# JBoss Enterprise Application Platform 6.1

## Guide d'administration et de configuration

À utiliser dans JBoss Enterprise Application Platform 6

Édition 3

Last Updated: 2017-10-26

# JBoss Enterprise Application Platform 6.1 Guide d'administration et de configuration

À utiliser dans JBoss Enterprise Application Platform 6
Édition 3

Nidhi Chaudhary

Lucas Costi

Russell Dickenson

Sande Gilda

Vikram Goyal

Eamon Logue

Darrin Mison

Scott Mumford

David Ryan

Misty Stanley-Jones

Keerat Verma

Tom Wells

## Notice légale



## Résumé

Cet ouvrage est un guide d'administration et de configuration de JBoss Enterprise Application Platform 6 et de ses correctifs.

# Table des matières

# PREFACE

# CHAPITRE 1. INTRODUCTION

## 1.1. JBOSS ENTERPRISE APPLICATION PLATFORM 6

JBoss Enterprise Application Platform 6 est une plate-forme middleware rapide, sécurisée et puissante construite sur des standards ouverts et compatibles avec Java Enterprise Edition 6. Elle intègre JBoss Application Server 7 avec un clustering de haute disponibilité, une puissante messagerie, une mise en cache distribuée et autres technologies pour créer une plate-forme stable, et évolutive.

La nouvelle structure modulaire permet que les services soient mis en place uniquement en fonction des besoins, ce qui va augmenter la vitesse de démarrage de façon importante. La console de gestion et l'interface de ligne de commande de gestion suppriment le besoin de modifier les fichiers de configuration XML manuellement, et rajoute la possibilité de script et d'automatiser les tâches. En outre, elle comprend des API et des frameworks de développement, que vous pouvez utiliser pour développer des applications Java EE puissantes, sécurisées et évolutives rapidement.

Report a bug

## 1.2. FONCTIONNALITÉS DE JBOSS ENTERPRISE APPLICATION PLATFORM 6

**Tableau 1.1. Fonctionnalités 6.1.0**

| Fonctionnalité | Description |
|---|---|
| Certification Java | Implémentation certifiée des spécifications de JBoss Enterprise Application Platform 6 Full Profil et Web Profile. |
| Domaine géré | • Un domaine géré procure une gestion centralisée d'instances de serveurs multiples et d'hôtes physiques, tandis qu'un serveur autonome autorise une instance de serveur unique.<br>• Les configurations, déploiement, liaisons de socket, modules, extensions, et propriétés système sont gérées par le groupe de serveurs.<br>• La sécurité des applications, qui comprend les domaines de sécurité, est gérée centralement pour une configuration simplifiée. |
| Console de gestion et Management CLI | Il y a de nouvelles interfaces pour gérer le domaine ou serveur autonome. Il n'est nul besoin de modifier les fichiers de configuration XML à la main. Le Management CLI offre également un mode lot, ce qui signifie que vous pourrez scripter et automatiser les tâches de gestion. |

| Fonctionnalité | Description |
| --- | --- |
| La disposition du répertoire est simplifiée | Le répertoire *modules/* contient maintenant les modules du serveur d'applications, au lieu d'utiliser les répertoires communs et spécifiques au serveur *lib/*. Les répertoires *domain/* et *standalone/* contiennent les artefacts et les fichiers de configuration pour le domaine et pour les déploiements autonomes. |
| Mécanisme de chargement de classes modulaire | Les modules sont chargés et déchargés à la demande pour plus de performance et de sécurité, des démarrages et redémarrages plus rapides. |
| Gestion de Sources de données simplifiée | Les pilotes de base de données peuvent être déployés comme tout autre service. En plus, les sources de données sont créées et gérées directement dans la Console de gestion ou le Management CLI. |
| Temps de démarrage et d'arrêt optimisés | JBoss Enterprise Application Platform 6 utilise moins de ressources et est très efficace dans son utilisation de ressources de système. Cela est particulièrement avantageux pour les développeurs. |

Report a bug

## 1.3. LES MODES D'OPÉRATION DE JBOSS ENTERPRISE APPLICATION PLATFORM 6

JBoss Enterprise Application Platform offre deux modes de fonctionnement des instances de JBoss Enterprise Application Platform. Il peut soit être démarré dans un *serveur autonome*, ou dans un *domaine géré*. Chaque mode est conçu en fonction des différents scénarios. Il permet de choisir entre une installation sur serveur unique ou gestion multi-serveurs coordonnée à effet de levier des besoins de votre entreprise et pour automatiser les processus et faciliter la gestion.

Le choix entre un domaine géré et des serveurs autonomes dépend de la façon dont vos serveurs sont gérés et non pas par rapport aux capacités de réponse aux demandes de l'utilisateur final. Cette distinction est particulièrement importante lorsqu'il s'agit de clusters de haute disponibilité (HA). Il est important de comprendre que la fonctionnalité HA est orthogonale à des serveurs autonomes en cours d'exécution ou d'un domaine géré. Autrement dit, un groupe de serveurs autonomes peut être configuré pour former un cluster HA.

Report a bug

## 1.4. LES SERVEURS AUTONOMES

Un serveur autonome correspond à l'un des deux modes opérationnels de JBoss Enterprise Application Platform. Un mode de serveur autonome est un processus indépendant qui ressemble au mode d'exécution unique des anciennes versions de JBoss Enterprise Application Platform.

L'instance de JBoss Enterprise Application Platform qui exécute en tant que serveur autonome est une instance unique, qui peut exécuter optionnellement dans une configuration clusterisée.

Report a bug

## 1.5. LES DOMAINES GÉRÉS

Un serveur autonome correspond à l'un des deux modes opérationnels d'une instance de JBoss Enterprise Application Platform. Il s'agit d'un mode qui permet de gérer plusieurs instances de JBoss Enterprise Application Platform à partir d'un seul point de contrôle.

Une collection de serveurs gérés de manière centralisée sont connus comme étant membres d'un domaine. Toutes les instances de JBoss Enterprise Application Platform du domaine partagent une politique commune de la gestion. Un domaine se compose d'un *contrôleur de domaine*, d'un ou plusieurs *contrôleurs hôte* et zéro ou plusieurs groupes de serveurs par l'hôte.

Un contrôleur de domaine est le point central d'où le domaine est contrôlé. Il s'assure que chaque serveur est configuré conformément à la politique de gestion du domaine. Le contrôleur de domaine est également un contrôleur hôte. Un contrôleur d'hôte est un procédé physique ou un hôte virtuel sur lequel est exécuté le script **`domain.sh`** ou **`domain.bat`**. À la différence du contrôleur de domaine, les contrôleurs hôte sont configurés pour déléguer des tâches de gestion du domaine à eux-mêmes. Le contrôleur hôte sur chaque hôte interagit avec le contrôleur de domaine pour contrôler le cycle de vie des instances de serveur d'applications s'exécutant sur l'hôte et d'aider du contrôleur de domaine pour les gérer. Chaque hôte peut contenir plusieurs groupes de serveurs. Un groupe de serveurs est un ensemble d'instances de serveur, avec JBoss Enterprise Application Platform installé dessus, géré et configuré comme un ensemble. Étant donné que le contrôleur de domaine gère la configuration et les applications qui sont déployées sur des groupes de serveurs, chaque serveur d'un groupe de serveurs partage la même configuration et les mêmes déploiements.

Il est possible pour le contrôleur de domaine, un contrôleur d'hôte unique, et pour plusieurs serveurs d'exécuter dans la même instance de la plate-forme EAP, sur le même système physique. Les contrôleurs hôte sont liés à des hôtes spécifiques physiques (ou virtuels). Vous pouvez exécuter plusieurs contrôleurs hôte sur le même matériel si vous utilisez des configurations différentes, afin que les ports et les autres ressources n'entrent pas en conflit.

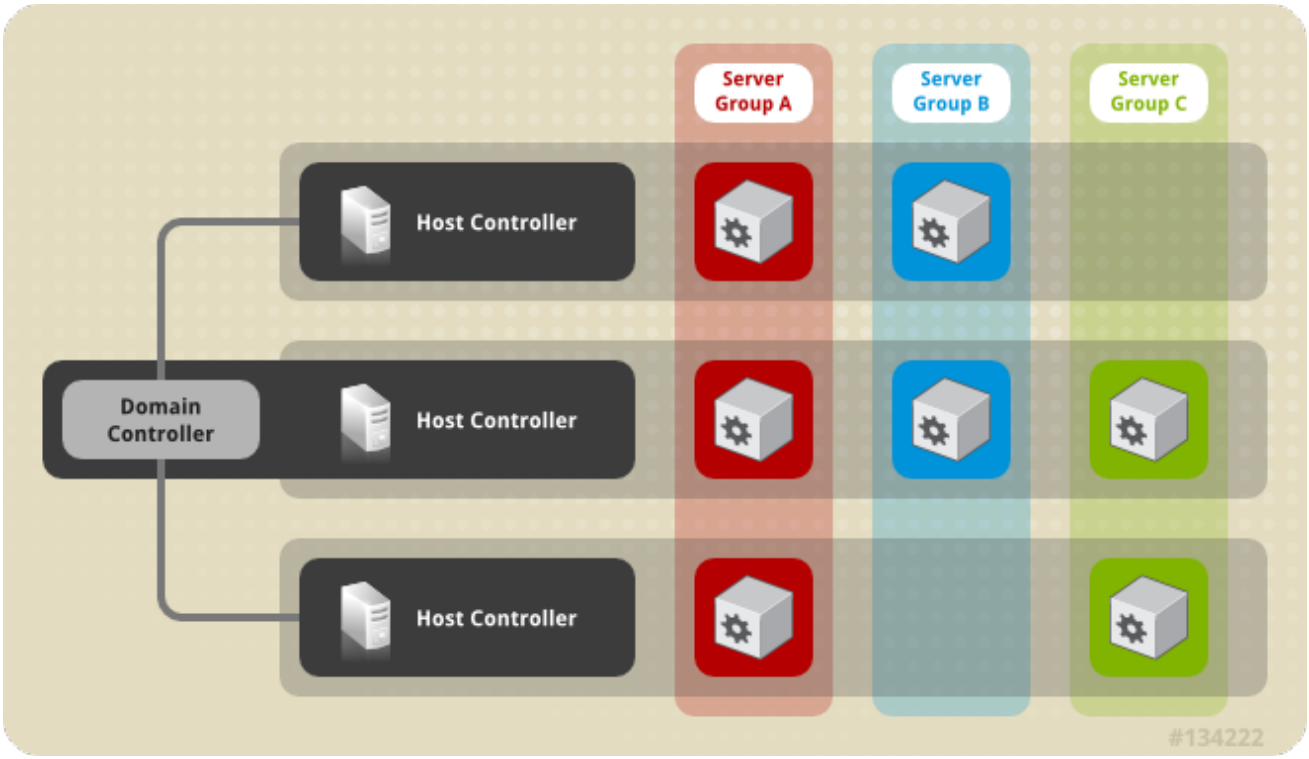



**Figure 1.1. Représentation graphique d'un domaine géré**

Report a bug

## 1.6. CONTRÔLEUR DE DOMAINE

Un contrôleur de domaine est une instance de serveur de JBoss Enterprise Application Platform qui agit en tant que point central de gestion pour un domaine. Une instance de contrôleur d'hôte est configurée pour agir en tant que contrôleur de domaine. Les responsabilités principales d'un contrôleur de gestion sont :

- Maintenir la politique centrale de gestion du domaine
- S'assurer que tous les contrôleurs soient mis au courant de leurs contenus actuels
- Assister tous les contrôleurs pour que toutes les instances en cours de JBoss Enterprise Application Platform soient configurées suivant cette politique

La politique de gestion centrale est stockée par défaut dans le fichier **`domain/configuration/domain.xml`**, dans le fichier d'installation JBoss Enterprise Application Server 6 non compressé, sur le système de fichiers de l'hôte du contrôleur de domaines.

On doit pouvoir trouver un fichier de domain.xml dans le répertoire de **`domain/configuration`** du contrôleur hôte qui est destiné à être exécuté comme contrôleur de domaine. Ce fichier n'est pas obligatoire pour les installations sur les contrôleurs hôtes non destinés à être conçus comme contrôleur de domaine. Cependant, la présence d'un fichier **`domain.xml`** sur un tel serveur ne nuit pas. Le fichier domain.xml contient la configuration des divers profils qui peuvent être configurés pour exécuter sur les instances de serveur d'un domaine. Une configuration de profil inclut la configuration détaillée des différents sous-systèmes qui composent un profil. La configuration de domaine inclut également la définition des groupes de sockets et la définition des groupes de serveurs.

Report a bug

## 1.7. ÉCHECS DE CONTRÔLEURS DE DOMAINES

Si un Contrôleur de domaines échoue pour une raison quelconque, vous pourrez configurer et promouvoir des Contrôleurs de domaines d'hôtes pour qu'ils se substituent à des contrôleurs de domaines.

Report a bug

## 1.8. CONTRÔLEUR HÔTE

Un contrôleur hôte est lancé, lorsque le script **`domain.sh`** ou **`domain.bat script`** est exécuté sur un hôte. Le responsabilité primaire d'un contrôleur hôte est la gestion de serveur. Il délègue des tâches de gestion de domaine et est responsable de lancer et d'arrêter les processus de serveurs d'applications individuels qui s'exécutent sur son hôte. Il interagit avec le contrôleur de domaines pour gérer la communication entre les serveurs et le contrôleur de domaines. Plusieurs contrôleurs hôte d'un domaine peuvent interagir avec un contrôleur de domaine unique. Par conséquent, tous les contrôleurs hôtes et les instances de serveurs exécutant en mode de domaine unique peuvent avoir un contrôleur de domaine unique et doivent appartenir au même domaine.

Chaque contrôleur hôte lit par défaut sa configuration à partir du fichier **`domain/configuration/host.xml`** situé dans le fichier d'installation de JBoss Enterprise Application Platfrom 6 décompressé sur le système de fichiers de son hôte. Le fichier **`host.xml`** contient les informations de configuration suivantes spécifiques à l'hôte particulier :

- Énumère les noms des instances de JBoss Enterprise Application Platform 6 sensées être exécutées à partir de l'installation.
- Une des configurations suivantes :
  - la façon dont le contrôleur contacte le contrôleur de domaines pour s'enregistrer lui-même et pour accéder à la configuration de domaine
  - la façon de rechercher et contacter un contrôleur de domaines éloigné
  - comment le contrôleur d'hôtes doit se persuader lui-même d'agir en tant que contrôleur de domaines
- Configuration d'éléments spécifiques à l'installation physique locale. Ainsi, les définitions d'interfaces nommées déclarées dans domain.xml peuvent être mappées vers une adresse IP spécifique à une machine dans host.xml. Les noms de chemins d'accès abstraits de domain.xml peuvent être mappés vers les chemins d'accès d'host.xml.

Report a bug

## 1.9. LES GROUPES DE SERVEURS

Un groupe de serveurs est une collection d'instances de serveur qui sont gérées et configurées ensemble. Dans un domaine géré, chaque instance de serveur d'applications appartient à un groupe de serveurs, même si c'est le seul membre. Les instances de serveur d'un groupe partagent la même configuration de profil et de déploiement de leur contenu. Les contrôleurs de domaines et les contrôleurs d'hôtes appliquent la configuration standard sur toutes les instances de serveurs de chaque groupe de serveur dans son domaine. Un domaine peut se composer de plusieurs groupes de serveurs. Différents groupes de serveurs peuvent être configurés avec différents profils et déploiements, par exemple dans un domaine avec différents niveaux de serveurs offrant différents services. Différents groupes de serveurs peuvent également avoir le même profil et les déploiements, comme par exemple, pour

supporter les scénarios de mise à jour d'applications qui s'enchaînent, quand une interruption totale du service est évitée grâce à la mise à niveau préalable de l'application sur un groupe de serveurs, suivie d'une mise à niveau d'un deuxième groupe de serveurs.

Voici un exemple de définition de groupe de serveurs :

```
<server-group name="main-server-group" profile="default">
 <socket-binding-group ref="standard-sockets"/>
  <deployments>
   <deployment name="foo.war_v1" runtime-name="foo.war"/>
   <deployment name="bar.ear" runtime-name="bar.ear"/>
  </deployments>
</server-group>
```

Un groupe de serveurs inclut les attributs obligatoires suivants :

- nom : le nom du groupe de serveurs
- profil : le nom du profil du groupe de serveurs

Un groupe de serveurs inclut les attributs optionnels suivants :

- socket-binding-group: le nom du groupe de liaisons de sockets par défaut à utiliser pour les serveurs dans le groupe. Ce nom peut être remplacé sur la base d'un serveur à la fois dans host.xml. Si le nom socket-binding-group n'est pas fourni dans l'élément server-group, il doit être donné pour chaque serveur dans le fichier host.xml.
- déployments : le contenu de déploiement à déployer sur les serveurs du groupe
- system-properties : les propriétés système à définir sur les serveurs du groupe
- jvm : les paramètres de configuration par défaut de tous les serveurs du groupe. Le contrôleur hôte fait fusionner ces paramètres dans n'importe quelle configuration fournie par host.xml pour établir les paramètres à utiliser dans la JVM du serveur.

Report a bug

## 1.10. ABOUT JBOSS ENTERPRISE APPLICATION PLATFORM 6 PROFILES

Le concept des profils qui ont été utilisées dans les versions précédentes d'Enterprise Application Platform n'est plus utilisé. JBoss Enterprise Application Platform 6 utilise maintenant un petit nombre de fichiers de configurations simples.

Les modules et les pilotes sont chargés en fonction des besoins, donc le concept du profil par défaut utilisé dans les anciennes versions de JBoss Enterprise Application Platform 6 où les profils étaient utilisés pour rendre le démarrage du serveur plus performant n'est pas très utile. Au moment du déploiement, les dépendances du module sont définies, ordonnancées, et résolues par le serveur ou le contrôleur du domaine, et chargées dans le bon ordre. Quand le déploiement est retiré, les modules sont retirés du chargement quand ils ne sont plus utiles à aucun déploiement.

Il est possible de désactiver les modules ou de supprimer le déploiement de pilotes ou autres services manuellement en retirant les sous-systèmes de la configuration. Cependant, dans la plupart des cas, cela n'est pas utile. Si aucune de vos applications utilisent un module, il ne sera pas chargé.

Report a bug

# CHAPITRE 2. GESTION DE SERVEUR D'APPLICATIONS

## 2.1. DÉMARRER JBOSS ENTERPRISE APPLICATION PLATFORM 6

### 2.1.1. Démarrer JBoss Enterprise Application Platform 6

Démarrer JBoss Enterprise Application Platform 6 d'une des manières suivantes :

- Section 2.1.3, « Démarrer JBoss Enterprise Application Platform 6 en tant que domaine géré »
- Section 2.1.2, « Démarrez JBoss EAP 6 comme un serveur autonome »

Report a bug

### 2.1.2. Démarrez JBoss EAP 6 comme un serveur autonome

**Résumé**

Cette rubrique couvre toutes les étapes à couvrir pour démarrer JBoss Enterprise Application Platform 6 en tant que serveur autonome.

**Procédure 2.1. Démarrer le Service de plate-forme comme serveur autonome.**

1. **Dans Red Hat Enterprise Linux.**
   Exécuter la commande suivante : ***EAP_HOME*/bin/standalone.sh**

2. **Dans Microsoft Windows Server**
   Exécuter la commande suivante : ***EAP_HOME*\bin\standalone.bat**

3. **Option : indiquer les paramètres supplémentaires.**
   Pour imprimer une liste de paramètres supplémentaires à passer aux scripts de démarrage, utiliser le paramètre ***-h***.

**Résultat**

L'instance du serveur autonome JBoss EAP 6 démarre.

Report a bug

### 2.1.3. Démarrer JBoss Enterprise Application Platform 6 en tant que domaine géré

**Procédure 2.2. Démarrer le Service de plate-forme comme serveur géré**

1. **Dans Red Hat Enterprise Linux.**
   Exécutez la commande : ***EAP_HOME*/bin/domain.sh**

2. **Dans Microsoft Windows Server**
   Exécutez la commande : ***EAP_HOME*\bin\domain.bat**

3. **En option : passez des paramètres supplémentaires au script de démarrage.**
   Pour obtenir une liste de paramètres que vous pourrez passer au script de démarrage, utilisez le paramètre ***-h***.

**Résultat**

L'instance du serveur géré JBoss EAP 6 démarre.

Report a bug

### 2.1.4. Démarrer la plateforme Enterprise Application Platform avec une Configuration différente.

Si vous n'indiquez pas de fichier de configuration, le serveur démarrera avec le fichier par défaut. Malgré tout, quand vous démarrez le serveur, vous pouvez spécifier Configuration manuelle. Le processus varie légèrement, suivant que vous utilisez un Domaine géré ou un Serveur autonome, et suivant le système d'exploitation que vous utilisez.

**Prérequis**

- Avant d'utiliser un fichier de configuration alternatif, préparez-le à l'aide de la configuration par défaut modèle. Pour un domaine géré, le fichier de configuration doit être placé dans ***EAP_HOME*/domain/configuration/**. Pour les Serveurs autonomes, le fichier de configuration devra être mis dans ***EAP_HOME*/standalone/configuration/**.

**NOTE**

Plusieurs exemples de configurations sont inclus dans les répertoires de configuration. Utiliser ces exemples pour activer des fonctionnalités supplémentaires, comme clustering ou l'API XTS de Transactions.

**Procédure 2.3. Démarrage de l'instance par une configuration différente**

1. **Domaine géré**
   Pour un Domaine géré, fournir le nom du fichier de configuration comme option du paramètre ***--domain-config***. Vous n'avez pas besoin de procurer le nom complet, si le fichier de configuration se trouve dans le répertoire ***EAP_HOME*/domain/configuration/**.

   **Exemple 2.1. Utilisation d'un fichier de configuration alternatif pour un Domaine géré dans Red Hat Enterprise Linux**

   ```
   [user@host bin]$ ./domain.sh --domain-config=domain-alternate.xml
   ```

   **Exemple 2.2. Utilisation d'un fichier de configuration alternatif pour un Domaine géré dans un serveur Microsoft Windows**

   ```
   C:\EAP_HOME\bin> domain.bat --domain-config=domain-alternate.xml
   ```

2. **Serveur autonome**
   Pour un Domaine autonome, fournir le nom du fichier de configuration comme option du paramètre ***--server-config***. Vous n'avez pas besoin de procurer le nom complet, si le fichier de configuration se trouve dans le répertoire ***EAP_HOME*/standalone/configuration/**.

   **Exemple 2.3. Utiliser un fichier de configuration alternatif pour un Serveur autonome Red Hat Enterprise Linux.**

```
[user@host bin]$ ./standalone.sh --server-config=standalone-alternate.xml
```

**Exemple 2.4. Utiliser un fichier de configuration alternatif pour un Serveur autonome Microsoft Windows.**

```
C:\EAP_HOME\bin> standalone.bat --server-config=standalone-alternate.xml
```

**Résultat**

La plateforme Enterprise Application Platform est maintenant en cours d'exécution, avec une configuration différente.

Report a bug

### 2.1.5. Stopper JBoss Enterprise Application Platform 6

La façon dont vous arrêtez la plate-forme JBoss Enterprise Application Platform 6 dépend de la façon dont elle a été lancée. Cette tâche couvre l'arrêt d'une instance qui a démarré de manière interactive, faire cesser une instance qui a été démarrée par un service et faire cesser une instance qui a été mise en arrière-plan par un script.

**NOTE**

Cette tâche ne règle pas l'arrêt d'un serveur ou d'un groupe de serveurs dans un Domaine géré. Pour ces scénarios, voir Section 2.2.3, « Stopper un serveur qui utilise une Console de gestion ».

**Procédure 2.4. Stopper une instance autonome de JBoss Enterprise Application Platform 6**

1. **Stopper une instance qui a été démarrée de façon interactive à partir d'une invite de commande.**
   Appuyez sur **`Ctrl-C`** dans le terminal où JBoss Enterprise Application Platform 6 exécute.

2. **Stopper une instance qui a démarré en tant que service de système d'exploitation.**
   Suivant votre système d'exploitation, utiliser une des procédures suivantes :

   - **Red Hat Enterprise Linux**
     Dans Red Hat Enterprise Linux, si vous avez écrit un script de service, utiliser sa fonction **`stop`**. Cela devra être inscrit dans le script. Ensuite, vous pourrez utiliser **`service scriptname stop`**, avec *scriptname* comme nom de script.

   - **Microsoft Windows Server**
     Dans Microsoft Windows, utiliser la commande **`net service`**, ou bien faîtes cesser le service à partir de l'applet **`Services`** qui se trouve dans le Panneau de contrôle.

3. **Stopper une instance qui exécute en arrière-plan (Red Hat Enterprise Linux)**

   a. Cherchez l'instance dans la liste de processus. Une option consiste à exécuter la commande **`ps aux |grep "[j]ava -server"`**. Cela renverra un résultat pour chaque

instance de JBoss Enterprise Application Platform 6 en cours d'exécution sur la machine locale.

b. Envoyer au processus le signal**TERM**, en exécutant **kill *process_ID***, avec *process_ID* comme numéro de deuxième champ de la commande **ps aux** ci-dessus.

**Résultat**

Chacune de ces solutions ferme la plate-forme JBoss Enterprise Application Platform 6 nettement, ce qui fait qu'aucune donnée n'est perdue.

Report a bug

### 2.1.6. Référence aux variables et arguments à passer à l'exécution du serveur

Le script de démarrage du serveur d'applications accepte l'ajout d'arguments et de variables en cours d'exécution. L'utilisation de ces paramètres permettent au serveur d'être démarré sous d'autres configurations que celles qui sont définies dans les fichiers de configuration **standalone.xml**, **domain.xml** et **host.xml**. Cela peut comprendre le démarrage du serveur par un ensemble de liaisons de sockets différent ou une configuration secondaire. Vous pourrez accéder à une liste des paramètres disponibles en passant la variable d'assistance au démarrage.

**Exemple 2.5.**

L'exemple suivant ressemble au démarrage de serveur expliqué dans Section 2.1.2, « Démarrez JBoss EAP 6 comme un serveur autonome », avec les variables **-h** ou **--help** en plus. Le résultats de cette variable d'assistance sont expliqués dans le tableau ci-dessous.

```
[localhost bin]$ standalone.sh -h
```

**Tableau 2.1. Tableau des arguments et variables du temps d'exécution**

| Argument ou Variable | Description |
|---|---|
| --admin-only | Définir le type d'exécution du serveur à ADMIN_ONLY. Cela le fera ouvrir les interfaces administratives et il pourra ainsi accepter les ordres de gestion, mais il ne pourra pas démarrer d'autres services de runtime ou accepter les demandes de l'utilisateur final. |
| -b=<value> | Définir la propriété système **jboss.bind.address** à la valeur donnée. |
| -b <value> | Définir la propriété système **jboss.bind.address** à la valeur donnée. |
| -b<interface>=<value> | Définir la propriété système **jboss.bind.address.<interface>** à la valeur donnée. |
| -c=<config> | Nommer le fichier de configuration du serveur à utiliser. La valeur par défaut est **standalone.xml**. |
| -c <config> | Nommer le fichier de configuration du serveur à utiliser. La valeur par défaut est **standalone.xml**. |

| Argument ou Variable | Description |
|---|---|
| --debug [<port>] | Activer le mode de débogage par un argument en option qui indique le port. Ne fonctionne que si le script de lancement le supporte. |
| -D<name>[=<value>] | Définir une propriété système. |
| -h | Afficher le message d'assistance et sortir. |
| --help | Afficher le message d'assistance et sortir. |
| -P=<url> | Télécharger les propriétés système de l'URL donné. |
| -P <url> | Télécharger les propriétés système de l'URL donné. |
| --properties=<url> | Télécharger les propriétés système de l'URL donné. |
| -S<name>[=<value>] | Définir une propriété de sécurité. |
| --server-config=<config> | Nommer le fichier de configuration du serveur à utiliser. La valeur par défaut est **`standalone.xml`**. |
| -u=<value> | Définir la propriété système **`jboss.default.multicast.address`** à la valeur donnée. |
| -u <value> | Définir la propriété système **`jboss.default.multicast.address`** à la valeur donnée. |
| -V | Afficher la version du serveur d'application et sortir. |
| -v | Afficher la version du serveur d'application et sortir. |
| --version | Afficher la version du serveur d'application et sortir. |

Report a bug

## 2.2. DÉMARRER ET ARRÊTER LES SERVEURS

### 2.2.1. Démarrer et Arrêter les Serveurs

Vous pouvez démarrer et arrêter les serveurs par le Management CLI ou la Console de gestion.

Si vous exécutez une instance de Serveur Autonome, vous pourrez éteindre le serveur par l'opération **`shutdown`** du Management CLI. Il n'y a pas d'équivalent dans la Console de gestion, car vous avez la possibilité de passer par votre système de fichiers pour fermer l'instance en cours.

Si vous exécutez un Domaine géré, la Console de gestion va vous permettre de démarrer ou de stopper sélectivement des serveurs spécifiques pour ce domaine.

Si vous souhaitez démarrer ou stopper un Domaine géré par le Management CLI, vous pouvez le faire en utilisant les commandes appropriées à la situation.

**Exemple 2.6. Démarrer/Stopper un Groupe de serveurs dans un Domaine géré via un Management CLI**

```
# start server group named main
/server-group=main:start-servers
# stop server group named main
/server-group=main:stop-servers
```

**Exemple 2.7. Démarrer/Stopper une Instance de serveur dans un Domaine géré via un Management CLI**

```
# start server instance named server-one
/host=master/server-config=server-one:start
# stop server instance named server-one
/host=master/server-config=server-one:stop
```

**Exemple 2.8. /Stopper un Hôte de serveur dans un Domaine géré via un Management CLI**

```
# stop server host named master
/host=master:shutdown
```

Report a bug

### 2.2.2. Démarrer un serveur par la Console de gestion

**Prérequis**

- Section 2.1.3, « Démarrer JBoss Enterprise Application Platform 6 en tant que domaine géré »
- Section 3.4.2, « Connectez-vous à la Console de management »

**Procédure 2.5. Démarrer le serveur**

1. **Naviguez dans `Server Instances` dans la Console de gestion**
   a. Sélectionner l'onglet **`Runtime`** en haut et à droite de la console.
   b. Sélectionner **Domain Status** → **Server Instances** à partir du menu à gauche de la console.

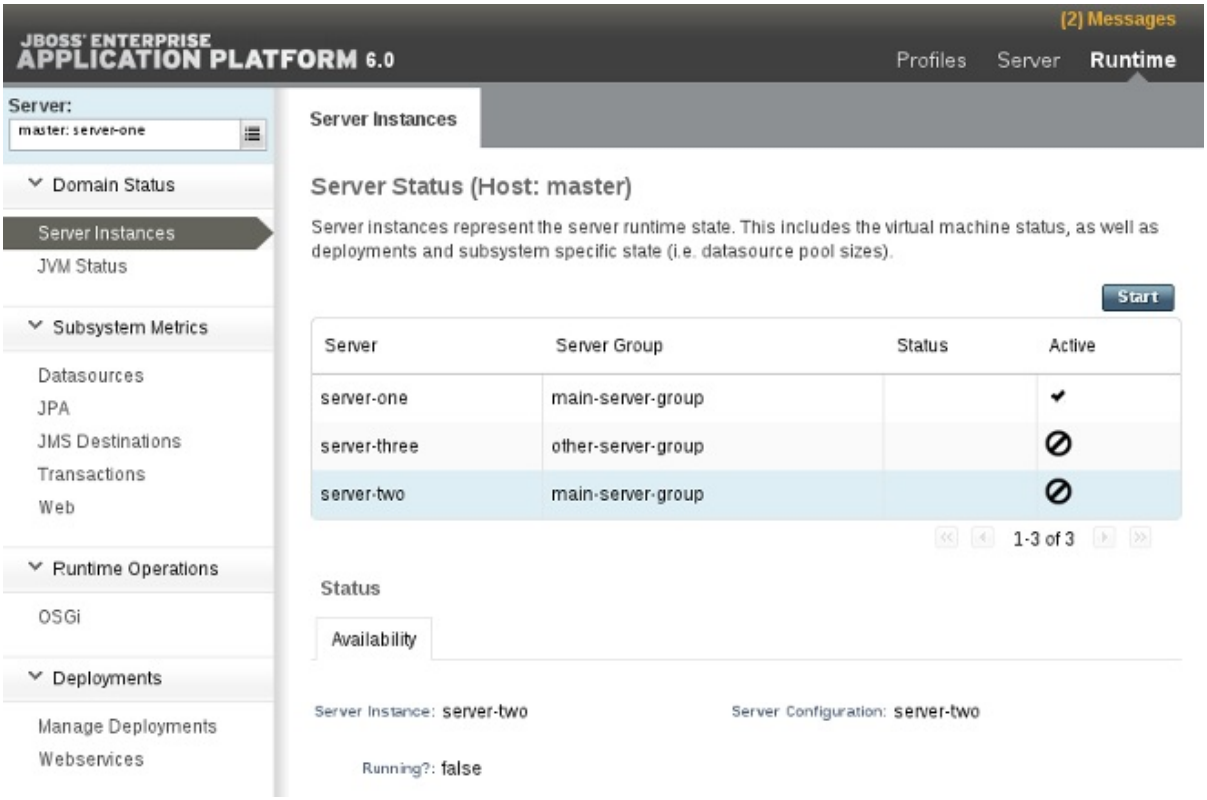



**Figure 2.1. Instances du serveur**

2. **Sélectionner un serveur**
   À partir de la liste **`Server Instances`**, sélectionner le serveur que vous souhaitez démarrer. Les serveurs qui sont en cours d'exécution sont indiqués.

3. **Cliquer sur le bouton `Start`**
   Cliquer sur le bouton **`Start`** en haut de la liste du serveur pour ouvrir la fenêtre de dialogue de confirmation. Cliquer sur le bouton **`Confirm`** pour démarrer le serveur.

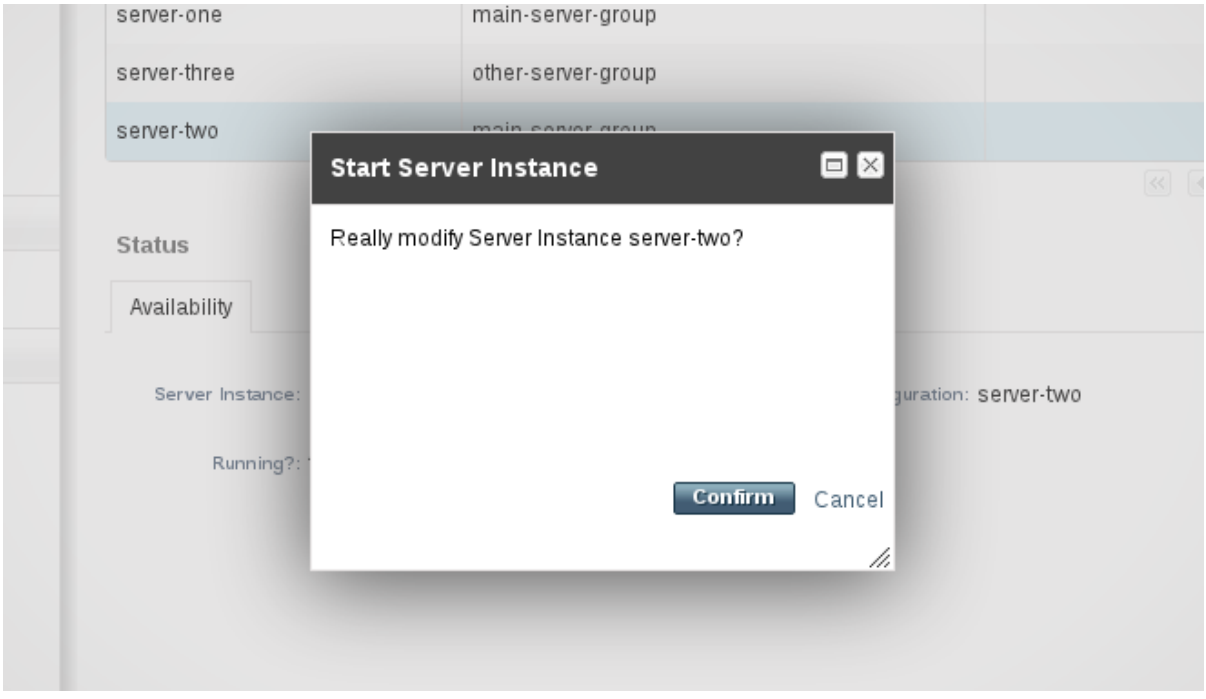



**Figure 2.2. Confirmer la modification de serveur**

**Résultat**

Le serveur sélectionné démarre et exécute.

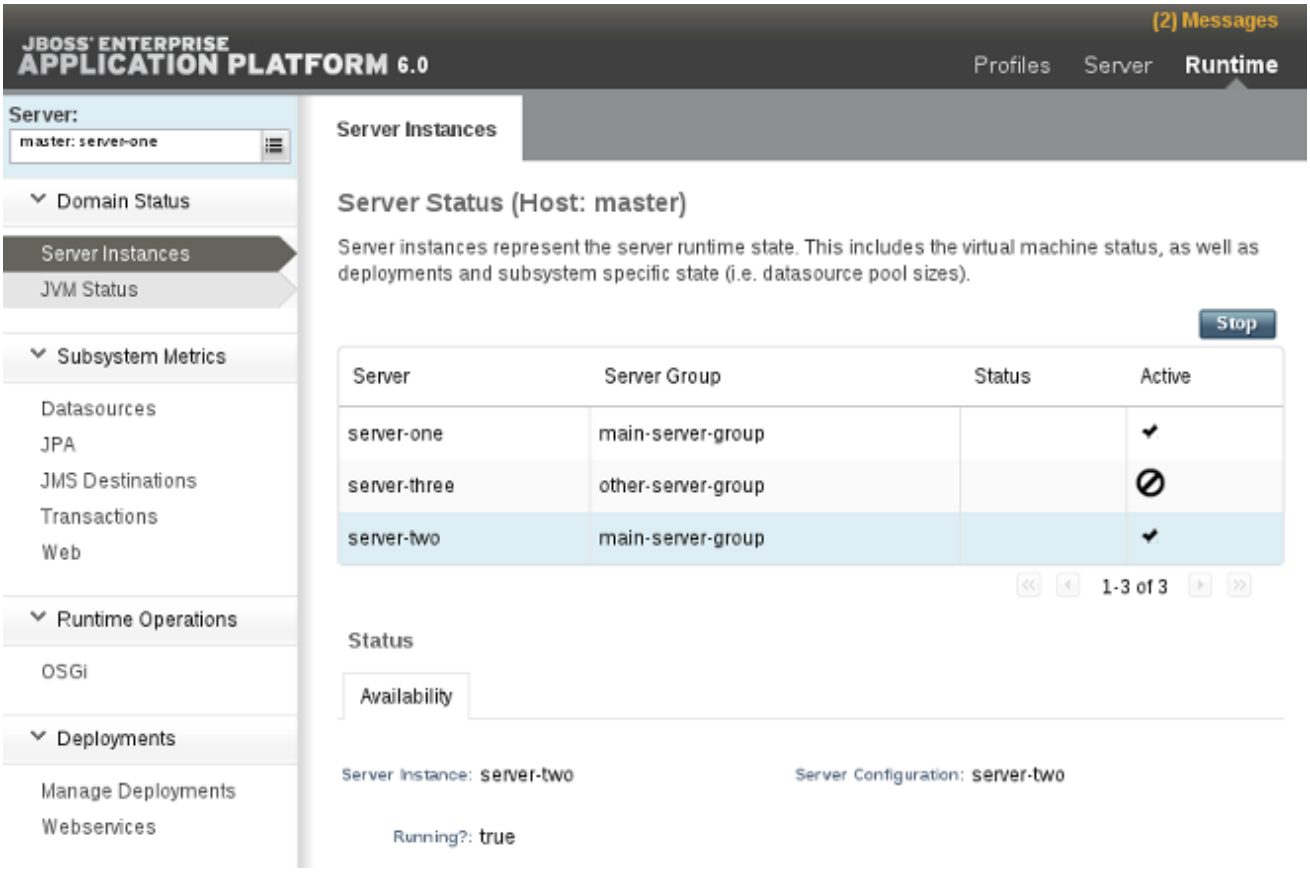



**Figure 2.3. Serveur démarré**

Report a bug

### 2.2.3. Stopper un serveur qui utilise une Console de gestion

#### Prérequis

- Section 2.1.3, « Démarrer JBoss Enterprise Application Platform 6 en tant que domaine géré »
- Section 3.4.2, « Connectez-vous à la Console de management »

#### Procédure 2.6. Stopper un serveur qui utilise une Console de gestion

1. **Naviguez dans `Server Instances` dans la Console de gestion**
   a. Sélectionner l'onglet **`Runtime`** en haut et à droite de la console.
   b. Sélectionner **Domain Status** → **Server Instances** à partir du menu à gauche de la console.

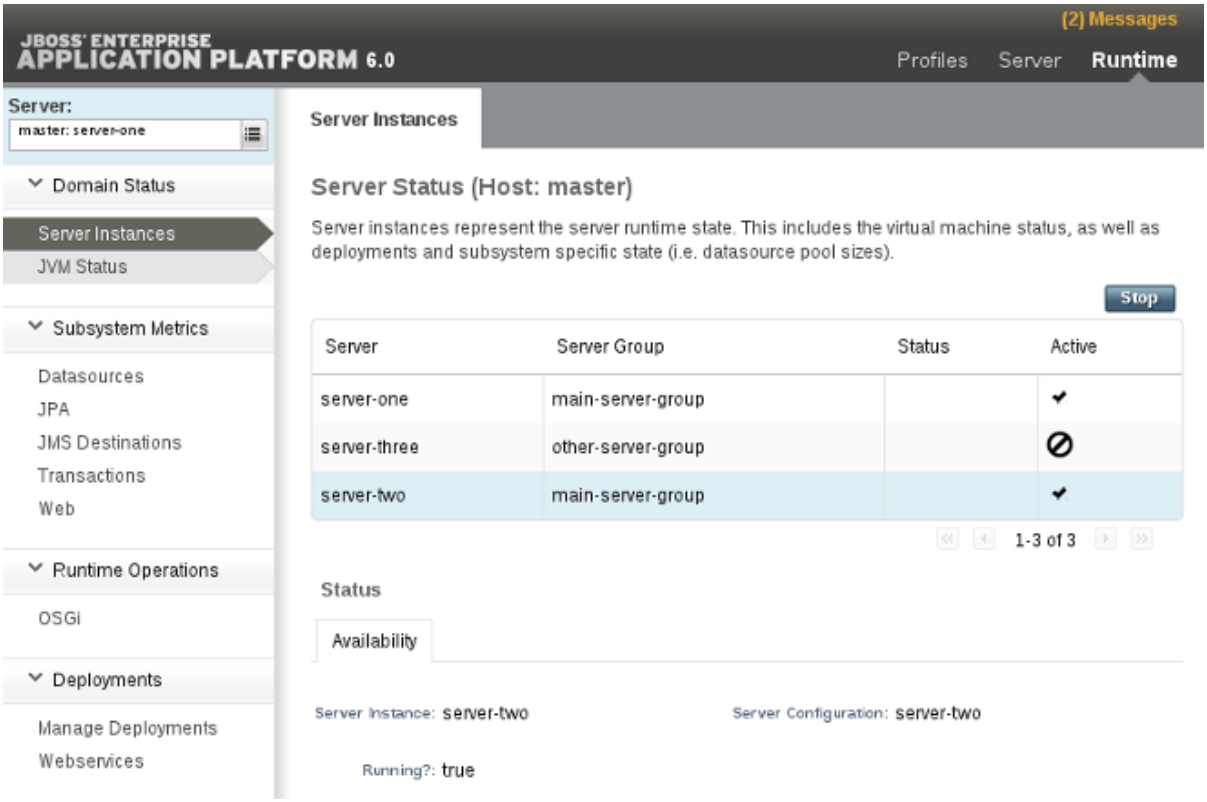



**Figure 2.4. Instances du serveur**

2. **Sélectionner un serveur**
   À partir de la liste **`Server Instances`**, sélectionner le serveur que vous souhaitez stopper. Les serveurs qui sont en cours d'exécution sont indiqués.

3. **Cliquez sur le bouton `Stop`.**
   Cliquer sur le bouton **`Stop`** en haut de la liste du serveur pour ouvrir la fenêtre de dialogue de confirmation. Cliquer sur le bouton **`Confirm`** pour stopper le serveur.

**Résultat**

Le serveur sélectionné est stoppé.

