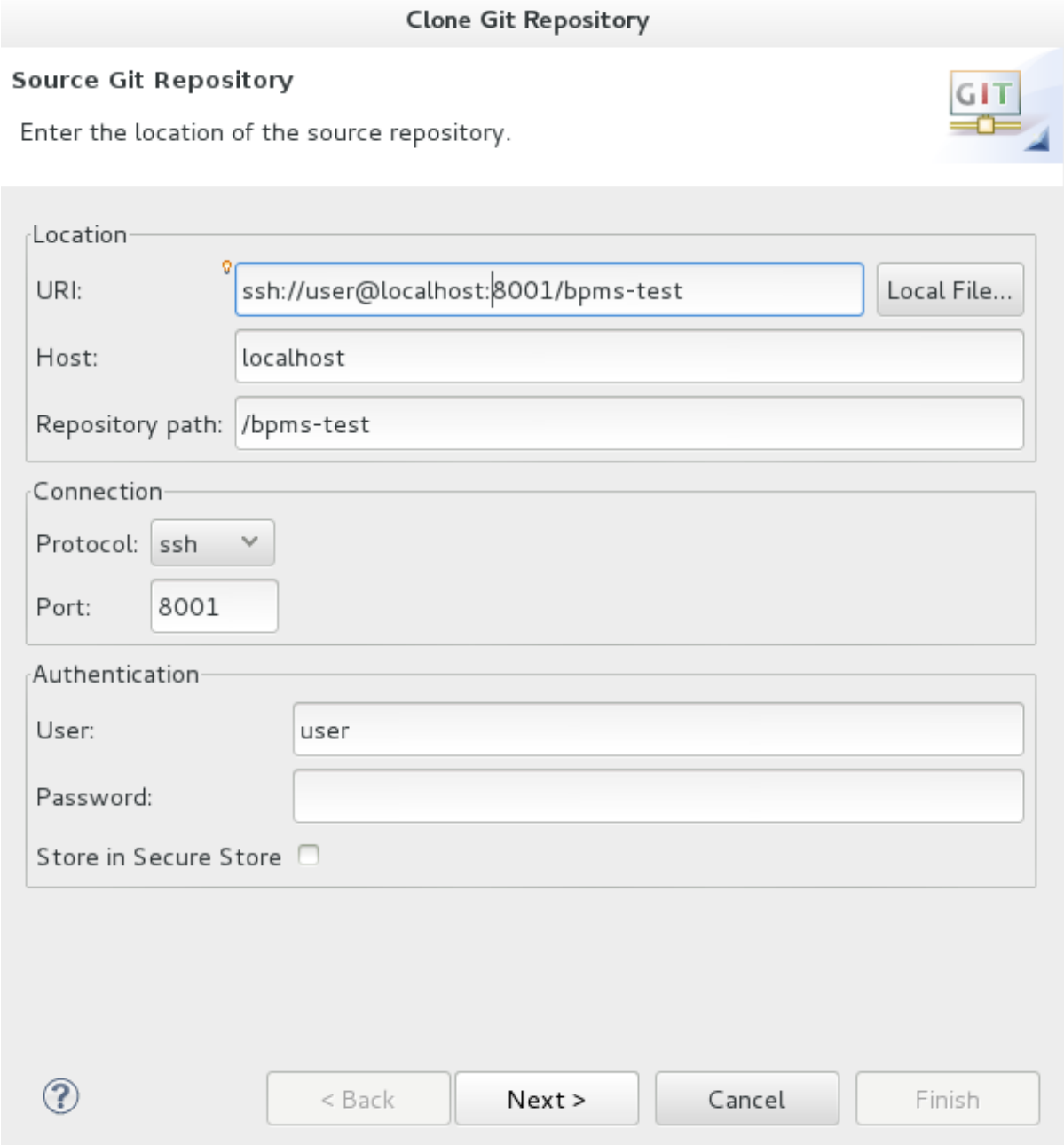
Vous pouvez cloner un référentiel Git distant ou importer un référentiel Git local.