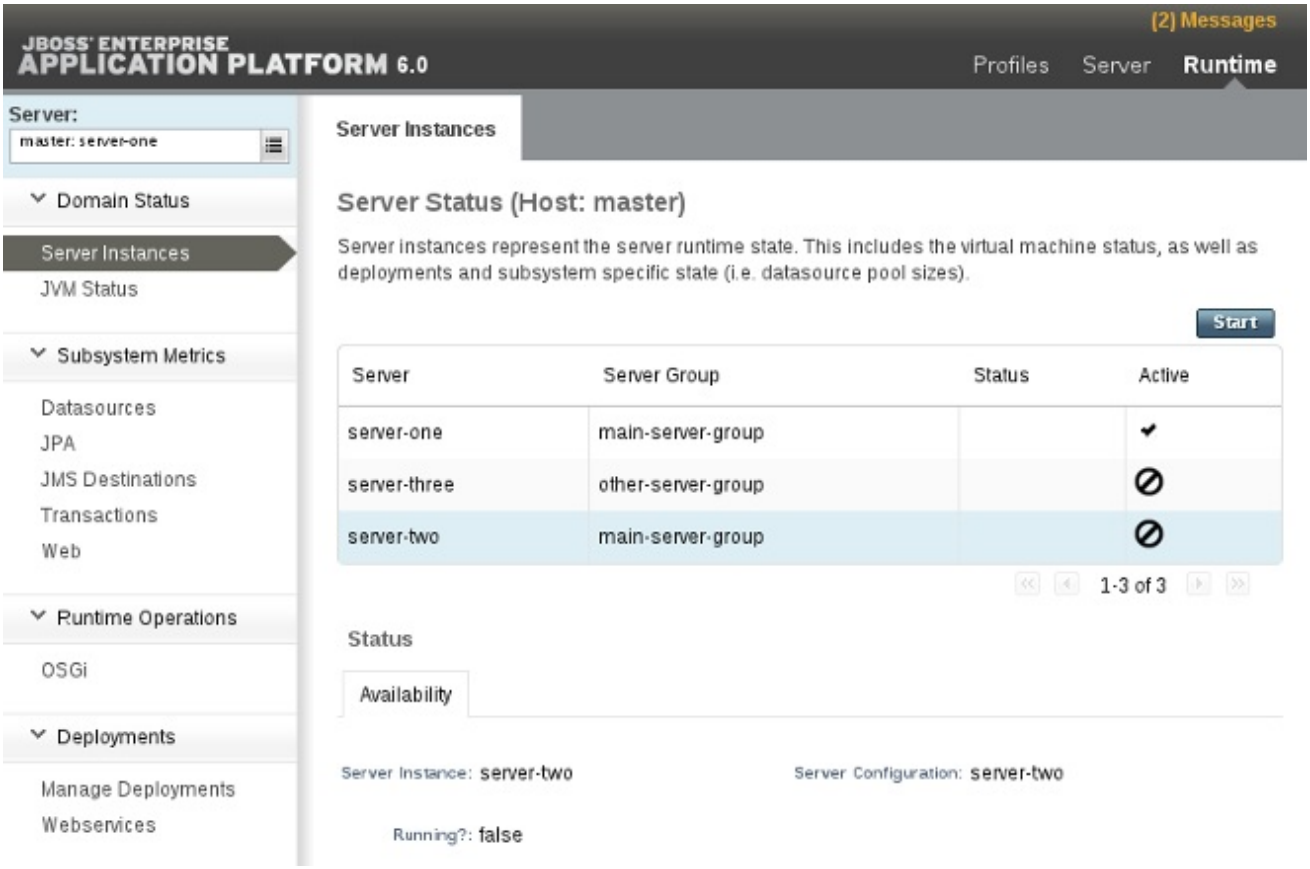



**Figure 2.5. Serveur stoppé**

Report a bug

## 2.3. CHEMINS D'ACCÈS AUX SYSTÈMES DE FICHIERS

### 2.3.1. Chemins d'accès aux systèmes de fichiers

JBoss Enterprise Application Platform 6 utilise des noms logiques pour les chemins de systèmes de fichiers. Les fichiers de configuration **`domain.xml`**, **`host.xml`** et **`standalone.xml`** incluent tous une section où les chemins d'accès peuvent être déclarés. D'autres sections de la configuration peuvent ensuite référencer ces chemins par leur nom logique, évitant la déclaration du chemin d'accès absolu pour chaque instance. Cela profite aux efforts de configuration et d'administration car cela permet à des configurations hôtes spécifiques de résoudre des noms logiques universels.

Par exemple, la configuration du sous-système de logging comprend une référence au chemin **`jboss.server.log.dir`** qui pointe vers le répertoire **`log`** du serveur.

**Exemple 2.9. Exemple de chemin d'accès relatif du répertoire de logging**

```
<file relative-to="jboss.server.log.dir" path="server.log"/>
```

JBoss Enterprise Application Platform 6 fournit un nombre de chemins d'accès standards automatiquement sans que l'utilisateur n'ait besoin de les configurer dans un fichier de configuration.

**Tableau 2.2. Chemins d'accès standard**

| Valeur | Description |
|---|---|
| **jboss.home** | Le répertoire root de la distribution JBoss EAP 6. |
| **user.home** | Le répertoire d'accueil de l'utilisateur. |
| **user.dir** | Le répertoire de travail actuel de l'utilisateur |
| **java.home** | Le répertoire d'installation de Java |
| **jboss.server.base.dir** | Le répertoire root d'une instance de serveur individuel. |
| **jboss.server.data.dir** | Le répertoire que le serveur va utiliser pour le stockage de fichier de données persistantes. |
| **jboss.server.log.dir** | Le répertoire que le serveur va utiliser pour le stockage de fichier de journalisation. |
| **jboss.server.tmp.dir** | Le répertoire que le serveur va utiliser pour le stockage de fichiers temporaires. |
| **jboss.domain.servers.dir** | Le répertoire sous lequel le contrôleur hôte va créer la zone de travail des instances de serveurs individuelles, dans un domaine géré. |

Les utilisateurs peuvent ajouter leurs propres chemins d'accès ou bien les remplacer tous sauf les cinq premiers en ajoutant l'élément **path** dans leur fichier de configuration. L'exemple suivant indique la nouvelle déclaration de chemin relatif qui se rapporte au répertoire root de l'instance du serveur individuel.

**Exemple 2.10. Format d'un chemin relatif**

```
<path name="examplename" path="example/path" relative-
to="jboss.server.data.dir"/>
```

La structure de la déclaration du chemin utilise les attributs suivants.

**Tableau 2.3. Attributs de chemin d'accès**

| Attribut | Description |
|---|---|
| **name** | Le nom du chemin d'accès. |
| **path** | Le chemin d'accès du système de fichier. Considéré comme chemin absolu, à moins que l'attribut **relative-to** ne soit spécifié, dans lequel cas, la valeur sera traitée comme étant relative à ce chemin. |

| Attribut | Description |
|---|---|
| **relative-to** | Un attribut optionnel qui indique le nom d'un autre nom anciennement nommé, ou bien qui correspond à un chemin standard défini par le système. |

Un élément **path** (chemin) d'un fichier de configuration **domain.xml** ne requiert que le nom de l'attribut. Il n'a pas besoin d'inclure des informations sur le chemin du système de fichiers, comme le montre l'exemple suivant.

**Exemple 2.11. Exemple de chemin de domaine**

```
<path name="example"/>
```

Cette configuration déclare simplement qu'il existe un chemin exemple nommée **exemple** auquel les autres parties de la configuration du **domain.xml** peuvent faire référence. L'emplacement du système de fichiers courant déclaré par l'**exemple** est spécifique aux fichiers de configuration **host.xml** respectifs des instances de l'hôte qui se joignent aux groupes de domaine. Si cette approche est utilisée, il doit y avoir un élément de chemin dans l'**host.xml** de chaque machine, qui indique le chemin du système de fichier.

**Exemple 2.12. Exemple de chemin d'hôte**

```
<path name="example" path="path/to/example" />
```

Un élément **path** d'un fichier **standalone.xml** doit inclure la spécification du chemin d'accès du système de fichier.

Report a bug

## 2.4. HISTORIQUE DU FICHIER DE CONFIGURATION

### 2.4.1. Les fichiers de configuration de JBoss Enterprise Application Platform 6

La configuration de JBoss Enterprise Application Platform 6 a considérablement changé depuis les dernières versions. Une des différences principale est l'utilisation d'une structure de fichier de configuration simplifiée, qui comprend un ou plusieurs des fichiers répertoriés ci-dessous :

**Tableau 2.4. Emplacements de Fichiers de configuration**

| Mode du serveur | Emplacement | But |
|---|---|---|
| domain.xml | ***EAP_HOME*/domain/configuration/domain.xml** | Il s'agit du fichier de configuration principal d'un domaine géré. Seul le master du domaine lit ce fichier. Il peut être supprimé pour les autres membres du domaine. |

| Mode du serveur | Emplacement | But |
|---|---|---|
| host.xml | ***EAP_HOME*/domain/configuration/host.xml** | Ce fichier contient les détails de configuration spécifiques à un hôte physique dans un domaine géré, tels que les interfaces réseau, les liaisons de sockets, le nom de l'hôte et d'autres détails spécifiques à l'hôte. Le fichier **host.xml** contient toutes les fonctionnalités de **hôte-master.xml** et **hôte-slave.xml**, qui sont décrits ci-dessous. Ce fichier n'est pas présent pour les serveurs autonomes. |
| host-master.xml | ***EAP_HOME*/domain/configuration/host-master.xml** | Ce fichier inclut tous les détails de configuration nécessaires pour exécuter un serveur en tant que serveur maître de domaine géré. Ce fichier n'est pas présent dans les serveurs autonomes. |
| host-slave.xml | ***EAP_HOME*/domain/configuration/host-slave.xml** | Ce fichier inclut tous les détails de configuration nécessaires pour exécuter un serveur en tant que serveur esclave de domaine géré. Ce fichier n'est pas présent dans les serveurs autonomes. |
| standalone.xml | ***EAP_HOME*/standalone/configuration/standalone.xml** | C'est le fichier de configuration par défaut pour un serveur autonome. Il contient toutes les informations sur le serveur autonome, y compris les sous-systèmes, réseautage, déploiements, les liaisons de sockets et autres détails configurables. Cette configuration est utilisée automatiquement lorsque vous démarrez votre serveur autonome. |

| Mode du serveur | Emplacement | But |
|---|---|---|
| standalone-full.xml | ***EAP_HOME*/standalone/configuration/standalone-full.xml** | Il s'agit d'un exemple de configuration pour un serveur autonome. Il prend en charge chaque sous-système possible à l'exception de ceux destinés à la haute disponibilité. Pour l'utiliser, arrêtez votre serveur et redémarrez à l'aide de la commande suivante : ***EAP_HOME*/bin/standalone.sh -c standalone-full.xml** |
| standalone-ha.xml | ***EAP_HOME*/standalone/configuration/standalone-ha.xml** | Cet exemple de fichier de configuration active tous les sous-systèmes par défaut et ajoute les **mod_cluster** et les sous-systèmes JGroups pour un serveur autonome, afin qu'il puisse participer à un cluster de haute disponibilité ou d'équilibrage de la charge. Ce fichier n'est pas applicable à un domaine géré. Pour utiliser cette configuration, arrêtez votre serveur et redémarrez le à l'aide de la commande suivante : ***EAP_HOME*/bin/standalone.sh -c standalone-ha.xml** |
| standalone-full-ha.xml | ***EAP_HOME*/standalone/configuration/standalone-full-ha.xml** | Il s'agit d'un exemple de configuration pour un serveur autonome. Il prend en charge chaque sous-système possible incluant ceux destinés à la haute disponibilité. Pour l'utiliser, arrêtez votre serveur et redémarrez à l'aide de la commande suivante : ***EAP_HOME*/bin/standalone.sh -c standalone-full-ha.xml** |

Ce sont les emplacements par défaut uniquement. Vous pouvez indiquer un fichier de configuration différent en cours d'exécution.

Report a bug

### 2.4.2. Historique du fichier de configuration

Les fichiers de configuration du serveur d'applications incluent **standalone.xml**, ainsi que les fichiers

**domain.xml** et **host.xml**. Alors que ces fichiers sont modifiables directement, la méthode recommandée consiste à configurer le modèle de serveur d'applications par les opérations de gestion disponibles, y compris Management CLI et la Console de gestion.

Pour aider à la maintenance et à la gestion de l'instance de serveur, le serveur d'applications crée une version horodatée du fichier de configuration original au moment du démarrage. Toutes les modifications de configuration supplémentaires suite aux opérations de gestion résultent à la sauvegarde automatique du fichier d'origine, et une copie de travail de l'instance est alors conservée en tant que référence et rollback. Cette fonctionnalité d'archivage s'étend à l'enregistrement, au chargement et à la suppression des snapshots de configuration du serveur pour autoriser les scénarios de rappel et de restauration.

- Section 2.4.3, « Démarrer le serveur par une ancienne configuration »
- Section 2.4.4, « Sauvegarder un snapshot de configuration par le Management CLI »
- Section 2.4.5, « Télécharger un snapshot de configuration »
- Section 2.4.6, « Supprimer un snapshot de configuration par le Management CLI »
- Section 2.4.7, « Lister tous les snapshots de configuration par le Management CLI »

Report a bug

### 2.4.3. Démarrer le serveur par une ancienne configuration

L'exemple suivant vous montre comment démarrer le serveur d'applications par une ancienne configuration dans un serveur autonome par **standalone.xml**. Le même concept s'applique à un domaine géré par **domain.xml** et **host.xml** respectivement.

Cet exemple nous remémore une ancienne configuration sauvegardée automatiquement par le serveur d'applications tandis que les opérations de gestion sont entrain de modifier le modèle de serveur.

1. Identifier la version de sauvegarde que vous souhaitez démarrer. Cet exemple va rappeler l'instance de modèle de serveur qui précédait la première modification qui a lieu suite à un démarrage réussi.

   ```
   EAP_HOME/configuration/standalone_xml_history/current/standalone.v1.
   xml
   ```

2. Démarrer le serveur par cette configuration du modèle de sauvegarde en passant le nom de fichier relatif sous **jboss.server.config.dir**.

   ```
   EAP_HOME/bin/standalone.sh --server-
   config=standalone_xml_history/current/standalone.v1.xml
   ```

**Résultat**

Le serveur d'applications démarre par la configuration sélectionnée.

Report a bug

### 2.4.4. Sauvegarder un snapshot de configuration par le Management CLI

**Résumé**

Les snapshots représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur.

L'exemple suivant utilise le fichier de configuration **`standalone.xml`**, mais le même processus s'applique aux fichiers de configuration **`domain.xml`** et **`host.xml`**.

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 2.7. Télécharger un Snapshot de configuration et Sauvegardez-le**

- **Sauvegarde d'un snapshot**
  Exécuter l'opération **`take-snapshot`** pour acquérir une copie de la configuration du serveur.

```
[standalone@localhost:9999 /] :take-snapshot
{
    "outcome" => "success",
    "result" =>
"/home/User/EAP_HOME/standalone/configuration/standalone_xml_history
/snapshot/20110630-172258657standalone.xml"
```

**Résultat**

Un snapshot de la configuration du serveur en cours a été sauvegardée.

Report a bug

### 2.4.5. Télécharger un snapshot de configuration

Les snapshots de configuration représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur. Le processus de chargement des snapshots ressemble à celui de la méthode pour Section 2.4.3, « Démarrer le serveur par une ancienne configuration », à partir de la ligne de commande et non pas l'interface Management CLI utilisée pour créer, lister et supprimer les snapshots.

L'exemple suivant utilise le fichier **`standalone.xml`**, mais le même processus s'applique aux fichiers **`domain.xml`** et **`host.xml`**.

**Procédure 2.8. Télécharger un snapshot de configuration**

1. Identifier le snapshot à télécharger. Cet exemple va rappeler le fichier suivant du répertoire de snapshots. Le chemin par défaut des fichiers de snapshot est le suivant.

```
EAP_HOME/standalone/configuration/standalone_xml_history/snapshot/20
110812-191301472standalone.xml
```

   Les snapshots sont exprimés par leurs chemins relatifs, selon lesquels l'exemple ci-dessus peut être écrit ainsi.

```
jboss.server.config.dir/standalone_xml_history/snapshot/20110812-
191301472standalone.xml
```

2. Démarrer le serveur par le snapshot de configuration sélectionné en passant le nom du fichier.

```
EAP_HOME/bin/standalone.sh --server-
config=standalone_xml_history/snapshot/20110913-
164449522standalone.xml
```

**Résultat**

Le serveur démarre à nouveau avec la configuration sélectionnée dans le snapshot téléchargé.

Report a bug

### 2.4.6. Supprimer un snapshot de configuration par le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

Les snapshots représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur.

Les exemples suivants utilisent le fichier **`standalone.xml`**, mais le même processus s'applique aux fichiers **`domain.xml`** et **`host.xml`**.

**Procédure 2.9. Supprimer un snapshot particulier**

1. Identifier le snapshot à effacer. Cet exemple va effacer le fichier suivant du répertoire de snapshots.

```
EAP_HOME/standalone/configuration/standalone_xml_history/snapshot/20
110630-165714239standalone.xml
```

2. Exécuter la commande **`:delete-snapshot`** pour supprimer un snapshot particulier, en spécifiant le nom du snapshot comme dans l'exemple ci-dessous.

```
[standalone@localhost:9999 /] :delete-snapshot(name="20110630-
165714239standalone.xml")
{"outcome" => "success"}
```

**Résultat**

Le snapshot a été supprimé.

**Procédure 2.10. Supprimer tous les snapshots**

- Exécuter la commande **`:delete-snapshot(name="all")`** pour supprimer tous les snapshots comme dans l'exemple ci-dessous.

```
[standalone@localhost:9999 /] :delete-snapshot(name="all")
{"outcome" => "success"}
```

**Résultat**

Tous les snapshots ont été supprimés.

Report a bug

### 2.4.7. Lister tous les snapshots de configuration par le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

Les snapshots représentent une copie d'un moment précis d'une configuration de serveur en cours. Ces copies peuvent être sauvegardées et chargées par l'administrateur.

L'exemple suivant utilise le fichier **`standalone.xml`**, mais le même processus s'applique aux fichiers **`domain.xml`** et **`host.xml`**.

**Procédure 2.11. Lister tous les snapshots de configuration**

- **Lister tous les snapshots**
  Lister tous les snapshots sauvegardés en exécutant la commande **`:list-snapshots`**.

```
[standalone@localhost:9999 /] :list-snapshots
{
    "outcome" => "success",
    "result" => {
        "directory" =>
"/home/hostname/EAP_HOME/standalone/configuration/standalone_xml_his
tory/snapshot",
        "names" => [
            "20110818-133719699standalone.xml",
            "20110809-141225039standalone.xml",
            "20110802-152010683standalone.xml",
            "20110808-161118457standalone.xml",
            "20110912-151949212standalone.xml",
            "20110804-162951670standalone.xml"
        ]
    }
}
```

**Résultat**

Les snapshots sont listés.

Report a bug

# CHAPITRE 3. INTERFACES DE GESTION

## 3.1. GESTION DU SERVEUR D'APPLICATIONS

JBoss Enterprise Application Platform 6 vous propose des outils de gestion multiples pour configurer et administrer votre implémentation suivant les besoins. Ces outils comprennent l'interface de commande CLI (Management Command Line Interface), comme exemples d'API de gestion pour permettre aux utilisateurs experts de développer leurs propres outils s'ils le désirent.

Report a bug

## 3.2. LES API (DE L'ANGLAIS APPLICATION PROGRAMMING INTERFACES) DE GESTION

**Clients de Gestion**

JBoss Enterprise Application Platform 6 offre trois approches différentes pour configurer et gérer des serveurs: une interface web, la ligne de commande et un ensemble de fichiers de configuration XML. Malgré que les méthodes recommandées pour la modification du fichier de configuration incluent la Console de Gestion et Management CLI, les modifications de configuration sont toujours synchronisées à travers les différentes vues et sont conservées dans les fichiers XML. Notez que les modifications apportées aux fichiers de configuration XML pendant l'exécution d'une instance de serveur seront remplacées par le modèle de serveur.

**HTTP API**

La Console de gestion est un exemple d'interface web construite avec Google Web Toolkit (GWT). La Console de gestion communique avec le serveur à l'aide de l'interface de gestion HTTP. Le point de terminaison HTTP API est le point d'entrée pour les clients de gestion basés sur le protocole HTTP, pour s'intégrer à la couche de gestion. Il utilise un protocole JSON encodé et un API de style RPC de-typed, pour décrire et exécuter des opérations de gestion en fonction d'un domaine géré ou d'un serveur autonome. L'API HTTP est utilisé par la console web, mais offre aussi des possibilités d'intégration pour un large éventail d'autres clients.

Le point de terminaison HTTP API est situé soit avec le contrôleur de domaine ou une instance de serveur autonome. Le point de terminaison HTTP API sert deux contextes différents; un pour l'exécution des opérations de gestion et de l'autre pour accéder à l'interface web. Par défaut, il s'exécute sur le port 9990.

**Exemple 3.1. Exemple de fichier de configuraiton HTTP API**

```
<management-interfaces>
    [...]
    <http-interface interface="management" port="9990"/>
<management-interfaces>
```

La console web est servie par le même port que l'API de gestion HTTP. Il est important de distinguer entre la Console de gestion accessible comme localhost par défaut, la Console de gestion accessible à distance par un hôte spécifique et une combinaison de port, et l'API du domaine exposé.

**Tableau 3.1. TableTitle**

| URL | Description |
| --- | --- |
| **`http://localhost:9990/console`** | La Console de gestion à laquelle accéde l'hôte local, et qui contrôle la configuration du Domaine géré. |
| **`http://`*`hostname`*`:9990/console`** | La Console de gestion accédée à distance, qui nomme l'hôte et qui contrôle la configuration du Domaine géré. |
| **`http://`*`hostname`*`:9990/management`** | Le HTTP Management API exécute sur le même port que la Console de gestion, affiche les mêmes valeurs et attributs bruts exposés à l'API. |

**API Natif**

Le Management CLI est un exemple d'outil d'API Natif. Cet outil de gestion est disponible à une instance de Serveur autonome ou à un Domaine, permettant ainsi à un utilisateur de se connecter à une instance du Serveur autonome ou au Contrôleur du domaine, et d'exécuter des opérations de gestion rendues disponibles par le modèle «de-typed».

Le point de terminaison de l'API Natif est le point d'entrée pour les clients de gestion qui s'appuient sur le protocole natif pour intégrer la couche de gestion. Il utilise un protocole binaire ouvert et une API style-RPC basée sur un très petit nombre de types Java pour décrire et exécuter des opérations de gestion. Il est utilisé par l'outil de gestion Management CLI, mais offre des capacités d'intégration pour un large éventail d'autres clients également.

Le point de terminaison d'API Natif est co-localisée avec un Contrôleur hôte ou un Serveur autonome. Il doit être activé pour utiliser le ManagementCLI. Par défaut, il s'exécute sur le port 9999.

**Exemple 3.2. Exemple de fichier de configuration d'API natif**

```
<management-interfaces>
    <native-interface interface="management" port="9999"/>
    [...]
<management-interfaces>
```

Report a bug

## 3.3. CONSOLE DE GESTION ET MANAGEMENT CLI

Dans JBoss Enterprise Application Platform 6, toutes les instances de serveurs et toutes les configurations sont gérées par les interfaces de gestion, et non pas par modification de fichiers XML. Malgré que les fichiers de configuration XML puissent toujours être édités, la gestion par les interfaces de gestion fournit une validation supplémentaire et des fonctionnalités avancées pour la gestion persistante des instances de serveurs. Les modifications apportées aux fichiers de configuration XML, tandis que l'instance de serveur est en cours d'exécution, seront remplacées par le modèle de serveur, et des commentaires XML ajoutés disparaîtront ainsi. Seules les interfaces de gestion doivent être utilisées pour modifier les fichiers de configuration pendant l'exécution d'une instance de serveur

Pour gérer les serveurs par une interface utilisateur graphique d'un navigateur web, utiliser la Console de gestion.

Pour gérer les serveurs par l'interface de ligne de commande, utiliser le Management CLI.

Report a bug

## 3.4. LA CONSOLE DE GESTION

### 3.4.1. Console de management

La Console de management est un outil administratif basé web pour la plateforme JBoss EAP.

Utilisez la Console de management pour démarrer et arrêter des serveurs, déployer et annuler le déploiement des applications, régler les paramètres du système et apporter des modifications persistantes à la configuration du serveur. La Console de management a également la capacité d'effectuer des tâches administratives, avec des notifications directes lorsque les modifications exigent que l'instance du serveur soit redémarrée ou rechargée.

Dans un Domaine géré, les instances de serveur et les groupes de serveurs d'un même domaine peuvent être gérés de façon centralisée à partir de la Console de management du contrôleur de domaine.

Report a bug

### 3.4.2. Connectez-vous à la Console de management

**Pré-requis**

- JBoss Enterprise Application Platform 6 doit être en cours d'exécution.

**Procédure 3.1. Connectez-vous à la Console de management**

1. **Naviguez vers la page de démarrage de la Console de management**
   Naviguez vers la Console de management. L'emplacement par défaut est http://localhost:9990/console/, où le port 9990 est prédéfini comme liaison de socket de Console de management.

2. **Connectez-vous à la Console de management**
   Saisir le nom d'utilisateur et le mot de passe du compte que vous avez déjà créé pour vous connecter à l'écran de connexion de la Console de management.

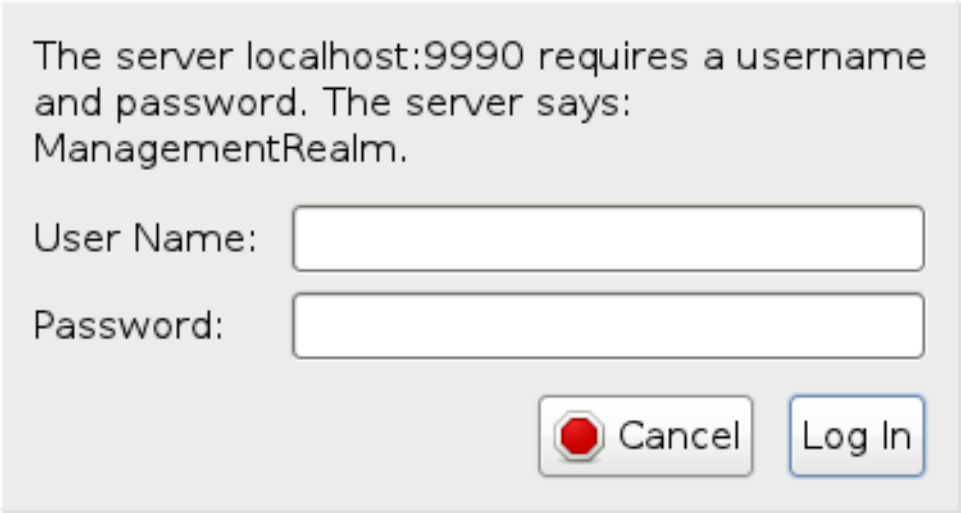



**Figure 3.1. Écran de connexion de la Console de management**

**Résultat**

Une fois connecté, une des pages de la Console de management apparaîtra :

**Domaine géré**

http://localhost:9990/console/App.html#server-instances

**Serveur autonome**

http://localhost:9990/console/App.html#server-overview

Report a bug

### 3.4.3. Changer la Langue de la Console de management

Les paramètres de configuration de la Console de management basée web utilisent l'anglais par défaut. Vous pouvez décider d'utiliser une des langues suivantes à la place.

**Langues prises en charge**

- Allemand (de)
- Chinois simplifié (zn-Hans)
- Portugais brézilien (pt-BR)
- Français (fr)
- Espagnol (es)
- Japonais (ja)

**Procédure 3.2. Changer la Langue de la Console des management basée-web**

1. **Connectez-vous à la Console de management.**
   Connectez-vous à la Console de management basée web.

2. **Ouvrir le dialogue de configuration.**
   Dans le coin gauche de l'écran, il y a une étiquette **`Settings`** de configuration. Cliquer sur cette étiquette pour ouvrir les paramètres de configuration de la Console de management.

3. **Sélectionner la langue désirée.**
   Sélectionner la langue désirée à partir de la case **`Locale`**. Puis sélectionner **`Save`**. Une autre case explicative vous demande de charger à nouveau l'application. Cliquer sur **`Confirm`**. Réactualiser votre navigateur pour pouvoir utiliser les nouveaux paramètres régionaux (Locale).

Report a bug

### 3.4.4. Configurer un Serveur par la Console de management

**Prérequis**

- Section 2.1.3, « Démarrer JBoss Enterprise Application Platform 6 en tant que domaine géré »
- Section 3.4.2, « Connectez-vous à la Console de management »

**Procédure 3.3. Configurer le serveur**

1. **Naviguez dans le panneau `Server Configuration` qui se trouve dans la Console de management**

   a. Sélectionnez l'onglet **`Server`** qui se trouve en haut et à droite de la console.

   b. Déployez l'élément de menu **Server Configurations** qui se trouve à gauche de la console et sélectionnez le serveur qui convient dans la liste.

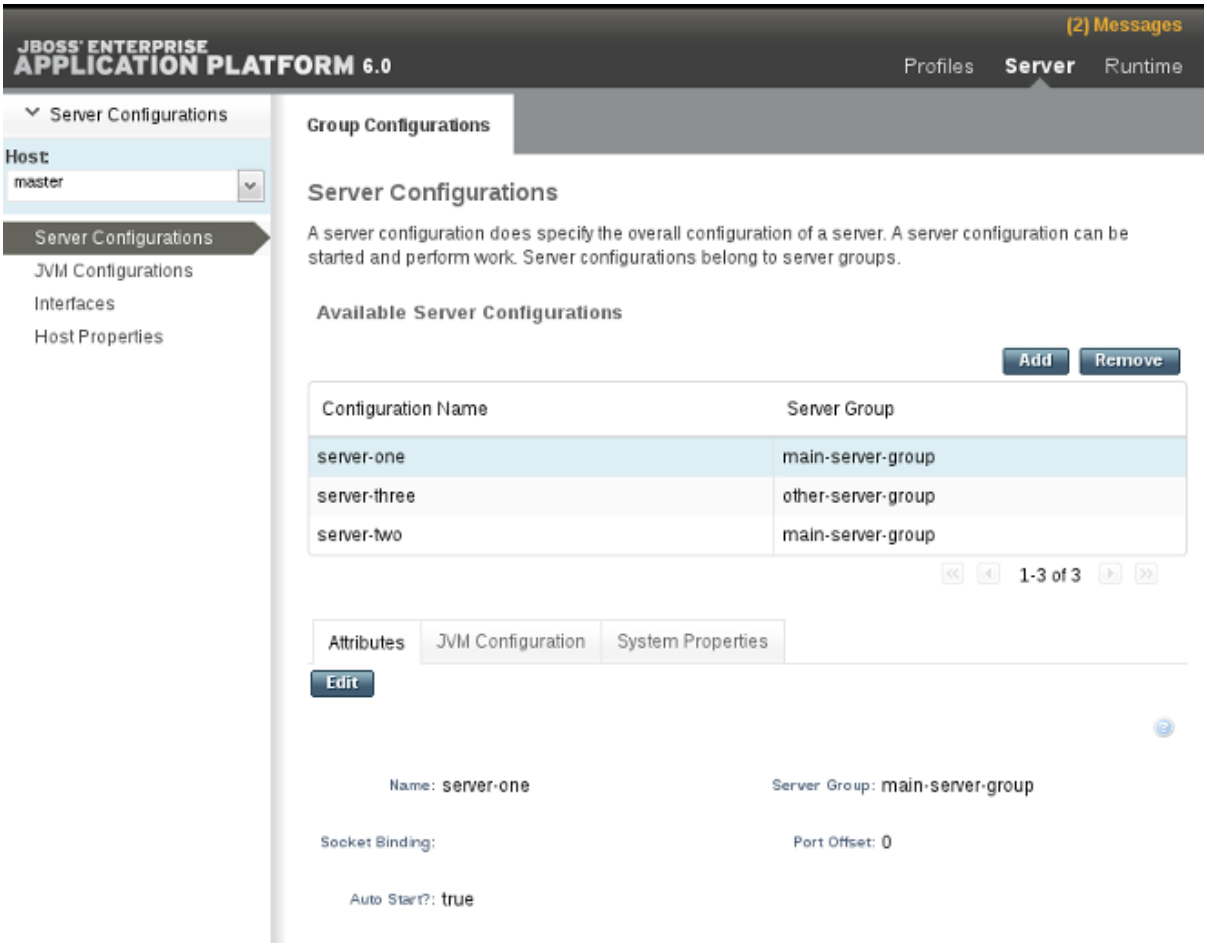



**Figure 3.2. Configuration du serveur**

2. **Modifier la configuration du serveur**

   a. Sélectionner le bouton **`Edit`** qui si trouve en dessous de la liste de serveurs.

   b. Procédez avec les changements des attributs de configuration.

   c. Sélectionnez le bouton **`Save`** qui se trouve en dessous de la liste du serveurs.

**Résultat**

La configuration du serveur a changé, et prendra effet la prochaine fois que le serveur démarre.

Report a bug

### 3.4.5. Ajouter un déploiement dans une Console de management

**Prérequis**

- Section 3.4.2, « Connectez-vous à la Console de management »

**Procédure 3.4. Ajouter et vérifier un déploiement**

1. **Naviguer dans le panneau `Manage Deployments` de la Console de management.**

   a. Sélectionner l'onglet **`Runtime`** en haut et à droite de la console.

   b. Pour un serveur autonome, il vous faudra étendre l'élément de menu **Server** qui se trouve à

gauche de la console et sélectionner **Manage Deployments**. Pour un domaine géré, étendre l'élément de menu **Domain** qui se trouve à gauche de la console, et sélectionner **Manage Deployments**.

Le panneau **`Manage Deployments`** apparaît.

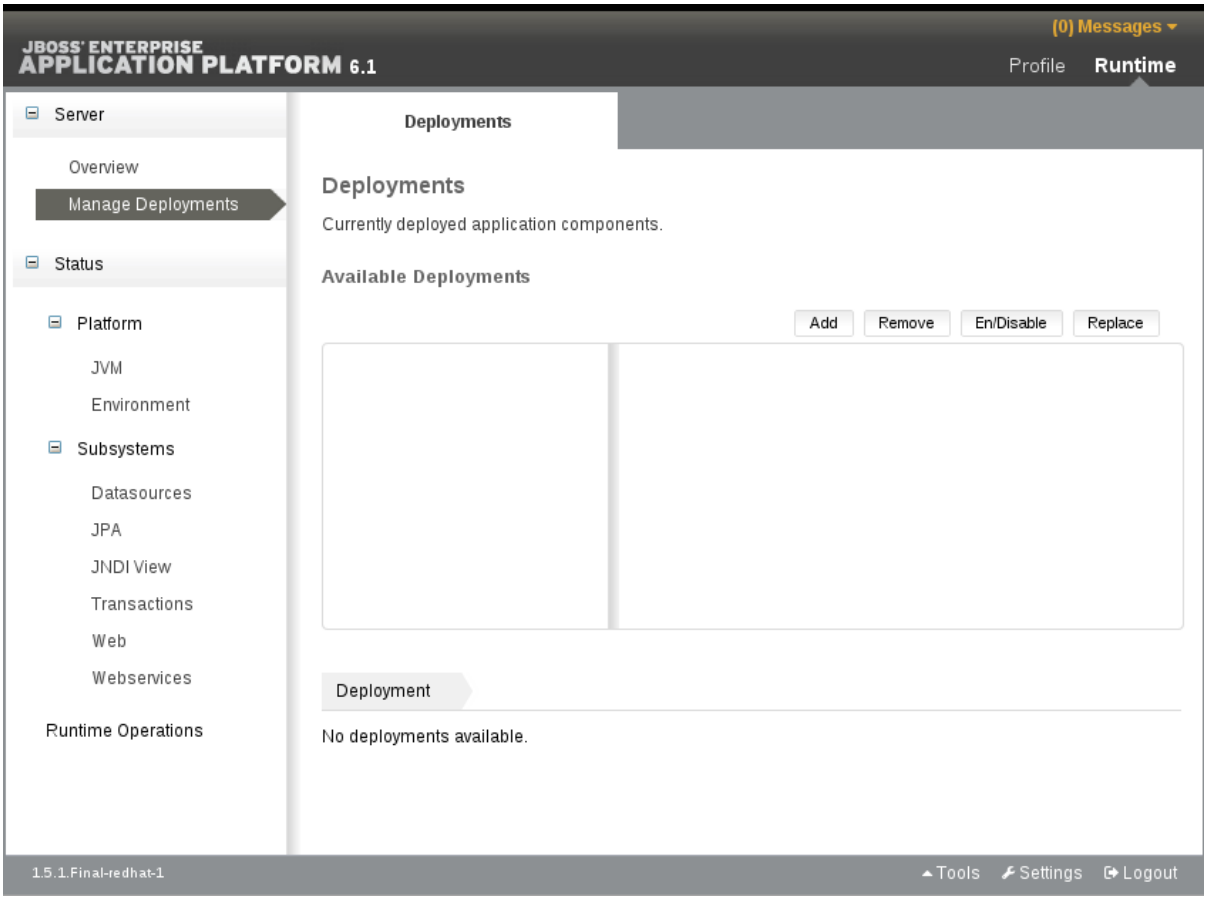



**Figure 3.3. Gestion des déploiements de domaines**

2. **Ajouter le contenu du déploiement.**
   Sélectionner le bouton **Add** en haut et à droite du panneau **`Deployments`**. Une boîte de dialogue **`Upload`** apparaîtra.

3. **Choisir un fichier à déployer**
   Dans la boîte de dialogue, sélectionner le bouton **`Choose File`**. Cherchez le fichier que vous souhaitez déployer, et sélectionnez-le en vue de son chargement. Sélectionner le bouton **`Next`** pour continuer une fois que le fichier a été sélectionné.

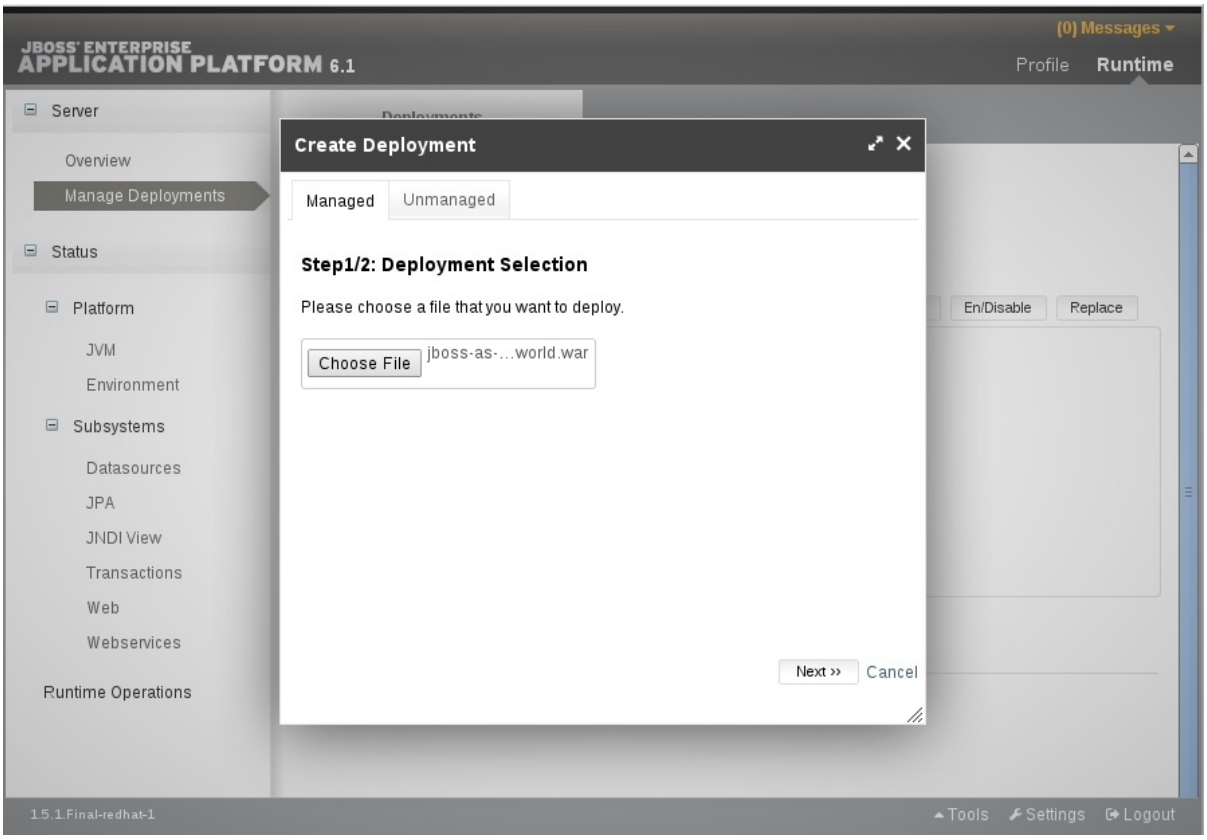



**Figure 3.4. Sélection du déploiement**

4. **Vérifier les noms de déploiement**
   Vérifier le nom du déploiement et le nom du runtime qui apparaît dans la boîte de dialogue **`Upload`**. Sélectionner le bouton **Save** pour charger le fichier une fois que les noms auront été vérifiés.

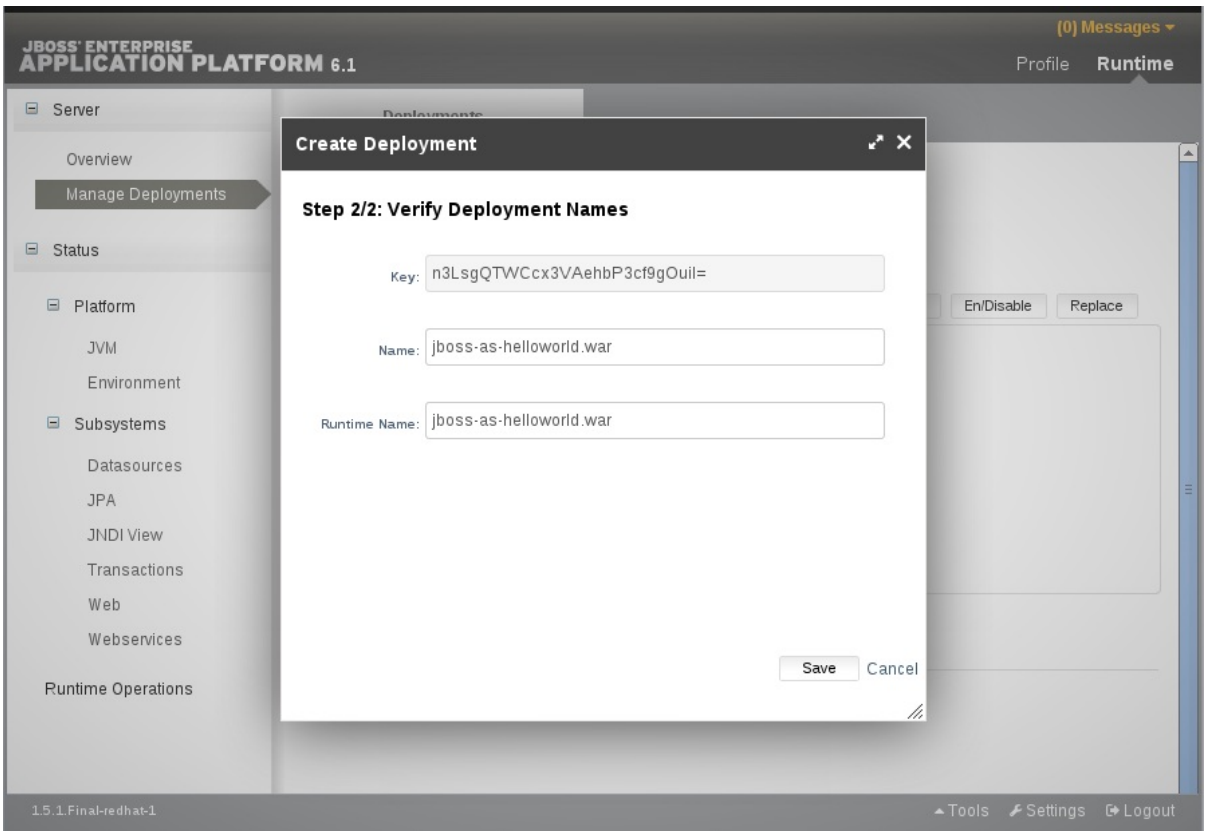



**Figure 3.5. Vérifier les noms de déploiement**

### Résultat

Le contenu sélectionné est téléchargé dans le serveur et est maintenant prêt à être déployé.

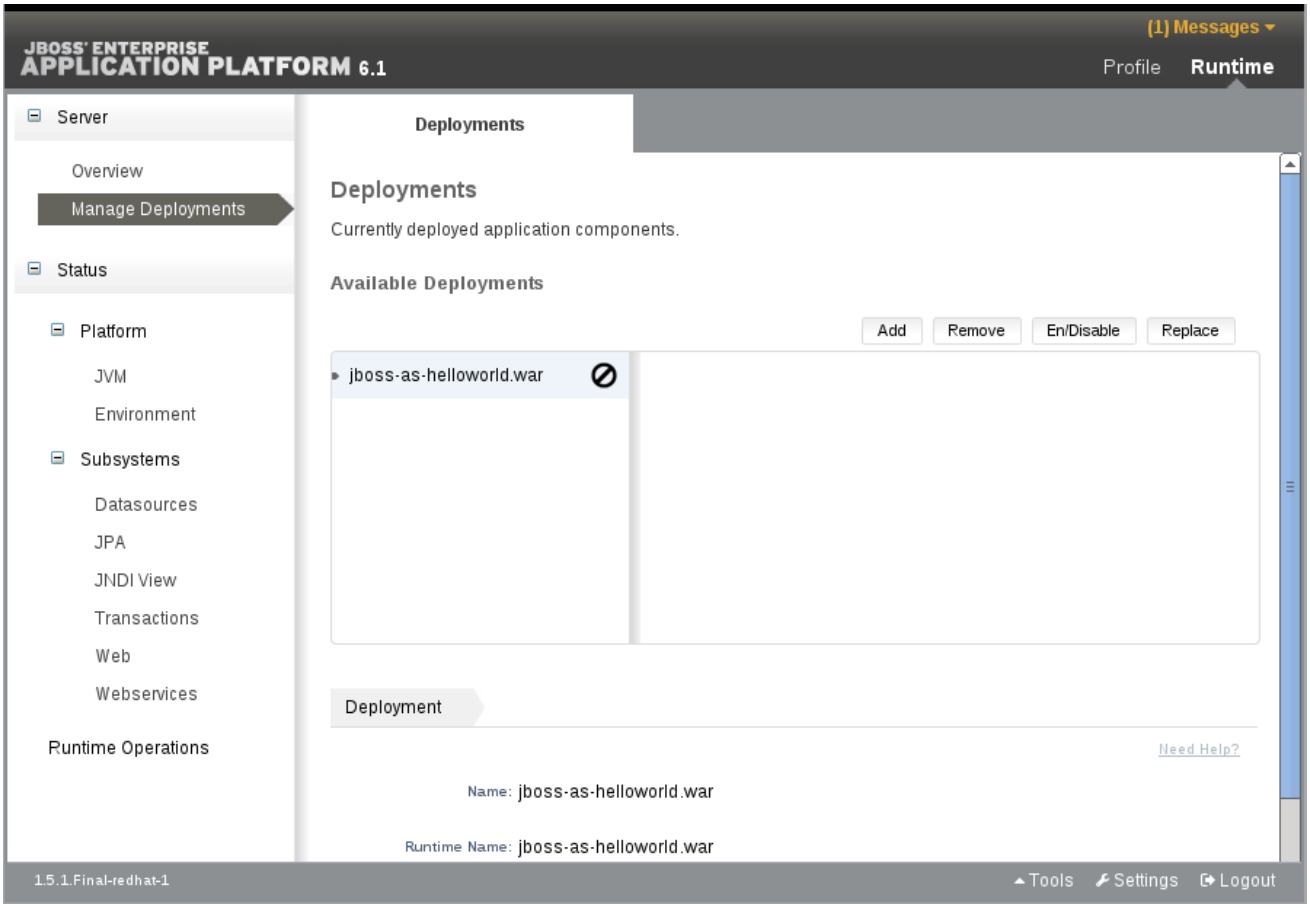



**Figure 3.6. Déploiement téléchargé dans une instance de serveur autonome**

Report a bug

## 3.4.6. Créer un nouveau serveur dans la Console de management

### Prérequis

- Section 2.1.3, « Démarrer JBoss Enterprise Application Platform 6 en tant que domaine géré »
- Section 3.4.2, « Connectez-vous à la Console de management »

### Procédure 3.5. Créer une nouvelle configuration de serveur

1. **Naviguez sur la page `Server Configuration` qui se trouve dans la Console de management**
   Sélectionnez l'onglet **`Server`** qui se trouve en haut et à droite de la console.

2. **Créer une nouvelle configuration**

   a. Sélectionner le bouton **Add** qui se trouve en haut du panneau **`Server Configuration`**.

   b. Modifier les paramètres de base du serveur dans la boîte de dialogue **`Create Server Configuration`**.

   c. Cliquez sur le bouton **Save** pour enregistrer la nouvelle configuration de votre serveur.

**Résultat**

Le nouveau serveur créé est listé dans **`Server Configurations`**.

Report a bug

## 3.4.7. Modifier les Niveaux de journalisation par défaut en utilisant la Console de management

**Procédure 3.6. Modifier les niveaux de journalisation**

1. **Naviguez dans le panneau `Logging` de la Console de gestion.**

   a. Si vous travaillez dans un domaine géré, sélectionner l'onglet **`Profiles`** en haut et à gauche de la console, puis, sélectionner le profil qui convient dans la liste déroulante à gauche de la console.

   b. Pour une domaine géré ou un serveur autonome, sélectionner **Core** → **Logging** à partir du menu qui se trouve à gauche de la console.

   c. Cliquer sur l'onglet **`Log Categories`** en haut de la console.

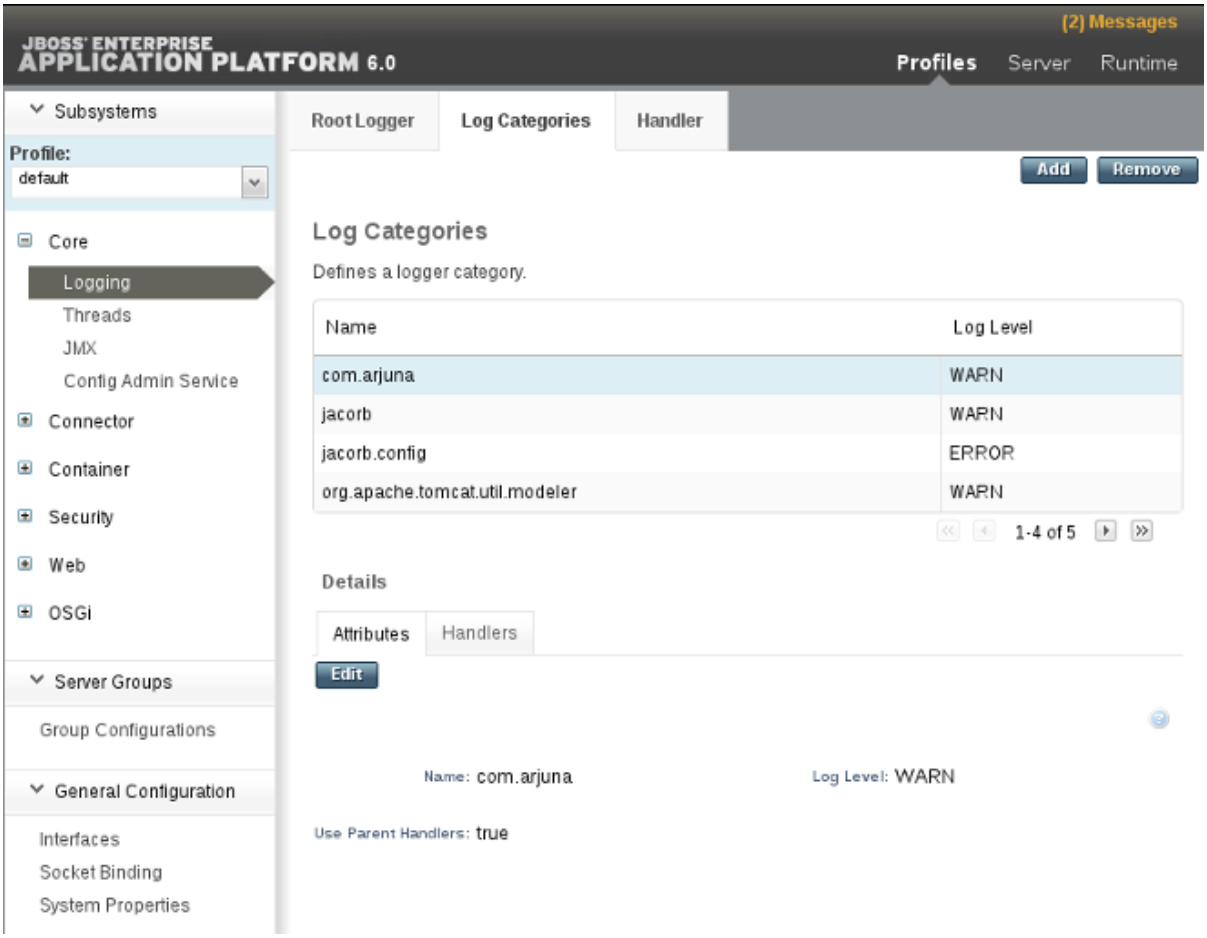



**Figure 3.7. Panneau de Logging**

2. **Modifier les informations du créateur du journal**
   Modifer les informations des entrées du tableau **`Log Categories`**.

   a. Sélectionner une entrée dans le tableau **`Log Categories`**, puis sélectionner le bouton **`Edit`** dans la section **`Details`** ci-dessous.

b. Définir le niveau de journalisation de la catégorie dans la zône déroulante **Log Level**. Sélectionner le bouton **Save** quand c'est fait.

**Résultat**

Les niveaux de journalisation des catégories qui conviennent sont maintenant mis à jour.

Report a bug

## 3.4.8. Créer un nouveau groupe de service dans la Console de management

**Prérequis**

- Section 3.4.2, « Connectez-vous à la Console de management »

**Procédure 3.7. Configurer et Ajouter un groupe de serveurs**

1. **Se rendre dans la vue Server Groups**
   Sélectionner l'onglet **Hosts** en haut et à droite.

2. Sélectionner l'onglet **Group Configurations** dans le menu **Server Groups** dans la colonne de gauche.

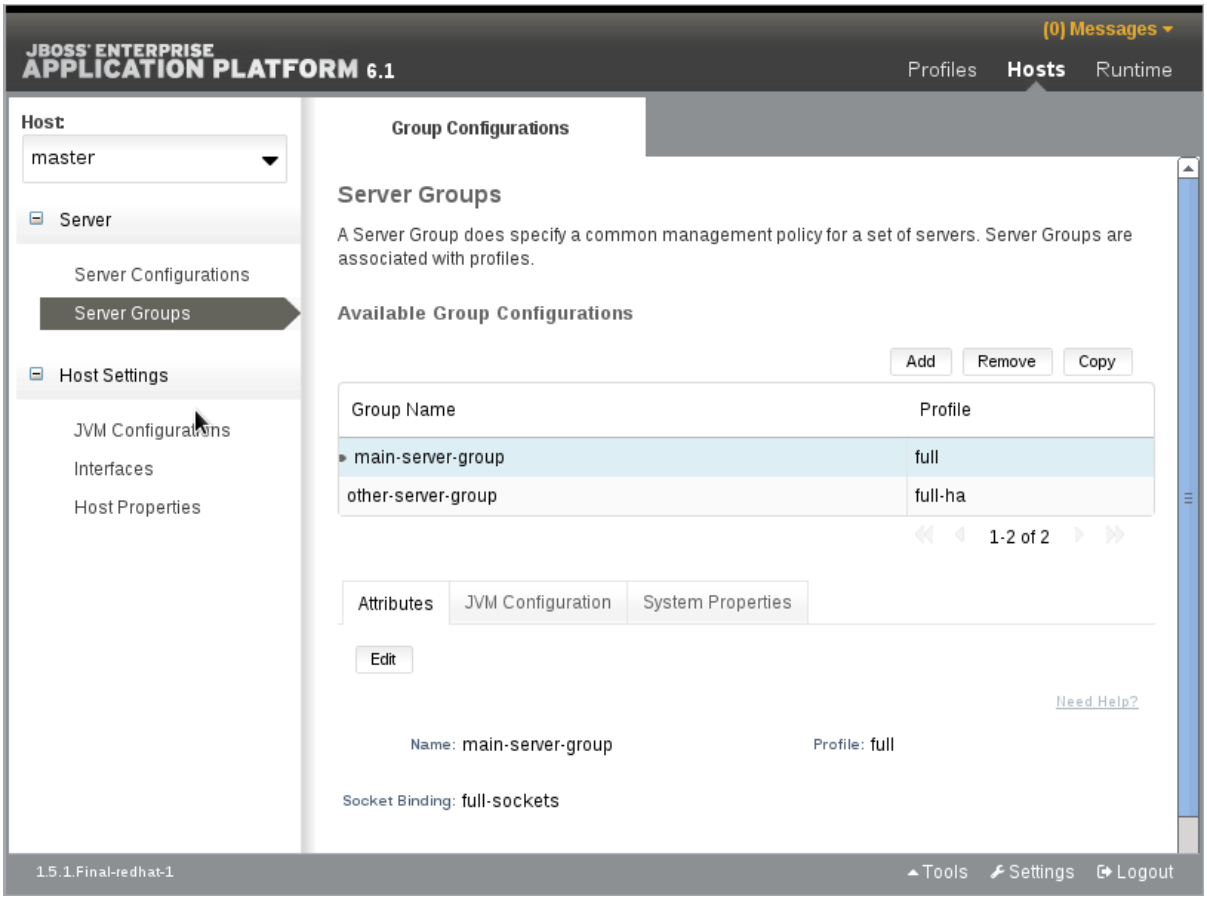



**Figure 3.8. La vue Server Groups**

3. **Ajouter un groupe de serveurs**
   Cliquer sur le bouton **Add** pour ajouter un nouveau groupe de serveurs.

4. **Configurer le groupe de serveurs**

a. Saisir un nom pour le groupe de serveurs

b. Sélectionner le profil dans lequel vous souhaitez ajouter le groupe de serveurs.

c. Sélectionner la liaison de socket dans laquelle vous souhaitez ajouter le groupe de serveurs.

d. Sélectionner le bouton **Save** pour sauvegarder le nouveau groupe.

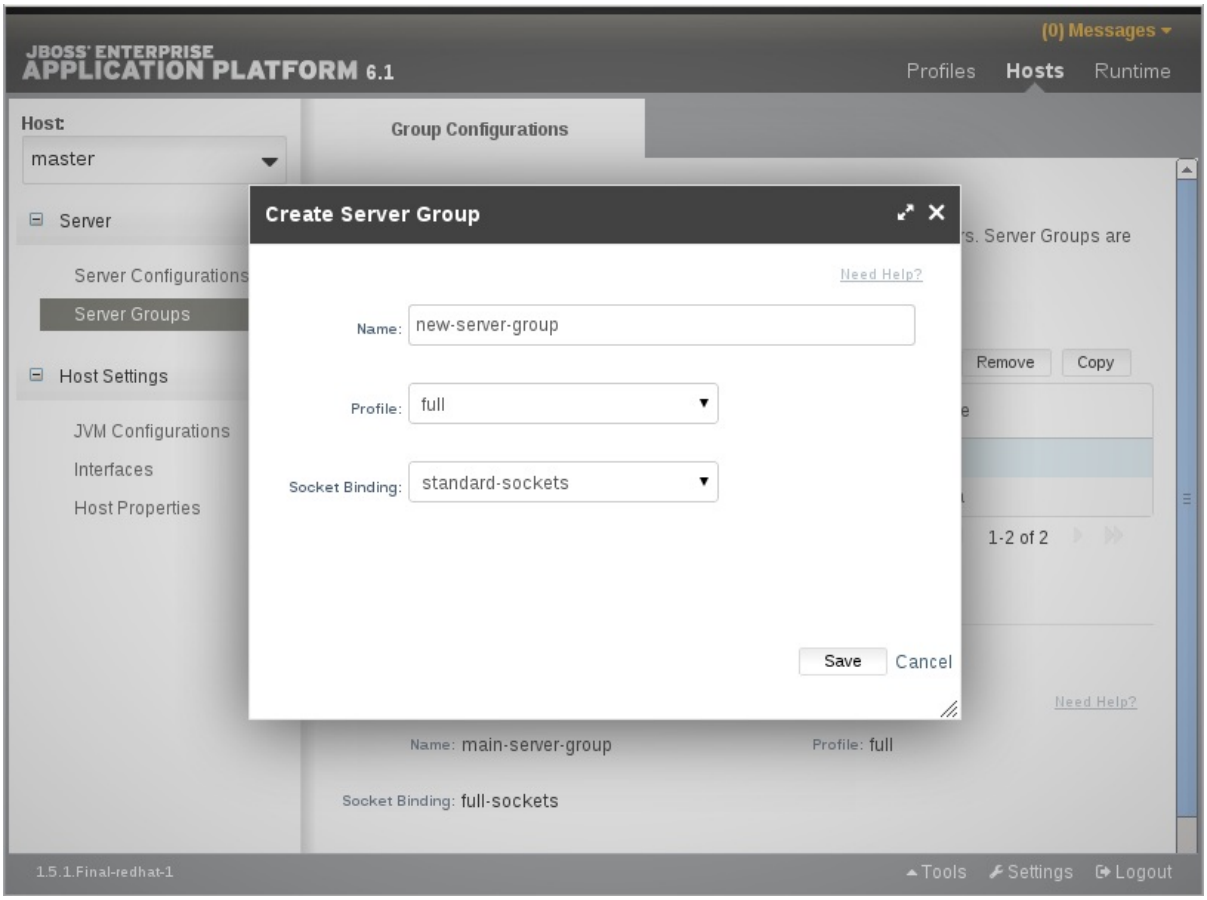



**Figure 3.9. Le dialogue `Create Server Group`**

**Résultat**

Le nouveau groupe de serveurs est visible dans la Console de management.

Report a bug

## 3.5. LE MANAGEMENT CLI

### 3.5.1. Gestion par interface en ligne de commande (CLI)

La Gestion par interface en ligne de commande (CLI) est un outil d'administration en ligne de commande pour JBoss Enterprise Application Platform 6.

Utiliser le Management CLI pour démarrer et stopper les serveurs, déployer et retirer les déploiements d'applications, configurer les paramètres du système, ou encore, effectuer d'autre tâches administratives. Les opérations peuvent être effectuées par mode de lots, ce qui permet à plusieurs tâches d'être exécutées en groupe.

Report a bug

### 3.5.2. Lancement du Management CLI

**Prérequis**

- Procéder à l'une des étapes suivantes avant d'accéder au Management CLI :
  - Section 2.1.2, « Démarrez JBoss EAP 6 comme un serveur autonome »
  - Section 2.1.3, « Démarrer JBoss Enterprise Application Platform 6 en tant que domaine géré »

**Procédure 3.8. Launch CLI in Linux or Windows**

- **Lancement du CLI dans Linux**
  Exécutez le fichier ***EAP_HOME*/bin/jboss-cli.sh** en saisissant ce qui suit dans la ligne de commande :

  ```
  $ EAP_HOME/bin/jboss-cli.sh
  ```

- **Lancement du CLI dans Windows**
  Exécutez le fichier ***EAP_HOME*\bin\jboss-cli.bat** en cliquant deux fois, ou en saisissant ce qui suit dans la ligne de commande :

  ```
  C:\>EAP_HOME\bin\jboss-cli.bat
  ```

Report a bug

### 3.5.3. Quitter le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.9. Quitter le Management CLI**

- **Exécutez la commande `quit`**
  Avec le Management CLI, saisir la commande **`quit`** :

  ```
  [domain@localhost:9999 /] quit
  Closed connection to localhost:9999
  ```

Report a bug

### 3.5.4. Se connecter à une instance de serveur géré par le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.10. Se connecter à une instance de serveur gérée**

- **Exécutez la commande `connect`**

Avec Management CLI, saisir la commande **connect** :

```
[disconnected /] connect
Connected to domain controller at localhost:9999
```

- Sinon, pour vous connecter à un serveur géré quand on démarre le Management CLI sur le système Linux, utiliser le paramètre ***--connect*** :

```
$ EAP_HOME/bin/jboss-cli.sh --connect
```

- Le paramètre ***--connect*** peut être utilisé pour indiquer l'hôte et le port du serveur. Pour connecter l'adresse **192.168.0.1** à la valeur du port **9999** ce qui suit s'applique :

```
$ EAP_HOME/bin/jboss-cli.sh --connect --
controller=192.168.0.1:9999
```

Report a bug

### 3.5.5. Comment obtenir de l'aide avec le Management CLI

**Résumé**

L'interface de gestion CLI dispose d'une boîte de dialogue d'assistance avec des options générales et des options sensibles au contexte. Les commandes d'assistance dépendant du contexte de l'opération nécessitent une connexion à un contrôleur de domaine ou à un serveur autonome. Ces commandes ne seront pas affichées dans la liste, sauf si la connexion a été établie.

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.11. General and Context-Sensitive Help**

1. **Exécuter la commande `help`**
   Avec le Management CLI, saisir la commande **help** :

```
[standalone@localhost:9999 /] help
```

2. **Obtenez de l'aide au niveau sensibilité du contexte**
   Avec le Management CLI, saisir la commande étendue **help -commands** :

```
[standalone@localhost:9999 /] help --commands
```

3. Pour plus d'informations sur une commande particulière, exécuter la commande **help** avec comme argument **'--help'**.

```
[standalone@localhost:9999 /]  deploy --help
```

**Résultat**

L'information d'assistance du CLI s'affichera.

Report a bug

### 3.5.6. Utiliser le Management CLI en Mode par lot

**Résumé**

Le traitement par lot permet à un certain nombre de requêtes d'être groupées par séquences et exécutées ensemble par unité. Si une des demandes opérationnelles d'une séquence échoue, tout le groupe d'opérations sera annulé.

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI »

**Procédure 3.12. Commandes et opérations en mode par lot**

1. **Saisir un mode par lot**
   Saisir le mode par lot par la commande **`batch`**.

   ```
   [standalone@localhost:9999 /] batch
   [standalone@localhost:9999 / #]
   ```

   Le mode par lot est indiqué par le signe (#) dans l'invite.

2. **Ajouter les demandes opérationnelles au lot**
   Une fois que vous serez en mode par lot, saisir les demandes opérationnelles comme d'habitude. Les demandes opérationnelles sont ajoutées au lot dans l'ordre de saisie.

   Voir Section 3.5.8, « Utiliser les opérations et les commandes du Management CLI » pour obtenir des informations sur la façon de formater les demandes opérationnelles.

3. **Exécuter le lot**
   Une fois que toute la séquence de demandes opérationnelles est saisie, exécuter le lot avec la commande **`run-batch`**.

   ```
   [standalone@localhost:9999 / #] run-batch
   The batch executed successfully.
   ```

   Voir Section 3.5.7, « Commandes CLI Mode Lot » pour obtenir une liste complète des commandes disponibles pour pouvoir travailler avec des lots.

4. **Les commandes de lots sont stockées dans des fichiers externes**
   Les commandes fréquemment exécutées peuvent être stockées dans un fichier texte externe et être chargées en passant le chemin d'accès complet au fichier comme argument à la commande **`batch`** ou exécutées directement en étant un argument à la commande **`run-batch`**.

   Vous pouvez créer un fichier de commande lot avec l'éditeur de texte. Chaque commande doit figurer sur une ligne par elle-même et la CLI doit être en mesure d'y accéder.

   La commande suivante chargera un fichier **`myscript.txt`** en mode lot. Toutes les commandes de ce fichier seront alors être prêtes à la modification ou à la suppression. De nouvelles commandes pourront être ajoutées. Les modifications effectuées au cours de cette session de lot ne persistera pas dans le fichier **`myscript.txt`**.

```
[standalone@localhost:9999 /] batch --file=myscript.txt
```

Ce qui suit exécutera instantanément les commandes lot stockées dans le fichier **myscript.txt**

```
[standalone@localhost:9999 /] run-batch --file=myscript.txt
```

**Résultat**

La séquence de demandes opérationnelles saisies est effectuée sous forme de lot.

Report a bug

### 3.5.7. Commandes CLI Mode Lot

Ce tableau vous fournit une liste de commandes lot valides pouvant être utilisées dans JBoss Enterprise Application Platform 6 CLI. Ces commandes ne peuvent être utilisées que pour travailler en lots.

**Tableau 3.2. Commandes CLI Mode Lot**

| Command Name | Description |
|---|---|
| list-batch | Liste des commandes et des opérations du lot en cours. |
| edit-batch-line line-number edited-command | Modifier une ligne du lot en cours en donnant un numéro de ligne à modifier et la commande éditée. Exemple: **edit-batch-line 2 data-source disable --name=ExampleDS**. |
| move-batch-line fromline toline | Réorganiser les lignes dans le lot en spécifiant le numéro de ligne à déplacer comme premier argument et sa nouvelle position comme deuxième argument. Exemple: **move-batch-line 3 1**. |
| remove-batch-line linenumber | Supprimer la commande de lot à la ligne indiquée. Exemple: **remove-batch-line 3**. |
| holdback-batch [batchname] | Vous pouvez reporter à plus tard ou stocker un lot en cours à l'aide de cette commande. Utiliser cette option si vous voulez soudainement exécuter quelque chose dans la CLI en dehors du lot. Pour revenir à ce lot en attente, tapez simplement **batch** à nouveau à la ligne de commande CLI.<br><br>Si vous fournissez un nom de lot en utilisant la commande **holdback-batch**, le lot sera stocké sous ce nom. Pour retourner au lot nommé, utilisez la commande **batch batchname**. L'appel de la commande **batch** sans un nom de lot va commencer un nouveau lot (sans nom). Il peut y avoir qu'un seul lot suspendu sans nom.<br><br>Pour voir une liste de tous les lots suspendus, utiliser la commande **batch -l**. |

| Command Name | Description |
|---|---|
| discard-batch | Rejète le lot actif en cours. |

Report a bug

### 3.5.8. Utiliser les opérations et les commandes du Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI »

**Procédure 3.13. Créer, configurer et exécuter les requêtes**

1. **Construire la demande opérationnelle**
   Les demandes opérationnelles facilitent une interaction de bas niveau dans le modèle de gestion. Il s'agit d'une façon contrôlée de modifier les configurations du serveur. Une demande opérationnelle se présente en trois parties :
   - une *adresse*, avec une barre oblique devant (`/`).
   - un *nom d'opération*, avec deux points (`:`).
   - un groupe optionnel de *paramètres*, entre parenthèses (`()`).

   a. **Déterminer l'adresse**
   La configuration est présentée sous forme de ressources auxquelles s'adresser et de façon hiérarchique. Chaque noeud de ressources procure un groupe différent d'opérations. Une adresse utilise la syntaxe suivante :

   ```
   /node-type=node-name
   ```

   - *node-type* correspond au type de noeud de ressource. Cela correspond à un nom d'élément dans la configuration XML.
   - *node-name* correspond au nom du nœud. Cela correspond à l'attribut `nom` de l'élément dans la configuration XML.
   - Séparer chaque niveau de l'arborescence de ressources par une barre oblique (`/`).

   Voir les fichiers de configuration XML pour déterminer l'adresse qui convient. Le fichier ***EAP_HOME*/standalone/configuration/standalone.xml** contient la configuration d'un serveur autonome et les fichiers ***EAP_HOME*/domain/configuration/domain.xml** ***EAP_HOME*/domain/configuration/host.xml** contiennent la configuration d'un domaine géré.

   **Exemple 3.3. Exemple d'adresses d'opérations**

   Pour procéder à une opération dans le sous-système de journalisation, utiliser l'adresse suivante dans la demande opérationnelle :

```
/subsystem=logging
```

Pour effectuer une opération sur la source de données Java, utiliser l'adresse suivante dans la demande opérationnelle :

```
/subsystem=datasources/data-source=java
```

b. **Déterminer l'opération**
Les opérations diffèrent avec chaque type de nœud de ressource. Une opération utilise la syntaxe suivante :

```
:operation-name
```

- *operation-name* correspond au nom de l'opération à demander.

Utiliser l'opération **`read-operation-names`** sur une adresse de ressources d'un serveur autonome pour lister les opérations disponibles.

**Exemple 3.4. Opérations disponibles**

Pour énumérer toutes les opérations disponibles du sous-système de journalisation, saisir la requête suivante dans une serveur autonome :

```
[standalone@localhost:9999 /] /subsystem=logging:read-
operation-names
{
    "outcome" => "success",
    "result" => [
        "add",
        "read-attribute",
        "read-children-names",
        "read-children-resources",
        "read-children-types",
        "read-operation-description",
        "read-operation-names",
        "read-resource",
        "read-resource-description",
        "remove",
        "undefine-attribute",
        "whoami",
        "write-attribute"
    ]
}
```

c. **Déterminer un paramètre**
Chaque opération a sans doute besoin de paramètres différents.

Les paramètres utilisent la syntaxe suivante :

```
(parameter-name=parameter-value)
```

- *parameter-name* correspond au nom du paramètre.
- *parameter-value* correspond à la valeur du paramètre.
- Les différents paramètres sont séparés par des virgules (`,`).

Afin de déterminer les paramètres qui conviennent, exécutez la commande **`read-operation-description`** sur un nœud de ressource, en faisant passer le nom de l'opération en tant que paramètre. Voir Exemple 3.5, « Déterminer les paramètres des opérations » pour plus de détails.

**Exemple 3.5. Déterminer les paramètres des opérations**

Afin de déterminer les paramètres qui conviennent pour l'opération **`read-children-types`** sur le sous-système de journalisation, saisir la commande **`read-operation-description`** comme suit :

```
[standalone@localhost:9999 /] /subsystem=logging:read-
operation-description(name=read-children-types)
{
    "outcome" => "success",
    "result" => {
        "operation-name" => "read-children-types",
        "description" => "Gets the type names of all the
children under the selected resource",
        "reply-properties" => {
            "type" => LIST,
            "description" => "The children types",
            "value-type" => STRING
        },
        "read-only" => true
    }
}
```

2. **Saisir toute la demande opérationnelle**
   Une fois que l'adresse, l'opération et tous les paramètres auront été sélectionnés, saisir la demande opérationnelle complète.

**Exemple 3.6. Exemple de demande opérationnelle**

```
[standalone@localhost:9999 /] /subsystem=web/connector=http:read-
resource(recursive=true)
```

**Résultat**

L'interface de gestion effectue la demande opérationnelle sur la configuration du serveur.

Report a bug

### 3.5.9. Références de Commandes de Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Résumé**

La section Section 3.5.5, « Comment obtenir de l'aide avec le Management CLI » décrit comment accéder aux fonctionnalités d'assistance du Management CLI. L'interface de gestion dispose d'une boîte de dialogue d'assistance avec des options générales et des options sensibles au contexte. Les commandes d'assistance dépendant du contexte de l'opération nécessitent une connexion à un contrôleur de domaine ou à un serveur autonome. Ces commandes ne seront pas affichées dans la liste, sauf si la connexion a été établie.

**Tableau 3.3.**

| Commande | Description |
|---|---|
| `batch` | Démarrer le mode par lot en créant un nouveau lot ou, selon les lots existants retenus, réactiver l'un d'entre eux. Si il n'y a pas de lot existant retenu, cette commande sans argument va commencer un nouveau lot .S'il y a un lot sans nom existant retenu, cette commande le réactivera. S'il y a des lots existants retenus, mais avec nom, il peuvent être activées en exécutant cette commande avec ce nom comme argument. |
| `cd` | Change le chemin du nœud en cours à l'argument. Le chemin du nœud en cours est utilisé comme adresse pour les requêtes opérationnelles qui ne contiennent pas la partie adresse. Si une opération n'inclut pas d'adresse, l'adresse incluse sera considérée comme relative au chemin du nœud en cours. Le chemin du nœud en cours peut finir en type de nœud. Dans ce cas, exécuter une opération en spécifiant un nom de nœud est suffisant, comme logging:read-resource. |
| `clear` | Efface l'écran |
| `command` | Vous permet d'ajouter, de supprimer ou de lister des commandes existantes de type standard. Une commande de type standard est une commande qui est assignée à un type de nœud spécifique et qui vous permet d'effectuer une opération disponible à une instance de ce type. Elle vous permet de modifier n'importe quelle propriété exposée par le type de n'importe quelle instance existante. |
| `connect` | Connecte le contrôleur à l'hôte et au port spécifiés. |
| `connection-factory` | Définit une usine de connexions |
| `data-source` | Gère les configurations de sources de données JDBC dans le sous-système de la source de données. |
| `deploy` | Déploie l'application désignée par le chemin d'accès au fichier ou bien, active une application qui est pré-existante, mais désactivée dans le référentiel. Si elle est exécutée sans argument, cette commande énumérera tous les déploiements existants. |

| Commande | Description |
|---|---|
| **help** | Affiche le message d'asistance. Peut être utilisé avec l'argument **--commands** pour fournir aux commandes données des résultats sensibles au contexte. |
| **history** | Affiche l'historique en mémoire de la commande CLI et affiche un statut pour savoir si l'expansion de l'historique est activée ou non. Peut être utilisé avec des arguments pour effacer, désactiver, ou activer l'expansion de l'historique selon les besoins. |
| **jms-queue** | Définit une file d'attente JMS dans le sous-système de messagerie. |
| **jms-topic** | Définit un topic dans le sous-système de messagerie. |
| **ls** | Lister les contenus du chemin d'accès au nœud. Par défaut, le résultat est imprimé dans des colonnes qui utilisent toute la largeur du terminal. Utiliser **-l** affichera les résultats sur la base d'un nom par ligne. |
| **pwd** | Affiche le chemin d'accès du nœud pour le nœud en cours |
| **quit** | Termine l'interface de ligne de commande. |
| **read-attribute** | Affiche la valeur et, suivant les arguments, la description de l'attribut d'une ressource gérée. |
| **read-operation** | Affiche la description d'une opération particulière, ou bien liste toutes les opérations si aucune n'est spécifiée. |
| **undeploy** | Annule une demande lorsque celle-ci est exécutée avec le nom de l'application prévue. Peut être exécuté avec des arguments pour supprimer également l'application du référentiel. Imprime la liste de tous les déploiements existants si exécutée sans application spécifiée. |
| **version** | Affiche la version de serveur d'application et les informations d'environnement. |
| **xa-data-source** | Gére la configuration de la source de données JDBC XA du sous-système de la source de données. |

Report a bug

## 3.5.10. Référence aux Opérations de Management CLI

### Exposer les opérations du Management CLI

Les opérations du Management CLI peuvent être exposées par l'opération **read-operation-names** décrite dans la rubrique Section 3.6.5, « Afficher les Noms de l'opération en utilisant le Management CLI ». Les descriptions des opérations peuvent être exposées par l'opération **read-operation-**

**descriptions** décrite dans la rubrique Section 3.6.4, « Affiche une description d'opération en utilisant le Management CLI ».

**Tableau 3.4. Les opérations de Management CLI**

| Nom de l'opération | Description |
|---|---|
| **add-namespace** | Ajoute un mappage de préfixe d'espace-nom à la mappe d'attribut d'espace-nom. |
| **add-schema-location** | Ajoute un schéma de mappage d'emplacement à la mappe d'attribut schema-locations. |
| **delete-snapshot** | Efface un snapshot de la configuratino serveur dans le répertoire de snapshots. |
| **full-replace-deployment** | Ajoute le contenu précédemment téléchargé déploiement à la liste de contenu disponible, remplace le contenu existant du même nom dans le runtime et supprime le contenu remplacé dans la liste de contenu disponible. Voir lien pour plus de renseignements. |
| **list-snapshots** | Liste les snapshots de la configuration du serveur sauvegardée dans le répertoire des snapshots. |
| **read-attribute** | Affiche la valeur d'un attribut d'une ressource sélectionnée. |
| **read-children-names** | Affiche les nom de tous les enfants d'une ressource donnée ayant le type donné. |
| **read-children-resources** | Affiche des informations sur tous les enfants d'une ressource d'un type donné. |
| **read-children-types** | Affiche les noms de types de tous les enfants pour la ressource sélectionnée. |
| **read-config-as-xml** | Lit la configuration actuelle et l'affiche en format XML. |
| **read-operation-description** | Affiche les détails d'une opération de la ressource donnée. |
| **read-operation-names** | Affiche les noms de toutes les opérations de la ressource donnée. |
| **read-resource** | Affiche les valeurs des attributs d'un modèle de ressource avec des informations complètes ou de base sur n'importe quelle ressource enfant. |
| **read-resource-description** | Indique la description des attributs d'une ressource, les types de dépendants et les opérations. |

| Nom de l'opération | Description |
| --- | --- |
| **reload** | Charge le serveur à nouveau en fermant tous les services et ne redémarrant. |
| **remove-namespace** | Supprime un mappage de préfixe d'espace-nom à la mappe d'attribut d'espace-nom. |
| **remove-schema-location** | Supprime un schéma de mappage d'emplacement à la mappe d'attribut schema-locations. |
| **replace-deployment** | Remplace le contenu existant du runtime par un contenu nouveau. Le nouveau contenu doit avoir été chargé auparavant dans le référentiel du contenu de déploiement. |
| **resolve-expression** | Opération qui accepte une expression comme entrée ou un string pouvant être compris comme une expression, et résolu en fonction du système locale de propriétés et des variables d'environnement. |
| **resolve-internet-address** | Prend un ensemble de critères de résolution d'interface et trouve une adresse IP sur une machine locale qui correspond au critère, ou échoue si aucune adresse IP correspondante n'est trouvée. |
| **server-set-restart-required** | Met le serveur en mode «restart-required» |
| **shutdown** | Ferme le serveur via un appel à **System.exit(0)**. |
| **start-servers** | Démarre tous les serveurs configurés dans un Domaine géré qui n'est pas actuellement en cours d'exécution. |
| **stop-servers** | Arrête tous les serveurs actuellement en cours d'exécution dans un Domaine géré. |
| **take-snapshot** | Prend un snapshot de la configuration du serveur et la sauvegarde dans le répertoire des snapshots. |
| **upload-deployment-bytes** | Indique si le contenu de déploiement du tableau d'octets inclus doit être ajouté au référentiel du contenu de déploiement. Notez que cette opération n'indique pas que le contenu doive être déployé au runtime. |
| **upload-deployment-stream** | Indique si le contenu de déploiement disponible dans l'index des flux entrants doit être ajouté au référentiel du contenu de déploiement. Notez que cette opération n'indique pas que le contenu doive être déployé au runtime. |
| **upload-deployment-url** | Indique si le contenu de déploiement disponible dans l'URL doit être ajouté au référentiel du contenu de déploiement. Notez que cette opération n'indique pas que le contenu doive être déployé au runtime. |

| Nom de l'opération | Description |
|---|---|
| **validate-address** | Valide l'adresse de l'opération |
| **write-attribute** | Indique la valeur d'un attribut d'une ressource sélectionnée. |

Report a bug

## 3.6. OPÉRATIONS DE MANAGEMENT CLI

### 3.6.1. Affiche les attributs d'une ressource par le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Résumé**

L'opération **read-attribute** est une opération globale utilisée pour lire la valeur d'exécution d'un attribut sélectionné. Peut être utilisée pour exposer uniquement les valeurs qui ont été définies par l'utilisateur, en ignorant toute valeur par défaut ou non définie. Les propriétés de la requête incluent les paramètres suivants.

**Propriétés de requêtes**

**name**

Le nom de l'attribut pour obtenir la valeur sous la ressource sélectionnée.

**include-defaults**

Un paramètre booléen qui peut être défini à **false** pour limiter les résultats de l'opération aux attributs qui ont été définis par l'utilisateur uniquement, et ignorer les valeurs par défaut.

**Procédure 3.14. Affiche la valeur de runtime en cours pour un attribut sélectionné.**

- **Exécuter l'opération read-attribute**
  À partir du Management CLI, utiliser l'opération **read-attribute** pour afficher la valeur d'un attribut de ressource. Pour plus d'informations sur les requêtes d'informations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes du Management CLI ».

```
[standalone@localhost:9999 /]:read-attribute(name=name-of-attribute)
```

Un avantage de l'opération **read-attribute** est la possibilité d'exposer la valeur d'exécution actuelle d'un attribut spécifique. Des résultats similaires peuvent être obtenus avec l'opération **read-attribute**, mais seulement avec l'addition de la propriété de requête **include-runtime**, et uniquement dans le cadre d'une liste de toutes les ressources disponibles pour ce nœud. L'opération **read-attribute** est destinée aux requêtes d'attribut précises, comme le montre l'exemple suivant.

**Exemple 3.7. Exécuter l'opération read-attribute pour exposer l'IP d'interface publique.**

Si vous connaissez le nom de l'attribut que vous souhaitez exposer, vous pouvez utiliser la commande **`read-attribute`** pour renvoyer la valeur exacte dans le runtime en cours.

```
[standalone@localhost:9999 /] /interface=public:read-
attribute(name=resolved-address)
{
    "outcome" => "success",
    "result" => "127.0.0.1"
}
```

L'attribut **`resolved-address`** est une valeur de runtime, donc il ne s'affiche pas dans les résultats de l'opération **`read-resource`** standard.

```
[standalone@localhost:9999 /] /interface=public:read-resource
{
    "outcome" => "success",
    "result" => {
        "any" => undefined,
        "any-address" => undefined,
        "any-ipv4-address" => undefined,
        "any-ipv6-address" => undefined,
        "inet-address" => expression "${jboss.bind.address:127.0.0.1}",
        "link-local-address" => undefined,
        "loopback" => undefined,
        "loopback-address" => undefined,
        "multicast" => undefined,
        "name" => "public",
        "nic" => undefined,
        "nic-match" => undefined,
        "not" => undefined,
        "point-to-point" => undefined,
        "public-address" => undefined,
        "site-local-address" => undefined,
        "subnet-match" => undefined,
        "up" => undefined,
        "virtual" => undefined
    }
}
```

Pour afficher **`resolved-address`** ou des autres valeurs de runtime, vous devrez utiliser la propriété de requête **`include-runtime`**.

```
[standalone@localhost:9999 /] /interface=public:read-resource(include-
runtime=true)
{
    "outcome" => "success",
    "result" => {
        "any" => undefined,
        "any-address" => undefined,
        "any-ipv4-address" => undefined,
        "any-ipv6-address" => undefined,
        "inet-address" => expression "${jboss.bind.address:127.0.0.1}",
        "link-local-address" => undefined,
```

```
        "loopback" => undefined,
        "loopback-address" => undefined,
        "multicast" => undefined,
        "name" => "public",
        "nic" => undefined,
        "nic-match" => undefined,
        "not" => undefined,
        "point-to-point" => undefined,
        "public-address" => undefined,
        "resolved-address" => "127.0.0.1",
        "site-local-address" => undefined,
        "subnet-match" => undefined,
        "up" => undefined,
        "virtual" => undefined
    }
}
```

**Résultat**

La valeur de l'attribut du runtime en cours est affichée.

Report a bug

### 3.6.2. Affiche l'utilisateur qui est actif dans le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Résumé**

L'opération **`whoami`** est une opération globale utilisée pour identifier les attributs de l'utilisateur actif. L'opération expose l'identité de nom d'utilisateur et le domaine qui leur est attribué. L'opération **`whoami`** est utile pour les administrateurs qui gèrent plusieurs comptes d'utilisateurs dans plusieurs domaines, ou pour aider à assurer le suivi des utilisateurs actifs à travers les instances de domaine avec plusieurs sessions de terminal et les comptes utilisateurs.

**Procédure 3.15. Affiche l'utilisateur qui est actif dans le Management CLI par l'opération `whoami`**

- **Exécuter l'opération `whoami`**
  À partir du Management CLI, utiliser l'opération **`whoami`** pour afficher le compte utilisateur actif.

  ```
  [standalone@localhost:9999 /] :whoami
  ```

  L'exemple suivant utilise la commande **`whoami`** dans une instance de serveur autonome pour montrer que l'utilisateur actif est le **`username`** , et que l'utilisateur est assigné au domaine **`ManagementRealm`**.

  **Exemple 3.8. Utiliser la commande `whoami` dans une instance autonome**

  ```
  [standalone@localhost:9999 /]:whoami
  {
  ```

```
    "outcome" => "success",
    "result" => {"identity" => {
        "username" => "username",
        "realm" => "ManagementRealm"
    }}
}
```

**Résultat**

Votre compte d'utilisateur actif en cours est affiché

Report a bug

### 3.6.3. Affiche les informations Système et Serveur dans le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.16. Affiche les informations Système et Serveur dans le Management CLI**

- **Exécuter la commande `version`**
  À partir du Management CLI, saisir la commande **`version`** :

```
[domain@localhost:9999 /] version
```

**Résultat**

Les informations sur la version de votre serveur d'applications et sur votre environnement s'afficheront.

Report a bug

### 3.6.4. Affiche une description d'opération en utilisant le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.17. Éxécuter la commande de Management CLI suivante**

- **Exécuter l'opération `read-operation-description`**
  À partir du Management CLI, utiliser **`read-operation-description`** pour afficher des informations sur l'opération. L'opération requiert des paramètres supplémentaires dans le format d'une paire clé-valeur pour indiquer quelle opération afficher. Pour plus d'informations sur les requêtes d'informations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes du Management CLI ».

```
[standalone@localhost:9999 /]:read-operation-description(name=name-
of-operation)
```

**Exemple 3.9. Afficher la description de l'opération «list-snapshots»**

L'exemple suivant vous montre la méthode utilisée pour décrire l'opération **`list-snapshots`**.

```
[standalone@localhost:9999 /] :read-operation-description(name=list-
snapshots)
{
    "outcome" => "success",
    "result" => {
        "operation-name" => "list-snapshots",
        "description" => "Lists the snapshots",
        "reply-properties" => {
            "type" => OBJECT,
            "value-type" => {
                "directory" => {
                    "type" => STRING,
                    "description" => "The directory where the snapshots
are stored"
                },
                "names" => {
                    "type" => LIST,
                    "value-type" => STRING,
                    "description" => "The names of the snapshots within
the snapshots directory"
                }
            }
        },
        "read-only" => false
    }
}
```

**Résultat**

La description est affichée pour l'opération choisie.

Report a bug

### 3.6.5. Afficher les Noms de l'opération en utilisant le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.18. Éxécuter la commande de Management CLI suivante**

- **Exécuter l'opération `read-operation-names`**
  À partir du Management CLI, utiliser l'opération **`read-operation-names`** pour afficher les noms des opérations disponibles. Pour plus d'informations sur les requêtes d'informations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes du Management CLI ».

  ```
  [standalone@localhost:9999 /]:read-operation-names
  ```

**Exemple 3.10. Afficher les noms de l'opération en utilisant le Management CLI**

L'exemple suivant vous montre la méthode utilisée pour décrire l'opération **`read-operation-names`**.

```
[standalone@localhost:9999 /]:read-operation-names
{
    "outcome" => "success",
    "result" => [
        "add-namespace",
        "add-schema-location",
        "delete-snapshot",
        "full-replace-deployment",
        "list-snapshots",
        "read-attribute",
        "read-children-names",
        "read-children-resources",
        "read-children-types",
        "read-config-as-xml",
        "read-operation-description",
        "read-operation-names",
        "read-resource",
        "read-resource-description",
        "reload",
        "remove-namespace",
        "remove-schema-location",
        "replace-deployment",
        "shutdown",
        "take-snapshot",
        "upload-deployment-bytes",
        "upload-deployment-stream",
        "upload-deployment-url",
        "validate-address",
        "write-attribute"
    ]
}
```

**Résultat**

Les noms d'opérations disponibles sont affichés.

Report a bug

### 3.6.6. Afficher les ressources disponibles en utilisant le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Résumé**

L'opération **`read-resource`** est une opération globale utilisée pour lire la valeur des ressources. Peut être utilisée pour exposer des informations complètes ou de base sur les ressources des nœuds en cours ou des nœuds enfants, ainsi qu'un ensemble de propriétés de requêtes qui peuvent étendre ou limiter l'étendue des résultats de l'opération. Les propriétés de la requête incluent les paramètres suivants.

**Propriétés de requêtes**

**`recursive`**

Pour savoir si on doit inclure récursivement des informations complètes sur les ressources enfant.

**`recursive-depth`**

La précision des informations de ressources enfant incluses

**`proxies`**

Si on doit inclure des ressources éloignées pour une recherche récursive. Par exemple, si on doit inclure les ressources niveau hôte à partir des Contrôleurs Hôtes esclave pour une demande de Contrôleur de domaines.

**`include-runtime`**

Si on doit inclure des attributs de runtime dans la réponse, comme des valeurs d'attributs qui ne proviennent pas de la configuration persistante. Cette propriété de requête est définie à false par défaut.

**`include-defaults`**

Une propriété de demande booléenne qui sert à activer ou à désactiver la lecture des attributs par défaut. Si définie sur false, seuls les attributs définis par l'utilisateur seront renvoyés, ignorant ainsi ceux qui sont non définis.

**Procédure 3.19. Éxécuter la commande de Management CLI suivante**

1. **Exécuter l'opération `read-resource`**
   Avec le Management CLI, faîtes l'opération **`read-resource`** pour afficher les ressources disponibles.