Procédure 8.4. Clonage d'un référentiel GIT distant

1. Démarrez le serveur Red Hat JBoss BRMS ou BPM Suite en sélectionnant le serveur à partir de l'onglet Serveur et en cliquant sur le bouton Start.
2. Démarrez en même temps le serveur Secure Shell, s'il n'est pas déjà en cours d'exécution, en utilisant la commande suivante. La commande est spécifique à Linux et Mac. Sur ces plateformes, si sshd a déjà démarré, cette commande échouera et vous pourrez ignorer cette étape.

```
/sbin/service sshd start
```

3. Dans JBoss Developer Studio, sélectionnez **File** → **Import...** (Fichier - Importer) et naviguez dans le fichier Git. Ouvrez le dossier Git et sélectionnez **Projects from Git** (Projets Git) puis cliquez sur **Next** (Suivant).
4. Sélectionnez la source du référentiel en tant que **Clone URI** et cliquez sur **Next** (Suivant).
5. Saisissez les détails du référentiel Git dans la prochaine fenêtre et cliquez sur **Next**.



Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

Store in Secure Store ☐



Figure 8.1. Informations sur le référentiel Git

6. Sélectionnez la branche que vous souhaitez importer dans la fenêtre suivante et cliquez sur **Next** (Suivant).
7. Pour définir le stockage local pour ce projet, saisissez (ou sélectionnez) un répertoire non-vide, apportez des modifications de configuration et cliquez sur **Next**.
8. Importez le projet en tant que projet général dans la fenêtre suivante et cliquez sur **Next** (Suivant). Nommez le projet et cliquez sur **Finish** (Terminer).

Procédure 8.5. Importer un référentiel local Git

1. Démarrez le serveur Red Hat JBoss BRMS ou BPM Suite en sélectionnant le serveur à partir de l'onglet Serveur et en cliquant sur le bouton **Start**.
2. Dans JBoss Developer Studio, sélectionnez **File** → **Import...** (Fichier - Importer) et naviguez dans le fichier Git. Ouvrez le dossier Git et sélectionnez **Projects from Git** (Projets Git) puis cliquez sur **Next** (Suivant).

3. Sélectionnez la source du référentiel en tant que **Existing local repository** et cliquez sur **Next (Suivant)**.

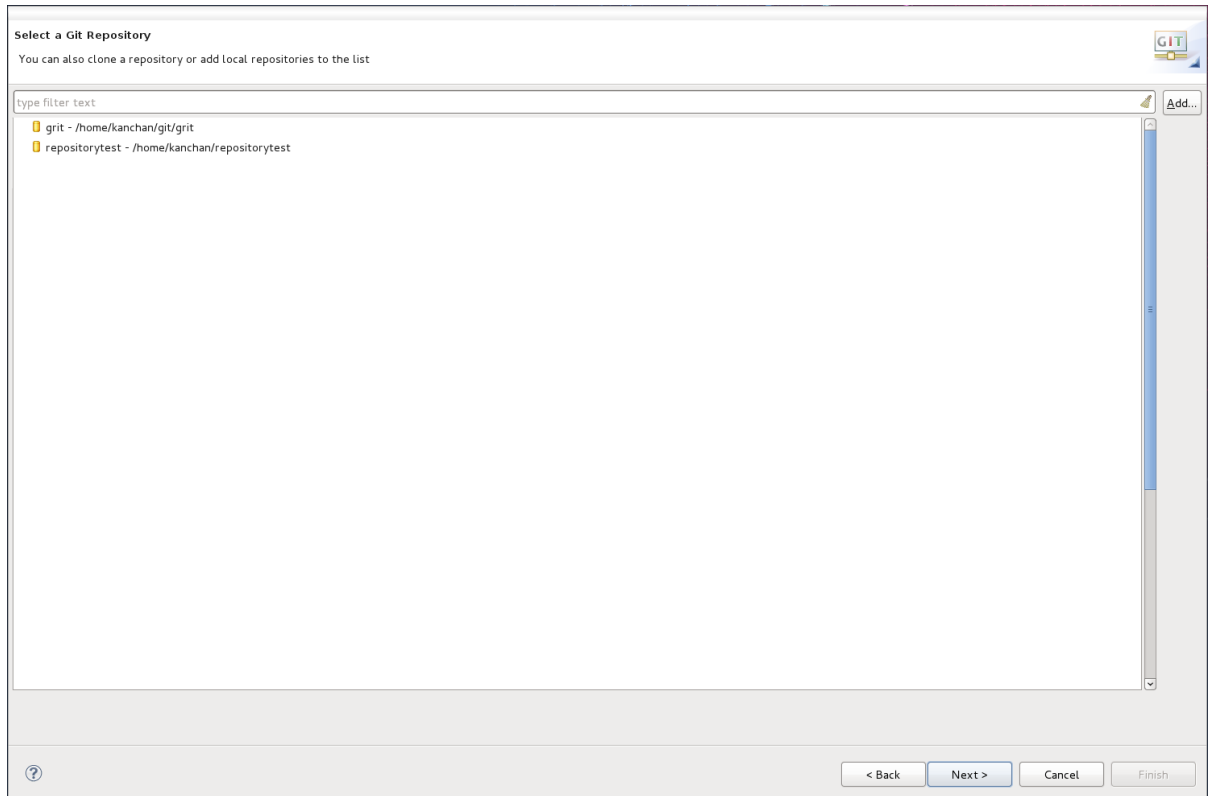


Figure 8.2. Informations sur le référentiel Git

4. Sélectionnez le référentiel qui doit être configuré à partir de la liste de référentiels disponibles et cliquez sur **Next**.
5. Dans la boîte de dialogue, sélectionnez le bouton **Import as general project** (Importer comme projet général) à partir de **Wizard for project import group** (Wizard pour groupe d'importation de projet) et cliquez sur **Next (Suivant)**. Nommez le projet et cliquez sur **Finish (Terminer)**.