   ```
   [standalone@localhost:9999 /]:read-resource
   ```

   L'exemple suivant vous montre comment il est possible d'utiliser l'opération **`read-resource`** dans une instance de serveur autonome pour exposer les informations de ressources générales. Les résultats ressemblent au fichier de configuration **`standalone.xml`**, qui affiche les ressources de système, les extensions, les interfaces et les sous-systèmes installés ou configurés pour l'instance du serveur. Ces résultats peuvent être interrogés directement.

   **Exemple 3.11. Exécuter l'opération `read-resource` niveau racine**

   ```
   [standalone@localhost:9999 /]:read-resource
   {
       "outcome" => "success",
       "result" => {
           "deployment" => undefined,
           "deployment-overlay" => undefined,
           "management-major-version" => 1,
           "management-micro-version" => 0,
           "management-minor-version" => 4,
           "name" => "longgrass",
           "namespaces" => [],
           "product-name" => "EAP",
   ```

```
        "product-version" => "6.1.0.GA",
        "release-codename" => "Janus",
        "release-version" => "7.2.0.Final-redhat-3",
        "schema-locations" => [],
        "system-property" => undefined,
        "core-service" => {
            "management" => undefined,
            "service-container" => undefined,
            "server-environment" => undefined,
            "platform-mbean" => undefined
        },
        "extension" => {
            "org.jboss.as.clustering.infinispan" => undefined,
            "org.jboss.as.connector" => undefined,
            "org.jboss.as.deployment-scanner" => undefined,
            "org.jboss.as.ee" => undefined,
            "org.jboss.as.ejb3" => undefined,
            "org.jboss.as.jaxrs" => undefined,
            "org.jboss.as.jdr" => undefined,
            "org.jboss.as.jmx" => undefined,
            "org.jboss.as.jpa" => undefined,
            "org.jboss.as.jsf" => undefined,
            "org.jboss.as.logging" => undefined,
            "org.jboss.as.mail" => undefined,
            "org.jboss.as.naming" => undefined,
            "org.jboss.as.pojo" => undefined,
            "org.jboss.as.remoting" => undefined,
            "org.jboss.as.sar" => undefined,
            "org.jboss.as.security" => undefined,
            "org.jboss.as.threads" => undefined,
            "org.jboss.as.transactions" => undefined,
            "org.jboss.as.web" => undefined,
            "org.jboss.as.webservices" => undefined,
            "org.jboss.as.weld" => undefined
        },
        "interface" => {
            "management" => undefined,
            "public" => undefined,
            "unsecure" => undefined
        },
        "path" => {
            "jboss.server.temp.dir" => undefined,
            "user.home" => undefined,
            "jboss.server.base.dir" => undefined,
            "java.home" => undefined,
            "user.dir" => undefined,
            "jboss.server.data.dir" => undefined,
            "jboss.home.dir" => undefined,
            "jboss.server.log.dir" => undefined,
            "jboss.server.config.dir" => undefined,
            "jboss.controller.temp.dir" => undefined
        },
        "socket-binding-group" => {"standard-sockets" =>
undefined},
        "subsystem" => {
            "logging" => undefined,
```

```
            "datasources" => undefined,
            "deployment-scanner" => undefined,
            "ee" => undefined,
            "ejb3" => undefined,
            "infinispan" => undefined,
            "jaxrs" => undefined,
            "jca" => undefined,
            "jdr" => undefined,
            "jmx" => undefined,
            "jpa" => undefined,
            "jsf" => undefined,
            "mail" => undefined,
            "naming" => undefined,
            "pojo" => undefined,
            "remoting" => undefined,
            "resource-adapters" => undefined,
            "sar" => undefined,
            "security" => undefined,
            "threads" => undefined,
            "transactions" => undefined,
            "web" => undefined,
            "webservices" => undefined,
            "weld" => undefined
        }
    }
}
```

2. **Exécuter l'opération `read-resource` pour un nœud enfant**
   L'opération **`read-resource`** peut être exécutée pour chercher les nœuds enfants à partir de la racine. La structure de l'opération commence par définir le nœud à exposer, puis s'ajoute à l'opération pour exécuter à ses côtés.

   ```
   [standalone@localhost:9999 /]/subsystem=web/connector=http:read-
   resource
   ```

   Dans l'exemple suivant, on peut exposer des informations de ressources spécifiques sur un composant de sous-système en redirigeant l'opération **`read-resource`** vers un nœud de sous-système web particulier.

   **Exemple 3.12. Exposer les ressources de nœud enfant à partir d'un nœud racine**

   ```
   [standalone@localhost:9999 /] /subsystem=web/connector=http:read-
   resource
   {
       "outcome" => "success",
       "result" => {
           "configuration" => undefined,
           "enable-lookups" => false,
           "enabled" => true,
           "executor" => undefined,
           "max-connections" => undefined,
           "max-post-size" => 2097152,
           "max-save-post-size" => 4096,
   ```

```
        "name" => "http",
        "protocol" => "HTTP/1.1",
        "proxy-name" => undefined,
        "proxy-port" => undefined,
        "redirect-port" => 443,
        "scheme" => "http",
        "secure" => false,
        "socket-binding" => "http",
        "ssl" => undefined,
        "virtual-server" => undefined
    }
}
```

Les mêmes résultats sont possibles en utilisant la commande **cd** pour naviguer dans les nœuds enfants et en exécutant l'opération **`read-resource`** directement.

**Exemple 3.13. Exposer les ressources de nœud enfant en changeant de répertoire**

```
[standalone@localhost:9999 /] cd subsystem=web
```

```
[standalone@localhost:9999 subsystem=web] cd connector=http
```

```
[standalone@localhost:9999 connector=http] :read-resource
{
    "outcome" => "success",
    "result" => {
        "configuration" => undefined,
        "enable-lookups" => false,
        "enabled" => true,
        "executor" => undefined,
        "max-connections" => undefined,
        "max-post-size" => 2097152,
        "max-save-post-size" => 4096,
        "name" => "http",
        "protocol" => "HTTP/1.1",
        "proxy-name" => undefined,
        "proxy-port" => undefined,
        "redirect-port" => 443,
        "scheme" => "http",
        "secure" => false,
        "socket-binding" => "http",
        "ssl" => undefined,
        "virtual-server" => undefined
    }
}
```

3. **Utiliser le paramètre récursif pour inclure des valeurs actives dans les résultats**

Le paramètre récursif peut être utilisé pour exposer les valeurs de tous les attributs, y compris les valeurs non persistantes, celles qui sont passés au démarrage, ou les autres attributs normalement actifs du modèle d'exécution.

```
[standalone@localhost:9999 /]/interface=public:read-
resource(include-runtime=true)
```

Par rapport à l'exemple précédent, l'inclusion de la propriété de requête **`include-runtime`** expose des attributs actifs supplémentaires, comme des octets envoyés ou des octets reçus par le connecteur HTTP.

**Exemple 3.14. Exposer des valeurs actives et supplémentaires par le paramètre `include-runtime`**

```
[standalone@localhost:9999 /] /subsystem=web/connector=http:read-
resource(include-runtime=true)
{
    "outcome" => "success",
    "result" => {
        "any" => undefined,
        "any-address" => undefined,
        "any-ipv4-address" => undefined,
        "any-ipv6-address" => undefined,
        "inet-address" => expression
"${jboss.bind.address:127.0.0.1}",
        "link-local-address" => undefined,
        "loopback" => undefined,
        "loopback-address" => undefined,
        "multicast" => undefined,
        "name" => "public",
        "nic" => undefined,
        "nic-match" => undefined,
        "not" => undefined,
        "point-to-point" => undefined,
        "public-address" => undefined,
        "resolved-address" => "127.0.0.1",
        "site-local-address" => undefined,
        "subnet-match" => undefined,
        "up" => undefined,
        "virtual" => undefined
    }
}
```

Report a bug

### 3.6.7. Afficher les descriptions de ressources disponibles en utilisant le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.20. Éxécuter la commande de Management CLI suivante**

1. **Exécuter l'opération `read-resource-description`**
   À partir du Management CLI, utiliser l'opération **`read-resource-description`** pour lire et afficher les noms des ressources disponibles. Pour plus d'informations sur les requêtes d'opérations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes du Management CLI ».

   ```
   [standalone@localhost:9999 /]:read-resource-description
   ```

2. **Utiliser les paramètres en option**
   L'opération **`read-resource-description`** autorise l'utilisation de paramètres supplémentaires.

   a. Utiliser le paramètre **`operations`** pour inclure les descriptions des opérations de la ressource.

   ```
   [standalone@localhost:9999 /]:read-resource-
   description(operations=true)
   ```

   b. Utiliser le paramètre **`inherited`** pour inclure ou pour exclure des descriptions des opérations héritées de ressource. L'état par défaut est true.

   ```
   [standalone@localhost:9999 /]:read-resource-
   description(inherited=false)
   ```

   c. Utiliser le paramètre **`recursive`** pour inclure les descriptions récursives des ressources dépendantes.

   ```
   [standalone@localhost:9999 /]:read-resource-
   description(recursive=true)
   ```

   d. Utiliser le paramètre **`locale`** pour obtenir la description des ressources. Si «null», la locale régionale par défaut sera utilisée.

   ```
   [standalone@localhost:9999 /]:read-resource-
   description(locale=true)
   ```

**Résultat**

Les descriptions des ressources disponibles sont affichées.

Report a bug

### 3.6.8. Charger à nouveau le serveur d'applications à l'aide du Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.21. Charger à nouveau le serveur d'applications**

- **Exécuter l'opération `reload`**

À partir du Management CLI, utiliser l'opération **`reload`** pour fermer le serveur via l'appel de système **`System.exit(0)`**. Pour plus d'informations sur les demandes d'opérations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes du Management CLI ».

```
[standalone@localhost:9999 /]:reload
{"outcome" => "success"}
```

**Résultat**

Le serveur d'applications charge à nouveau en fermant tout d'abord tous les services et en démarrant le runtime à nouveau. Le Management CLI reconnecte automatiquement.

Report a bug

### 3.6.9. Fermer le serveur d'applications à l'aide du Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.22. Fermer le Serveur d'applications**

- **Exécuter l'opération `shutdown`**
  - À partir du Management CLI, utiliser l'opération **`shutdown`** pour fermer le serveur via l'appel de système **`System.exit(0)`**. Pour plus d'informations sur les requêtes d'opérations, voir le sujet Section 3.5.8, « Utiliser les opérations et les commandes du Management CLI ».
    - En mode autonome, utiliser la commande suivante :

      ```
      [standalone@localhost:9999 /]:shutdown
      ```

    - En mode domaine, utiliser la commande suivante avec le nom d'hôte qui convient :

      ```
      [domain@localhost:9999 /]/host=master:shutdown
      ```

  - Pour vous connecter à une instance CLI détachée et pour fermer le serveur, exécuter la commande suivante :

    ```
    jboss-admin.sh --connect command=:shutdown
    ```

  - Pour vous connecter à une instance CLI éloignée et pour fermer le serveur, exécuter la commande suivante :

    ```
    [disconnected /] connect IP_ADDRESS
    Connected to IP_ADDRESS:PORT_NUMBER
    [192.168.1.10:9999 /] :shutdown
    ```

    Remplacer l' *IP_ADDRESS* par l'adresse IP de votre instance.

**Résultat**

Le serveur d'applications se ferme. Le Management CLI sera disconnecté car le runtime n'est pas disponible.

Report a bug

### 3.6.10. Configurer un attribut à l'aide du Management CLI

#### Prérequis

- Section 3.5.2, « Lancement du Management CLI »

#### Résumé

L'opération **`write-attribute`** est une opération globale utilisée pour écrire ou modifier un attribut de la ressource sélectionnée. Vous pouvez utiliser l'opération pour rendre les modifications persistantes ou pour modifier les paramètres de configuration de vos instances de serveur géré. Les propriétés de la requête incluent les paramètres suivants.

#### Propriétés de requêtes

**`name`**

Le nom de l'attribut pour définir la valeur sous la ressource sélectionnée.

**`value`**

La valeur désirée de l'attribut sous la ressource sélectionnée. Peut être «null» si le modèle sous-jacent supporte des valeurs «null».

#### Procédure 3.23. Configurer un attribut de ressource à l'aide du Management CLI

- **Exécuter l'opération`write-attribute`**
  À partir du Management CLI, utiliser l'opération **`write-attribute`** pour modifier la valeur d'un attribut de ressource. L'opération peut être exécutée dans le nœud dépendant de la ressource, ou bien dans le nœud root du Management CLI où le chemin de ressource complet est spécifié.

> **Exemple 3.15. Désactiver le scanner de déploiement par l'opération `write-attribute`**
>
> L'exemple suivant utilise l'opération **`write-attribute`** pour désactiver le scanner de déploiement. L'opération est exécutée à partir d'un nœud root, en utilisant l'onglet de complétion de tâche pour pouvoir peupler le chemin de ressources qui convient.
>
> ```
> [standalone@localhost:9999 /] /subsystem=deployment-
> scanner/scanner=default:write-attribute(name=scan-enabled,value=false)
> {"outcome" => "success"}
> ```
>
> Le résultat de l'opération peut être confirmé directement par l'opération **`read-attribute`**.
>
> ```
> [standalone@localhost:9999 /] /subsystem=deployment-
> scanner/scanner=default:read-attribute(name=scan-enabled)
> {
>     "outcome" => "success",
>     "result" => false
> }
> ```

Les résultats peuvent également être confirmés en listant tous les attributs de ressources du nœud disponibles par l'opération **`read-resource`**. Dans l'exemple suivant, cette configuration particulière vous montre que l'attribut **`scan-enabled`** est maintenant défini à **`false`**.

```
[standalone@localhost:9999 /] /subsystem=deployment-
scanner/scanner=default:read-resource
{
    "outcome" => "success",
    "result" => {
        "auto-deploy-exploded" => false,
        "auto-deploy-xml" => true,
        "auto-deploy-zipped" => true,
        "deployment-timeout" => 600,
        "path" => "deployments",
        "relative-to" => "jboss.server.base.dir",
        "scan-enabled" => false,
        "scan-interval" => 5000
    }
}
```

**Résultat**

L'attribut de ressource est mis à jour.

Report a bug

## 3.7. HISTORIQUE DE LA COMMANDE MANAGEMENT CLI

### 3.7.1. Utiliser l'Historique de commande à l'aide du Management CLI.

Le Management CLI contient une fonctionnalité d'historique de commande qui est activée par défaut dans l'installation du serveur d'applications. L'historique est conservé à la fois en tant qu'archive dans la mémoire volatile de la session CLI active, et est ajouté le fichier de journalisation qui sauvegarde automatiquement dans le répertoire d'accueil de l'utilisateur sous le nom **`.jboss-cli-history`**. Le fichier de l'historique est configuré par défaut pour enregistrer un maximum de 500 commandes CLI.

La commande **`history`** elle-même renverra l'historique de la session en cours, ou si accompagnée d'arguments, elle pourra désactiver, activer ou supprimer l'historique de la mémoire de session. Le Management CLI vous donne également la possibilité d'utiliser les flèches de votre clavier pour naviguer dans l'historique des commandes et des opérations.

**Fonctions de l'historique du Management CLI**

- Section 3.7.2, « Afficher l'Historique de commandes à l'aide du Management CLI. »
- Section 3.7.3, « Effacer l'Historique de commandes à l'aide du Management CLI. »
- Section 3.7.4, « Désactiver l'Historique de commandes à l'aide du Management CLI. »
- Section 3.7.5, « Activer l'Historique de commandes à l'aide du Management CLI. »

Report a bug

### 3.7.2. Afficher l'Historique de commandes à l'aide du Management CLI.

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.24. Afficher l'Historique de commandes à l'aide du Management CLI.**

- **Exécuter la commande `history`**
  À partir du Management CLI, saisir la commande **`history`** :

```
[standalone@localhost:9999 /] history
```

**Résultat**

L'historique de la commande CLI stocké en mémoire depuis le démarrage du CLI ou la commande de suppression de l'historique est affiché.

Report a bug

### 3.7.3. Effacer l'Historique de commandes à l'aide du Management CLI.

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.25. Effacer l'Historique de commandes à l'aide du Management CLI.**

- **Exécuter la commande `history --clear`**
  À partir du Management CLI, saisir la commande **`version`** :

```
[standalone@localhost:9999 /] history --clear
```

**Résultat**

L'historique des commandes enregistré depuis le démarrage du CLI est supprimé de la mémoire de session. L'historique de la commande est toujours présent dans le fichier **`.jboss-cli-history`** qui est sauvegardé dans le répertoire d'accueil de l'utilisateur.

Report a bug

### 3.7.4. Désactiver l'Historique de commandes à l'aide du Management CLI.

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.26. Désactiver l'Historique de commandes à l'aide du Management CLI.**

- **Exécuter la commande `history --disable`**
  À partir du Management CLI, saisir la commande **`history --disable`** :

```
[standalone@localhost:9999 /] history --disable
```

**Résultat**

Les commandes passées dans le CLI ne seront pas enregistrées dans la mémoire ou dans un fichier **`.jboss-cli-history`** sauvegardé dans le répertoire d'accueil de l'utilisateur.

Report a bug

### 3.7.5. Activer l'Historique de commandes à l'aide du Management CLI.

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Procédure 3.27. Activer l'Historique de commandes à l'aide du Management CLI.**

- **Exécuter la commande `history --enable`**
  À partir du Management CLI, saisir la commande **`history --enable`** :

  ```
  [standalone@localhost:9999 /] history --enable
  ```

**Résultat**

Les commandes passées dans le CLI seront enregistrées dans la mémoire ou dans un fichier **`.jboss-cli-history`** sauvegardé dans le répertoire d'accueil de l'utilisateur.

Report a bug

# CHAPITRE 4. GESTION DES UTILISATEURS

## 4.1. CRÉATION D'UTILISATEUR

### 4.1.1. Ajouter l'utilisateur d'origine dans les interfaces de gestion

**Aperçu**

La gestion des interfaces de gestion de la plate-forme JBoss EAP 6 sont sécurisées par défaut, et il n'y a pas de comptes d'utilisateur disponibles au départ, à moins que vous ayez installé la plateforme par l'installateur graphique. Il s'agit d'une précaution de sécurité, afin de prévenir les atteintes à la sécurité des systèmes à distance pour cause de simples erreurs de configuration. L'accès non HTTP local est protégé par un mécanisme SASL, par une négociation qui a lieu entre le client et le serveur chaque fois que le client se connecte pour la première fois à partir de l'hôte local.

Cette tâche décrit comment créer l'utilisateur d'administration d'origine, qui peut utiliser la console de gestion basée web et les instances éloignées du Management CLI pour configurer et administrer la plate-forme JBoss EAP 6 à partir de systèmes distants. Pour plus d'informations sur la configuration de sécurité par défaut, reportez-vous à Section 10.7.1, « Configuration Sécurité Utilisateur par défaut ».

**NOTE**

La communication HTTP avec la plate-forme EAP est considérée «communication à distance», même si le trafic prend sa source sur l'hôte local. Par conséquent, vous devez créer au moins un utilisateur afin de pouvoir utiliser la console de gestion. Si vous essayiez d'accéder à la console de gestion avant d'ajouter un utilisateur, vous recevriez une erreur parce qu'il n'y a pas de déploiement consistent tant que l'utilisateur n'a pas été ajouté.

**Procédure 4.1. Créer l'utilisateur administratif d'origine pour les interfaces de gestion distantes**

1. **Invoquer le script `add-user.sh` ou `add-user.bat` script.**
   Remplacer le répertoire ***`JJP_HOME`*`/bin/`**. Invoquer le script qui convient à votre système d'exploitation.

   **Red Hat Enterprise Linux**

   ```
   [user@host bin]$ ./add-user.sh
   ```

   **Microsoft Windows Server**

   ```
   C:\bin>  add-user.bat
   ```

2. **Choisissez d'utiliser un utilisateur Management.**
   Sélectionner l'option **a** pour ajouter un utilisateur Management. Cet utilisateur sera ajouté au domaine **`ManagementRealm`** et il sera autorisé à effectuer des opérations de gestion par la Management Console basée web ou par le Management CLI basé sur la ligne de commande. Autre alternative, **b**, ajoutera l'utilisateur au domaine **`ApplicationRealm`**, et ne fournit aucune permission particulière. Ce domaine est fourni pour être utilisé avec des applications.

3. **Choisissez le domaine pour l'utilisateur :**

L'invite suivante se rapporte au domaine où l'utilisateur sera ajouté. Pour un utilisateur ayant les permissions de gérer JBoss EAP 6, choisissez la valeur par défaut, qui est **`ManagementRealm`**.

4. **Saisir le nom d'utilisateur et le mot de passe que vous souhaitez.**
   Lorsque vous y êtes invité, saisir le domaine de sécurité, le nom d'utilisateur et le mot de passe. Appuyer sur **`ENTER`** sélectionne le domaine **`ManagementRealm`** par défaut, qui permet à l'utilisateur d'administrer la plate-forme JBoss EAP 6 en utilisant les interfaces de gestion. Vous devez ajouter au moins un utilisateur dans le domaine. Vous êtes invité à confirmer l'information. Si vous êtes satisfait, saisir **`yes`** .

5. **Décidez si l'utilisateur représente une instance de serveur de JBoss Enterprise Application Platform 6 à distance.**
   En plus des administrateurs, un autre type d'utilisateur qui a parfois besoin d'être ajouté à JBoss Enterprise Application Platform 6 dans le domaine **`ManagementRealm`** est un utilisateur qui représente une autre instance de JBoss Enterprise Application Platform 6, et qui a besoin d'être authentifié pour rejoindre un groupement en tant que membre. L'invite suivante vous permet de désigner votre utilisateur supplémentaire dans ce but. Si vous sélectionnez **`yes`**, on vous donnera une valeur **`secret`** de hachage, qui représentera le mot de passe de l'utilisateur, que vous aurez besoin d'ajouter dans un fichier de configuration différent. Dans le but de cette tâche, répondre **no** à cette question.

6. **Saisir des utilisateurs supplémentaires.**
   Vous pouvez saisir des utilisateurs supplémentaires si vous le souhaitez, en répétant la procédure. Vous pouvez également les ajouter à tout moment sur un système en cours d'exécution. Au lieu de choisir le domaine de sécurité par défaut, vous pouvez ajouter des utilisateurs d'autres domaines afin d'ajuster leurs autorisations.

7. **Créer des utilisateurs en mode non interactif.**
   Vous pouvez créer des utilisateurs en mode non interactif, en l'indiquant dans chaque paramètre de ligne de commande. Cette approche n'est pas recommandée sur les systèmes partagés, parce que les mots de passe seront visibles dans les fichiers de journalisation (log) et dans les fichiers d'historique. La syntaxe de la commande, pour le domaine de gestion, est la suivante :

   ```
   [user@host bin]$ ./add-user.sh usernamepassword
   ```

   Pour utiliser le domaine d'application, utiliser le paramètre **`-a`**.

   ```
   [user@host bin]$ ./add-user.sh -a usernamepassword
   ```

8. Vous pouvez supprimer la sortie normale du script utilisateur ajouter en passant le paramètre **`--silent`**. Cela s'applique uniquement si un minimum de paramètres, **`nom d'utilisateur`** et **`mot de passe`**, ont été indiqués. Le message d'erreur apparaîtra toujours.

**Résultat**

Tout utilisateur que vous ajoutez est activé dans les domaines de sécurité que vous avez indiqués. Les utilisateurs actifs dans le domaine **`ManagementRealm`** sont en mesure de gérer la plateforme JBoss EAP 6 à partir de systèmes éloignés.

Report a bug

### 4.1.2. Ajout d'un utilisateur dans l'interface de gestion

Use the same procedure outlined in Section 4.1.1, « Ajouter l'utilisateur d'origine dans les interfaces de gestion ».

Report a bug

# CHAPITRE 5. RÉSEAU ET CONFIGURATION DE PORT

## 5.1. INTERFACES

### 5.1.1. Les interfaces

Le serveur d'applications utilise des références d'interface nommées dans la configuration. Cela permet à la configuration de référencer les déclarations d'interface individuelle avec des noms logiques, au lieu de référencer toutes les informations sur l'interface à chaque utilisation. L'utilisation de noms logiques permettent également de conserver une homogénéité des références de groupe pour les interfaces nommées, quand les instances de serveur d'un domaine géré peut contenir des informations d'interface diverses à travers les machines. En utilisant cette méthodologie, chaque instance de serveur peut correspondre à un groupe de nom logique, ce qui facilite l'administration du groupe d'interfaces dans son ensemble.

Une interface réseau est déclarée en spécifiant un nom logique et un critère de sélection pour l'interface physique. Le serveur d'applications est fourni avec une configuration par défaut pour une gestion et un nom d'interface publique. Dans cette configuration, le groupe de l'interface publique est destiné à être utilisé par toute communication de réseau liée à l'application comme Web Messagerie. Le groupe interface de gestion est destiné à tous les composants et les services requis par la couche de gestion, y compris HTTP Management Endpoint. Les noms d'interface sont fournis en tant que suggestions uniquement, alors que n'importe quel nom de groupe peut être substitué ou créé selon les besoins.

Les fichiers de configuration **`domain.xml`**, **`host.xml`** et **`standalone.xml`** comprennent tous des déclarations d'interface. Les critères de déclaration peuvent référencer une adresse générique ou spécifier un ensemble de caractéristiques qu'une interface ou une adresse doit avoir pour pouvoir établir une correspondance valide. Les exemples suivants illustrent plusieurs configurations possibles de déclarations d'interface, généralement définies soit dans le fichier de configuration **`standalone.xml`** ou **`host.xml`**. Cela permet à des groupes d'hôtes distants de maintenir leurs propres attributs d'interfaces spécifiques, tout en permettant une référence aux groupes d'interfaces dans le fichier de configuration **`domain.xml`** du contrôleur de domaine.

Le premier exemple montre une valeur d'**`inet-address`** indiquée pour le groupes de noms relatifs **`management`** et **`public`**.

**Exemple 5.1. Un groupe d'interfaces créé avec une adresse inet**

```
<interfaces>
  <interface name="management">
   <inet-address value="127.0.0.1"/>
  </interface>
  <interface name="public">
   <inet-address value="127.0.0.1"/>
  </interface>
</interfaces>
```

Dans l'exemple suivant, un groupe global interface utilise l'élément **`any-address`** pour déclarer une adresse générique.

**Exemple 5.2. Un groupe golbal créé avec une déclaration générique**

```
<interface name="global">
   <!-- Use the wild-card address -->
   <any-address/>
</interface>
```

L'exemple suivant déclare une carte d'interface de réseau sous un groupe relatif avec un nom **externe**.

**Exemple 5.3. Un groupe externe créé avec un NIC**

```
<interface name="external">
   <nic name="eth0"/>
</interface>
```

Dans l'exemple suivant, une déclaration est créée comme groupe par défaut pour un besoin spécifique. Dans ce cas, les caractéristiques des éléments supplémentaires définissent les conditions pour que l'interface soit une correspondance valide. Cela permet la création de groupes de déclaration d'interface très spécifique, avec la possibilité de les référencer de manière prédéfinie, réduisant ainsi le temps de configuration et d'administration sur plusieurs instances de serveur.

**Exemple 5.4. Un groupe par défaut créé avec des valeurs conditionnelles spécifiques**

```
<interface name="default">
   <!-- Match any interface/address on the right subnet if it's
        up, supports multicast, and isn't point-to-point -->
   <subnet-match value="192.168.0.0/16"/>
   <up/>
   <multicast/>
   <not>
      <point-to-point/>
   </not>
</interface>
```

Alors que les déclarations d'interface peuvent être faites et modifiées dans les fichiers de configuration des sources, le Management CLI et la Console de gestion fournissent un environnement sécurisé, contrôlé et persistant pour les modifications de configuration.

Report a bug

### 5.1.2. Configurer les interfaces

Les configurations de l'interface par défaut des fichiers de configuration **standalone.xml** et **host.xml** offrent trois interfaces nommées avec jetons d'interfaces relatives pour chacune. Vous pouvez utiliser la Console de gestion ou le Management CLI pour configurer des attributs et valeurs

supplémentaires indiquées dans le tableau ci-dessous. Vous pouvez également remplacer les liaisons d'interfaces relatives par des valeurs spécifiques selon les besoins. Notez que si vous le faites, vous serez incapable de passer une valeur d'interface en cours d'exécution de serveur, car **-b** peut seulement substituer une valeur relative.

**Exemple 5.5. Configuration d'interface par défaut**

```
        <interfaces>
            <interface name="management">
                <inet-address
value="${jboss.bind.address.management:127.0.0.1}"/>
            </interface>
            <interface name="public">
                <inet-address value="${jboss.bind.address:127.0.0.1}"/>
            </interface>
            <interface name="unsecure">
                <inet-address
value="${jboss.bind.address.unsecure:127.0.0.1}"/>
            </interface>
        </interfaces>
```

**Tableau 5.1. Attributs et valeurs d'interface**

| Élément d'interface | Description |
|---|---|
| **any** | Élément vide de type exclusion d'adresse, utilisé pour forcer le critère de sélection. |
| **any-address** | Élément vide indiquant que les sockets qui utilisent cette interface doivent être liés à une adresse générique. L'adresse générique IPv6 (::) sera utilisée à moins que la propriété système java.net.preferIpV4Stack soit définie sur true, dans lequel cas, l'adresse générique (0.0.0.0) IPv4 sera utilisée. Si un socket est lié à une adresse anylocal IPv6 sur une machine dual-stack, il pourra accepter le trafic IPv6 et IPv4 ; si lié à l'adresse IPv4 anylocal (mappées IPv4), il ne peut accepter que le trafic IPv4. |
| **any-ipv4-address** | Élément vide indiquant que les sockets qui utilisent cette interface doivent être liés à une adresse générique (0.0.0.0) IPv4. |
| **any-ipv6-address** | Élément vide indiquant que les sockets qui utilisent cette interface doivent être liés à une adresse générique (::). IPv6. |
| **inet-address** | Soit une adresse IP en notation à points IPV6 ou IPV4, ou un nom d'hôte qui puisse être résolu. |
| **link-local-address** | Élément vide indiquant qu'une partie du critère de sélection d'une interface doit si oui ou non il a une adresse associée local-link. |

| Élément d'interface | Description |
|---|---|
| **loopback** | Élément vide indiquant qu'une partie du critère de sélection d'une interface est de savoir s'il s'agit oui ou non d'une interface de loopback. |
| **loopback-address** | Une adresse de loopback qui ne peut pas réellement être configurée sur l'interface de loopback de la machine. Diffère d'inet-addressType car la valeur donnée sera utilisée même si aucune carte réseau possédant l'adresse IP associée ne peut être trouvée. |
| **multicast** | Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être si oui ou non il y a un support multi-diffusion. |
| **nic** | Le nom d'une interface de réseau (e.g. eth0, eth1, lo). |
| **nic-match** | Une expression standard à laquelle faire correspondre les noms des interfaces de réseau disponibles sur la machine pour trouver une interface qui convienne. |
| **not** | Élément vide de type exclusion d'adresse, utilisé pour forcer le critère de sélection. |
| **point-to-point** | Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être si elle a oui ou non une interface d'un point à un autre. |
| **public-address** | Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être si elle a oui ou non une adresse publiquement routable. |
| **site-local-address** | Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être ou non une adresse associée à à son site-local |
| **subnet-match** | Une adresse IP réseau et le nombre de bits dans le préfixe de réseau de l'adresse, sous la forme « / » ; par exemple"192.168.0.0/16". |
| **up** | Élément vide indiquant qu'une partie du critère de sélection d'une interface est active ou non. |
| **virtual** | Élément vide indiquant qu'une partie du critère de sélection d'une interface doit être ou non une interface virtuelle. |

- **Configuration des attributs d'une interface**
  Sélectionner le Management CLI ou la Console de gestion pour configurer vos attributs d'interface selon les besoins.

  - **Configurer les attributs d'interface par le Management CLI**
    Utiliser le Management CLI pour ajouter de nouvelles interfaces et pour écrire des nouvelles valeurs dans dans les attributs de l'interface.

    a. **Ajouter une nouvelle interface**

Utiliser l'opération **add** (ajouter) pour créer une nouvelle interface selon les besoins. Vous pouvez exécuter cette commande à partir de la racine de la session de Management CLI, qui, dans l'exemple suivant, crée un nouveau titre de nom d'interface *interfacename*, avec **inet-address** déclaré ainsi *12.0.0.2*.

```
/interface=interfacename/:add(inet-address=12.0.0.2)
```

b. **Modifier les attributs d'une interface**
Utiliser l'opération **write** pour écrire une nouvelle valeur dans un attribut. Vous pouvez utiliser l'onglet de complétion pour terminer la chaîne de commande en cours, ainsi que pour exposer les attributs disponibles. L'exemple suivant met à jour la valeur de l' **inet-address** à *12.0.0.8*

```
/interface=interfacename/:write(inet-address=12.0.0.8)
```

c. **Modifier les attributs d'une interface**
Confirmer que les valeurs ont changé en exécutant **read-resource** accompagné du paramètre **include-runtime=true** pour exposer toutes les valeurs actives en cours du modèle de serveur.

```
[standalone@localhost:9999 interface=public] :read-
resource(include-runtime=true)
```

- **Configurer les attributs d'interface par la Console de gestion**
Utiliser la Console de gestion pour ajouter des nouvelles interfaces et pour écrire des nouvelles valeurs dans les attributs de l'interface.

a. **Connectez-vous à la Console de gestion.**
Connectez-vous à la Console de gestion de votre instance de Serveur autonome ou Managed Domain (Domaine géré).

b. **Si vous utilisez le Managed Domain, sélectionner le profuil qui convient.**
Sélectionner l'onglet **Profiles** en haut à droite, puis sélectionner le profil qui convient à partir du menu **Profile** en haut et à gauche de l'écran suivant.

c. **Sélectionner l'item Interfaces à partir du menu de navigation.**
Sélectionner l'item de menu **Interfaces** à partir du menu de navigation.

d. **Ajouter une nouvelle interface**

i. Cliquer sur le bouton **Add** (ajouter).

ii. Saisir les valeurs qui conviennent pour les **Name** (Nom), **Inet Address** (Adresse Inet) et **Address Wildcard** (Adresse générique).

iii. Cliquer sur le bouton **Save** pour terminer.

e. **Modifier les attributs d'une interface**

i. Sélectionner l'interface à modifier et cliquer sur le bouton **Edit**.

ii. Saisir les valeurs qui conviennent pour les **Name** (Nom), **Inet Address** (Adresse Inet) et **Address Wildcard** (Adresse générique).

iii. Cliquer sur le bouton **Save** pour terminer.

Report a bug

## 5.2. GROUPES DE LIAISONS DE SOCKETS

### 5.2.1. Groupes de liaisons de sockets

Les liaisons de sockets et les groupes de liaisons de sockets vous permettent de définir les ports réseau et leur relation aux interfaces réseau utiles à votre configuration JBoss Enterprise Application Platform 6.

Une liaison de socket est une configuration nommée pour un socket. Les déclarations pour ces configurations nommées se trouvent dans les deux fichiers de configuration **`domain.xml`** et **`standalone.xml`**. D'autres sections de la configuration peuvent alors faire référence à ces sockets par leur nom logique, plutôt que d'avoir à inclure tous les détails de la configuration de socket. Cela permet de référencer des configurations de sockets relatives qui peuvent varier sur des machines différentes.

Les liaisons de sockets sont rassemblées sous un groupe de liaisons du sockets. Un groupe de liaisons de sockets est une collection de déclarations de liaisons de sockets qui sont regroupées sous un nom logique. Le groupe peut ensuite être référencé dans l'ensemble de la configuration. Un serveur autonome contient uniquement un de ces groupes, alors qu'une instance managée de domaine peut contenir plusieurs groupes. Vous pouvez créer un groupe de liaison du socket pour chaque groupe de serveurs dans le domaine géré, ou partager un groupe de liaisons de sockets entre plusieurs groupes de serveurs.

Les groupes de noms permettent à des références simplifiées d'être utilisées pour des groupes particuliers de liaisons de sockets lors de la configuration des groupes de serveurs, dans le cas d'un domaine géré. Une autre utilisation courante est pour la configuration et la gestion de plusieurs instances du serveur autonome sur un seul système. Les exemples suivants montrent les groupes de liaison de sockets par défaut dans les fichiers de configuration des instances du domaine et du serveur autonome.

**Exemple 5.6. Liaisons de sockets par défaut pour la configuration du serveur autonome.**

Les groupes de liaisons de sockets par défaut dans le fichier de configuration **`standalone.xml`** sont groupés sous **`standard-sockets`**. Ce groupe est aussi référencé dans l'interface **`publique`** , en utilisant la même méthodologie de référencement logique.

```
<socket-binding-group name="standard-sockets" default-
interface="public">
    <socket-binding name="http" port="8080"/>
    <socket-binding name="https" port="8443"/>
    <socket-binding name="jacorb" port="3528"/>
    <socket-binding name="jacorb-ssl" port="3529"/>
    <socket-binding name="jmx-connector-registry" port="1090"
interface="management"/>
    <socket-binding name="jmx-connector-server" port="1091"
interface="management"/>
    <socket-binding name="jndi" port="1099"/>
    <socket-binding name="messaging" port="5445"/>
    <socket-binding name="messaging-throughput" port="5455"/>
    <socket-binding name="osgi-http" port="8090"
interface="management"/>
    <socket-binding name="remoting" port="4447"/>
```

```
    <socket-binding name="txn-recovery-environment" port="4712"/>
    <socket-binding name="txn-status-manager" port="4713"/>
</socket-binding-group>
```

**Exemple 5.7. Liaisons de sockets par défaut pour la configuration du serveur autonome.**

Les groupes de liaisons de sockets par défaut dans le fichier de configuration **domain.xml** contiennent quatre groupes : **standard-sockets**, **ha-sockets**, **full-sockets** et **full-ha-sockets**. Ces groupes sont également référencés dans une interface appelée **public**.

```
<socket-binding-groups>
    <socket-binding-group name="standard-sockets" default-
interface="public">
        <!-- Needed for server groups using the 'default' profile  -->
        <socket-binding name="ajp" port="8009"/>
        <socket-binding name="http" port="8080"/>
        <socket-binding name="https" port="8443"/>
        <socket-binding name="osgi-http" interface="management"
port="8090"/>
        <socket-binding name="remoting" port="4447"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
        </outbound-socket-binding>
    </socket-binding-group>
    <socket-binding-group name="ha-sockets" default-interface="public">
        <!-- Needed for server groups using the 'ha' profile  -->
        <socket-binding name="ajp" port="8009"/>
        <socket-binding name="http" port="8080"/>
        <socket-binding name="https" port="8443"/>
        <socket-binding name="jgroups-mping" port="0" multicast-
address="${jboss.default.multicast.address:230.0.0.4}" multicast-
port="45700"/>
        <socket-binding name="jgroups-tcp" port="7600"/>
        <socket-binding name="jgroups-tcp-fd" port="57600"/>
        <socket-binding name="jgroups-udp" port="55200" multicast-
address="${jboss.default.multicast.address:230.0.0.4}" multicast-
port="45688"/>
        <socket-binding name="jgroups-udp-fd" port="54200"/>
        <socket-binding name="modcluster" port="0" multicast-
address="224.0.1.105" multicast-port="23364"/>
        <socket-binding name="osgi-http" interface="management"
port="8090"/>
        <socket-binding name="remoting" port="4447"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
        </outbound-socket-binding>
    </socket-binding-group>
    <socket-binding-group name="full-sockets" default-
interface="public">
```

```
        <!-- Needed for server groups using the 'full' profile  -->
        <socket-binding name="ajp" port="8009"/>
        <socket-binding name="http" port="8080"/>
        <socket-binding name="https" port="8443"/>
        <socket-binding name="jacorb" interface="unsecure" port="3528"/>
        <socket-binding name="jacorb-ssl" interface="unsecure"
port="3529"/>
        <socket-binding name="messaging" port="5445"/>
        <socket-binding name="messaging-group" port="0" multicast-
address="${jboss.messaging.group.address:231.7.7.7}" multicast-
port="${jboss.messaging.group.port:9876}"/>
        <socket-binding name="messaging-throughput" port="5455"/>
        <socket-binding name="osgi-http" interface="management"
port="8090"/>
        <socket-binding name="remoting" port="4447"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
        </outbound-socket-binding>
    </socket-binding-group>
    <socket-binding-group name="full-ha-sockets" default-
interface="public">
        <!-- Needed for server groups using the 'full-ha' profile  -->
        <socket-binding name="ajp" port="8009"/>
        <socket-binding name="http" port="8080"/>
        <socket-binding name="https" port="8443"/>
        <socket-binding name="jacorb" interface="unsecure" port="3528"/>
        <socket-binding name="jacorb-ssl" interface="unsecure"
port="3529"/>
        <socket-binding name="jgroups-mping" port="0" multicast-
address="${jboss.default.multicast.address:230.0.0.4}" multicast-
port="45700"/>
        <socket-binding name="jgroups-tcp" port="7600"/>
        <socket-binding name="jgroups-tcp-fd" port="57600"/>
        <socket-binding name="jgroups-udp" port="55200" multicast-
address="${jboss.default.multicast.address:230.0.0.4}" multicast-
port="45688"/>
        <socket-binding name="jgroups-udp-fd" port="54200"/>
        <socket-binding name="messaging" port="5445"/>
        <socket-binding name="messaging-group" port="0" multicast-
address="${jboss.messaging.group.address:231.7.7.7}" multicast-
port="${jboss.messaging.group.port:9876}"/>
        <socket-binding name="messaging-throughput" port="5455"/>
        <socket-binding name="modcluster" port="0" multicast-
address="224.0.1.105" multicast-port="23364"/>
        <socket-binding name="osgi-http" interface="management"
port="8090"/>
        <socket-binding name="remoting" port="4447"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
        </outbound-socket-binding>
```

```
    </socket-binding-group>
</socket-binding-groups>
```

Les instances de liaisons de sockets peuvent être créées et modifiées dans les fichiers source **`standalone.xml`** et **`domain.xml`** du répertoire de l'application. La méthode recommandée pour gérer les liaisons consiste à utiliser la Console de gestion ou le Management CLI. Les avantages à utiliser la Console de gestion est l'interface utilisateur graphique avec une écran de Groupe de liaisons de sockets dédié dans la section **`Configuration générale`** . Le Management CLI propose un API et un flux de travail basés ligne de commande qui permet le traitement par lots et l'utilisation de scripts aux niveaux supérieurs et inférieurs de la configuration de serveur d'applications. Les deux interfaces permettent la persistance des modifications ou bien leur enregistrement dans la configuration du serveur.

Report a bug

### 5.2.2. Configurer les liaisons de sockets

Les liaisons de sockets peuvent être définies dans des groupes de liaisons sockets uniques. Le serveur autonome contient un de ces groupes, le groupe **`standard-sockets`**, et ne peut pas créer de groupes supplémentaires. À la place, vous pouvez créer des fichiers de configuration de serveur autonome alternatif. Pour le domaine géré cependant, vous pouvez créer des groupes de liaisons de sockets et configurer les liaisons de sockets qu'ils contiennent selon vos besoins. Le tableau suivant montre les attributs disponibles pour chaque liaison de sockets.

**Tableau 5.2. Attributs de liaisons de sockets**

| Composant | Description | Rôle |
|---|---|---|
| Nom | Nom logique de la configuration de socket qui doit être utilisée ailleurs dans la configuration. | Requis |
| Port | Port de base auquel un socket basé sur cette configuration doit être lié. Notez que les serveurs peuvent être configurés pour substituer cette valeur de base en appliquant une incrémentation ou décrémentation à toutes les valeurs de port. | Requis |
| Interface | Nom logique de l'interface à laquelle un socket basé sur cette configuration doit être lié. Si non défini, la valeur de l'attribut "default-interface" du groupe de liaison de sockets enveloppant servira. | Option |
| Adresse multi-diffusion | Si le socket doit être utilisé en multi-diffusion, c'est l'adresse multi-diffusion qu'il vous faut. | Option |

| Composant | Description | Rôle |
|---|---|---|
| Port multi-diffusion | Lié au cycle de vie de la conversation. Le scope de la conversation correspond aux longueurs de la requête et de la session, et est contrôlé par l'application. | Option |
| Port fixe | Si les contextes ne correspondent pas à vos besoins, vous pourrez définir des scopes personnalisés. | Option |

- **Configurer des liaisons de sockets dans des Groupes de liaisons de sockets**
  Sélectionner le Management CLI ou la Console de gestion pour configurer vos liaisons de sockets selon les besoins.
  - **Configurer les liaisons de sockets par le Management CLI**
    Sélectionner le Management CLI pour configurer les liaisons de sockets.
    a. **Ajouter un nouvelle liaison de sockets**
    Utiliser l'opération **`add`** (ajouter) pour créer une nouvelle configuration d'adresse si nécessaire. Vous pouvez exécuter cette commande à partir de la racine de la session de Management CLI, qui, dans l'exemple suivant, crée une nouvelle liaison de sockets intitulée *newsocket*, avec un attribut **`port`** déclaré comme *1234*. Les exemples s'appliquent à la fois pour la modification des serveurs autonomes et des serveurs gérés sur la liaison de sockets **`standard-sockets`** comme montré ci-dessous.

```
[domain@localhost:9999 /] /socket-binding-group=standard-
sockets/socket-binding=newsocket/:add(port=1234)
```

    b. **Modifier les attributs de modèle**
    Utiliser l'opération **`write`** pour écrire une nouvelle valeur dans un attribut. Vous pouvez utiliser l'onglet de complétion pour terminer la chaîne de commande en cours, ainsi que pour exposer les attributs disponibles. L'exemple suivant met à jour la valeur du **`port`** à *2020*

```
[domain@localhost:9999 /] /socket-binding-group=standard-
sockets/socket-binding=newsocket/:write-
attribute(name=port,value=2020)
```

    c. **Confirmer les attributs de modèle**
    Confirmer que les valeurs ont changé en exécutant **`read-resource`** accompagné du paramètre **`include-runtime=true`** pour exposer toutes les valeurs actives en cours du modèle de serveur.

```
[domain@localhost:9999 /] /socket-binding-group=standard-
sockets/socket-binding=newsocket/:read-resource
```

  - **Configurer les liaisons de sockets par la Console de gestion**
    Utiliser le Management CLI pour configurer les liaisons de sockets.

a. **Connectez-vous à la Console de gestion.**
Connectez-vous à la Console de gestion de votre domaine géré ou de votre serveur autonome.

b. **Sélectionner le `Profile tab`**
Sélectionner l'onglet **`Profiles`** en haut et à droite.

c. **Sélectionner l'élément `Socket Binding` à partir du menu de navigation.**
Sélectionner l'élément de menu **`Socket Binding`** à partir du menu de navigation. Si vous utilisez un domaine géré, sélectionner le groupe désiré dans le menu **`Socket Binding Groups`**.

d. **Ajouter un nouvelle liaison de sockets**

i. Cliquer sur le bouton **`Add`** (ajouter).

ii. Saisir les valeurs qui conviennent pour les **`Name`** (Nom), **`Port`** (Port) et **`Binding Group`** (Groupe de liaisons).

iii. Cliquer sur le bouton **`Save`** pour terminer.

e. **Modifier les attributs d'une interface**

i. Sélectionner la liaison de sockets à modifier et cliquer sur le bouton **`Edit`**.

ii. Saisir les valeurs qui conviennent pour les **`Name`** (Nom), **`Interface`** (Interface) ou **`Port`** (Port).

iii. Cliquer sur le bouton **`Save`** pour terminer.

Report a bug

### 5.2.3. Ports de réseau utilisés par la plateforme JBoss EAP 6

Les ports de réseau utilisés par la configuration par défaut de la plateforme JBoss EAP 6 dépendent de plusieurs facteurs:

- Le fait que vos groupes de serveurs utilisent le groupe de liaisons de sockets par défaut , ou un groupe de liaisons de sockets personnalisé.
- Des exigences de vos déploiements individuels.

**NOTE**

Un décalage de port numérique peut être configuré pour atténuer les conflits de ports lorsque vous exécutez plusieurs serveurs sur un même serveur physique. Si votre serveur utilise un décalage de port numérique, ajouter le décalage au numéro de port par défaut pour le groupe de liaisons de socket de son groupe de serveurs. Par exemple, si le port HTTP du groupe de liaisons de socket est 8080 et si votre serveur utilise un décalage de port de 100, son port HTTP sera 8180.

À moins d'instruction particulière, les ports utilisent le protocole TCP.

**Groupes de liaison de socket par défaut**

- `full-ha-sockets`
- `full-sockets`
- `ha-sockets`
- `standard-sockets`

**Tableau 5.3. Référence aux Groupes de liaison de socket par défaut**

| Nom | Port | Port Multidiffusion | Description | full-ha-sockets | full-sockets | ha-socket | standard-socket |
|---|---|---|---|---|---|---|---|
| `ajp` | 8009 | | Protocole Apache JServ. Utilisé pour le clustering HTTP et pour l'équilibrage des charges. | Oui | Oui | Oui | Oui |
| `http` | 8080 | | Le port par défaut des applications déployées. | Oui | Oui | Oui | Oui |
| `https` | 8443 | | Connexion cryptée-SSL entre les applications déployées et les clients. | Oui | Oui | Oui | Oui |
| `jacorb` | 3528 | | Services CORBA pour les transactions JTS et autres services dépendants-ORB. | Oui | Oui | Non | Non |
| `jacorb-ssl` | 3529 | | Services CORBA cryptés-SSL. | Oui | Oui | Non | Non |
| `jgroups-diagnostics` | | 7500 | Multicast. Utilisé pour la découverte de paires dans les groupements HA. | Oui | Non | Oui | Non |
| `jgroups-mping` | | 45700 | Multicast. Utilisé pour découvrir l'appartenance de groupe d'orgine dans un cluster HA. | Oui | Non | Oui | Non |

| Nom | Port | Port Multidiffusion | Description | full-ha-sockets | full-sockets | ha-socket | standard-socket |
|---|---|---|---|---|---|---|---|
| **jgroups-tcp** | 7600 | | Découverte de paires unicast dans les groupements HA avec TCP. | Oui | Non | Oui | Non |
| **jgroups-tcp-fd** | 57600 | | Utilisé pour la détection des échecs en TCP. | Oui | Non | Oui | Non |
| **jgroups-udp** | 55200 | 45688 | Découverte de paires unicast dans les groupements HA avec TCP. | Oui | Non | Oui | Non |
| **jgroups-udp-fd** | 54200 | | Utilisé pour la détection des échecs par UDP. | Oui | Non | Oui | Non |
| **messaging** | 5445 | | Service JMS. | Oui | Oui | Non | Non |
| **messaging-group** | | | Référencé par la difffusion HornetQ JMS et les groupes Discovery | Oui | Oui | Non | Non |
| **messaging-throughput** | 5455 | | Utilisé par JMS à distance. | Oui | Oui | Non | Non |
| **mod_cluster** | | 23364 | Port multicast de communication entre l'équilibreur de charge HTTP et JBoss Enterprise Application Platform. | Oui | Non | Oui | Non |
| **osgi-http** | 8090 | | Utilisé par les composants internes qui utilisent le sous-système OSGi. | Oui | Oui | Oui | Oui |
| **remoting** | 4447 | | Utilisé pour l'invocation EJB. | Oui | Oui | Oui | Oui |

| Nom | Port | Port Multidiffusion | Description | full-ha-sockets | full-sockets | ha-socket | standard-socket |
|---|---|---|---|---|---|---|---|
| `txn-recovery-environment` | 4712 | | Gestionnaire de recouvrement des transactions JTA. | Oui | Oui | Oui | Oui |
| `txn-status-manager` | 4713 | | Gestionnaire des transactions JTA / JTS. | Oui | Oui | Oui | Oui |

**Ports de gestion**

En plus des groupes de liaisons de socket, chaque contrôleur d'hôte ouvre deux ports supplémentaires pour la gestion:

- 9990 - Le port de Console de gestion
- 9999 - Le port utilisé par la Console de gestion et par le Management API

Report a bug

### 5.2.4. Valeurs de décalage des ports pour les Groupes de liaison de sockets par défaut

Les valeurs de décalage des port (Port offsets) est un décalage chiffré qui vient s'ajouter aux valeurs de port données par le groupe de liaisons de sockets pour ce serveur. Cela permet à un seul serveur d'hériter les liaisons de sockets du groupe de serveurs auquel il appartient, avec un décalage pour veiller à ce qu'il n'entre pas en conflit avec les autres serveurs du groupe. Par exemple, si le port HTTP du groupe de liaisons de socket est 8080 et si votre serveur utilise un port offset de 100, son port HTTP sera 8180.

Report a bug

### 5.2.5. Configurer les Port Offset (valeurs de décalages de ports)

- **Configurer les Port Offset (valeurs de décalages de ports)**
  Sélectionner le Management CLI ou la Console de gestion pour configurer vos ports offsets.
  - **Configurer Port Offsets par le Management CLI**
    Sélectionner le Management CLI pour configurer vos ports offsets.
    a. **Modifier les Ports Offsets**
    Utiliser la commande **`write`** pour écrire une nouvelle valeur d'attribut de port offset. L'exemple suivant met à jour la valeur **`socket-binding-port-offset`** du *server-two* à *250*. Ce serveur est un membre du groupe d'hôte local par défaut.Vous aurez besoin d'un redémarrage pour que les changements puissent voir lieu.

```
[domain@localhost:9999 /] /host=master/server-config=server-
two/:write-attribute(name=socket-binding-port-
offset,value=250)
```

b. **Confirmer les attributs de Port Offset**
Confirmer que les valeurs ont changé en exécutant **`read-resource`** accompagné du paramètre **`include-runtime=true`** pour exposer toutes les valeurs actives en cours du modèle de serveur.

```
[domain@localhost:9999 /] /host=master/server-config=server-
two/:read-resource(include-runtime=true)
```

- **Configurer Port Offsets par la Console de Management**
Sélectionner la Console de Management pour configurer vos ports offsets.

a. **Connectez-vous à la Console de gestion.**
Connectez-vous à la Console de gestion de votre domaine géré.

b. **Sélectionner l'onglet `Hosts`**
Sélectionnez l'onglet **`Hosts`** qui se trouve en haut et à droite.

c. **Modifier les attributs de Port Offset**

i. Sélectionner le serveur dans la section **`Configuration Name`** et cliquer sur le bouton **`Edit`**.

ii. Saisir les valeurs que vous désirez dans le champ **`Port Offset`**.

iii. Cliquer sur le bouton **`Save`** pour terminer.

Report a bug

## 5.3. IPV6

### 5.3.1. Configurer les Préférences de JVM Stack d'IPv6 Networking

**Résumé**

Cette section couvre le IPv6 Networking de l'installation JBoss Enterprise Application Platform 6.

**Procédure 5.1. Désactiver la propriété IPv4 Stack Java**

1. Ouvrir le fichier qui convient à l'installation :

- **Pour le serveur autonome :**
Ouvrir ***`EAP_HOME`*`/bin/standalone.conf`**.

- **Pour un domaine géré :**
Ouvrir ***`EAP_HOME`*`/bin/domain.conf`**.

2. Modifier la propriété IPv4 Stack Java à false :

```
-Djava.net.preferIPv4Stack=false
```

-

Par exemple :

```
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
   JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -
Djava.net.preferIPv4Stack=false
   -Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.client.gcInterval=3600000
   -Dsun.rmi.dgc.server.gcInterval=3600000 -
Djava.net.preferIPv6Addresses=true"
fi
```

Report a bug

### 5.3.2. Configurer les déclarations d'interface du réseautage IPv6

**Résumé**

Suivre ces étapes de configuration de l'adresse inet d'interface dans IPv6 par défaut:

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 »
- Section 3.4.2, « Connectez-vous à la Console de management »

**Procédure 5.2. Configurer l'interface du réseautage IPv6**

1. Cliquer sur l'onglet **`Profile`** qui se trouve en haut et à droite de la console.
2. Sélectionner l'option **`Interfaces`** qui se trouve sous **`General Configuration`**.
3. Sélectionner l'interface réseau nommée à configurer.
4. Cliquer sur le bouton **`Edit`**.
5. Définir l'adresse inet à:

   ```
   ${jboss.bind.address.management:[ADDRESS]}
   ```

6. Cliquez sur le bouton **`Enregistrer`** pour enregistrer les modifications.
7. Démarrer le serveur à nouveau pour implémenter les changements.

Report a bug

### 5.3.3. Configurer les Préférences JVM Stacks des adresses IPv6

**Résumé**

Cette section couvre la configuration de l'installation JBoss Enterprise Application Platform 6 pour qu'elle favorise les adresses IPv6 à travers les fichiers de configuration.

**Procédure 5.3. Configurer l'installation JBoss Enterprise Application Platform 6 pour qu'elle favorise les adresses IPv6.**

1. Ouvrir le fichier qui convient à l'installation :
   - **Pour le serveur autonome :**
     Ouvrir ***EAP_HOME*/bin/standalone.conf**.
   - **Pour un domaine géré :**
     Ouvrir ***EAP_HOME*/bin/domain.conf**.
2. Ajouter la propriété Java suivante aux options de la Java VM

   ```
   -Djava.net.preferIPv6Addresses=true
   ```

   Par exemple :

   ```
   # Specify options to pass to the Java VM.
   #
   if [ "x$JAVA_OPTS" = "x" ]; then
      JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -
   Djava.net.preferIPv4Stack=false
      -Dorg.jboss.resolver.warning=true -
   Dsun.rmi.dgc.client.gcInterval=3600000
      -Dsun.rmi.dgc.server.gcInterval=3600000 -
   Djava.net.preferIPv6Addresses=true"
   fi
   ```

Report a bug

# CHAPITRE 6. GESTION DE SOURCES DE DONNÉES

## 6.1. INTRODUCTION

### 6.1.1. JDBC

L'API JDBC est la norme qui définit comment les bases de données sont accessibles aux applications Java. Une application configure une source de données qui référence un pilote JDBC. Le code de l'application peut alors s'inscrire au pilote, à la place de la base de données. Le pilote convertit le code dans le langage de base de données. Cela signifie que si le pilote qui convient est installé, une application pourra être utilisée sans base de données supportée.

La norme JDBC 4.0 est définie ici : http://jcp.org/en/jsr/detail?id=221.

Pour démarrer avec JDBC et avec les sources de données, voir la section sur le Pilote JDBC du Guide d'administration et de configuration de JBoss Enterprise Application Platform 6.

Report a bug

### 6.1.2. Bases de données supportées dans JBoss Enterprise Application Platform 6

Liste de bases de données JDBC conformes supportées par JBoss Enterprise Application Platform 6 : http://www.redhat.com/resourcelibrary/articles/jboss-enterprise-application-platform-supported-configurations.

Report a bug

### 6.1.3. Types de sources de données

Les deux grands types de ressources sont dénommées **`les sources de données`** et **`les sources de données XA`**.

Les sources de données non-XA sont utilisées pour les applications qui n'utilisent pas de transactions, ou les applications qui utilisent des transactions avec une base de données simple.

Les sources de données XA sont utilisées par les applications dont les transactions sont réparties à travers plusieurs bases de données. Les sources de données XA rajoutent un niveau supplémentaires.

Vous n'avez qu'à indiquer le type de source de données quand vous la créez dans la Console de gestion ou le Management CLI.

Report a bug

### 6.1.4. L'exemple de source de données

JBoss Enterprise Application Platform 6 inclut une base de données H2. C'est un système de gestion de base de données relationnelle léger qui fournit aux développeurs la possibilité de créer rapidement des applications et c'est l'exemple de source de données exemple pour la plate-forme.

**AVERTISSEMENT**

Toutefois, elle ne devrait pas être utilisée dans un environnement de production. C'est une source de données très petite, autonome qui prend en charge toutes les normes nécessaires pour le test et la création d'applications, mais qui n'est pas fiable ou suffisamment évolutive pour utilisation en production.

Pour obtenir une liste de sources de données certifiées ou prises en charges, consulter Section 6.1.2, « Bases de données supportées dans JBoss Enterprise Application Platform 6 ».

Report a bug

### 6.1.5. Déploiement des fichiers -ds.xml

Dans JBoss Enterprise Application Platform 6, les sources de données sont définies comme ressources du sous-système du serveur. Dans les versions précédentes, on avait besoin d'un fichier de configuration de source de données **`*-ds.xml`** dans le répertoire de déploiement de la configuration du serveur. Les fichiers **`*-ds.xml`** peuvent encore être déployés dans JBoss Enterprise Application Platform 6, selon le schéma suivant : http://docs.jboss.org/ironjacamar/schema/datasources_1_1.xsd.

**AVERTISSEMENT**

Cette fonctionnalité doit être utilisée pour le développement uniquement. Elle n'est pas conseillée en production car non supportée par les outils de gestion et d'admin de JBoss.

**IMPORTANT**

Il est obligatoire d'utiliser une référence à une entrée de <driver> (pilote) déjà déployé / défini quand on déploie les fichiers **`*-ds.xml`**.

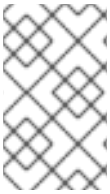
Report a bug

## 6.2. PILOTES JDBC

### 6.2.1. Installer un pilote JDBC avec la Console de gestion

**Résumé**

Avant que votre application puisse se connecter à une source de données JDBC, les pilotes JDBC de votre fournisseur de source de données doivent être installés dans un endroit où la plate-forme d'applications EAP puisse les utiliser. Le serveur d'applications JBoss Enterprise vous permet de déployer ces pilotes tout comme tout autre déploiement. Cela signifie que, si vous utilisez un domaine géré, vous pourrez les déployer sur plusieurs serveurs dans un groupe de serveurs.



**NOTE**

La meilleure méthode d'installation des pilotes JDBC est de les installer comme module principal. Pour installer le pilote JDBC comme module principal, voir: Section 6.2.2, « Installer un Pilote JDBC comme Core Module ».

**Prérequis**

Vous devrez remplir les conditions suivantes avant de pouvoir effectuer cette tâche :

- Télécharger le pilote JDBC de votre fournisseur de base de données.

**Procédure 6.1. Déployer le pilote JDBC**

1. **Accéder à la Console de gestion**
   Section 3.4.2, « Connectez-vous à la Console de management »
2. **Déployez le fichier JAR dans votre serveur ou groupe de serveurs**
   Si vous utilisez un domaine géré, déployez le fichier JAR dans un groupe de serveurs. Sinon, déployez-le sur votre serveur. Voir Section 9.2.2, « Déployer une application par la Console de gestion ».

**Résultat :**

Le pilote JDBC est déployé, et est disponible pour vos applications.

Report a bug

### 6.2.2. Installer un Pilote JDBC comme Core Module

**Prérequis**

Vous devrez remplir les conditions suivantes avant de pouvoir effectuer cette tâche :

- Télécharger le pilote JDBC de votre base de données fournisseur. Voici les adresses de téléchargement pour le pilote JDBC : Section 6.2.3, « Adresses des téléchargements de pilotes JDBC ».
- En extraire l'archive

**Procédure 6.2. Installer un Pilote JDBC comme Core Module**

1. Créer une structure de chemin d'accès de fichier sous le répertoire ***EAP_HOME*/modules/**. Ainsi, pour un pilote MySQL JDBC, créer une structure de répertoire comme suit : ***EAP_HOME*/modules/com/mysql/main/**.
2. Copier le pilote JDBC dans le sous-répertoire **main/**.
3. Dans le sous-répertoire **main/**, créer un fichier **module.xml** qui ressemble à l'exemple suivant :

**Exemple 6.1. Exemple de fichier module.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">
  <resources>
```

```
    <resource-root path="mysql-connector-java-5.1.15.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Le nom du module, **com.mysql**, doit correspondre à la structure du répertoire du module.

4. Démarrer le serveur.
5. Démarrer le Management CLI.
6. Exécuter la commande CLI suivante pour ajouter le module de pilote JDBC comme pilote :

```
/subsystem=datasources/jdbc-driver=DRIVER_NAME:add(driver-
name=DRIVER_NAME,driver-module-name=MODULE_NAME,driver-xa-
datasource-class-name=XA_DATASOURCE_CLASS_NAME)
```

**Exemple 6.2. Exemple de Commande CLI**

```
/subsystem=datasources/jdbc-driver=mysql:add(driver-
name=mysql,driver-module-name=com.mysql,driver-xa-datasource-
class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)
```

**Résultat**

Le pilote JDBC est maintenant installé et configuré comme Core Module. Il est maintenant prêt à être référencé par les sources de données d'application.

Report a bug

### 6.2.3. Adresses des téléchargements de pilotes JDBC

Le tableau suivant donne les adresses de téléchargement standard pour les pilotes JDBC de bases de données communes à la plate-forme JBoss EAP. Ces liens pointent vers des sites de tiers qui ne sont pas contrôlés, ni surveillés activement par Red Hat. Pour les pilotes les plus à jour de votre base de données, consultez le site Web et la documentation du fournisseur de votre base de données

**Tableau 6.1. Adresses des téléchargements du pilote JDBC**

| Fournisseur | Adresse de téléchargement |
|---|---|
| MySQL | http://www.mysql.com/products/connector/ |
| PostgreSQL | http://jdbc.postgresql.org/ |
| Oracle | http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html |

| Fournisseur | Adresse de téléchargement |
|---|---|
| IBM | http://www-306.ibm.com/software/data/db2/java/ |
| Sybase | http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect |
| Microsoft | http://msdn.microsoft.com/data/jdbc/ |

Report a bug

### 6.2.4. Accès aux Classes Spécifique du fournisseur

**Résumé**

Cette section couvre les étapes à suivre pour utiliser les classes spécifiques JDBC. Cela est utile quand une application a besoin d'utiliser une fonctionnalité spécifique à un fournisseur ne faisant pas partie de l'API JDBC.

**AVERTISSEMENT**

Ceci est une procédure d'utilisation avancée. Seules les applications qui ont besoin d'une fonctionnalité que l'on ne peut pas trouver dans l'API JDBC doivent implémenter cette procédure.

**IMPORTANT**

Ce processus est requis pour le mécanisme de ré-authentification et pour accéder aux classe spécifiques au fournisseur.

**IMPORTANT**

Suivre les directives de l'API spécifique au fournisseur de près, car la connexion est contrôlée par le conteneur IronJacamar.

**Prérequis**

- Section 6.2.2, « Installer un Pilote JDBC comme Core Module ».

**Procédure 6.3. Ajouter une dépendance à l'application**

- **Configurer le fichier `MANIFEST.MF`**
  a. Ouvrir le fichier **`META-INF/MANIFEST.MF`** de l'application dans un éditeur de texte.
  b. Ajouter une dépendance au module JDBC et sauvegarder le fichier.

```
Dépendences : MODULE_NAME
```

**Exemple 6.3. Exemple de dépendance**

```
Dépendences: com.mysql
```

- a. **Créer un fichier `jboss-deployment-structure.xml`**
  Créer un fichier **`jboss-deployment-structure.xml`** dans le dossier **`META-INF/`** ou **`WEB-INF`** de l'application.

**Exemple 6.4. Exemple de fichier `jboss-deployment-structure.xml`**

```
<jboss-deployment-structure>
  <deployment>
    <dependencies>
      <module name="com.mysql" />
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

**Exemple 6.5. Accède à l'API spécifique du fournisseur**

L'exemple ci-dessous accède à l'API MySQL.

```
import java.sql.Connection;
import org.jboss.jca.adapters.jdbc.WrappedConnection;

  Connection c = ds.getConnection();
  WrappedConnection wc = (WrappedConnection)c;
  com.mysql.jdbc.Connection mc = wc.getUnderlyingConnection();
```

Report a bug

## 6.3. NON-XA DATASOURCES

### 6.3.1. Créer une source de données Non-XA avec les Interfaces de gestion

**Résumé**

Cette section explique les étapes à suivre pour créer une source de données non-XA, en utilisant la Console de gestion ou le Management CLI.

**Prérequis**

- Le serveur JBoss Enterprise Application Platform 6 doit être en cours d'exécution.

**NOTE**

Avant la version 10.2 de la source de données Oracle, le paramètre <no-tx-separate-pools/> était requis, car le mélange de connexions transactionnelles et non-transactionnelles auraient créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

**Procédure 6.4. Créer une source de données en utilisant le Management CLI ou la Console de gestion**

- **Management CLI**
  a. Lancer l'outil CLI et connectez-vous à votre serveur.
  b. Exécuter la commande suivante pour créer une source de données non-XA, et configurer les variables comme il se doit :

```
data-source add --name=DATASOURCE_NAME --jndi-name=JNDI_NAME --driver-name=DRIVER_NAME  --connection-url=CONNECTION_URL
```

  c. Activer la source de données :

```
data-source enable --name=DATASOURCE_NAME
```

- **Console de gestion**
  a. Connectez-vous à la Console de gestion.
  b. **Naviguez dans le panneau `Datasources` qui se trouve dans la Console de gestion**
    i. **Mode autonome**
    Sélectionnez l'onglet **`Profil`** qui se trouve en haut et à droite de la console.
    **Mode Domaine**
    A. Sélectionnez l'onglet **`Profiles`** qui se trouve en haut et à droite de la console.
    B. Sélectionner le profil qui convient à partir du menu déroulant en haut à gauche.
    C. Étendre le menu **Subsystems** qui se trouve à gauche de la console.
    ii. Sélectionner **Connector** → **Datasources** à partir du menu à gauche de la console.

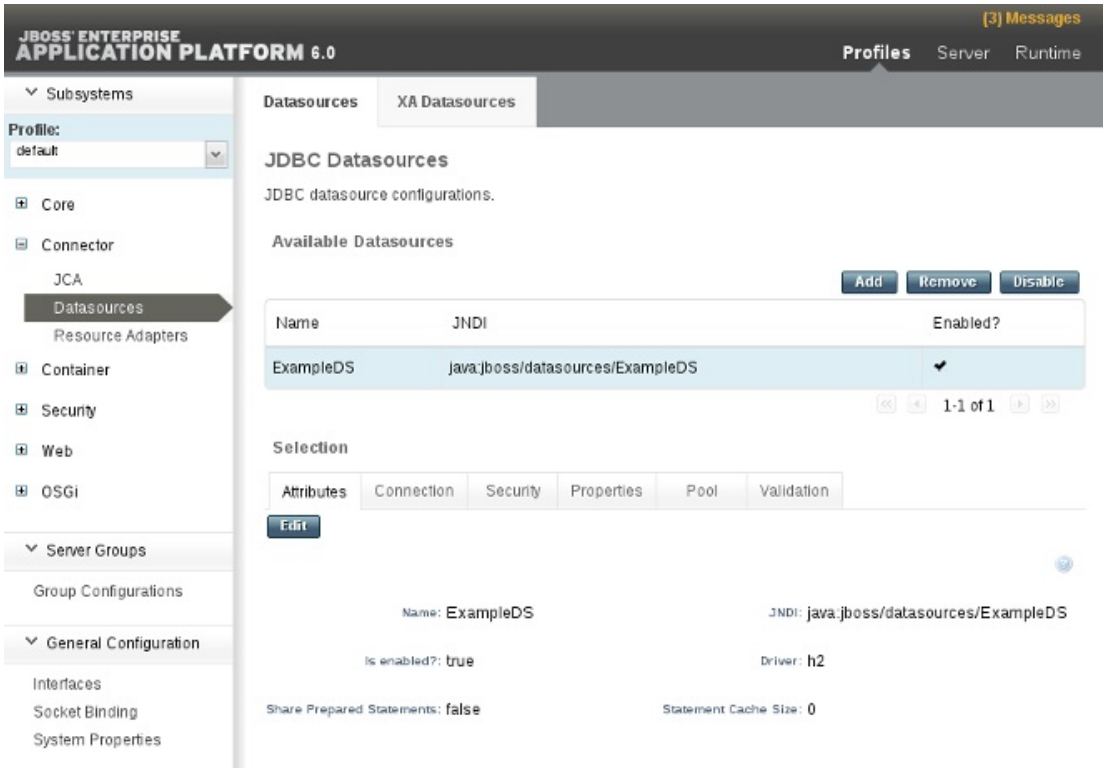



**Figure 6.1. Panneau de sources de données**

c. **Créer une nouvelle source de données**

i. Sélectionner le bouton **Add** qui se trouve en haut du panneau **`Datasources`**.

ii. Saisir les attributs de la nouvelle source de données de l'assistant **`Create Datasource`** et appuyez sur **`Next`**.

iii. Saisir les informations sur le pilote JDBC dans l'assistant **`Create Datasource`** et appuyez sur **`Next`**.

iv. Saisir les paramètres de connexion dans l'assistant **`Create Datasource`** et appuyez sur **`Done`**.

**Résultat**

La source de données non-Xa a été ajoutée au serveur. Elle est maintenant visible dans le fichier **`standalone.xml`** ou le fichier **`domain.xml`**, ainsi que dans les interfaces de gestion.

Report a bug

## 6.3.2. Modifier une source de données Non-XA avec les Interfaces de gestion

**Résumé**

Cette section explique les étapes à suivre pour modifier une source de données non-XA, en utilisant la Console de gestion ou le Management CLI.

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 ».

**NOTE**

Les sources de données non-XA peuvent être intégrées avec les transactions JTA. Pour intégrer la source de données dans JTA, veillez à ce que le paramètre **`jta`** soit défini à **`true`**.

**Procédure 6.5. Modifier une Source de données non-XA**

- **Management CLI**
  a. Section 3.5.2, « Lancement du Management CLI ».
  b. Utiliser la commande **`write-attribute`** pour configurer un attribut de source de données :

```
/subsystem=datasources/data-source=DATASOURCE_NAME:write-attribute(name=ATTRIBUTE_NAME,value=ATTRIBUTE_VALUE)
```

  c. Charger à nouveau le serveur pour confirmer les changements :

```
:reload
```

- **Console de gestion**
  a. Section 3.4.2, « Connectez-vous à la Console de management ».
  b. **Naviguez dans le panneau `Datasources` qui se trouve dans la Console de gestion**
    i. - **Mode autonome**
       Sélectionnez l'onglet **`Profil`** qui se trouve en haut et à droite de la console.
       - **Mode Domaine**
         A. Sélectionnez l'onglet **`Profiles`** qui se trouve en haut et à droite de la console.
         B. Sélectionner le profil qui convient à partir du menu déroulant en haut à gauche.
         C. Étendre le menu **Subsystems** qui se trouve à gauche de la console.
    ii. Sélectionner **Connector** → **Datasources** à partir du menu à gauche de la console.

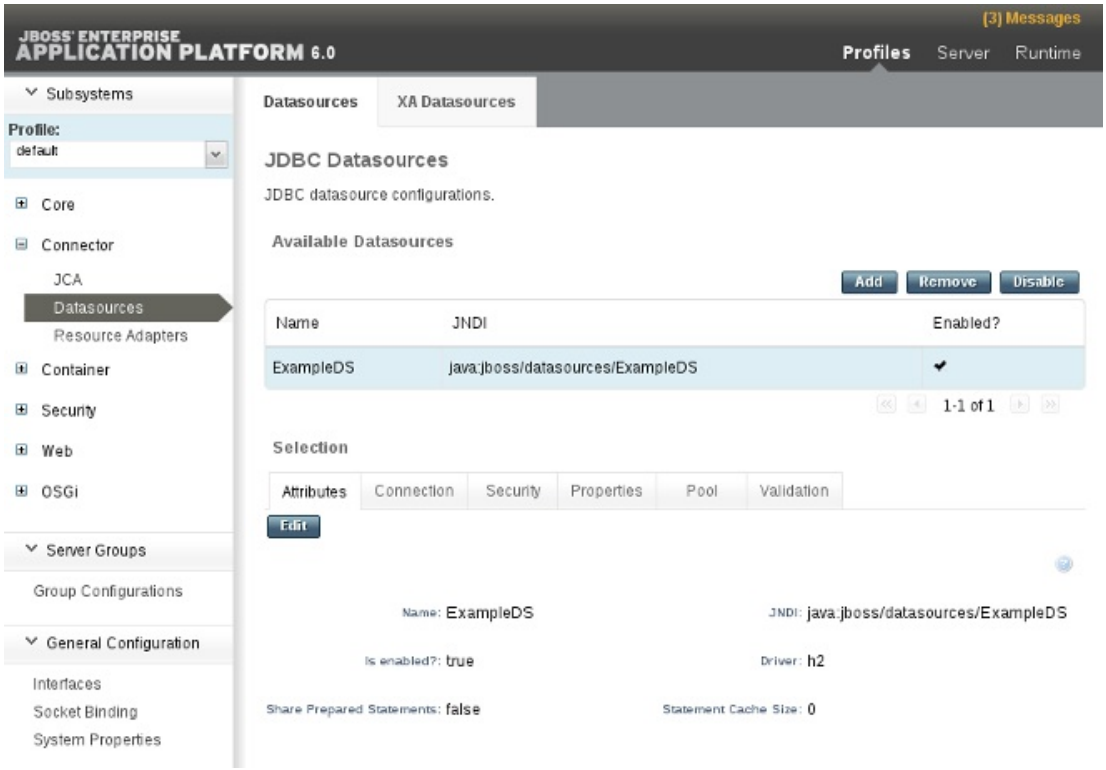



**Figure 6.2. Panneau de sources de données**

c. **Modifier la source de données**

i. Sélectionner la source de données qui convient à partir de la liste **`Available Datasources`**. Les attributs de la source de données sont affichés dans le panneau **`Attributes`** ci-dessous.

ii. Sélectionner le bouton **`Edit`** pour modifier les attributs de la source de données.

iii. Modifier les attributs de la source de données et sélectionner le bouton **`Save`** quand c'est fait.

**Résultat**

La source de données non-Xa a été configurée. Elle est maintenant visible dans le fichier **`standalone.xml`** ou le fichier **`domain.xml`**, ainsi que dans les interfaces de gestion.

- Pour créer une nouvelle source de données. voir : Section 6.3.1, « Créer une source de données Non-XA avec les Interfaces de gestion ».
- Pour supprimer la source de données, voir Section 6.3.3, « Supprimer une source de données Non-XA avec les Interfaces de gestion ».

Report a bug

### 6.3.3. Supprimer une source de données Non-XA avec les Interfaces de gestion

**Résumé**

Cette section couvre les étapes à suivre pour supprimer une source de données XA de JBoss Enterprise Application Platform 6, par la Console de gestion ou le Management CLI.

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 ».

**Procédure 6.6. Supprimer un source de données non-XA**

- **Management CLI**
  a. Section 3.5.2, « Lancement du Management CLI ».
  b. Exécuter la commande suivante pour supprimer un source de données non-XA :

```
data-source remove --name=DATASOURCE_NAME
```

- **Console de gestion**
  a. Section 3.4.2, « Connectez-vous à la Console de management ».
  b. **Naviguez dans le panneau `Datasources` qui se trouve dans la Console de gestion**
     i. - **Mode autonome**
        Sélectionnez l'onglet **`Profil`** qui se trouve en haut et à droite de la console.
        - **Mode Domaine**
          A. Sélectionnez l'onglet **`Profiles`** qui se trouve en haut et à droite de la console.
          B. Sélectionner le profil qui convient à partir du menu déroulant en haut à gauche.
          C. Étendre le menu **Subsystems** qui se trouve à gauche de la console.
     ii. Sélectionner **Connector** → **Datasources** à partir du menu à gauche de la console.

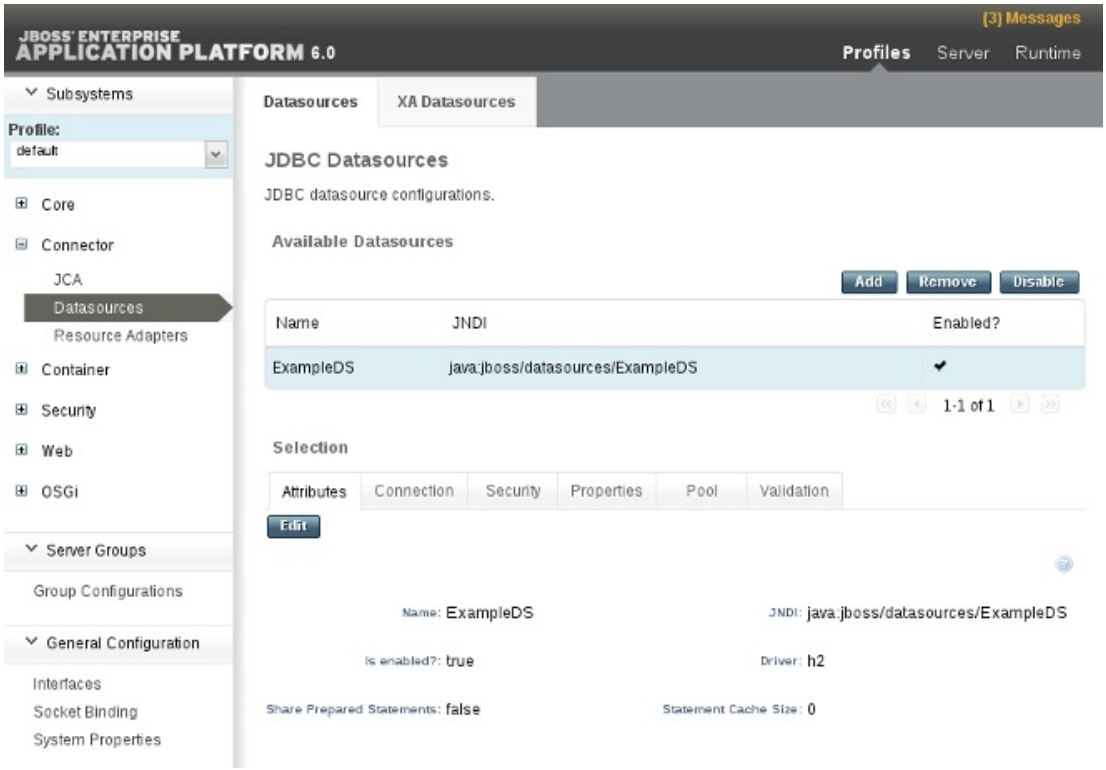



**Figure 6.3. Panneau de sources de données**

c. Sélectionner la source de données enregistrée à supprimer, et cliquer sur le bouton **`Remove`** qui se trouve en haut et à droite de la console.

**Résultat**

La source de données non-XA a été supprimée dans le serveur.

- Pour créer une nouvelle source de données. voir : Section 6.3.1, « Créer une source de données Non-XA avec les Interfaces de gestion ».

Report a bug

## 6.4. XA DATASOURCES

### 6.4.1. Créer une source de données XA avec les Interfaces de gestion

**Résumé**

Cette section explique les étapes à suivre pour créer une source de données XA, en utilisant la Console de gestion ou le Management CLI.

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 ».

**NOTE**

Avant la version 10.2 de la source de données Oracle, le paramètre <no-tx-separate-pools/> était requis, car le mélange de connexions transactionnelles et non-transactionnelles auraient créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

**Procédure 6.7. Créer une source de données XA en utilisant le Management CLI ou la Console de gestion**

- **Management CLI**

  a. Section 3.5.2, « Lancement du Management CLI ».

  b. Exécuter la commande suivante pour créer une source de données XA, et configurer les variables comme il se doit :

  ```
  xa-data-source add --name=XA_DATASOURCE_NAME --jndi-
  name=JNDI_NAME --driver-name=DRIVER_NAME --xa-datasource-
  class=XA_DATASOURCE_CLASS
  ```

  c. **Configurer les propriétés de la source de données XA**

  i. **Définir le nom du serveur**
  Exécuter la commande suivante pour configurer le nom du serveur de l'hôte :

  ```
  /subsystem=datasources/xa-data-
  source=XA_DATASOURCE_NAME/xa-datasource-
  properties=ServerName:add(value=HOSTNAME)
  ```

  ii. **Définir le nom de la base de données**
  Exécuter la commande suivante pour configurer le nom de la base de données :

  ```
  /subsystem=datasources/xa-data-
  source=XA_DATASOURCE_NAME/xa-datasource-
  properties=DatabaseName:add(value=DATABASE_NAME)
  ```

  d. Activer la source de données :

  ```
  xa-data-source enable --name=XA_DATASOURCE_NAME
  ```

- **Console de gestion**

  a. Section 3.4.2, « Connectez-vous à la Console de management ».

  b. **Naviguez dans le panneau `Datasources` qui se trouve dans la Console de gestion**

  i. - **Mode autonome**
  Sélectionnez l'onglet **`Profil`** qui se trouve en haut et à droite de la console.

     - **Mode Domaine**

       A. Sélectionnez l'onglet **`Profiles`** qui se trouve en haut et à droite de la console.

       B. Sélectionner le profil qui convient à partir du menu déroulant en haut à gauche.

       C. Étendre le menu **Subsystems** qui se trouve à gauche de la console.

  ii. Sélectionner **Connector** → **Datasources** à partir du menu à gauche de la console.

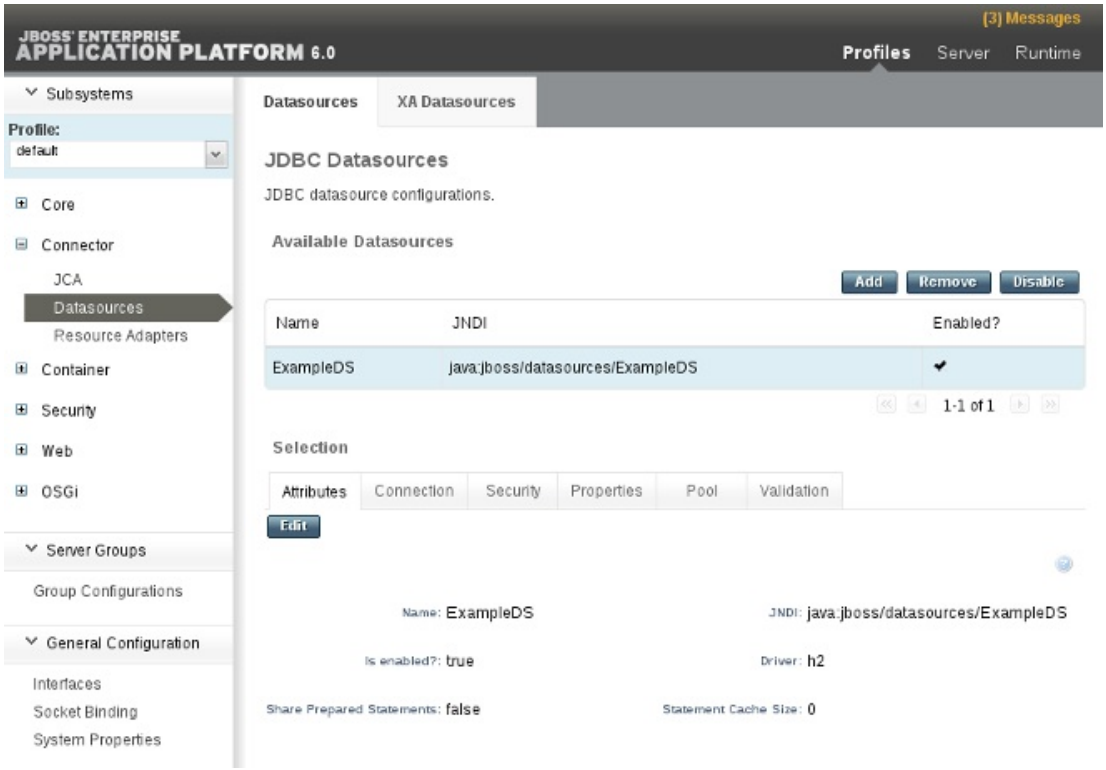



**Figure 6.4. Panneau de sources de données**

c. Sélectionner le panneau **`XA Datasource`**.

d. **Créer une nouvelle source de données XA**

i. Sélectionner le bouton **Add** qui se trouve en haut du panneau **`Datasources`**.

ii. Saisir les attributs de la nouvelle source de données XA de l'assistant **`Create XA Datasource`** et appuyez sur **`Next`**.

iii. Saisir les informations sur le pilote JDBC dans l'assistant **`Create XA Datasource`** et appuyez sur **`Next`**.

iv. Modifier les propriétés et appuyez sur **`Next`**.

v. Saisir les paramètres de connexion dans l'assistant **`Create XA Datasource`** et appuyez sur **`Done`**.

**Résultat**

La source de données XA a été ajoutée au serveur. Elle est maintenant visible dans le fichier **`standalone.xml`** ou le fichier **`domain.xml`**, ainsi que dans les interfaces de gestion.

- Pour configurer la source de données davantage, voir : Section 6.4.2, « Modifier une base de données XA par les interfaces de gestion ».
- Pour supprimer la source de données, voir Section 6.4.3, « Supprimer une source de données XA avec les Interfaces de gestion ».

Report a bug

### 6.4.2. Modifier une base de données XA par les interfaces de gestion

**Résumé**

Cette section explique les étapes à suivre pour modifier une source de données XA, en utilisant la Console de gestion ou le Management CLI.

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 ».

**Procédure 6.8. Modifier une source de données XA en utilisant le Management CLI ou la Console de gestion**

- **Management CLI**
  a. Section 3.5.2, « Lancement du Management CLI ».
  b. **Configurer les attributs de source de données XA**
  Utiliser la commande **`write-attribute`** pour configurer un attribut de source de données :

```
/subsystem=datasources/xa-data-source=XA_DATASOURCE_NAME:write-attribute(name=ATTRIBUTE_NAME,value=ATTRIBUTE_VALUE)
```

  c. **Configurer les propriétés de la source de données XA**
  Exécuter la commande suivante pour configurer une sous-ressource de source de données XA :

```
/subsystem=datasources/xa-data-source=DATASOURCE_NAME/xa-datasource-properties=PROPERTY_NAME:add(value=PROPERTY_VALUE)
```

  d. Charger à nouveau le serveur pour confirmer les changements :

```
:reload
```

- **Console de gestion**
  a. Section 3.4.2, « Connectez-vous à la Console de management ».
  b. **Naviguez dans le panneau `Datasources` qui se trouve dans la Console de gestion**
    i. ■ **Mode autonome**
    Sélectionnez l'onglet **`Profil`** qui se trouve en haut et à droite de la console.
    ■ **Mode Domaine**
    A. Sélectionnez l'onglet **`Profiles`** qui se trouve en haut et à droite de la console.
    B. Sélectionner le profil qui convient à partir du menu déroulant en haut à gauche.
    C. Étendre le menu **Subsystems** qui se trouve à gauche de la console.

ii. Sélectionner **Connector** → **Datasources** à partir du menu à gauche de la console.

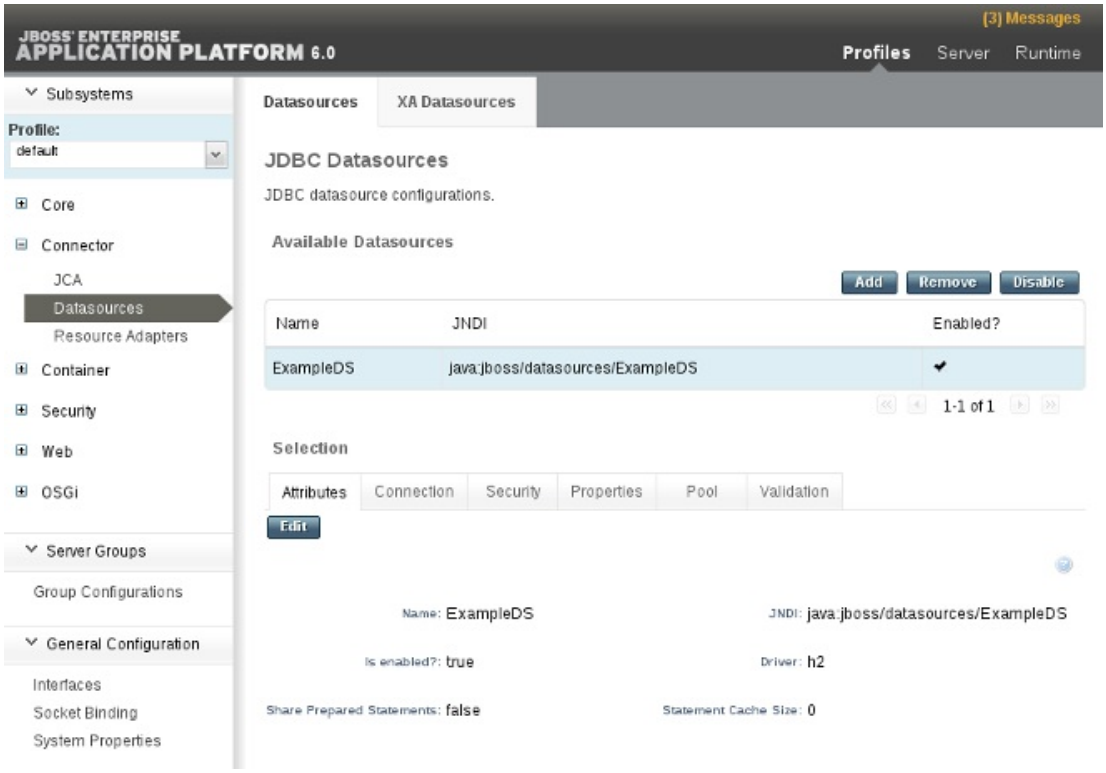



**Figure 6.5. Panneau de sources de données**

c. Sélectionner le panneau **`XA Datasource`**.

d. **Modifier la source de données**

i. Sélectionner la source de données XA qui convient à partir de la liste **`Available XA Datasources`**. Les attributs de la source de données XA sont affichés dans le panneau **`Attributes`** ci-dessous.

ii. Sélectionner le bouton **`Edit`** pour modifier les attributs de la source de données.

iii. Modifier les attributs de la source de données XA et sélectionner le bouton **`Save`** quand c'est fait.

**Résultat**

La source de données XA a été configurée. Les changements sont maintenant visibles dans le fichier **`standalone.xml`** ou le fichier **`domain.xml`**, ainsi que dans les interfaces de gestion.

- Pour créer une nouvelle source de données, voir : Section 6.4.1, « Créer une source de données XA avec les Interfaces de gestion ».
- Pour supprimer la source de données, voir Section 6.4.3, « Supprimer une source de données XA avec les Interfaces de gestion ».

Report a bug

### 6.4.3. Supprimer une source de données XA avec les Interfaces de gestion

**Résumé**

Cette section couvre les étapes à suivre pour supprimer une source de données XA de JBoss Enterprise Application Platform 6, par la Console de gestion ou le Management CLI.

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 ».

**Procédure 6.9. Supprimer une source de données XA en utilisant le Management CLI ou la Console de gestion**

- **Management CLI**
  a. Section 3.5.2, « Lancement du Management CLI ».
  b. Exécuter la commande suivante pour supprimer une source de données:

```
xa-data-source remove --name=XA_DATASOURCE_NAME
```

- **Console de gestion**
  a. Section 3.4.2, « Connectez-vous à la Console de management ».
  b. **Naviguez dans le panneau `Datasources` qui se trouve dans la Console de gestion**
     i. - **Mode autonome**
        Sélectionnez l'onglet **`Profil`** qui se trouve en haut et à droite de la console.
        - **Mode Domaine**
          A. Sélectionnez l'onglet **`Profiles`** qui se trouve en haut et à droite de la console.
          B. Sélectionner le profil qui convient à partir du menu déroulant en haut à gauche.
          C. Étendre le menu **Subsystems** qui se trouve à gauche de la console.
     ii. Sélectionner **Connector → Datasources** à partir du menu à gauche de la console.

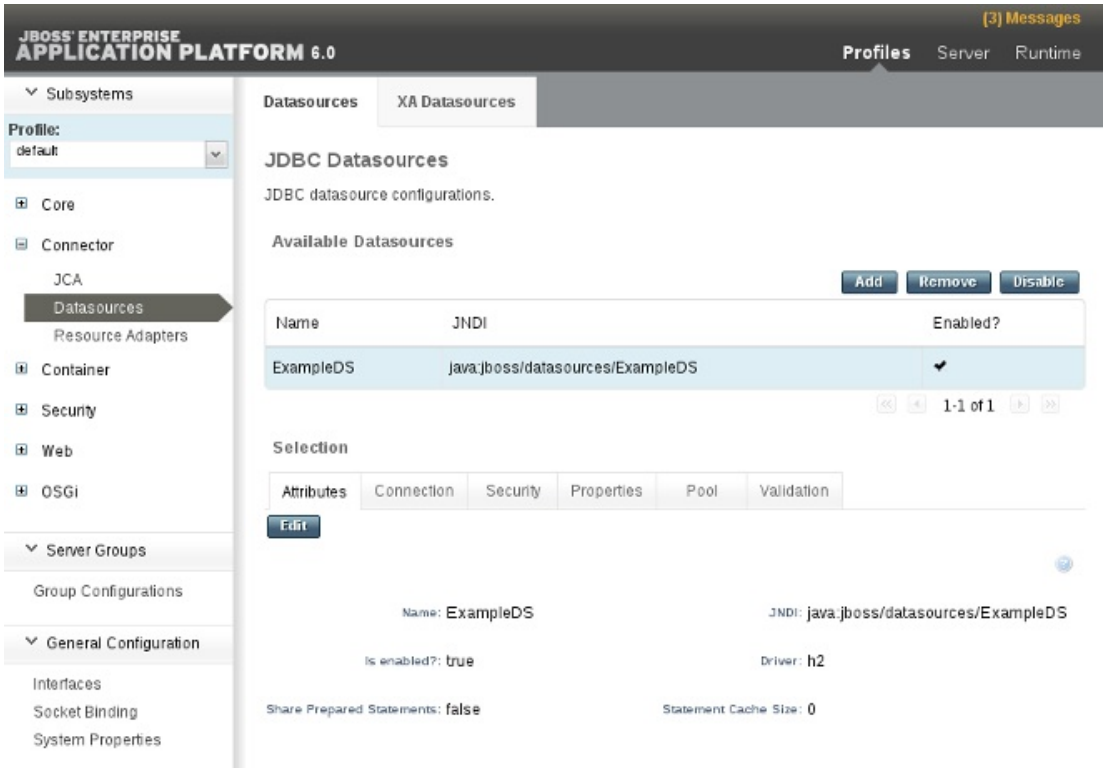



**Figure 6.6. Panneau de sources de données**

c. Sélectionner le panneau **`XA Datasource`**.

d. Sélectionner la source de données XA enregistrée à supprimer, et cliquer sur le bouton **`Remove`** qui se trouve en haut et à droite de la console.

**Résultat**

La source de données XA a été supprimée dans le serveur.

- Pour ajouter une nouvelle source de données, voir : Section 6.4.1, « Créer une source de données XA avec les Interfaces de gestion ».

Report a bug

## 6.4.4. XA Recovery

### 6.4.4.1. Les modules de recouvrement XA

Chaque ressource XA a besoin d'un module de recouvrement associé avec sa configuration. Le module de recouvrement doit étendre la classe **`com.arjuna.ats.jta.recovery.XAResourceRecovery`**.

JBoss Enterprise Application Platform fournit des modules de recouvrement pour les ressources JDBC et JMS XA. Pour ces types de ressources, les modules de recouvrement sont automatiquement enregistrés. Si vous devez utiliser un module personnalisé, vous pourrez l'enregistrer dans votre source de données.

Report a bug

### 6.4.4.2. Configurer les modules de recouvrement

Pour la plupart des ressources JDBC et JMS, le module de recouvrement est automatiquement associé à la ressource. Dans de tels cas, vous aurez uniquement besoin de configurer les options qui permettent au module de recouvrement de se connecter à vos ressources afin de procéder au recouvrement.

Pour les ressources personnalisées qui ne sont ni JDBC, ni JMS, contacter Red Hat Global Services pour obtenir des informations sur les configurations qui sont prises en charge.

Chacun de ces attributs de configuration peut être défini lors de la création de la source de données, ou par la suite. Vous pouvez les définir en utilisant la Console de gestion basée web ou le Management CLI par ligne de commande. Se référer à Section 6.4.1, « Créer une source de données XA avec les Interfaces de gestion » and Section 6.4.2, « Modifier une base de données XA par les interfaces de gestion » pour obtenir des informations sur la façon de configurer des sources de données XA.

Voir les tableaux suivants pour les attributs de configuration de sources de données, et pour obtenir des informations sur la configuration spécifique à certains fournisseurs de bases de données.

**Tableau 6.2. Attributs de configuration générale**

| Attribut | Description |
|---|---|
| recovery-username | Le nom d'utilisateur qui doit être utilisé par le module de recouvrement pour se connecter à la ressource de recouvrement. |
| recovery-password | Le mot de passe qui doit être utilisé par le module de recouvrement pour se connecter à la ressource de recouvrement. |
| recovery-security-domain | Le domaine de sécurité qui doit être utilisé par le module de recouvrement pour se connecter à la ressource de recouvrement. |
| recovery-plugin-class-name | i vous devez utiliser un module de recouvrement personnalisé, définissez cet attribut au nom complet de classe du module. Le module doit étendre la classe **`com.arjuna.ats.jta.recovery.XAResourceRecovery`**. |
| recovery-plugin-properties | Si vous utilisez un module de récupération personnalisée qui requiert des propriétés à définir, définissez cet attribut à la liste de paires *key=value* séparée par des virgules pour les propriétés. |

### Informations de configuration spécifiques au fournisseur

#### Oracle

Si la source de données Oracle n'est pas configurée correctement, vous apercevrez sans doute les erreurs suivantes dans votre sortie de journalisation :

```
WARN  [com.arjuna.ats.jta.logging.loggerI18N]
[com.arjuna.ats.internal.jta.recovery.xarecovery1] Local
XARecoveryModule.xaRecovery  got XA exception
javax.transaction.xa.XAException, XAException.XAER_RMERR
```

Pour résoudre cette erreur, veillez à ce que l'utilisateur Oracle configuré dans ***`recovery-username`*** ait bien accès aux tableaux utiles au recouvrement. L'énoncé SQL suivant affiche les attributions d'instances correctes Oracle 11g ou Oracle 10g R2 corrigées pour le bogue 5945463 d'Oracle.

```
GRANT SELECT ON sys.dba_pending_transactions TO recovery-username;
```

```
GRANT SELECT ON sys.pending_trans$ TO recovery-username;
GRANT SELECT ON sys.dba_2pc_pending TO recovery-username;
GRANT EXECUTE ON sys.dbms_xa TO recovery-username;
```

Si vous utilisez une version Oracle 11g antérieure à 11g, modifier l'énoncé final **EXECUTE** ainsi:

```
GRANT EXECUTE ON sys.dbms_system TO recovery-username;
```

**PostgreSQL**

Voir la documentation PostgreSQL pour obtenir des instructions sur la façon d'activer des transactions (ex. XA) préparées. La version 8.4-701 du pilote JDBC de PostgreSQL a un bogue dans **`org.postgresql.xa.PGXAConnection`** qui empêche le recouvrement dans certaines situations. Cela a été résolu dans les nouvelles versions.

**MySQL**

Basé sur http://bugs.mysql.com/bug.php?id=12161, le recouvrement de transactions XA ne fonctionnait pas dans certaines versions de MySQL 5. Cela a été corrigé dans MySQL 6.1. VOir l'URL du bogue ou la documentation MySQL pour obtenir davantage d'informations.

**IBM DB2**

IBM DB2 s'attend à ce que la méthode **`XAResource.recover`** soit appelée uniquement pendant la phase de resynchronisation désignée qui se produit lorsque le serveur d'applications est redémarré après un accident ou une panne. Il s'agit d'une décision de conception pour l'implémentation de DB2, qui est en dehors du dessein de cette documentation.

Report a bug

## 6.5. SÉCURITÉ DES BASES DE DONNÉES

### 6.5.1. Sécurité des bases de données

La meilleure solution de sécuriser les bases de données est soit l'utilisation des domaines de sécurité, soit les mots de passe. Vous trouverez un exemple de chacun ci-dessous. Pour plus d'informations, consulter :

- Domaines de sécurité : Section 10.6.1, « Les domaines de sécurité ».
- Mots de passe : Section 10.12.1, « Sécurisation des chaînes sensibles des fichiers en texte clair ».

**Exemple 6.6. Exemple de domaine de sécurité**

```
<security>
   <security-domain>mySecurityDomain</security-domain>
</security>
```

**Exemple 6.7. Exemple de mots de passe**

```
<security>
```

```
    <user-name>admin</user-name>

<password>${VAULT::ds_ExampleDS::password::N2NhZDYzOTMtNWE0OS00ZGQ0LWE4M
mEtMWNlMDMyNDdmNmI2TElORV9CUkVBS3ZhdWx0}</password>
</security>
```

Report a bug

## 6.6. CONFIGURATION DES SOURCES DE DONNÉES

### 6.6.1. Paramètres de source de données

**Tableau 6.3. Les paramètres de source de données communs aux sources XA ou non-XA**

| Paramètre | Description |
|---|---|
| jndi-name | Le nom JNDI unique pour la source de données. |
| pool-name | Le nom du pool de gestion de la source de données. |
| activé | Indique si la source de données est activée. |
| use-java-context | Indique si on doit relier la source de données au JNDI global. |
| spy | Activer la fonctionnalité **`spy`** sur la couche JDBC. Cela journalisera tout le trafic JDBC dans la source de données. Le paramètre **`logging-category`** doit également être défini à **`org.jboss.jdbc`**. |
| use-ccm | Activer le gestionnaire de connexion cache. |
| new-connection-sql | Un énoncé SQL qui exécute quand la connexion est ajoutée au pool de connexion. |
| transaction-isolation | Un parmi :<br>• TRANSACTION_READ_UNCOMMITTED<br>• TRANSACTION_READ_COMMITTED<br>• TRANSACTION_REPEATABLE_READ<br>• TRANSACTION_SERIALIZABLE<br>• TRANSACTION_NONE |
| url-delimiter | Le délimiteur d'URLs d'une connexion url pour les bases de données clusterisées HA (Haute disponibilité). |

| Paramètre | Description |
| --- | --- |
| url-selector-strategy-class-name | Une classe qui implémente l'interface **`org.jboss.jca.adapters.jdbc.URLSelectorStrategy`**. |
| sécurité | Contient les éléments enfant qui représentent les paramètres de sécurité. Consulter Tableau 6.8, « Paramètres de sécurité ». |
| validation | Contient les éléments enfant qui représentent les paramètres de validation. Consulter Tableau 6.9, « Paramètres de validation ». |
| timeout | Contient les éléments enfant qui représentent les paramètres de timeout. Consulter Tableau 6.10, « Paramètre de timeout ». |
| énoncé | Contient les éléments enfant qui représentent les paramètres d'énoncé. Consulter Tableau 6.11, « Paramètres d'instruction ». |

**Tableau 6.4. Paramètres de source de données non-xa**

| Paramètre | Description |
| --- | --- |
| jta | Active l'intégration JTA pour les sources de données non-XA. Ne s'applique pas aux sources de données XA. |
| connection-url | L'URL de connexion du pilote JDBC. |
| driver-class | Le nom complet de la classe de pilote JDBC. |
| connection-property | Propriétés de connexion arbitraires passées à la méthode **`Driver.connect(url,props)`**. Chaque connection-property indique une paire name/value. Le nom de la propriété provient du nom, et la valeur provient du contenu de l'élément. |
| pool | Contient les éléments enfant qui représentent les paramètres de pooling. Consulter Tableau 6.6, « Les paramètres de pool communs aux sources XA ou non-XA ». |

**Tableau 6.5. Paramètres de source de données XA**

| Paramètre | Description |
| --- | --- |

| Paramètre | Description |
| --- | --- |
| xa-datasource-property | Une propriété pour assigner la classe d'implémentation **`XADataSource`**. Spécifié par *name=value*. Si une méthode setter existe, dans le format **`set`*`Name`***, la propriété sera définie en appelant une méthode setter sous le format **`set`*`Name`*`(`*`value`*`)`**. |
| xa-datasource-class | Le nom complet de la classe d'implémentation de **`javax.sql.XADataSource`**. |
| pilote | Unique référence au module de chargeur de classe qui contient le pilote JDBC. Le format accepté est *driverName#majorVersion.minorVersion*. |
| xa-pool | Contient des éléments enfant qui représentent les paramètres de pooling. Consulter Tableau 6.6, « Les paramètres de pool communs aux sources XA ou non-XA » et Tableau 6.7, « Paramètres du pool XA ». |
| recouvrement | Contient des éléments enfant qui représentent les paramètres de recouvrement. Consulter Tableau 6.12, « Paramètres de recouvrement ». |

**Tableau 6.6. Les paramètres de pool communs aux sources XA ou non-XA**

| Paramètre | Description |
| --- | --- |
| min-pool-size | Le nombre minimum de connexions contenues par un pool. |
| max-pool-size | Le nombre maximum de connexions qu'un pool peut contenir |
| Pré-remplissage | Indique si l'on doit essayer de pré-remplir un pool de connexion. Un élément vide indique une valeur **`true`**. La valeur par défaut est **`false`**. |
| use-strict-min | Indique si la taille du pool est stricte. **`false`** par défaut. |
| flush-strategy | Indique si le pool doit être vidé en cas d'erreur. Les valeurs valides sont :<br>• FailingConnectionOnly<br>• IdleConnections<br>• EntirePool<br>La valeur par défaut est **`FailingConnectionOnly`**. |

| Paramètre | Description |
| --- | --- |
| allow-multiple-users | Indique si plusieurs utilisateurs pourront accéder à la source de données à travers la méthode getConnection (utilisateur, mot de passe), et si le type de pool interne devrait expliquer ce comportement. |

**Tableau 6.7. Paramètres du pool XA**

| Paramètre | Description |
| --- | --- |
| is-same-rm-override | Indique si la classe **`javax.transaction.xa.XAResource.isSameRM(XAResource)`** retourne **`true`** ou **`false`**. |
| entrelacement | Indique si on doit activer l'entrelacement pour les fabriques de connexion XA. |
| no-tx-separate-pools | Indique si on doit créer des sous-répertoires distincts pour chaque contexte. Cela est nécessaire pour les sources de données Oracle, qui ne permettent pas aux connexions XA d'être utilisées à la fois à l'intérieur et à l'extérieur d'une transaction de JTA |
| pad-xid | Indique si on doit remplir le Xid. |
| wrap-xa-resource | Indique si on doit inclure XAResource dans une instance **`org.jboss.tm.XAResourceWrapper`**. |

**Tableau 6.8. Paramètres de sécurité**

| Paramètre | Description |
| --- | --- |
| user-name | Le nom d'utilisation pour créer une nouvelle connexion. |
| mot de passe | Le mot de passe à utiliser pour créer une nouvelle connexion |
| security-domain | Contient le nom d'un gestionnaire de sécurité JAAS, qui gère l'authentification. Ce nom correspond à l'attribut application-policy/name de la configuration de connexion JAAS. |
| reauth-plugin | Définit un plugin d'authentification à nouveau pour la ré authentification de connexions physiques. |

**Tableau 6.9. Paramètres de validation**

| Paramètre | Description |
| --- | --- |
| valid-connection-checker | Une mise en œuvre d'interface **`org.jboss.jca.adaptors.jdbc.ValidConnectionChecker`** qui fournit une méthode **`SQLException.isValidConnection(Connection e)`** pour valider une connexion. Une exception signifie que la connexion est détruite. Cela remplace le paramètre **`check-valid-connection-sql`** s'il est présent. |
| check-valid-connection-sql | Un énoncé SQL pour vérifier la validité d'un pool de connexion. Peut être appelé quand une connexion gérée est tirée d'un pool. |
| validate-on-match | Indique si la validation niveau de connexion est effectuée lorsqu'une fabrique de connexion tente de faire correspondre une connexion gérée avec un ensemble donné.<br><br>Indiquer "true" pour **`validate-on-match`** n'est pas normalement fait en conjonction avec "true" pour **`background-validation`**. **`Validate-on-match`** est utile quand un client doit avoir une connexion vallidée avant utilisation. Ce paramètre est à true par défaut. |
| background-validation | Indique que les connexions sont validées sur un thread d'arrière plan. La validation d'arrière plan est une optimisation de performance si non utilisée en conjonction avec **`validate-on-match`**. Si **`validate-on-match`** est sur true, utiliser **`background-validation`** peut aboutir à des contrôles qui n'ont plus cours. La validation d'arrière plan ne vous donne pas la possibilité d'obtenir des mauvaises connexions pour le client (une connexion se détériore entre le balayage de validation et avant qu'elle soit remise au client), dont l'application client doit prendre en compte cette possibilité. |
| background-validation-millis | La durée, en millisecondes, d'exécution de la validation d'arrière-plan. |
| use-fast-fail | Si true, l'allocation de connexion échouera dès la première tentative, quand la connexion est invalide. La valeur par défaut est **`false`**. |
| stale-connection-checker | Une instance de **`org.jboss.jca.adapters.jdbc.StaleConnectionChecker`** qui fournit une méthode de booléen **`isStaleConnection(SQLException e)`**. Si cette méthode renvoie **`true`**, l'exception sera contenue dans une **`org.jboss.jca.adapters.jdbc.StaleConnectionException`**, une sous-classe de **`SQLException`**. |

| Paramètre | Description |
| --- | --- |
| exception-sorter | Une instance de **`org.jboss.jca.adapters.jdbc.ExceptionSorter`** qui fournit une méthode de booléen **`isExceptionFatal(SQLException e)`**. Cette méthode valide le fait que l'exception doit être envoyée à toutes les instances de **`javax.resource.spi.ConnectionEventListener`** comme message **`connectionErrorOccurred`**. |

**Tableau 6.10. Paramètre de timeout**

| Paramètre | Description |
| --- | --- |
| use-try-lock | Utilise **`tryLock()`** à la place de **`lock()`**. Tente d'obtenir le verrou pour le nombre de secondes configuré, avant le timeout, au lieu d'échouer immédiatement quand le verrou n'est pas rendu disponible. La valeur par défaut est de **`60`** secondes. Par exemple, pour définir un timeout de 5 minutes, définir **`<use-try-lock>`**300**`</use-try-lock>`**. |
| blocking-timeout-millis | La durée, en millisecondes, de blocage en attendant une connexion. Après de délai, l'exception est envoyée. Cela aura pour effet de bloquer uniquement tandis qu'on attend un permis de connexion, et cela n'aura pas pour effet de lancer une exception si la création d'une nouvelle connexion prend trop de temps. La valeur par défaut est de 30000, ce qui correspond à 30 secondes. |
| idle-timeout-minutes | La durée maximum, en minutes, avant qu'une connexion inutile puisse être fermée. La durée maximum dépend du temps de balayage de l'idleRemover, qui correspond à la moitié du **`idle-timeout-minutes`** le plus petit de n'importe quel pool. |
| set-tx-query-timeout | Indique si on doit définir le timeout d'interrogation par rapport au temps qui reste avant le timeout de transaction. Si aucune transaction n'existe, on utilisera le timeout de recherche qui a été configuré. La valeur par défaut est **`false`**. |
| query-timeout | Timeout pour les recherches, en secondes. La valeur par défaut est «no timeout». |

| Paramètre | Description |
| --- | --- |
| allocation-retry | Le nombre de tentatives de connexions avant d'envoyer une connexion. La valeur par défaut est `0`, pour qu'une exception puisse être envoyée à la première défaillance. |
| allocation-retry-wait-millis | Le temps, en millisecondes, qu'il faut attendre avant de retenter d'allouer une connexion. La valeur par défaut est 5 000, soit 5 secondes. |
| xa-resource-timeout | Si la valeur est non nulle, elle passe à la méthode `XAResource.setTransactionTimeout`. |

**Tableau 6.11. Paramètres d'instruction**

| Paramètre | Description |
| --- | --- |
| track-statements | Indique si l'on doit vérifier les instructions non fermées lorsqu'une connexion est renvoyée à un pool ou qu'une instruction est retournée dans le cache d'instruction préparée. Si false, les instructions ne seront pas suivies.<br><br>**Valeurs valides**<br>• `true`: les instructions et les ensembles de résultats sont suivis, et un avertissement sera émis s'ils ne sont pas fermés.<br>• `false`: ni les instructions, ni les ensembles de résultats ne seront suivis.<br>• `nowarn`: les instructions sont suivies, mais il n'y a aucun avertissement. Valeur par défaut. |
| prepared-statement-cache-size | Le nombre d'instructions préparées par connexion, dans le cache LRU (Least Recently Used / Utilisé le moins souvent récemment). |
| share-prepared-statements | Indique si le fait de demander la même instruction deux fois sans la fermer utilise la même instruction préparée sous-jacente. La valeur par défaut est `false`. |

**Tableau 6.12. Paramètres de recouvrement**

| Paramètre | Description |
| --- | --- |
| recover-credential | Une paire nom d'utilisateur/mot de passe ou domaine de sécurité pour le recouvrement. |

| Paramètre | Description |
| --- | --- |
| recover-plugin | Une mise en œuvre de la classe **`org.jboss.jca.core.spi.recoveryRecoveryPlugin`** à utiliser pour le recouvrement. |

Report a bug

### 6.6.2. Les URL de connexion de sources de données

**Tableau 6.13.**

| Source de données | URL de connexion |
| --- | --- |
| PostgreSQL | jdbc:postgresql://*SERVER_NAME*:*PORT*/*DATABASE_NAME* |
| MySQL | jdbc:mysql://*SERVER_NAME*:*PORT*/*DATABASE_NAME* |
| Oracle | jdbc:oracle:thin:*@ORACLE_HOST*:*PORT*:*ORACLE_SID* |
| IBM DB2 | jdbc:db2://*SERVER_NAME*:*PORT*/*DATABASE_NAME* |
| Microsoft SQLServer | jdbc:microsoft:sqlserver://*SERVER_NAME*:*PORT*;DatabaseName=*DATABASE_NAME* |

Report a bug

### 6.6.3. Extensions de sources de données

Les déploiements de sources de données peuvent utiliser plusieurs extensions de l'adaptateur de ressources JDBC pour améliorer la validation de la connexion, et vérifier si l'exception doit rétablir la connexion. Ces extensions sont les suivantes :

**Tableau 6.14. Extensions de sources de données**

| Extension de sources de données | Paramètre de configuration | Description |
| --- | --- | --- |
| org.jboss.jca.adapters.jdbc.spi.ExceptionSorter | <exception-sorter> | Vérifie si SQLException est fatale pour la connexion sur laquelle l'exception a été lancée |

| Extension de sources de données | Paramètre de configuration | Description |
|---|---|---|
| org.jboss.jca.adapters.jdbc.spi.StaleConnection | <stale-connection-checker> | Met les SQLExceptions caduques dans une **`org.jboss.jca.adapters.jdbc.StaleConnectionException`** |
| org.jboss.jca.adapters.jdbc.spi.ValidConnection | <valid-connection-checker> | Vérifie si une connexion est valide pour être utilisée par l'application |

JBoss Enterprise Application Platform 6 comprend également les implémentations de ces extensions pour plusieurs bases de données prises en charge.

**Implémentations des extensions**

**Générique**

- org.jboss.jca.adapters.jdbc.extensions.novendor.NullExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.novendor.NullStaleConnectionChecker
- org.jboss.jca.adapters.jdbc.extensions.novendor.NullValidConnectionChecker
- org.jboss.jca.adapters.jdbc.extensions.novendor.JDBC4ValidConnectionChecker

**PostgreSQL**

- org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker

**MySQL**

- org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLReplicationValidConnectionChecker
- org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker

**IBM DB2**

- org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionChecker
- org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker

**Sybase**

- org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectionChecker

**Microsoft SQLServer**

- org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker

**Oracle**

- org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter
- org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker

Report a bug

### 6.6.4. Voir les statistiques de bases de données

Vous pourrez voir des statistiques de sources de données définies dans **`jdbc`** et **`pool`** qui utilisent des versions modifiées des commandes ci-dessous :

**Procédure 6.10.**

- ```
  /subsystem=datasources/data-source=ExampleDS/statistics=jdbc:read-resource(include-runtime=true)
  ```

  ```
  /subsystem=datasources/data-source=ExampleDS/statistics=pool:read-resource(include-runtime=true)
  ```

**NOTE**

Veillez à spécifier l'argument ***`include-runtime=true`***, car tous les statistiques sont des informations de runtime et la valeur par défaut est **`false`**.

Report a bug

### 6.6.5. Statistiques de bases de données

**Statistiques principales**

Le tableau suivant montre une liste des statistiques principaux de sources de données pris en charge :

**Tableau 6.15. Statistiques principales**

| Nom | Description |
|---|---|
| **`ActiveCount`** | Le nombre de connexions actives. Chacune des connexions est soit utilisée par une autre application ou disponible dans le pool |
| **`AvailableCount`** | Le nombre de connexions disponibles dans le pool |
| **`AverageBlockingTime`** | Le durée moyenne passée à bloquer l'obtention d'un verrou exclusif sur le pool. La valeur est en millisecondes. |
  ```

| Nom | Description |
| --- | --- |
| **AverageCreationTime** | Le durée moyenne passée à créer une connexion. La valeur est en millisecondes. |
| **CreatedCount** | Le nombre de connexions créées. |
| **DestroyedCount** | Le nombre de connexions détruites. |
| **InUseCount** | Le nombre de connexions actuellement utilisées. |
| **MaxCreationTime** | La durée maximum pour créer une connexion. La valeur est en millisecondes. |
| **MaxUsedCount** | Le nombre maximum de connexions utilisées |
| **MaxWaitCount** | Le nombre maximum de requêtes attendant une connexion en même temps. |
| **MaxWaitTime** | Le durée maximum à attendre un verrou exclusif sur le pool. |
| **TimedOut** | Le nombre de connexions expirées. |
| **TotalBlockingTime** | Le durée à attendre un verrou exclusif sur le pool. La valeur est en millisecondes. |
| **TotalCreationTime** | La durée passée à créer des connexions. La valeur est en millisecondes. |
| **WaitCount** | Le nombre de requêtes en attente de connexion. |

### Statistiques JDBC

Le tableau suivant montre une liste des statistiques JDBC de sources de données pris en charge :

**Tableau 6.16. Statistiques JDBC**

| Nom | Description |
| --- | --- |
| **PreparedStatementCacheAccessCount** | Le nombre de fois qu'un cache d'énoncé a été accédé. |
| **PreparedStatementCacheAddCount** | Le nombre d'énoncés ajoutés au cache de l'énoncé. |
| **PreparedStatementCacheCurrentSize** | Le nombre d'énoncés préparés et que l'on peut appeler, actuellement mis en cache dans un cache d'énoncé. |

| Nom | Description |
| --- | --- |
| **PreparedStatementCacheDeleteCount** | Le nombre d'énoncés rejetés du cache. |
| **PreparedStatementCacheHitCount** | Le nombre de fois que des énoncés de cache ont été utilisés. |
| **PreparedStatementCacheMissCount** | Le nombre de fois qu'une requête d'énoncé a pu être réglée par un énoncé d'un cache. |

Report a bug

## 6.7. EXEMPLES DE SOURCES DE DONNÉES

### 6.7.1. L'exemple de source de données PostgreSQL

**Exemple 6.8.**

L'exemple ci-dessous est une configuration de source de données PostgreSQL. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:jboss/PostgresDS" pool-name="PostgresDS">
    <connection-
url>jdbc:postgresql://localhost:5432/postgresdb</connection-url>
    <driver>postgresql</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidCon
nectionChecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptio
nSorter"></exception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="postgresql" module="org.postgresql">
      <xa-datasource-class>org.postgresql.xa.PGXADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **module.xml** pour la source de données PostgreSQL ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-9.1-902.jdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Report a bug

### 6.7.2. Exemple de source de données PostgreSQL XA

**Exemple 6.9.**

L'exemple ci-dessous est une configuration de source de données PostgreSQL XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:jboss/PostgresXADS" pool-
name="PostgresXADS">
    <driver>postgresql</driver>
    <xa-datasource-property name="ServerName">localhost</xa-datasource-
property>
    <xa-datasource-property name="PortNumber">5432</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">postgresdb</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidCon
nectionChecker">
      </valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptio
nSorter">
      </exception-sorter>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="postgresql" module="org.postgresql">
      <xa-datasource-class>org.postgresql.xa.PGXADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **module.xml** pour la source de données PostgreSQL XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-9.1-902.jdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Report a bug

### 6.7.3. Exemple de source de données MySQL

**Exemple 6.10.**

L'exemple ci-dessous est une configuration de source de données MySQL. La source de données a été activée, un utilisateur a été ajouté, et des options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:jboss/MySqlDS" pool-name="MySqlDS">
    <connection-url>jdbc:mysql://mysql-
localhost:3306/jbossdb</connection-url>
    <driver>mysql</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionC
hecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"
></exception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="mysql" module="com.mysql">
      <xa-datasource-
class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-
class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **module.xml** pour la source de données MySQL ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.0.8-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Report a bug

### 6.7.4. Exemple de source de données MySQL XA

**Exemple 6.11.**

L'exemple ci-dessous est une configuration de source de données MySQL XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:jboss/MysqlXADS" pool-
name="MysqlXADS">
  <driver>mysql</driver>
    <xa-datasource-property name="ServerName">localhost</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">mysqldb</xa-datasource-
property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionC
hecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"
></exception-sorter>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="mysql" module="com.mysql">
      <xa-datasource-
class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-
class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données MySQL XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.0.8-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Report a bug

### 6.7.5. L'exemple de source de données Oracle

**NOTE**

Avant la version 10.2 de la source de données Oracle, le paramètre <no-tx-separate-pools/> était requis, car le mélange de connexions transactionnelles et non-transactionnelles auraient créé une erreur. Ce paramètre n'est plus requis pour certaines applications.

**Exemple 6.12.**

L'exemple ci-dessous est une configuration de source de données Oracle. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:/OracleDS" pool-name="OracleDS">
    <connection-url>jdbc:oracle:thin:@localhost:1521:XE</connection-
url>
    <driver>oracle</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectio
nChecker"></valid-connection-checker>
      <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectio
nChecker"></stale-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorte
r"></exception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="oracle" module="com.oracle">
      <xa-datasource-
class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>
```

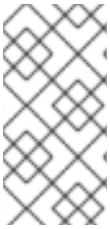
```
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **module.xml** pour la source de données Oracle ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.oracle">
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

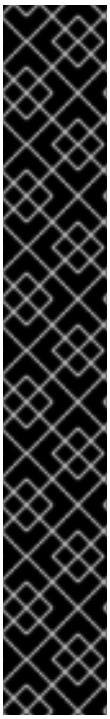
Report a bug

### 6.7.6. L'exemple de source de données Oracle XA



**NOTE**

Avant la version 10.2 de la source de données Oracle, le paramètre <no-tx-separate-pools/> était requis, car le mélange de connexions transactionnelles et non-transactionnelles auraient créé une erreur. Ce paramètre n'est plus requis pour certaines applications.



**IMPORTANT**

Les paramètres de configuration doivent être appliqués pour un utilisateur qui accède à une source de données Oracle XA pour que le recourvrement XA fonctionne correctement. La valeur **user** est une valeur définir par l'utilisateur pour pouvoir se connecter à partir de JBoss à Oracle :

- GRANT SELECT ON sys.dba_pending_transactions TO user;
- GRANT SELECT ON sys.pending_trans$ TO user;
- GRANT SELECT ON sys.dba_2pc_pending TO user;
- GRANT EXECUTE ON sys.dbms_xa TO user; (If using Oracle 10g R2 (patched) or Oracle 11g)

  OU

  GRANT EXECUTE ON sys.dbms_system TO user; (If using an unpatched Oracle version prior to 11g)

**Exemple 6.13.**

L'exemple ci-dessous est une configuration de source de données Oracle XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:/XAOracleDS" pool-name="XAOracleDS">
    <driver>oracle</driver>
    <xa-datasource-property name="URL">jdbc:oracle:oci8:@tc</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
      <no-tx-separate-pools />
    </xa-pool>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectio
nChecker"></valid-connection-checker>
      <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectio
nChecker"></stale-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorte
r"></exception-sorter>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="oracle" module="com.oracle">
      <xa-datasource-
class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données Oracle XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.oracle">
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Report a bug

### 6.7.7. Exemple de source de données Microsoft SQLServer

**Exemple 6.14.**

L'exemple ci-dessous est une configuration de source de données Microsoft SQLServer. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:/MSSQLDS" pool-name="MSSQLDS">
    <connection-
url>jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=MyDatabase</c
onnection-url>
    <driver>sqlserver</driver>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionC
hecker"></valid-connection-checker>
    </validation>
  </datasource>
  <drivers>
    <driver name="sqlserver" module="com.microsoft">
      <xa-datasource-
class>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</xa-datasource-
class>
    </driver>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données Microsoft SQLServer ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.microsoft">
  <resources>
    <resource-root path="sqljdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Report a bug

### 6.7.8. Exemple de source de données Microsoft SQLServer XA

**Exemple 6.15.**

L'exemple ci-dessous est une configuration de source de données Microsoft SQLServer XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:/MSSQLXADS" pool-name="MSSQLXADS">
    <driver>sqlserver</driver>
    <xa-datasource-property name="ServerName">localhost</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">mssqldb</xa-datasource-
property>
    <xa-datasource-property name="SelectMethod">cursor</xa-datasource-
property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
    </xa-pool>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionC
hecker"></valid-connection-checker>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="sqlserver" module="com.microsoft">
      <xa-datasource-
class>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</xa-datasource-
class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données Microsoft SQLServer XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.microsoft">
  <resources>
    <resource-root path="sqljdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Report a bug

### 6.7.9. Exemple de source de données IBM DB2

**Exemple 6.16.**

L'exemple ci-dessous est une configuration de source de données IBM DB2. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:/DB2DS" pool-name="DB2DS">
    <connection-url>jdbc:db2:ibmdb2db</connection-url>
    <driver>ibmdb2</driver>
    <pool>
      <min-pool-size>0</min-pool-size>
      <max-pool-size>50</max-pool-size>
    </pool>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionCheck
er"></valid-connection-checker>
      <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionCheck
er"></stale-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter"></e
xception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="ibmdb2" module="com.ibm">
      <xa-datasource-class>com.ibm.db2.jdbc.DB2XADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données IBM DB2 ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.ibm">
  <resources>
    <resource-root path="db2jcc.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Report a bug

### 6.7.10. Exemple de source de données IBM DB2 XA

**Exemple 6.17.**

L'exemple ci-dessous est une configuration de source de données IBM DB2 XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:/DB2XADS" pool-name="DB2XADS">
    <driver>ibmdb2</driver>
    <xa-datasource-property name="DatabaseName">ibmdb2db</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
    </xa-pool>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionCheck
er"></valid-connection-checker>
      <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionCheck
er"></stale-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter"></e
xception-sorter>
    </validation>
    <recovery>
      <recovery-plugin class-
name="org.jboss.jca.core.recovery.ConfigurableRecoveryPlugin">
        <config-property name="EnableIsValid">false</config-property>
        <config-property name="IsValidOverride">false</config-property>
        <config-property name="EnableClose">false</config-property>
      </recovery-plugin>
    </recovery>
  </xa-datasource>
  <drivers>
    <driver name="ibmdb2" module="com.ibm">
      <xa-datasource-class>com.ibm.db2.jcc.DB2XADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données IBM DB2 XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.ibm">
  <resources>
    <resource-root path="db2jcc.jar"/>
    <resource-root path="db2jcc_license_cisuz.jar"/>
    <resource-root path="db2jcc_license_cu.jar"/>
  </resources>
  <dependencies>
```

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

Report a bug

### 6.7.11. L'exemple de source de données Sybase

**Exemple 6.18.**

L'exemple ci-dessous est une configuration de source de données Sybase. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <datasource jndi-name="java:jboss/SybaseDB" pool-name="SybaseDB"
enabled="true">
    <connection-url>jdbc:sybase:Tds:localhost:5000/DATABASE?
JCONNECT_VERSION=6</connection-url>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectio
nChecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorte
r"></exception-sorter>
    </validation>
  </datasource>
  <drivers>
    <driver name="sybase" module="com.sybase">
      <datasource-
class>com.sybase.jdbc2.jdbc.SybDataSource</datasource-class>
      <xa-datasource-class>com.sybase.jdbc3.jdbc.SybXADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données Sybase ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.sybase">
  <resources>
    <resource-root path="jconn2.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
```

```
</module>
```

Report a bug

### 6.7.12. L'exemple de source de données Sybase

**Exemple 6.19.**

L'exemple ci-dessous est une configuration de source de données Sybase XA. La source de données a été activée, un utilisateur a été ajouté et les options de validation ont été définies.

```
<datasources>
  <xa-datasource jndi-name="java:jboss/SybaseXADS" pool-
name="SybaseXADS" enabled="true">
    <xa-datasource-property name="NetworkProtocol">Tds</xa-datasource-
property>
    <xa-datasource-property name="ServerName">myserver</xa-datasource-
property>
    <xa-datasource-property name="PortNumber">4100</xa-datasource-
property>
    <xa-datasource-property name="DatabaseName">mydatabase</xa-
datasource-property>
    <security>
      <user-name>admin</user-name>
      <password>admin</password>
    </security>
    <validation>
      <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectio
nChecker"></valid-connection-checker>
      <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorte
r"></exception-sorter>
    </validation>
  </xa-datasource>
  <drivers>
    <driver name="sybase" module="com.sybase">
      <datasource-
class>com.sybase.jdbc2.jdbc.SybDataSource</datasource-class>
      <xa-datasource-class>com.sybase.jdbc3.jdbc.SybXADataSource</xa-
datasource-class>
    </driver>
  </drivers>
</datasources>
```

L'exemple ci-dessous est un fichier **`module.xml`** pour la source de données Sybase XA ci-dessus.

```
<module xmlns="urn:jboss:module:1.1" name="com.sybase">
  <resources>
    <resource-root path="jconn2.jar"/>
  </resources>
```

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

Report a bug

# CHAPITRE 7. CONFIGURATION DES MODULES

## 7.1. INTRODUCTION

### 7.1.1. Modules

Un Module est un regroupement logique des classes utilisées pour la gestion de dépendance et le chargement de classe. JBoss Enterprise Application Platform 6 identifie deux types de modules, parfois appelés modules statiques et dynamiques. La seule différence entre les deux est la façon dont ils sont emballés. Tous les modules offrent les mêmes caractéristiques.

**Modules statiques**

Les modules statiques sont prédéfinis dans le répertoire ***EAP_HOME*/modules/** du serveur d'application. Chaque sous-répertoire représente un module et contient un ou plusieurs fichiers JAR et un fichier de configuration (**module.xml**). Le nom de ce module est défini dans le fichier **module.xml**. Toutes les API fournies par le serveur d'application sont fournies en tant que modules statiques, comme les API Java EE ainsi que d'autres API telles que JBoss Logging, par exemple.

La création de modules statiques personnalisés peut être utile si plusieurs applications sont déployées sur un même serveur utilisant les mêmes bibliothèques de tierce partie. Au lieu d'un regroupement de ces bibliothèques pour chaque application, un module contenant ces bibliothèques peut être créé et installé par l'administrateur JBoss. Les applications peuvent ensuite déclarer une dépendance explicite sur les modules statiques personnalisés.

**Modules dynamiques**

Les modules dynamiques sont créés et chargés par le serveur d'application pour chaque déploiement JAR ou WAR (ou sous-déploiement d'un EAR). Le nom d'un module dynamique est dérivé du nom de l'archive déployée. Comme les déploiements sont chargés sous forme de modules, ils peuvent configurer des dépendances et peuvent être utilisés comme dépendances par d'autres déploiements.

Les modules ne sont chargés qu'en fonction des besoins. Cela a généralement lieu quand une application est déployée avec des dépendances implicites ou explicites.

Report a bug

### 7.1.2. Modules globaux

Un module global est un module que JBoss Enterprise Application Platform 6 fournit comme dépendance pour chaque application. Chaque module peut être composé en l'ajoutant à la liste du serveur d'applications des modules globaux. Il n'est nul besoin de faire des changements au module.

Report a bug

### 7.1.3. Les Dépendances de modules

Une dépendance de module est une déclaration qui indique qu'un module a besoin des classes d'un autre module pour pouvoir fonctionner. Les modules peuvent déclarer leurs dépendances sur un certain nombre d'autres modules. Quand le serveur d'applications charge un module, le chargeur de classes de module traite les dépendances de ce module et ajoute les classes de chaque dépendance à son chemin de classe. Si une dépendance particulière est introuvable, le module ne pourra pas charger.

Les applications déployées (JAR et WAR) sont chargées comme modules dynamiques et utilisent des dépendances pour accéder aux API fournis par JBoss Enterprise Application Platform 6.

Il y a deux types de dépendances : explicite et implicite.

Les dépendances explicites sont déclarées dans la configuration par le développeur. Les modules statiques peuvent déclarer des dépendances dans le fichier modules.xml. Les modules dynamiques peuvent avoir des dépendances déclarées dans les descripteurs de déploiement MANIFEST.MF ou jboss-deployment-structure.xml..

Les dépendances explicites peuvent être spécifiées comme étant optionnelles. Une erreur de chargement de dépendance optionnelle n'entraînera pas l'annulation d'un chargement de module. Cependant, si la dépendance est rendue disponible par la suite, elle ne sera PAS ajoutée au chemin de classe du module. Les dépendances doivent être rendues disponibles quand le module est chargé.

Les dépendances implicites sont ajoutées automatiquement par le serveur d'applications quand on trouve certaines conditions ou métadonnées dans un déploiement. Les API Java EE 6 fournis avec JBoss Enterprise Application Platform sont des exemples de modules ajoutés par détection de dépendances implicites dans les déploiements.

Les déploiements peuvent également être configurés de façon à exclure des dépendances implicites particulières. Il vous faut pour cela le fichier de déploiement jboss-deployment-structure.xml. C'est normalement le cas quand une application empaquète une version spécifique de bibliothèque que le serveur d'applications tentera d'ajouter comme dépendance implicite.

Un chemin de classe de module ne contient que ses propres classes et celles de ses dépendances immédiates. Un module n'est pas en mesure d'accéder aux classes des dépendances. Cependant, un module peut indiquer quand une dépendance explicite est exportée. Une dépendance exportée est fournie à tout module qui dépend du module qui l'exporte.

> **Exemple 7.1. Les dépendances de module**
>
> Le Module A dépend du Module B et le Module B dépend su Module C. Le Module A peut accéder aux classes du Module B, et le Module B peut accéder aux classes du Module C. Le Module C ne peut pas accéder aux classes du Module C à moins que :
>
> - Le Module A déclare une dépendance explicite sur le Module C, ou bien
> - Le Module B exporte ses dépendances sur le Module C.