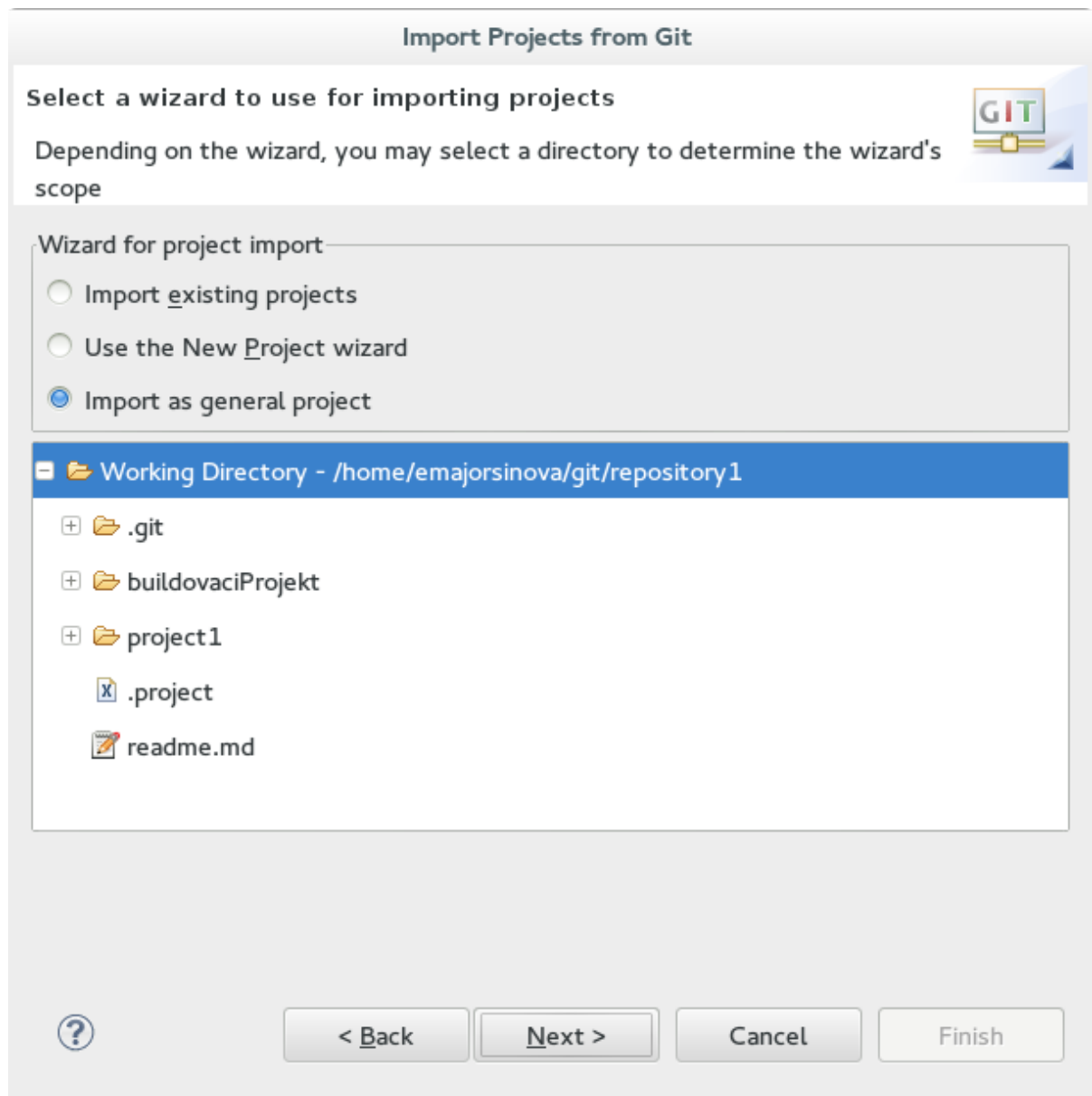


Figure 8.3. Wizard pour Importation de Projet

CHAPITRE 9. CORRECTIFS ET MISE À NIVEAU DE RED HAT JBOSS BPM SUITE

9.1. CORRECTIFS ET MISES À NIVEAU

Les correctifs de Red Hat JBoss BPM Suite peuvent être sous forme de mise à jour asynchrone ou planifiée :

- Mises à jour asynchrones : correctifs ponctuels publiés en dehors d'un cycle de mise à jour normal du produit existant. Ces mises à jour peuvent inclure des correctifs de sécurité, ainsi que d'autres correctifs ponctuels fournis par GSS (Red Hat Global Support Services) pour résoudre ces problèmes particuliers.
- Mises à jour planifiées : les correctifs cumulatifs d'un produit existant, qui incluent toutes les mises à jour développées auparavant pour cette version du produit.

Les correctifs de Red Hat JBoss BPM Suite peuvent être téléchargés à partir de <https://access.redhat.com/downloads/>.

Les fichiers suivants sont inclus dans la version de correction JBoss BRMS et JBoss BPM Suite.

- Clients JBoss BRMS - `jboss-brms-<version>-patch.zip`
- Clients JBoss BPM Suite - `jboss-bpmsuite-<version>-patch.zip`
- Mises à jour des référentiels Maven (identiques pour les clients JBoss BRMS et JBoss BPM Suite) - `jboss-brms-bpmsuite-<version>-incremental-maven-repository.zip`

9.2. APPLIQUER DES CORRECTIFS DANS RED HAT JBOSS BPM SUITE 6.2.

Dans Red Hat JBoss BPM Suite, l'outil de correction client est distribué comme un fichier zip comprenant des scripts `.sh` et `.bat` simples, permettant une application de mises à jour facilitée et automatique sur une installation JBoss BPM Suite 6.1 (ou version ultérieure).



IMPORTANT

L'outil de correction est à utiliser avec JBoss BPM Suite 6.1 ou version ultérieure, et ne doit pas être utilisé sur les versions antérieures. Pour plus d'informations, veuillez consulter <https://access.redhat.com/articles/1455733>.

Le script nécessite deux paramètres obligatoires : `<path-to-distribution-root>` et `<type-of-distribution>`. Par exemple, la commande suivante applique les mises à jour au lot JBoss EAP suivant :



NOTE

Les mises à jour de corrections ne doivent pas être appliquées lors de l'exécution d'une instance de Red Hat JBoss BPM Suite. Assurez-vous que le serveur est éteint avant d'exécuter la commande suivante.

```
$ ./apply-updates.sh ~/EAP_HOME/jboss-eap-6.4 eap6.x
```

Les types de distribution suivants sont pris en charge :

- eap6.x
- eap6.x-bc
- eap6.x-kie-server
- eap6.x-dashbuilder
- generic
- generic-bc
- generic-kie-server
- generic-dashbuilder
- was8
- was8-bc
- was8-kie-server
- was8-dashbuilder
- wls12c
- wls12c-bc
- wls12c-kie-server
- wls12c-dashbuilder
- brms-engine
- planner-engine
- supplementary-tools

Les outils de migration et démarrage rapide sont également inclus dans le correctif et peuvent être téléchargés en tant que fichier zip.