Report a bug

### 7.1.4. Isolement du chargeur de classes d'un sous-déploiement

Chaque sous-déploiement d'EAR (Archive Enterprise) est un module dynamique possédant son propre chargeur de classe. Par défaut, un sous-déploiement peut accéder aux ressources d'autres sous-déploiements.

Si un sous-déploiement ne doit pas accéder aux ressources d'autres sous-déploiements (une isolation de sous-déploiement est alors requise), alors cela pourra être activé.

Report a bug

## 7.2. DÉSACTIVER L'ISOLEMENT DE MODULE DE SOUS-DÉPLOIEMENT POUR TOUS LES DÉPLOIEMENTS

Cette tâche montre aux administrateurs du serveur comment désactiver l'isolement du module de sous-déploiement dans le serveur d'applications. Cela affecte tous les déploiements.

**AVERTISSEMENT**

Cette tâche requiert que vous éditiez les fichiers de configuration XML du serveur. Le serveur doit être arrêté avant cela. Ce n'est que temporaire car les outils administratifs de version finale supporteront ce type de configuration.

1. **Arrêter le serveur**
   Arrêter le server de JBoss Enterprise Application Platform.

2. **Ouvrir le fichier de configuration du serveur**
   Ouvrir le fichier de configuration du serveur dans un éditeur de texte

   Ce fichier sera différent pour un domaine géré ou un serveur autonome. De plus, des emplacements et des noms de fichiers non-défauts peuvent être utilisés. Les fichiers de configuration par défaut sont **`domain/configuration/domain.xml`** et **`standalone/configuration/standalone.xml`** pour les domaines gérés et les serveurs autonomes respectivement.

3. **Chercher la configuration de sous-système EE**
   Chercher la configuration de sous-système EE. L'élément **`<profile>`** du fichier de configuration contient plusieurs éléments du sous-système. L'élément du sous-système a comme espace-nom **`urn:jboss:domain:ee:1.0`**.

   ```
   <profile>

      ...

      <subsystem xmlns="urn:jboss:domain:ee:1.0" />

      ...
   ```

   La configuration par défaut a une balise en fermeture automatique unique mais une configuration personnalisée peut avoir des balises d'ouverture ou de fermeture distinctes (éventuellement avec d'autres éléments è l'intérieur) comme ceci :

   ```
   <subsystem xmlns="urn:jboss:domain:ee:1.0" ></subsystem>
   ```

4. **Remplacer les balises en fermeture automatique si nécessaire**
   Si l'élément de sous-système EE est une balise en fermeture automatique unique, remplacez-la par les balises d'ouverture ou de fermeture qui conviennent ainsi :

   ```
   <subsystem xmlns="urn:jboss:domain:ee:1.0" ></subsystem>
   ```

5. **Ajouter l'élément ear-deployments-isolated**
   Ajouter l'élément **`ear-subdeployments-isolated`** comme dépendant de l'élément du sous-système EE et ajouter le contenu de **`false`** comme suit :

   ```
   <subsystem xmlns="urn:jboss:domain:ee:1.0" ><ear-subdeployments-isolated>false</ear-subdeployments-isolated></subsystem>
   ```

6. **Démarrer ler serveur**
   Lancer à nouveau le serveur JBoss Enterprise Application Platform pour qu'il commence d'exécuter avec la nouvelle configuration.

**Résultat :**

Le serveur va maintenant exécuter avec l'isolement de module de sous-déploiement désactivé pour tous les déploiements.

Report a bug

## 7.3. AJOUTER UN MODULE À TOUS LES DÉPLOIEMENTS

Cette tâche montre comment les administrateurs JBoss peuvent définir une liste de modules globaux.

**Prérequis**

1. Vous devez connaître le nom des modules qui ont été ajoutés comme modules globaux. Voir Section 7.4.1, « Modules inclus » pour obtenir la liste des modules statiques inclus dans JBoss Enterprise Application Platform 6. Si le module est dans un autre déploiement, voir Section 7.4.2, « Nommage de modules dynamiques » pour déterminer le nom du module.

**Procédure 7.1. Ajouter un module à la liste des modules globaux**

1. Connectez-vous à la console de gestion. Section 3.4.2, « Connectez-vous à la Console de management »

2. Naviguez dans le panneau **`EE Subsystem`**.

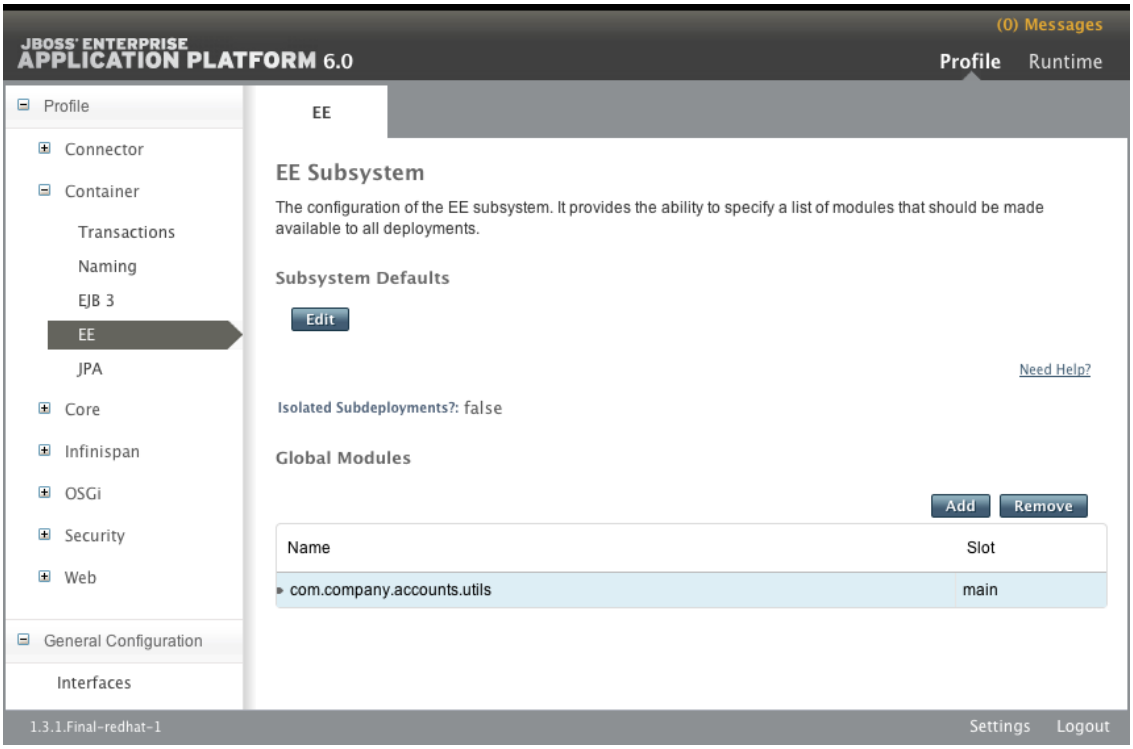



**Figure 7.1. Panneau EE Subsystem**

3. Cliquer sur le bouton **Add** dans la section **Global Modules**. La boîte de dialogue **Create Module** apparaîtra.

4. Saisir alors le nom du module et le slot de module, en option.

5. Cliquer sur le bouton **Enregistrer** pour ajouter le nouveau module global, ou bien cliquer sur le lien **Annuler** pour annuler.

   - Si vous cliquez sur le bouton **Enregistrer**, la boîte de dialogue va se fermer et le module spécifié sera ajouté à la liste des modules globaux.

   - Si vous cliquez sur le bouton **Annuler**, la boîte de dialogue se fermera et il n'y aura aucun changement.

**Résultat**

Les modules ajoutés à la liste des modules globaux seront ajoutés en tant que dépendances à chaque déploiement.

Report a bug

## 7.4. RÉFÉRENCE

### 7.4.1. Modules inclus

- **asm.asm**
- **ch.qos.cal10n**
- **com.google.guava**
- **com.h2database.h2**

- com.sun.jsf-impl
- com.sun.jsf-impl
- com.sun.xml.bind
- com.sun.xml.messaging.saaj
- gnu.getopt
- javaee.api
- javax.activation.api
- javax.annotation.api
- javax.api
- javax.ejb.api
- javax.el.api
- javax.enterprise.api
- javax.enterprise.deploy.api
- javax.faces.api
- javax.faces.api
- javax.inject.api
- javax.interceptor.api
- javax.jms.api
- javax.jws.api
- javax.mail.api
- javax.management.j2ee.api
- javax.persistence.api
- javax.resource.api
- javax.rmi.api
- javax.security.auth.message.api
- javax.security.jacc.api
- javax.servlet.api
- javax.servlet.jsp.api

- javax.servlet.jstl.api
- javax.transaction.api
- javax.validation.api
- javax.ws.rs.api
- javax.wsdl4j.api
- javax.xml.bind.api
- javax.xml.jaxp-provider
- javax.xml.registry.api
- javax.xml.rpc.api
- javax.xml.soap.api
- javax.xml.stream.api
- javax.xml.ws.api
- jline
- net.sourceforge.cssparser
- net.sourceforge.htmlunit
- net.sourceforge.nekohtml
- nu.xom
- org.antlr
- org.apache.ant
- org.apache.commons.beanutils
- org.apache.commons.cli
- org.apache.commons.codec
- org.apache.commons.collections
- org.apache.commons.io
- org.apache.commons.lang
- org.apache.commons.logging
- org.apache.commons.pool
- org.apache.cxf

- **org.apache.httpcomponents**
- **org.apache.james.mime4j**
- **org.apache.log4j**
- **org.apache.neethi**
- **org.apache.santuario.xmlsec**
- **org.apache.velocity**
- **org.apache.ws.scout**
- **org.apache.ws.security**
- **org.apache.ws.xmlschema**
- **org.apache.xalan**
- **org.apache.xerces**
- **org.apache.xml-resolver**
- **org.codehaus.jackson.jackson-core-asl**
- **org.codehaus.jackson.jackson-jaxrs**
- **org.codehaus.jackson.jackson-mapper-asl**
- **org.codehaus.jackson.jackson-xc**
- **org.codehaus.woodstox**
- **org.dom4j**
- **org.hibernate**
- **org.hibernate.envers**
- **org.hibernate.infinispan**
- **org.hibernate.validator**
- **org.hornetq**
- **org.hornetq.ra**
- **org.infinispan**
- **org.infinispan.cachestore.jdbc**
- **org.infinispan.cachestore.remote**
- **org.infinispan.client.hotrod**

- **org.jacorb**
- **org.javassist**
- **org.jaxen**
- **org.jboss.as.aggregate**
- **org.jboss.as.appclient**
- **org.jboss.as.cli**
- **org.jboss.as.clustering.api**
- **org.jboss.as.clustering.common**
- **org.jboss.as.clustering.ejb3.infinispan**
- **org.jboss.as.clustering.impl**
- **org.jboss.as.clustering.infinispan**
- **org.jboss.as.clustering.jgroups**
- **org.jboss.as.clustering.service**
- **org.jboss.as.clustering.singleton**
- **org.jboss.as.clustering.web.infinispan**
- **org.jboss.as.clustering.web.spi**
- **org.jboss.as.cmp**
- **org.jboss.as.connector**
- **org.jboss.as.console**
- **org.jboss.as.controller**
- **org.jboss.as.controller-client**
- **org.jboss.as.deployment-repository**
- **org.jboss.as.deployment-scanner**
- **org.jboss.as.domain-add-user**
- **org.jboss.as.domain-http-error-context**
- **org.jboss.as.domain-http-interface**
- **org.jboss.as.domain-management**
- **org.jboss.as.ee**

- **org.jboss.as.ee.deployment**
- **org.jboss.as.ejb3**
- **org.jboss.as.embedded**
- **org.jboss.as.host-controller**
- **org.jboss.as.jacorb**
- **org.jboss.as.jaxr**
- **org.jboss.as.jaxrs**
- **org.jboss.as.jdr**
- **org.jboss.as.jmx**
- **org.jboss.as.jpa**
- **org.jboss.as.jpa.hibernate**
- **org.jboss.as.jpa.hibernate**
- **org.jboss.as.jpa.hibernate.infinispan**
- **org.jboss.as.jpa.openjpa**
- **org.jboss.as.jpa.spi**
- **org.jboss.as.jpa.util**
- **org.jboss.as.jsr77**
- **org.jboss.as.logging**
- **org.jboss.as.mail**
- **org.jboss.as.management-client-content**
- **org.jboss.as.messaging**
- **org.jboss.as.modcluster**
- **org.jboss.as.naming**
- **org.jboss.as.network**
- **org.jboss.as.osgi**
- **org.jboss.as.platform-mbean**
- **org.jboss.as.pojo**
- **org.jboss.as.process-controller**

- **org.jboss.as.protocol**
- **org.jboss.as.remoting**
- **org.jboss.as.sar**
- **org.jboss.as.security**
- **org.jboss.as.server**
- **org.jboss.as.standalone**
- **org.jboss.as.threads**
- **org.jboss.as.transactions**
- **org.jboss.as.web**
- **org.jboss.as.webservices**
- **org.jboss.as.webservices.server.integration**
- **org.jboss.as.webservices.server.jaxrpc-integration**
- **org.jboss.as.weld**
- **org.jboss.as.xts**
- **org.jboss.classfilewriter**
- **org.jboss.com.sun.httpserver**
- **org.jboss.common-core**
- **org.jboss.dmr**
- **org.jboss.ejb-client**
- **org.jboss.ejb3**
- **org.jboss.iiop-client**
- **org.jboss.integration.ext-content**
- **org.jboss.interceptor**
- **org.jboss.interceptor.spi**
- **org.jboss.invocation**
- **org.jboss.ironjacamar.api**
- **org.jboss.ironjacamar.impl**
- **org.jboss.ironjacamar.jdbcadapters**

- **org.jboss.jandex**
- **org.jboss.jaxbintros**
- **org.jboss.jboss-transaction-spi**
- **org.jboss.jsfunit.core**
- **org.jboss.jts**
- **org.jboss.jts.integration**
- **org.jboss.logging**
- **org.jboss.logmanager**
- **org.jboss.logmanager.log4j**
- **org.jboss.marshalling**
- **org.jboss.marshalling.river**
- **org.jboss.metadata**
- **org.jboss.modules**
- **org.jboss.msc**
- **org.jboss.netty**
- **org.jboss.osgi.deployment**
- **org.jboss.osgi.framework**
- **org.jboss.osgi.resolver**
- **org.jboss.osgi.spi**
- **org.jboss.osgi.vfs**
- **org.jboss.remoting3**
- **org.jboss.resteasy.resteasy-atom-provider**
- **org.jboss.resteasy.resteasy-cdi**
- **org.jboss.resteasy.resteasy-jackson-provider**
- **org.jboss.resteasy.resteasy-jaxb-provider**
- **org.jboss.resteasy.resteasy-jaxrs**
- **org.jboss.resteasy.resteasy-jsapi**
- **org.jboss.resteasy.resteasy-multipart-provider**

- org.jboss.sasl
- org.jboss.security.negotiation
- org.jboss.security.xacml
- org.jboss.shrinkwrap.core
- org.jboss.staxmapper
- org.jboss.stdio
- org.jboss.threads
- org.jboss.vfs
- org.jboss.weld.api
- org.jboss.weld.core
- org.jboss.weld.spi
- org.jboss.ws.api
- org.jboss.ws.common
- org.jboss.ws.cxf.jbossws-cxf-client
- org.jboss.ws.cxf.jbossws-cxf-factories
- org.jboss.ws.cxf.jbossws-cxf-server
- org.jboss.ws.cxf.jbossws-cxf-transports-httpserver
- org.jboss.ws.jaxws-client
- org.jboss.ws.jaxws-jboss-httpserver-httpspi
- org.jboss.ws.native.jbossws-native-core
- org.jboss.ws.native.jbossws-native-factories
- org.jboss.ws.native.jbossws-native-services
- org.jboss.ws.saaj-impl
- org.jboss.ws.spi
- org.jboss.ws.tools.common
- org.jboss.ws.tools.wsconsume
- org.jboss.ws.tools.wsprovide
- org.jboss.xb

- **org.jboss.xnio**
- **org.jboss.xnio.nio**
- **org.jboss.xts**
- **org.jdom**
- **org.jgroups**
- **org.joda.time**
- **org.junit**
- **org.omg.api**
- **org.osgi.core**
- **org.picketbox**
- **org.picketlink**
- **org.python.jython.standalone**
- **org.scannotation.scannotation**
- **org.slf4j**
- **org.slf4j.ext**
- **org.slf4j.impl**
- **org.slf4j.jcl-over-slf4j**
- **org.w3c.css.sac**
- **sun.jdk**

Report a bug

### 7.4.2. Nommage de modules dynamiques

Tous les déploiements sont chargés en tant que modules par JBoss Enterprise Application Platform 6 et sont nommés en fonction des conventions suivantes :

1. Les déploiements des fichiers WAR et JAR sont nommés selon le format suivant :

   ```
   deployment.DEPLOYMENT_NAME
   ```

   Par exemple, **inventory.war** et **store.jar** auront les mêmes noms de module que **deployment.inventory.war** et **deployment.store.jar** respectivement.

2. Les sous-déploiements des Archives Enterprise sont nommés selon le format suivant :

   ```
   deployment.EAR_NAME.SUBDEPLOYMENT_NAME
   ```

Ainsi, le sous-déploiement **`reports.war`** qui se trouve dans l'archive enterprise **`accounts.ear`** aura le nom de module du **`deployment.accounts.ear.reports.war`**.

Report a bug

# CHAPITRE 8. VALVES GLOBALES

## 8.1. VALVES

Une Valve est une classe Java insérée dans le pipeline de traitement de demande d'une application. Elle est insérée dans le pipeline avant les filtres servlet. Les Valves peuvent apporter des modifications à la demande avant de les passer ou d'effectuer tout autre traitement comme l'authentification ou annuler la demande. Les Valves sont généralement empaquetées dans une application.

Les versions 6.1.0 et supérieures prennent en charge les valves globales.

Report a bug

## 8.2. VALVES GLOBALES

Une valve globale est une valve insérée dans le pipeline de traitement de requête de toutes les applications déployées. Une valve est rendue globale lorsqu'elle est mise en paquetage et qu'elle est installée comme module statique dans Boss Enterprise Application Platform 6. Les valves globales sont configurées dans le sous-système web.

Seules les versions 6.1.0 et supérieures prennent en charge les valves globales.

Report a bug

## 8.3. LES VALVES D'AUTHENTIFICATION

Une valve d'authentification est une valve qui authentifie les informations d'identification d'une requête. Cette valve est une sous-classe de **`org.apache.catalina.authenticator.AuthenticatorBase`** et elle remplace la méthode **`authenticate()`**

Elle peut être utilisée pour implémenter des schémas d'authentification supplémentaires.

Report a bug

## 8.4. INSTALLATION D'UNE VALVE GLOBALE

Les valves globales doivent être empaquetées et installées sous forme de modules statiques dans JBoss Enterprise Application Platform 6. Cette tâche vous montre comment installer le module.

**Prérequis :**

- La valve doit déjà avoir été créée et empaquetée dans un fichier JAR.
- Un fichier **`module.xml`** doit déjà avoir été créé pour le module.

  Voir Section 6.2.2, « Installer un Pilote JDBC comme Core Module » pour obtenir un exemple de fichier **`module.xml`**.

**Procédure 8.1. Installer un Module Global**

1. **Créer un répertoire d'installation de module**
   Vous devrez créer un répertoire pour le module à installer dans le répertoire de modules du serveur d'applications.

```
EAP_HOME/modules/system/layers/base/MODULENAME/main
```

```
$ mkdir -P
/usr/share/jboss/modules/system/layers/base/MyValveModule/main
```

2. **Copier les fichiers**
   Copier le JAR et les fichiers **`module.xml`** dans le répertoire créé dans l'étape 1.

```
$ cp MyValves.jar modules.xml
/usr/share/jboss/modules/system/layers/base/MyValveModule/main
```

Les classes Valve déclarées dans le module sont maintenant disponibles et peuvent être configurées dans le sous-système web.

Report a bug

## 8.5. CONFIGURATION D'UNE VALVE GLOBALE

Les valves globales sont activées et configurées dans le sous-système web par l'intermédiaire du JBoss CLI.

**Procédure 8.2. Configuration d'une Valve globale**

1. **Activer la Valve**
   Utiliser l'opération **add** pour ajouter une nouvelle saisie Valve.

```
/subsystem=web/valve=VALVENAME:add(module="MODULENAME",class-
name="CLASSNAME")
```

   Vous devrez indiquer les valeurs suivantes :

   - **`VALVENAME`**, le nom utilisé pour cette valve dans la configuration de l'application.
   - **`MODULENAME`**, le module qui contient la valeur en cours de configuration.
   - **`CLASSNAME`**, le nom de classe de la valve spécifique dans le module.

```
/subsystem=web/valve=clientlimiter:add(module="clientlimitermodule",
class-name="org.jboss.samplevalves.restrictedUserAgentsValve")
```

2. **En option : spécifier les paramètres**
   Si la valve a des paramètres de configuration, spécifier les dans l'opération **`add-param`**.

```
/subsystem=web/valve=testvalve:add-param(param-name="NAME", param-
value="VALUE")
```

```
/subsystem=web/valve=testvalve:add-param(
   param-name="restricteduseragents",
   param-value="^.*MS Web Services Client Protocol.*$"
)
```

Cette valve est maintenant activée et configurée pour toutes les applications déployées.

Report a bug

# CHAPITRE 9. DÉPLOIEMENT D'APPLICATIONS

## 9.1. LES DÉPLOIEMENTS D'APPLICATIONS

JBoss Enterprise Application Platform 6 dispose d'une gamme d'options de déploiement et de configuration d'application pour répondre à la fois aux environnements administratifs et de développement. Pour les administrateurs, la Console de gestion et le Management CLI offrent un graphisme et des interfaces de ligne de commande idéals pour gérer le déploiement des applications dans un environnement de production. Pour les développeurs, la gamme des options de testing de déploiement d'application incluent un scanner de déploiement hautement configurable de système de fichiers, l'utilisation d'un IDE comme JBoss Developer Studio, ou le déploiement et l'annulation du déploiement via Maven.

**Administration**

- **Console de gestion**
  - Section 9.2.2, « Déployer une application par la Console de gestion »
  - Section 9.2.3, « Retirer le déploiement d'une application à l'aide de la Console de gestion »
- **Management CLI**
  - Section 9.3.2, « Déployer une application dans un domaine géré à l'aide du Management CLI »
  - Section 9.3.4, « Déployer une application dans une Serveur autonome à l'aide du Management CLI »
  - Section 9.3.3, « Supprimer le déploiement d'une application dans un domaine géré à l'aide du Management CLI »
  - Section 9.3.5, « Supprimer le déploiement d'une application dans un serveur autonome à l'aide du Management CLI »
  - Section 9.3.1, « Gérer le déploiement d'une application à l'aide du Management CLI »

**Développement**

- **Scanner de déploiement**
  - Section 9.4.7, « Configurer le scanneur de déploiement »
  - Section 9.4.2, « Déployer une application dans une instance de serveur autonome par un scanneur de déploiement »
  - Section 9.4.3, « Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanneur de déploiement »
  - Section 9.4.4, « Redéploiement d'une application dans une instance de serveur autonome par le scanneur de déploiement »
  - Section 9.4.8, « Configurer le scanneur de déploiement avec le Management CLI »

- Section 9.4.6, « Référence pour attributs de scanneur de déploiement »
- Section 9.4.5, « Référence pour les fichiers de marquage de scanneur de déploiement »

- **Maven**
  - Section 9.5.2, « Déployer une application dans Maven »
  - Section 9.5.3, « Supprimer le déploiement d'une application dans Maven »

Report a bug

## 9.2. DÉPLOYER AVEC LA CONSOLE DE GESTION

### 9.2.1. Gérer le déploiement d'une application à l'aide de la Console de gestion

Le déploiement d'applications par l'intermédiaire de la Console de gestion vous donne l'avantage d'une interface graphique facile à utiliser. Vous pouvez voir en un coup de œil quelles applications sont déployées sur votre serveur ou les groupes de serveurs, et vous pouvez désactiver ou supprimer des applications dans le référentiel de contenu selon les besoins.

Report a bug

### 9.2.2. Déployer une application par la Console de gestion

**Prérequis**

- Section 3.4.2, « Connectez-vous à la Console de management »
- Section 3.4.5, « Ajouter un déploiement dans une Console de management »

**Procédure 9.1. Déployer une application par la Console de gestion**

1. **Naviguer dans le panneau `Manage Deployments` de la Console de gestion.**
   a. Sélectionner l'onglet **`Runtime`** en haut et à droite de la console.
   b. Sélectionner l'option **Deployments** → **Manage Deployments** à partir du menu à gauche de la console.
2. **Déployer une application**
   La méthode de déploiement variera suivant que vous déployez dans une instance de serveur autonome ou dans un domaine géré.
   - **Déployer dans une instance de serveur autonome.**
     Le tableau **`Deployments`** affiche tous les déploiements d'applications disponibles et leurs statuts.

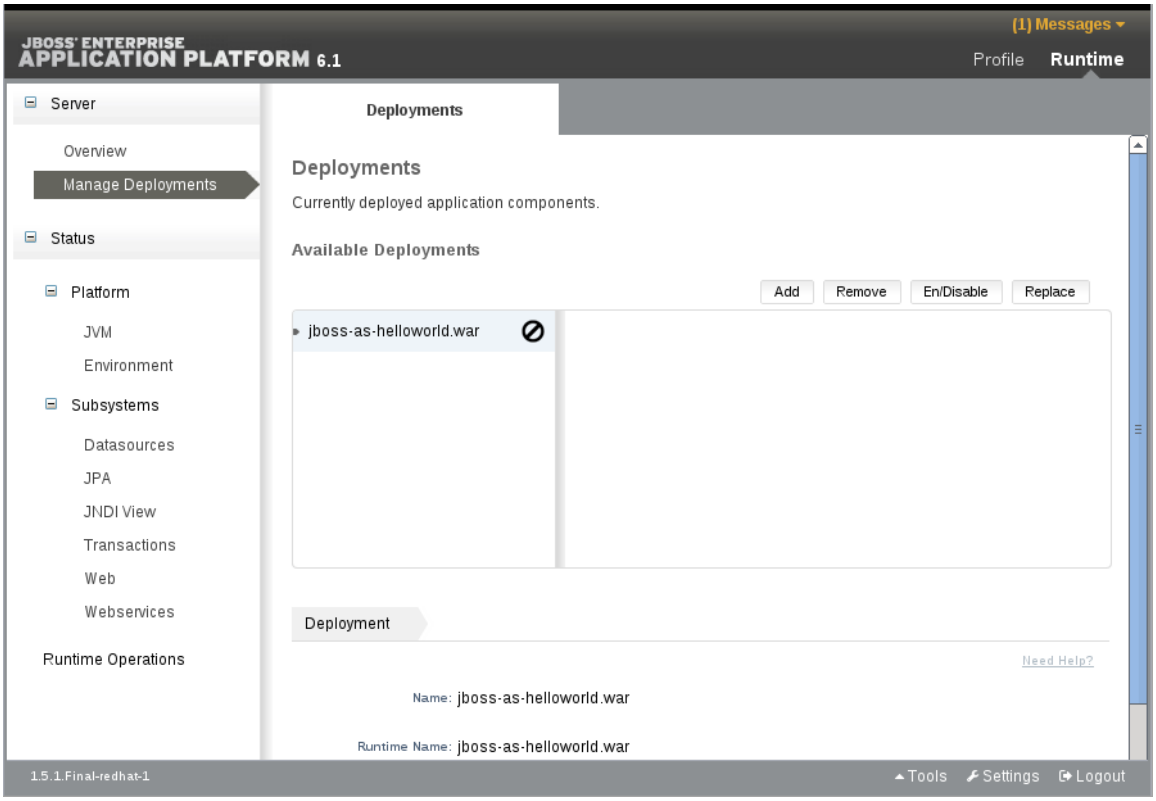



**Figure 9.1. Déploiements disponibles**

a. **Activer l'application dans une instance de serveur autonome**
Cliquer sur le bouton **`Enable`** du tableau **`Deployments`** pour activer le déploiement de l'application.

b. **Confirmer**
Cliquer sur le bouton **`confirm`** pour confirmer que l'application puisse être activée dans l'instance du serveur.


**Figure 9.2. Déploiements disponibles dans un serveur autonome.**

- **Déployer dans un domaine géré.**
La section **`Deployment Content`** contient un tableau **`Content Repository`** qui affiche tous les déploiements d'applications et leurs statuts.

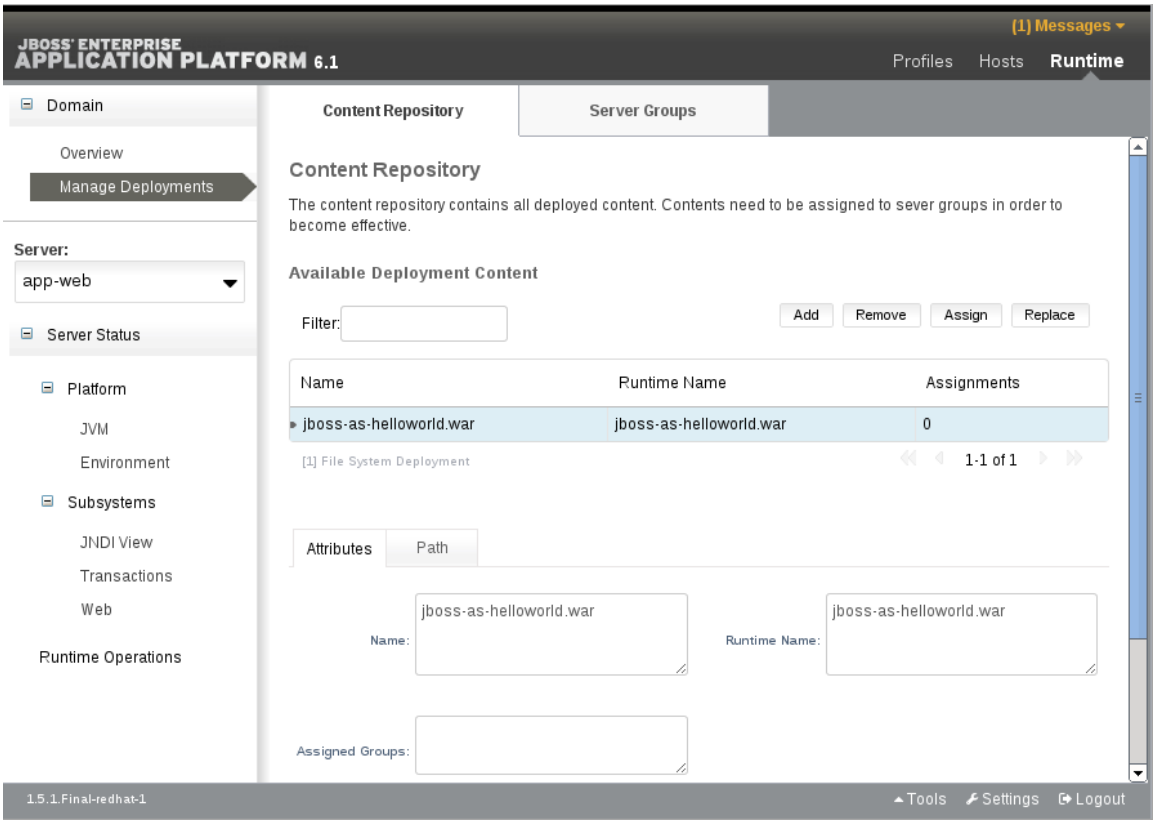



**Figure 9.3. Déploiements disponibles dans un domaines géré.**

a. **Activer l'application dans un domaine géré**
Cliquer sur le bouton **`Add to Groups`** dans le tableau **`Content Repository`**.

b. **Sélectionner les groupes de serveurs**
Cocher les cases pour chaque groupe de serveurs dans lesquels vous souhaitez ajouter l'application et cliquer sur le bouton **Save** pour continuer.

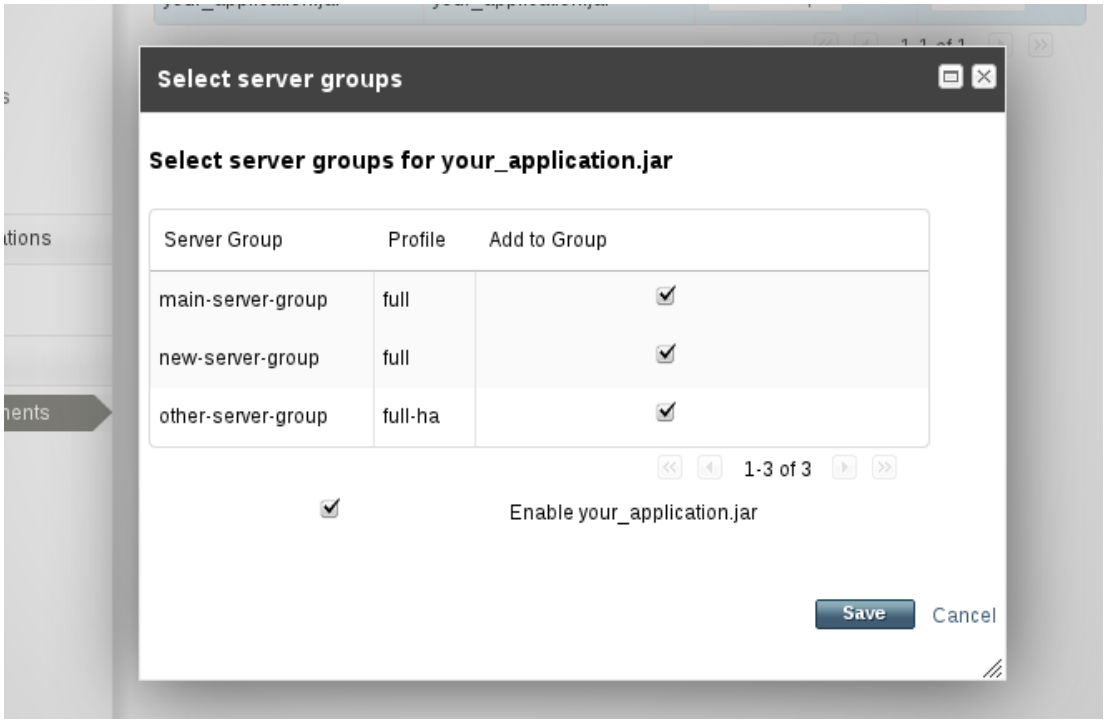



**Figure 9.4. Sélectionnner les groupes de serveurs pour le déploiement de l'application**

c. **Confirmer**
Cliquer sur l'onglet **`Server Group Deployments`** pour afficher le tableau **`Server Groups`**. Votre application sera maintenant déployée dans les groupes de serveurs que vous avez sélectionnés.

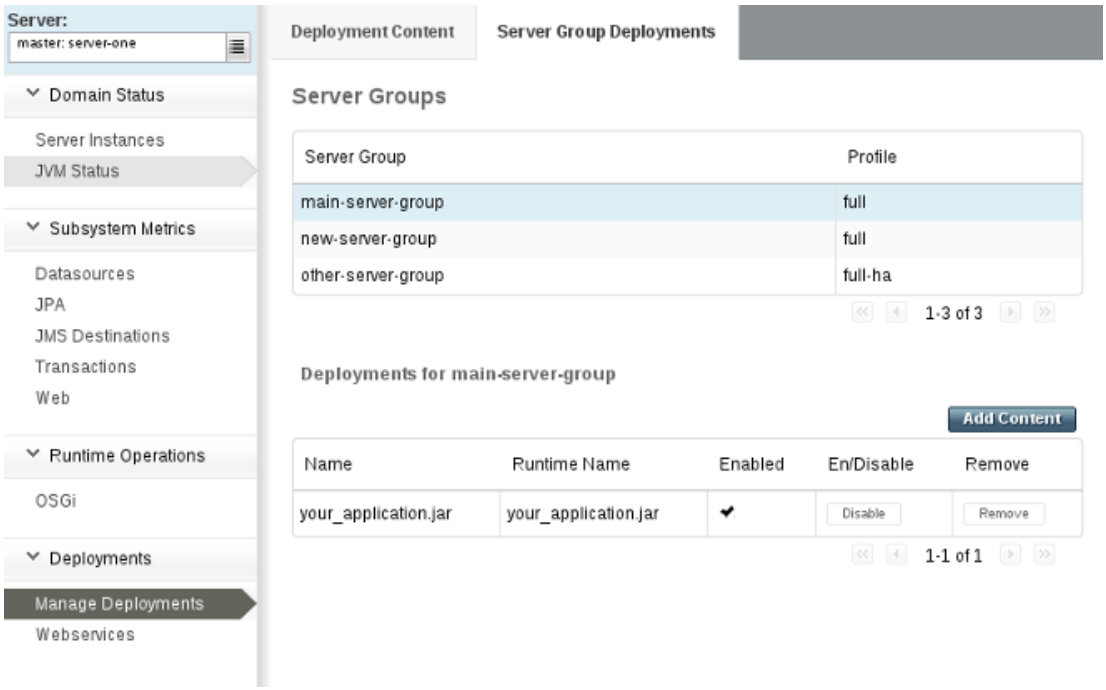



**Figure 9.5. Confirmation du déploiement d'applications dans les groupes de serveurs.**

**Résultat**

L'application est déployée sur le serveur qui convient ou dans le groupe de serveurs qui convient.

Report a bug

### 9.2.3. Retirer le déploiement d'une application à l'aide de la Console de gestion

**Prérequis**

- Section 3.4.2, « Connectez-vous à la Console de management »
- Section 3.4.5, « Ajouter un déploiement dans une Console de management »
- Section 9.2.2, « Déployer une application par la Console de gestion »

**Procédure 9.2. Retirer le déploiement d'une application à l'aide de la Console de gestion**

1. **Naviguer dans le panneau `Manage Deployments` de la Console de gestion.**

   a. Sélectionner l'onglet **`Runtime`** en haut et à droite de la console.

   b. Sélectionner l'option **Deployments** → **Manage Deployments** à partir du menu à gauche de la console.

2. **Supprimer le déploiement d'une application**
   La méthode de suppression du déploiement variera suivant que vous déployez dans une instance de serveur autonome ou dans un domaine géré.

- **Supprimer le déploiement d'une instance de serveur autonome.**
  Le tableau **`Deployments`** affiche tous les déploiements d'applications disponibles et leurs statuts.

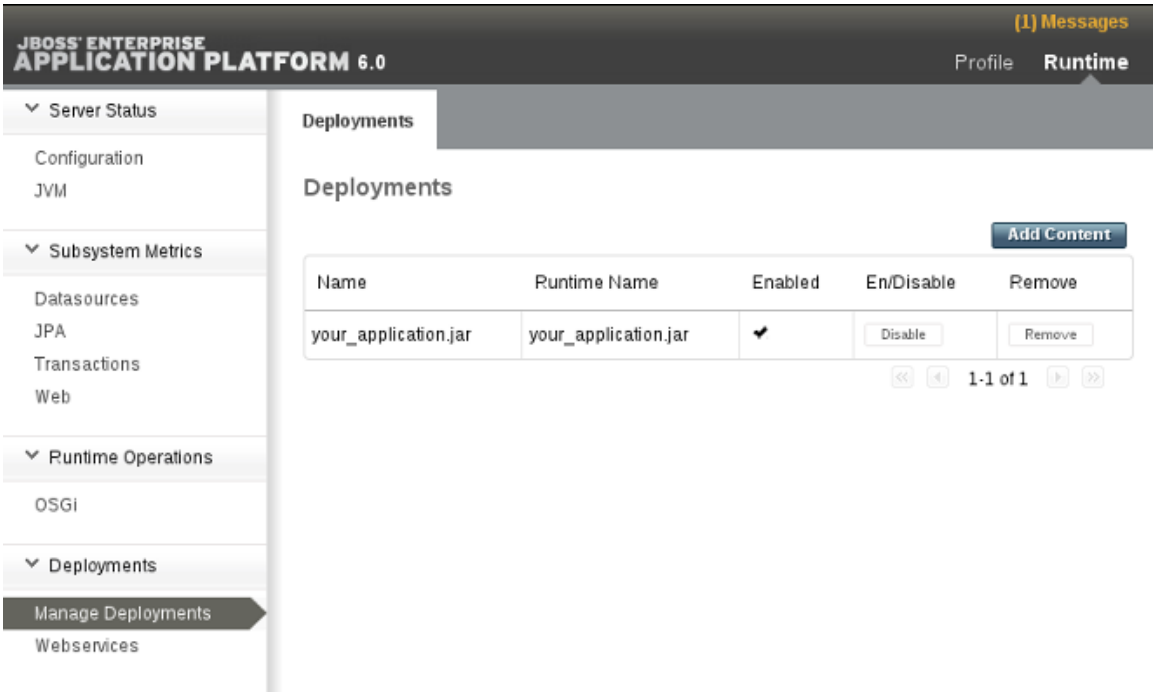



**Figure 9.6. Déploiements disponibles**

a. **Désactiver l'application dans une instance de serveur autonome**
   Cliquer sur le bouton **`Disable`** du tableau **`Deployments`** pour désactiver le déploiement de l'application.

b. **Confirmer que vous souhaitez désactiver l'application**
   Cliquer sur le bouton **`confirm`** pour confirmer que l'application puisse être désactivée dans l'instance du serveur.

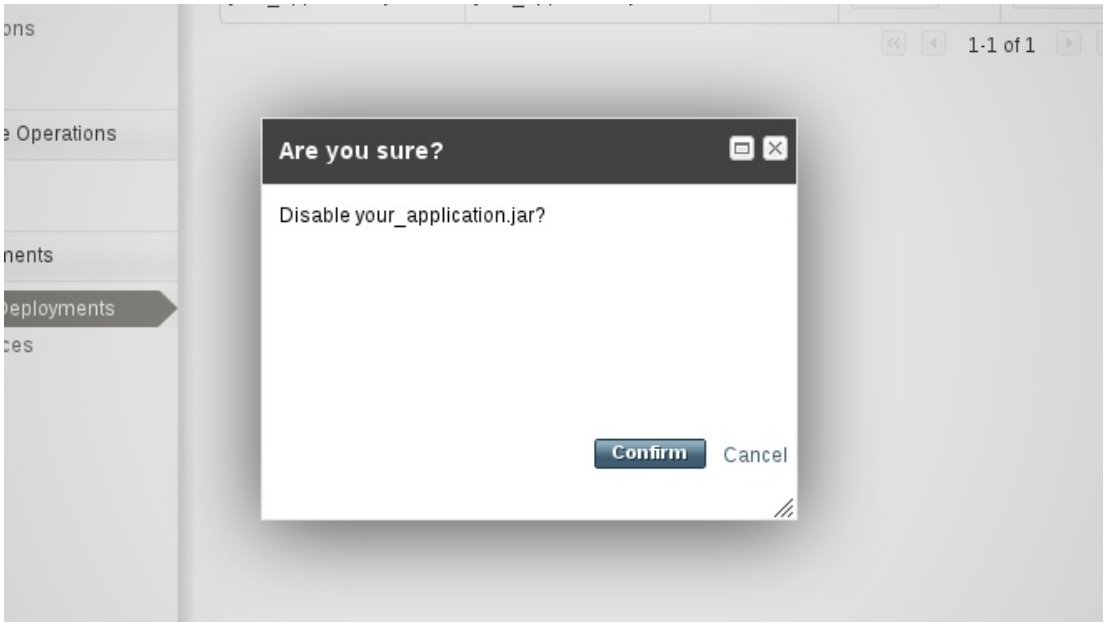



**Figure 9.7. Confirmer l'application à désactiver**

- **Supprimer le déploiement à partir d'un domaine géré**

La section **`Deployment Content`** contient un tableau **`Content Repository`** qui affiche tous les déploiements d'applications et leurs statuts.

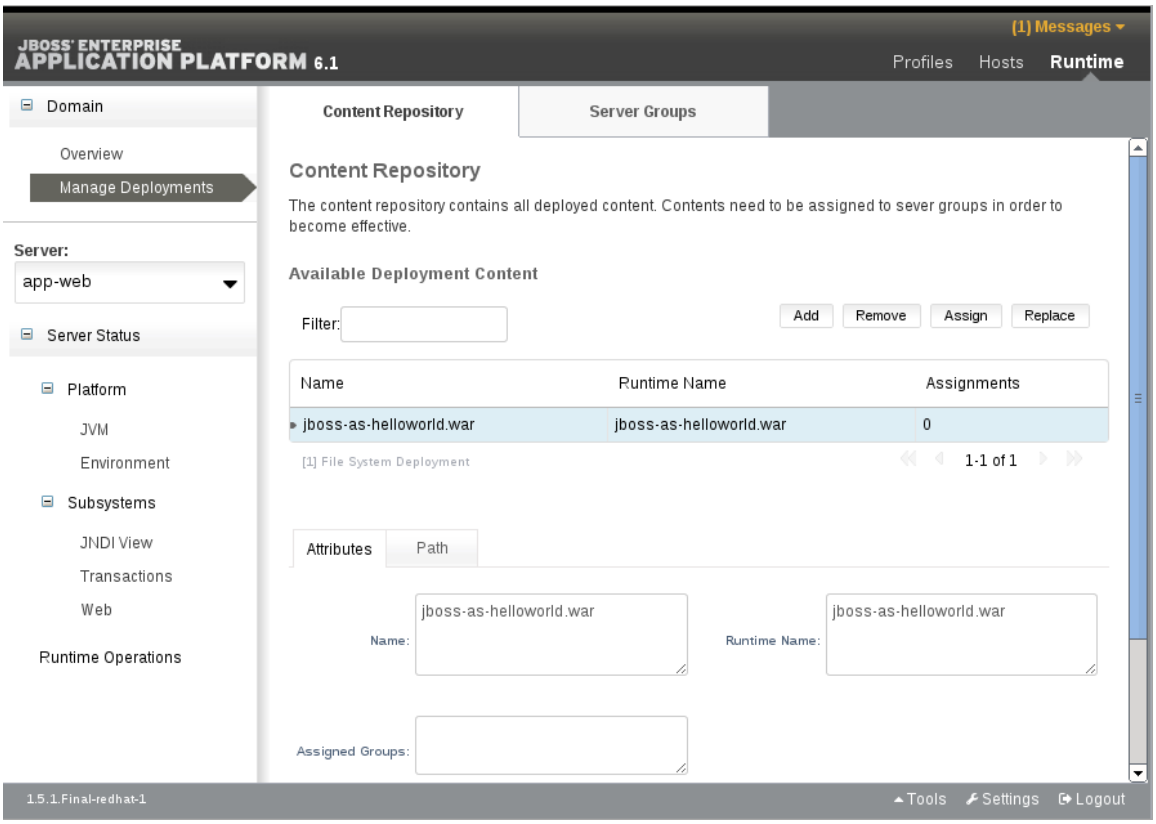



**Figure 9.8. Déploiements disponibles dans un domaines géré.**

a. **Désactiver l'application dans un domaine géré**
Cliquer sur l'onglet **`Server Group Deployments`** pour afficher les groupes de serveurs et le statut de leurs applications déployées.