NOTE

Seules les mises à jour de BRMS/BPM Suite sont incluses dans la distribution de correctifs. Les correctifs de EAP doivent être appliqués à l'aide du mécanisme de correction EAP. Veuillez consulter le [Guide d'installation Red Hat JBoss EAP](#).

Fonctionnalité de sauvegarde

Avant d'appliquer une mise à jour, le script client effectue une sauvegarde de la distribution spécifiée et copie le fichier de distribution ou le répertoire dans le sous-répertoire **backup/<current - timestamp>**. Le répertoire de sauvegarde de niveau supérieur est créé au même niveau de fichier système que le **apply-updates**.

Fonctionnalité liste noire

L'outil de correction client fournit une fonctionnalité liste noire permettant d'indiquer au script les fichiers devant être mis à jour. Cette fonctionnalité est très utile et permet de conserver vos fichiers de configuration et d'éviter qu'ils ne soient supprimés automatiquement par le processus de mise à jour. Vous pouvez également indiquer des fichiers autres que des fichiers de configuration, si besoin.

Pour spécifier les fichiers *de la liste noire*, ouvrez le fichier **blacklist.txt** présent dans la distribution de correction. Saisissez le chemin d'accès relatif aux fichiers ne devant pas être mis à jour. Chaque fichier doit être précisé sur une ligne individuelle.

```
# lines with a '#' are comment lines, like this one
# blank lines are ignored

# we have made changes to the web.xml that must be preserved
WEB-INF/web.xml

# this file has custom modifications
styles/base.css
```

Les fichiers spécifiés dans le fichier **blacklist.txt** et dont le contenu du correctif est mis à jour ne seront pas pris en compte par l'outil de mise à jour. À la place, l'outil copie le nouveau fichier mis à jour au même emplacement et y ajoute le suffixe **new**. Par exemple, après l'exécution de l'outil de correction, ces deux fichiers existeront dans le dossier **styles**, continuant avec le fichier **blacklist.txt** dans l'exemple ci-dessus.

```
$ ls styles
base.css base.css.new
```

Il est maintenant de votre responsabilité de comparer le contenu des deux documents et de fusionner les modifications.

Que se passe-t-il si vous souhaitez conserver des fichiers qui ne sont plus distribués ? Placez-les également dans le fichier **blacklist.txt**. L'outil de mise à jour de correction ne supprimera pas ces fichiers, et créera à la place un fichier marqueur vide avec le suffixe **removed**. Vous pourrez ensuite décider de conserver ou supprimer ces fichiers manuellement.

Sur la base du même exemple, si le fichier **base.css** a été supprimé et qu'il était listé dans le fichier **blacklist.txt**, le contenu du répertoire des styles sera, suite à l'exécution de l'outil de correction, semblable à ce qui suit :

```
$ ls styles
base.css base.css.removed
```

9.3. CORRIGER D'AUTRES PLATEFORMES ET APPLICATIONS

Les commandes suivantes peuvent être utilisées pour mettre à jour d'autres plateformes prises en charge et applications courantes dans Red Hat JBoss BPM Suite.

Patch EAP 6.x Business Central WAR

```
$ ./apply-updates.[sh|bat] <some-path>/jboss-eap-
6.4/standalone/deployments/business-central.war eap6.x-bc
```

Patch Generic KIE Server WAR

```
$ ./apply-updates.[sh|bat] <some-path-to-tomcat-home>/webapps/kie-  
server.war generic-kie-server
```

Patch Whole WebLogic 12c Bundle

```
$ ./apply-updates.[sh|bat] <path-to-unzipped-wls12c-bundle> wls12c
```

Patch Planner Engine Bundle

```
$ ./apply-updates.[sh|bat] <path-to-unzipped-planner-bundle> planner-  
engine
```

ANNEXE A. HISTORIQUE DES VERSIONS

Version 6.2.0-2.1	Mon Feb 15 2016	Red Hat Localization Services
Fichiers de traduction synchronisés avec les sources XML 6.2.0-2		
Version 6.2.0-2	Mon Nov 30 2015	Tomas Radej
Ajout d'une remarque concernant les versions dans l'historique des versions, correction des dates du journal des modifications.		
Version 6.2.0-1	Mon Nov 20 2015	Tomas Radej
Version initiale créée pour la version 6.2.0 de JBoss BPM Suite.		