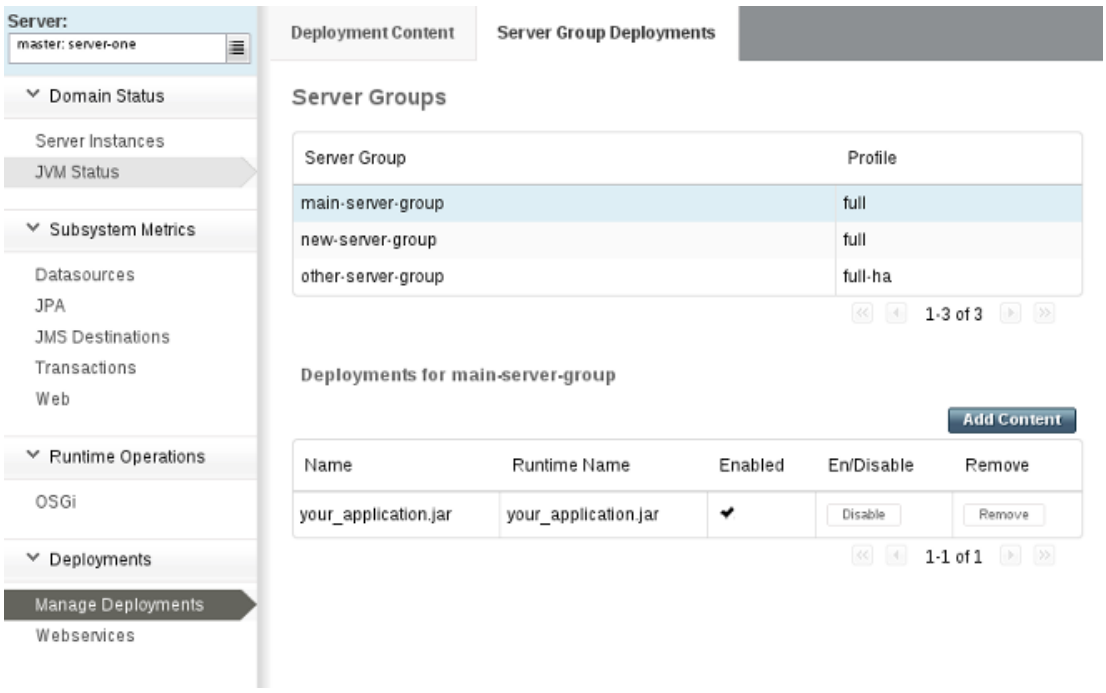



**Figure 9.9. Déploiements du groupe de serveurs**

b. **Sélectionner le groupe de serveurs**

Cliquer sur le nom du serveur dans le tableau **`Server Group`** pour supprimer un déploiement.

c. **Désactiver l'application à partir du serveur sélectionné**
Cliquer sur le bouton **`disable`** pour désactiver l'application d'un serveur sélectionné.

d. **Confirmer que vous souhaitez désactiver l'application**
Cliquer sur le bouton **`confirm`** pour confirmer que l'application puisse être désactivée dans l'instance du serveur.

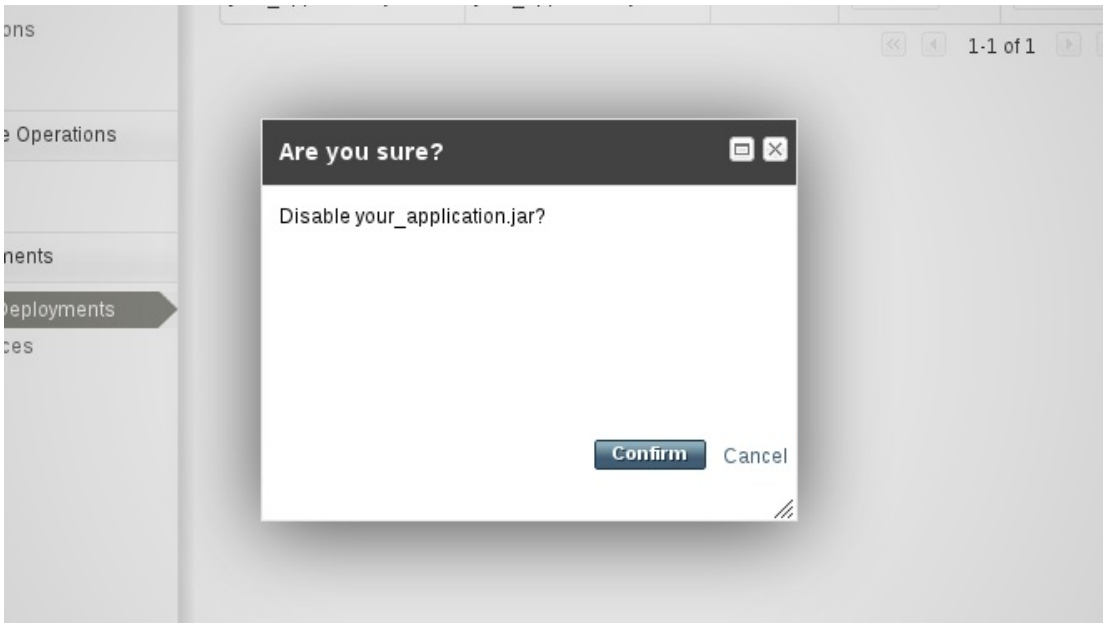



**Figure 9.10. Confirmer l'application à désactiver**

e. **Répéter la suppression du déploiement pour les groupes de serveurs qui restent.**
Répéter au besoin pour d'autres groupes de serveur. Le statut de l'application est confirmé pour chaque groupe de serveurs dans le tableau **`Deployments`**.

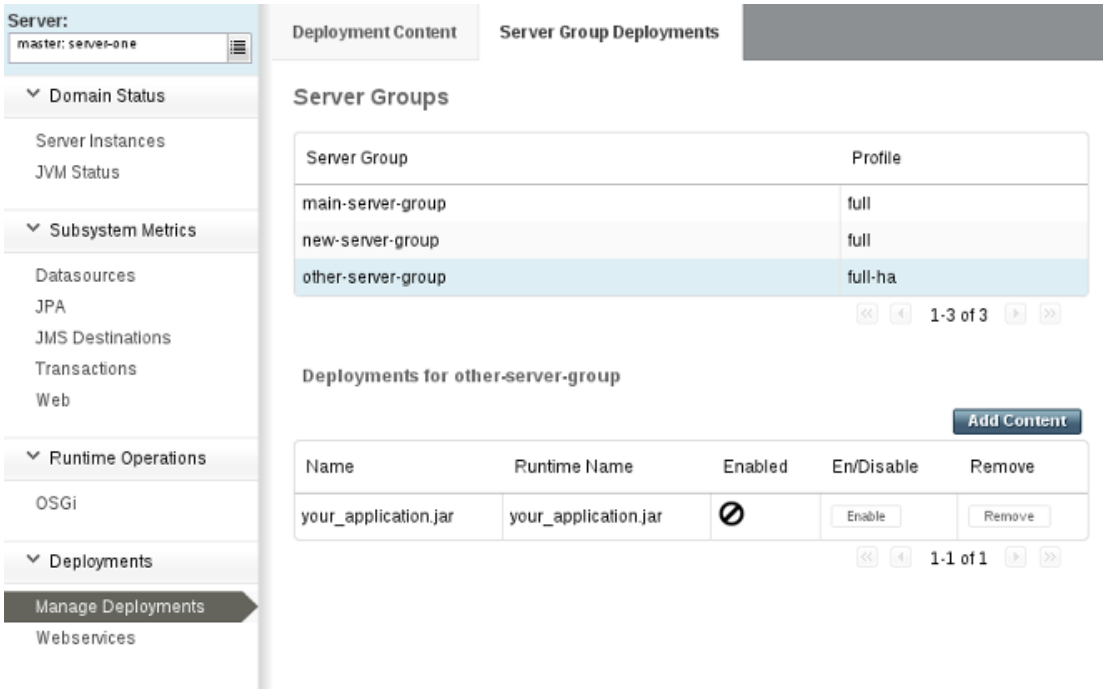



**Figure 9.11. Confirmation du déploiement d'applications à partir d'un groupe de serveurs.**

**Résultat**

L'application n'est pas déployée à partir d'un serveur qui convient ou d'un groupe de serveurs.

Report a bug

## 9.3. DÉPLOYER AVEC LE MANAGEMENT CLI

### 9.3.1. Gérer le déploiement d'une application à l'aide du Management CLI

Le déploiement d'applications par l'intermédiaire du Management CLI vous donne l'avantage d'une interface à ligne de commande facile à utiliser. Vous pouvez utiliser les capacités de scripting pour configurer le déploiement d'applications spécifiques et des scénarios de gestion. Vous pouvez gérer le statut déploiement d'un serveur dans le cas d'une instance autonome, ou un réseau entier de serveurs dans le cas d'un domaine géré.

Report a bug

### 9.3.2. Déployer une application dans un domaine géré à l'aide du Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI »

**Procédure 9.3. Déployer une application dans un domaine géré**

- **Exécutez la commande `deploy`**

Par l'intermédiaire du Management CLI, saisir la commande **deploy** ainsi que le chemin d'accès vers le déploiement de l'application. Inclure le paramètre ***--all-server-groups*** afin de déployer tous les groupes de serveurs.

```
[domain@localhost:9999 /] deploy /path/to/test-application.war --
all-server-groups
'test-application.war' déployé..
```

- Sinon, définir des groupes de serveurs particuliers de déploiement avec le paramètre ***--server-groups***.

```
[domain@localhost:9999 /] deploy /path/to/test-application.war --
server-groups server_group_1, server_group_2
'test-application.war' déployé.
```

**Résultat**

L'application indiquée est maintenant déployée dans un groupe de serveurs de votre domaine géré.

Report a bug

### 9.3.3. Supprimer le déploiement d'une application dans un domaine géré à l'aide du Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI »
- Section 9.3.2, « Déployer une application dans un domaine géré à l'aide du Management CLI »

**Procédure 9.4. Supprimer le déploiement d'une application dans un domaine géré**

- **Exécutez la commande undeploy**
  Par l'intermédiaire du Management CLI, saisir la commande **undeploy** ainsi que le nom de fichier du déploiement de l'application. On peut retirer le déploiement de l'application à partir de n'importe quel groupe de serveur dans lequel elle a été déployée à l'origine en ajoutant le paramètre ***--all-relevant-server-groups***.

```
[domain@localhost:9999 /]undeploytest-application.war--all-relevant-
server-groupsSuccessfully undeployed test-application.war.
```

**Résultat**

L'application spécifiée n'est plus déployée maintenant.

Report a bug

### 9.3.4. Déployer une application dans une Serveur autonome à l'aide du Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI »

**Procédure 9.5. Déployer une application dans un serveur autonome**

- **Exécutez la commande `deploy`**
  Avec Management CLI, saisir la commande **`deploy`** avec le chemin d'accès vers le déploiement de l'application.

```
[standalone@localhost:9999 /] deploy /path/to/test-application.war
'test-application.war' déployé.
```

**Résultat**

L'application indiquée est maintenant déployée dans un serveur autonome.

Report a bug

### 9.3.5. Supprimer le déploiement d'une application dans un serveur autonome à l'aide du Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »
- Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI »
- Section 9.3.4, « Déployer une application dans une Serveur autonome à l'aide du Management CLI »

**Procédure 9.6. Supprimer le déploiement d'une application dans un serveur autonome**

- **Exécutez la commande `undeploy`**
  Avec Management CLI, saisir la commande **`undeploy`** avec le nom du fichier du déploiement de l'application.

```
[standalone@localhost:9999 /] undeploy test-application.war
Successfully undeployed test-application.war.
```

**Résultat**

L'application spécifiée n'est plus déployée maintenant.

Report a bug

## 9.4. DÉPLOYER AVEC LE SCANNEUR DE DÉPLOIEMENT

### 9.4.1. Gérer le déploiement d'applications dans le scanneur de déploiement

Déployer des applications dans une instance de serveur autonome par l'intermédiaire d'un scanneur de déploiement vous permet de créer et de tester des applications d'une manière adaptée aux cycles de

développement rapides. Vous pouvez configurer le scanneur de déploiement en fonction de vos besoins de fréquence de déploiement et de comportement pour une variété de types d'applications.

Report a bug

### 9.4.2. Déployer une application dans une instance de serveur autonome par un scanneur de déploiement

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 »

**Résumé**

Cette tâche présente une méthode de déploiement des applications dans une instance de serveur autonome par le scanneur de déploiement. Comme il est indiqué dans la section Section 9.1, « Les déploiements d'applications », cette méthode est retenue pour la commodité des développeurs, où les méthodes de Management Console ou de Management CLI sont recommandées pour la gestion des applications dans les environnements de production.

**Procédure 9.7. Utiliser le scanneur de déploiement pour déployer les applications.**

1. **Copier le contenu dans le dossier de déploiement**
   Copier le fichier de l'application dans un dossier de déploiement qui se situe ***EAP_HOME*/standalone/deployments/**.

2. **Modes d'analyses de déploiements**
   Le déploiement d'une application varie entre un mode d'analyse de déploiement manuel ou automatique.

   - **Déploiement automatique**
     Le scanneur de déploiement saisit un changement d'état d'un dossier et créer un fichier de marquage, comme expliqué dans la section Section 9.4.5, « Référence pour les fichiers de marquage de scanneur de déploiement ».

   - **Déploiement manuel**
     Le scanneur de déploiement a besoin d'un fichier de marqueurs pour déclencher le processus de dépoiement. L'exemple suivant utilise la commande Unix **touch** pour créer un nouveau fichier **.dodeploy**.

     **Exemple 9.1. Déploiement par la commande touch**

     ```
     [user@host bin]$
     touch$EAP_HOME/standalone/deployments/example.war.dodeploy
     ```

**Résultat**

Le fichier de l'application est déployé sur le serveur d'applications. Un fichier de marquage est créé dans le dossier de déploiement pour indiquer la réussite du déploiement, et l'application est marquée comme **Enabled** dans la console de gestion.

**Exemple 9.2. Contenu du dossier de déploiement après le déploiement.**

```
example.war
example.war.deployed
```

Report a bug

### 9.4.3. Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanneur de déploiement

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 »
- Section 9.4.2, « Déployer une application dans une instance de serveur autonome par un scanneur de déploiement »

**Résumé**

Cette tâche présente une méthode de retrait de déploiement d'applications en provenance d'une instance de serveur autonome qui ont été déployées par le scanneur de déploiement. Comme il est indiqué dans la section Section 9.1, « Les déploiements d'applications », cette méthode est retenue pour la commodité des développeurs, où les méthodes de Management Console ou de Management CLI sont recommandées pour la gestion des applications dans les environnements de production.

**NOTE**

Le scanneur de déploiement ne devrait pas servir en conjonction avec d'autres méthodes de déploiement pour la gestion des applications. Les applications supprimées du serveur d'applications par la console de gestion seront retirées du runtime sans affecter les fichiers de marquage ou l'application contenue dans le répertoire de déploiement. Pour minimiser le risque de redéploiement accidentelle ou autres erreurs, utilisez le Management CLI et la Console de gestion pour l'administration dans des environnements de production.

**Procédure 9.8. Retirer le déploiement d'une application à l'aide d'une des méthodes suivantes**

- **Supprimer le déploiement de l'application**
  Il existe deux méthodes pour supprimer un déploiement d'application suivant que vous souhaitiez supprimer l'application d'un dossier de déploiement ou bien que vous souhaitiez uniquement modifier son statut de déploiement.
  - **Retirer un déploiement par suppression du fichier de marquage**
    Supprimer le fichier de marquage **`example.war.deployed`** de l'application qui est déployée pour commencer le retrait de déploiement de l'application du runtime.

    **Résultat**

    Le scanneur de déploiement retire l'application et crée un fichier de marquage **`example.war.undeployed`**. L'application demeure dans le dossier de déploiement.

  - **Retirer le déploiement en supprimant l'application**
    Supprimer l'application depuis le répertoire de déploiement pour encourager le scanneur de déploiement à commencer le retrait du déploiement de l'application du runtime.

**Résultat**

Le scanneur de déploiement retire l'application et crée un fichier de marquage **`filename.filetype.undeployed`**. L'application n'est plus présente dans le dossier de déploiement.

**Résultat**

Le fichier de l'application n'est plus déployé dans le serveur d'applications et n'est plus visible dans l'écran **`Deployments`** de la Console de gestion.

Report a bug

### 9.4.4. Redéploiement d'une application dans une instance de serveur autonome par le scanneur de déploiement

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 »
- Section 9.4.2, « Déployer une application dans une instance de serveur autonome par un scanneur de déploiement »

**Résumé**

Cette tâche présente une méthode de retrait de déploiement d'applications dans une instance de serveur autonome qui a été déployée par le scanneur de déploiement. Comme il est indiqué dans la section Section 9.1, « Les déploiements d'applications », cette méthode est retenue pour la commodité des développeurs, où les méthodes de Management Console ou de Management CLI sont recommandées pour la gestion des applications dans les environnements de production.

**Procédure 9.9. Déployer à nouveau une application dans un serveur autonome**

- **Redéploiement de l'application**
  Il existe trois méthodes possibles pour redéployer une application déployée par le scanner de déploiement. Ces méthodes déclenchent le scanneur de déploiement pour initier un cycle de déploiement, et peuvent être choisies en fonction des préférences personnelles.
  - **Redéploiement par modification du fichier de marquage**
    Déclencher le redéploiement du scanneur de déploiement en modifiant l'horodatage d'accès du fichier. Dans l'exemple Linux suivant, on utilise une commande Unix **`touch`**.

    **Exemple 9.3. Redéployer par la commande Unix `touch`**

    ```
    [user@host bin]$
    touchEAP_HOME/standalone/deployments/example.war.dodeploy
    ```

    **Résultat**

    Le scanneur de déploiement a détecté un changement dans le fichier de marquage et a déployé à nouveau l'application. Un nouveau fichier de marquage **`.deployed`** remplace le précédent.

  - **Déployer à nouveau en créant un nouveau fichier de marquage `.dodeploy`**

Déclencher le redéploiement du scanneur de déploiement en créant un nouveau fichier de marquage **`.dodeploy`**. Voir les instructions de déploiement du manuel qui se trouvent dans Section 9.4.2, « Déployer une application dans une instance de serveur autonome par un scanneur de déploiement ».

- **Déployer à nouveau en supprimant le fichier de marquage**
  Comme décrit dans Section 9.4.5, « Référence pour les fichiers de marquage de scanneur de déploiement », la suppression du fichier de marquage **`.deployed`** va déclencher un retrait de déploiement et créera un marqueur **`.undeployed`**. Supprimer le marqueur de suppression de déploiement déclenchera le cycle de déploiement à nouveau. Voir Section 9.4.3, « Supprimer le déploiement d'une application dans une instance de serveur autonome par un scanneur de déploiement » pour obtenir des informations supplémentaires.

**Résultat**

Le fichier de l'application est déployé à nouveau.

Report a bug

## 9.4.5. Référence pour les fichiers de marquage de scanneur de déploiement

**Les fichiers de marquage**

Les fichiers de marquage font partie du sous-système du scanneur de déploiement. Ces fichiers indiquent le statut d'une application dans un répertoire de déploiement de l'instance du serveur autonome. Un fichier de marquage a le même nom que l'application, avec un suffixe de fichier qui indique l'état du déploiement de l'application. Le tableau suivant définit les types et les réponses de chaque fichier de marquage.

> **Exemple 9.4. Exemple de fichier de marquage**
>
> L'exemple suivant montre une fichier de marquage d'une instance déployée réussie pour une application nommée **`testapplication.war`**.
>
> ```
> testapplication.war.deployed
> ```

**Tableau 9.1. Définitions de types de fichiers de marquage**

| Suffixe du nom de fichier | Origine | Description |
|---|---|---|
| **`.dodeploy`** | Utilisateur généré | Indique que le contenu doit être déployé or redéployé dans le runtime. |
| **`.skipdeploy`** | Utilisateur généré | Désactive l'autodéploiement d'une application si présent. Utile pour bloquer temporairement l'auto-déploiement d'un contenu explosé, empêchant ainsi le risque de modifications de contenu imcomplétes d'être rendues live. Peut être utile pour le contenu compressé, bien que le scanneur détecte les changements en cours dans le contenu compressé et attend qu'ils soient terminés. |

| Suffixe du nom de fichier | Origine | Description |
|---|---|---|
| **.isdeploying** | Système généré | Indique l'initiation du déploiement. Le fichier de marquage sera effacé quand le processus de déploiement sera complété. |
| **.deployed** | Système généré | Indique que le contenu a été déployé. Le déploiement du contenu sera supprimé si le fichier est effacé. |
| **.failed** | Système généré | Indique les échecs de déploiement. Le fichier de marquage contient des informations sur la cause de l'échec. Si le fichier de marquage est supprimé, le contenu sera rendu visible dans l'auto-déploiement à nouveau. |
| **.isundeploying** | Système généré | Indique une réponse suite à la suppression d'un fichier **.deployed**. Le déploiement du contenu sera supprimé et le marqueur sera effacé automatiquement dès complition. |
| **.undeployed** | Système généré | Indique si le contenu a été déployé. La suppression du fichier de marquage n'a pas d'impact sur le re-déploiement du contenu. |
| **.pending** | Système généré | Indique que les instructions de déploiement devront être envoyées au serveur suite à la résolution d'un problème qui a été detecté. Ce marqueur sert de blocage de déploiement global. Le scanneur ne demandera pas au serveur de déployer ou de supprimer un déploiement de contenu tant que cette condition existe. |

Report a bug

### 9.4.6. Référence pour attributs de scanneur de déploiement

Le scanneur de déploiement contient les attributs suivants, qui sont exposés dans le Management CLI et qui peuvent être configurés par l'opération **write-attribute**. Pour plus d'informations sur les options de configuration, se référer à la section suivante Section 9.4.8, « Configurer le scanneur de déploiement avec le Management CLI ».

**Tableau 9.2. Attributs de scanneur de déploiement**

| Nom | Description | Type | Valeur par défaut |
|---|---|---|---|
| **auto-deploy-exploded** | Permet le déploiement automatique d'un contenu éclaté sans nécessiter un fichier de marquage **.dodeploy**. Recommandé pour les scénarios de développement de base uniquement pour empêcher le déploiement d'applications éclatées de se produire lors de changements du développeur ou du système d'exploitation. | Booléen | False |

| Nom | Description | Type | Valeur par défaut |
|---|---|---|---|
| **auto-deploy-xml** | Autorise le déploiement automatique d'un contenu XML sans besoin de fichier de marquage **.dodeploy**. | Booléen | True |
| **auto-deploy-zipped** | Autorise le déploiement automatique d'un contenu compressé sans besoin de fichier de marquage **.dodeploy**. | Booléen | True |
| **deployment-timeout** | La durée nécessaire en secondes pour que le scanneur de déploiement puisse permettre un déploiement avant annulation. | Long | 60 |
| **path** | Définit le chemin du système de fichier à scanner. Si l'attribut **relative-to** est spécifié, la valeur **path** agira comme un ajout relatif à ce répertoire ou chemin d'accès. | String | Déploiements |
| **relative-to** | Référence à un chemin de système de fichier défini dans la section **paths** du fichier de configuration XML du serveur. | String | jboss.server.base.dir |
| **scan-enabled** | Autorise le scanning automatique des applications par **scan-interval** au démarrage. | Booléen | True |
| **scan-interval** | L'intervalle en millisecondes entre les balayages de référentiels. Une valeur inférieure à 1 empêche le scanneur d'opérer au démarrage. | Int | 5000 |

Report a bug

### 9.4.7. Configurer le scanneur de déploiement

Le scanner de déploiement peut être configuré à l'aide de la Console de gestion ou le Management CLI. Vous pouvez créer un nouveau scanneur de déploiement ou bien gérer les attributs existants de scanneur, c'est à dire l'intervalle de balayage, l'emplacement du dossier de déploiement et les types de fichiers d'application qui déclencheront un déploiement.

Report a bug

### 9.4.8. Configurer le scanneur de déploiement avec le Management CLI

**Prérequis**

- Section 3.5.2, « Lancement du Management CLI »

**Résumé**

Bien qu'il existe plusieurs méthodes de configuration du scanneur de déploiement, le Management CLI permet d'exposer et de modifier les attributs par l'utilisation de scripts de lots (batch scripts) ou en temps réel. Vous pouvez modifier le comportement du scanneur de déploiement par l'utilisation de l'attribut lecture **`read-attribute`** et des opérations de ligne de commande **`write-attribute`**. Davantage d'informations sur les attributs de scanneur de déploiement sont définis dans la rubrique Section 9.4.6, « Référence pour attributs de scanneur de déploiement ».

Le scanneur de déploiement est un sous-système de JBoss Enterprise Application Platform 6, que vous pouvez voir dans **`standalone.xml`**.

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
    <deployment-scanner path="deployments" relative-
to="jboss.server.base.dir" scan-interval="5000"/>
</subsystem>
```

**Procédure 9.10. Configurer le scanneur de déploiement**

1. **Détermine les attributs de scanner de déploiement à configurer**
   Pour configurer le scanneur de déploiement par le Management CLI, vous devrez tout d'abord exposer les noms d'attribut qui conviennent. Vous pouvez faire cela grâce à l'opération **`read-resources`** au noeud root, ou bien par la commande **`cd`** pour passer au noeud dépendant du sous-système. Vous pouvez également afficher les attributs par la commande **`ls`** à ce niveau.

   - **Exposer les attributs de scanneur de déploiement par l'opération `read-resource`**
     Utiliser l'opération **`read-resource`** pour exposer les attributs définis par la ressource de scanneur de déploiement par défaut.

     ```
     [standalone@localhost:9999 /]/subsystem=deployment-
     scanner/scanner=default:read-resource
     {
         "outcome" => "success",
         "result" => {
             "auto-deploy-exploded" => false,
             "auto-deploy-xml" => true,
             "auto-deploy-zipped" => true,
             "deployment-timeout" => 60,
             "path" => "deployments",
             "relative-to" => "jboss.server.base.dir",
             "scan-enabled" => true,
             "scan-interval" => 5000
         }
     }
     ```

   - **Exposer les attributs de scanneur de déploiement par la commande `ls`**
     Utiliser la commande **`ls`** avec l'argument **`-l`** en option pour afficher une table de résultats qui incluent des attributs de nœud, des valeurs et types de sous-système. Vous pouvez en apprendre davantage sur la commande **`ls`** et ses arguments en exposant l'entrée **`ls --help`**. Pour plus d'informations sur le menu help du Management CLI, voir la section Section 3.5.5, « Comment obtenir de l'aide avec le Management CLI ».

     ```
     [standalone@localhost:9999 /] ls -l /subsystem=deployment-
     ```

```
scanner/scanner=default
ATTRIBUTE            VALUE                 TYPE
auto-deploy-exploded false                 BOOLEAN
auto-deploy-xml      true                  BOOLEAN
auto-deploy-zipped   true                  BOOLEAN
deployment-timeout   60                    LONG
path                 deployments           STRING
relative-to          jboss.server.base.dir STRING
scan-enabled         true                  BOOLEAN
scan-interval        5000                  INT
```

2. **Configurer le scanneur de déploiement par l'opération `write-attribute`**
   Une fois que vous avez déterminé le nom de l'attribut à modifier, utiliser la commande **`write-attribute`** pour spécifier le nom de l'attribut et la nouvelle valeur à indiquer. Les exemples suivants sont tous exécutés au niveau du noeud dépendant, qui peut être acccédé en utilisant la commande **cd** et la saisie semi-automatique via la touche TAB pour passer au noeud de scanneur par défaut.

```
[standalone@localhost:9999 /] cd subsystem=deployment-
scanner/scanner=default
```

   a. **Active le déploiement automatique du contenu explosé.**
      Utiliser l'opération **`write-attribute`** pour activer le déploiement automatique du contenu d'application explosé.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=auto-deploy-exploded,value=true)
{"outcome" => "success"}
```

   b. **Désactiver le déploiement automatique du contenu XML**
      Utiliser l'opération **`write-attribute`** pour désactiver le déploiement automatique du contenu d'application explosé.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=auto-deploy-xml,value=false)
{"outcome" => "success"}
```

   c. **Désactiver le déploiement automatique du contenu compressé**
      Utiliser la commande **`write-attribute`** pour désactiver le déploiement automatique du contenu d'applications compressé.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=auto-deploy-zipped,value=false)
{"outcome" => "success"}
```

   d. **Configurer l'attribut du chemin d'accès**
      Utiliser l'opération **`write-attribute`** pour modifier l'attribut de chemin d'accès, pour substituer la valeur de l'exemple **`newpathname`** par un nouveau nom de chemin d'accès que le scanneur de déploiement puisse surveiller. Noter que le serveur aura besoin que le nouveau chargement prenne effet.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=path,value=newpathname)
```

```
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
```

e. **Configurer l'attribut du chemin relatif**
Utiliser l'opération **`write-attribute`** pour modifier la référence relative du chemin du système de fichier ainsi définie dans la sections des chemins d'accès du fichier de configuration XML. Noter que le serveur aura besoin que le nouveau chargement prenne effet.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=relative-to,value=new.relative.dir)
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
```

f. **Désactiver le scanneur de déploiement**
Utiliser l'opération **`write-attribute`** pour désactiver le scanneur de déploiement en définissant la valeur **`scan-enabled`** à false.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=scan-enabled,value=false)
{"outcome" => "success"}
```

g. **Changer l'intervalle de balayage**
Utiliser l'opération **`write-attribute`** pour modifier l'intervalle de balayage de 5000 millisecondes à 10000 millisecondes.

```
[standalone@localhost:9999 scanner=default] :write-
attribute(name=scan-interval,value=10000)
{"outcome" => "success"}
```

**Résultat**

Vos modifications de configuration sont sauvegardés dans le scanneur de déploiement.

Report a bug

## 9.5. DÉPLOYER AVEC MAVEN

### 9.5.1. Gestion du déploiement d'applications dans Maven

Le déploiement d'applications dans Maven vous permet d'incorporer un cycle de déploiement dans votre flux de développement existant.

Report a bug

### 9.5.2. Déployer une application dans Maven

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 »

**Résumé**

Cette tâche vous montre une méthode pour déployer des applications dans Maven. L'exemple fourni utilise l'application **`jboss-as-helloworld.war`** qui se trouve dans la collection JBoss Enterprise Application Platform 6 Quick Starts. Le projet **`helloworld`** contient un fichier POM qui initialise le **`jboss-as-maven-plugin`**. Ce pugin fournit des simples opérations pour déployer ou supprimer le déploiement d'applications vers ou en provenance su serveur d'applications.

**Procédure 9.11. Déployer une application dans Maven.**

1. **Exécuter la commande de déploiement de Maven dans une session de terminal**
   Ouvrir la session de terminal et naviguez dans le répertoire qui contient les exemples Quickstart.

2. Exécuter la commande de déploiement Maven pour déployer l'application. Si l'application est déjà en cours d'exécution, elle sera redéployée.

   ```
   [localhost]$ mvn package jboss-as:deploy
   ```

3. **Confirmer le déploiement de l'application**

   - **Voir le résultat dans la fenêtre du terminal**
     Le déploiement peut être confirmé si vous regardez les entrées de journalisation de l'opération dans la fenêtre du terminal.

     **Exemple 9.5. Confirmation Maven pour l'application helloworld**

     ```
     [INFO] ------------------------------------------------------------------
     [INFO] BUILD SUCCESSFUL
     [INFO] ------------------------------------------------------------------
     [INFO] Total time: 3 seconds
     [INFO] Finished at: Mon Oct 10 17:22:05 EST 2011
     [INFO] Final Memory: 21M/343M
     [INFO] ------------------------------------------------------------------
     ```

   - **Voir les résultats dans la fenêtre du terminal du serveur**
     Le déploiement peut également être confirmé dans le flux de statut de l'instance du serveur d'applications actives.

     **Exemple 9.6. Confirmation du serveur d'applications pour l'application helloworld**

```
17:22:04,922 INFO  [org.jboss.as.server.deployment] (pool-1-
thread-3) Contenu ajouté dans
/home/username/EAP_HOME/standalone/data/content/2c/39607b0c8dbc
6a36585f72866c1bcfc951f3ff/content
17:22:04,924 INFO  [org.jboss.as.server.deployment] (MSC
service thread 1-1) Starting deployment of "jboss-as-
helloworld.war"
17:22:04,954 INFO  [org.jboss.weld] (MSC service thread 1-3)
Processing CDI deployment: jboss-as-helloworld.war
17:22:04,973 INFO  [org.jboss.weld] (MSC service thread 1-2)
Starting Services for CDI deployment: jboss-as-helloworld.war
17:22:04,979 INFO  [org.jboss.weld] (MSC service thread 1-4)
Starting weld service
17:22:05,051 INFO  [org.jboss.web] (MSC service thread 1-2)
registering web context: /jboss-as-helloworld
17:22:05,064 INFO  [org.jboss.as.server.controller] (pool-1-
thread-3) Deployed "jboss-as-helloworld.war"
```

**Résultat**

L'application est déployée dans le serveur d'applications.

Report a bug

### 9.5.3. Supprimer le déploiement d'une application dans Maven

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 »

**Résumé**

Cette tâche vous montre une méthode pour supprimer le déploiement d'applications dans Maven. L'exemple fourni utilise l'application **`jboss-as-helloworld.war`** qui se trouve dans la collection Enterprise Application Server Quick Starts. Le projet **`helloworld`** contient un fichier POM qui initialise le **`jboss-as-maven-plugin`**. Ce plug-in fournit des simples opérations pour déployer ou supprimer le déploiement d'applications vers ou en provenance du serveur d'applications.

**Procédure 9.12. Supprimer un déploiement d'application dans Maven**

1. **Exécuter la commande de déploiement de Maven dans une session de terminal**
   Ouvrir la session de terminal et naviguez dans le répertoire qui contient les exemples Quickstart.

   **Exemple 9.7. Passes au répertoire d'application helloworld**

   ```
   [localhost]$ cd /path/to/EAP_Quickstarts/helloworld
   ```

2. Exécuter la commande de suppression de déploiement.

   ```
   [localhost]$ mvn jboss-as:undeploy
   ```

3. **Confirmer la suppression du déploiement de l'application**

   - **Voir le résultat dans la fenêtre du terminal**
     La suppression du déploiement peut être confirmée si on observe les entrées de journalisation de la fenêtre de terminal.

     **Exemple 9.8. Confirmation Maven pour l'application helloworld**

```
[INFO] ------------------------------------------------------------------
[INFO] Building JBoss AS Quickstarts: Helloworld
[INFO]    task-segment: [jboss-as:undeploy]
[INFO] ------------------------------------------------------------------
[INFO] [jboss-as:undeploy {execution: default-cli}]
[INFO] Executing goal undeploy for
/home/username/EAP_Quickstarts/helloworld/target/jboss-as-
helloworld.war on server localhost (127.0.0.1) port 9999.
Oct 10, 2011 5:33:02 PM org.jboss.remoting3.EndpointImpl
<clinit>
INFO: JBoss Remoting version 3.2.0.Beta2
Oct 10, 2011 5:33:02 PM org.xnio.Xnio <clinit>
INFO: XNIO Version 3.0.0.Beta2
Oct 10, 2011 5:33:02 PM org.xnio.nio.NioXnio <clinit>
INFO: XNIO NIO Implementation Version 3.0.0.Beta2
[INFO] ------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------
[INFO] Total time: 1 second
[INFO] Finished at: Mon Oct 10 17:33:02 EST 2011
[INFO] Final Memory: 11M/212M
[INFO] ------------------------------------------------------------------
```

   - **Voir les résultats dans la fenêtre du terminal du serveur**
     La suppression du déploiement peut également être confirmée dans le flux de statut de l'instance du serveur d'applications actives.

     **Exemple 9.9. Confirmation du serveur d'applications pour l'application helloworld**

```
17:33:02,334 INFO  [org.jboss.weld] (MSC service thread 1-3)
Stopping weld service
17:33:02,342 INFO  [org.jboss.as.server.deployment] (MSC
service thread 1-3) Stopped deployment jboss-as-helloworld.war
in 15ms
17:33:02,352 INFO  [org.jboss.as.server.controller] (pool-1-
thread-5) Undeployed "jboss-as-helloworld.war"
```

- 

**Résultat**

La suppression du déploiement de l'application provient du serveur d'applications.

Report a bug

## 9.6. CONTRÔLER L'ORDRE DES APPLICATIONS DÉPLOYÉES DANS JBOSS APPLICATION PLATFORM

JBoss Enterprise Application Platform 6 offre un contrôle à grain fin sur l'ordre de déploiement d'applications lors du démarrage du serveur. L'ordre strict de déploiement d'applications présentes dans plusieurs fichiers ear peut être activé avec la persistance de cet ordre après un redémarrage.

**Procédure 9.13. Contrôle de l'ordre de déploiement dans EAP 6.0.X**

1. Crée des scripts CLI qui déploient et retirent les déploiements d'applications dans un ordre séquentiel quand le serveur est à l'Arrêt/Démarrage.

2. CLI prend également en charge le concept de mode batch qui permet de grouper les commandes et les opérations et les exécuter ensemble comme une unité atomique. Si au moins une des commandes ou opérations échoue, toutes les autres commandes et opérations exécutées avec succès dans le lot seront annulées.

**Procédure 9.14. Contrôler l'ordre de déploiement dans EAP 6.1.X**

La nouvelle fonctionnalité nommée Inter Deployment Dependencies d'EAP 6.1.X vous permet de déclarer des dépendances entre les niveaux supérieurs de déploiement.

1. Créer (s'il n'existe pas encore) un fichier **`jboss-all.xml`** dans le dossier **`app.ear/META-INF`** où **`app.ear`** est l'archive d'application qui dépend d'une autre archive d'application à déployer avant.

2. Éffectuer une saisie **`jboss-deployment-dependencies`** dans ce fichier comme indiqué ci-dessous. Notez que dans la liste ci-dessous, **`framework.ear`** est l'application de dépendance qui doit être déployée avant que l'archive d'application **`app.ear`** ne le soit.

```
<jboss umlns="urn:jboss:1.0">
  <jboss-deployment-dependencies xmlns="urn:jboss:deployment-
dependencies:1.0">
    <dependency name="framework.ear" />
  </jboss-deployment-dependencies>
</jboss>
```

Report a bug

# CHAPITRE 10. SÉCURISER JBOSS ENTERPRISE APPLICATION PLATFORM

## 10.1. LA SÉCURITÉ DU SOUS-SYSTÈME

Le sous-système de sécurité fournit l'infrastructure pour la fonctionnalité de sécurité de JBoss Enterprise Application Platform. La plupart des éléments de configuration ne doivent pas souvent utiliser *deep-copy-subject-mode*. De plus, vous pouvez configurer des propriétés de sécurité dans tout le système. L'ensemble de la configuration se soucie des *domaines de sécurité*.

**Mode Deep Copy**

Si le mode Deep Copy est désactivé (par défaut), la copie d'une structure de données de sécurité se réfère l'original uniquement au lieu de copier toute la structure de données. Ce comportement est plus efficace, mais prône à la corruption des données si plusieurs threads possédant la même identité effacent le sujet lors d'un vidage ou d'une déconnexion.

Le mode Deep Copy entraîne la copie totale de la structure des données et de toutes ses données associées, si elles sont marquées comme «clonables». C'est plus sûr niveau thread, mais moins efficace.

**Propriétés de sécurité dans tout le système**

Vous pouvez définir des propriétés de sécurité dans tout le système, qui sont appliquées à **`java.security.Security class`**.

**Security Domain**

Un domaine de sécurité est un ensemble de configuration de sécurité déclarative *Java Authentication and Authorization Service (JAAS)* qu'une ou plusieurs applications utilisent pour contrôler l'authentification, l'autorisation, l'auditing de sécurité et le mapping de sécurité. Trois domaines de sécurité sont inclus par défaut: **`jboss-ejb-policy`**, **`jboss-web-policy`**, et **`other`**. Vous pouvez créer autant de domaines de sécurité que vous souhaitez pour accommoder les besoins de vos applications.

Report a bug

## 10.2. STRUCTURE DU SOUS-SYSTÈME DE SÉCURITÉ

Le sous-système de sécurité est configuré dans le domaine géré ou le fichier de configuration autonome. La plupart des éléments de configuration peuvent être configurés à l'aide de la Console de gestion via web ou du Management CLI sur console. Voici le code XML qui représente un exemple de sous-système de sécurité.

**Exemple 10.1. Exemple de configuration de sous-système de sécurité**

```
<subsystem xmlns="urn:jboss:domain:security:1.2">
 <security-management>
  ...
 </security-management>
 <security-domains>
        <security-domain name="other" cache-type="default">
            <authentication>
                <login-module code="Remoting" flag="optional">
                    <module-option name="password-stacking"
```

```
value="useFirstPass"/>
                </login-module>
                <login-module code="RealmUsersRoles" flag="required">
                    <module-option name="usersProperties"
value="${jboss.domain.config.dir}/application-users.properties"/>
                    <module-option name="rolesProperties"
value="${jboss.domain.config.dir}/application-roles.properties"/>
                    <module-option name="realm"
value="ApplicationRealm"/>
                    <module-option name="password-stacking"
value="useFirstPass"/>
                </login-module>
            </authentication>
        </security-domain>
        <security-domain name="jboss-web-policy" cache-type="default">
            <authorization>
                <policy-module code="Delegating" flag="required"/>
            </authorization>
        </security-domain>
        <security-domain name="jboss-ejb-policy" cache-type="default">
            <authorization>
                <policy-module code="Delegating" flag="required"/>
            </authorization>
        </security-domain>
    </security-domains>
    <vault>
      ...
    </vault>
</subsystem>
```

Les éléments **<security-management>**, **<subject-factory>** et **<security-properties>** ne font pas partie de la configuration par défaut. Les éléments **<subject-factory>** et **<security-properties>** ont été dépréciés à partir de JBoss Enterprise Application Platform 6.1.

Report a bug

## 10.3. CONFIGURER LE SOUS-SYSTÈME DE SÉCURITÉ

Vous pouvez configurer le sous-système de sécurité par le Management CLI ou par la Console de gestion basée web.

Chaque élément de niveau supérieur du sous-système de sécurité contient des informations sur un aspect différent la la configuration de la sécurité. Voir Section 10.2, « Structure du sous-système de sécurité » pour obtenir un exemple de configuration de sous-système.

**<security-management>**

Cette section remplace les comportemens de haut niveau du sous-système de sécurité. Chaque paramètre est optionnel. Il est rare de modifier ces paramètres sauf pour le mode de sujet Deep Copy.

| Option | Description |
| --- | --- |
| deep-copy-subject-mode | Indiquez si l'on doit copier ou lier les tokens de sécurité pour la sécurité des threads. |
| authentication-manager-class-name | Indiquer un nom de classe d'implémentation AthentificationManager alternatif à utiliser. |
| default-callback-handler-class-name | Spécifie un nom de classe global pour l'implémentation de CallbackHandler à utiliser avec les modules de connexion. |
| authorization-manager-class-name | Indiquer un nom de classe d'implémentation AthorizationManager alternatif à utiliser. |
| audit-manager-class-name | Indiquer un nom de classe d'implémentation Audit Manager alternatif à utiliser. |
| identity-trust-manager-class-name | Indiquer un nom de classe d'implémentation IdentityTrustManager alternatif à utiliser. |
| mapping-manager-class-name | Indiquer la nom de classe d'implémentation MappingManager à utiliser. |

**<subject-factory>**

L'usine de sujets contrôle la création d'instances de sujets. Elle utilise sans doute le gestionnaire d'authentification pour vérifier l'appelant. L'utilisation principale du sujet est la création d'un sujet par les composants JCA. Il est rare que l'on ait besoin de modifier l'usine de sujets.

**<security-domains>**

Un élément de conteneur qui contient plusieurs domaines de sécurité. Un domaine de sécurité peut contenir des informations sur des modules d'authentification, d'autorisation, de mappage, et d'auditing, ainsi qu'une autorisation JASPI et une configuration JSSE. Votre application indique ainsi un domaine de sécurité pour gérer ses informations de sécurité.

**<security-properties>**

Contient des noms et valeurs définis dans la classe java.security.Security

Report a bug

## 10.4. MODE DE SUJET DEEP COPY

Si le mode de sujet Deep Copy (*deep copy subject mode*) est désactivé (par défaut), copier une structure de données de sécurité fait une référence à l'original, au lieu de copier toute la structure de données. Ce comportement est plus efficace, mais prône à la corruption des données si plusieurs threads possédant la même identité effacent le sujet lors d'un vidage ou d'une déconnexion.

Le mode de sujet Deep Copy entraîne la copie totale de la structure des données et tout ce que ses données associées, quand elles sont marquées «clonables». C'est plus sûr niveau thread, mais moins efficace.

Le mode de sujet Deep Copy est configuré dans le cadre du sous-système de sécurité.

Report a bug

## 10.5. ACTIVER LE MODE DE SUJET DEEP COPY

Vous pouvez activer le mode de sécurité Deep Copy à partir de la console de gestion basée web ou le Management CLI.

**Procédure 10.1. Activer le mode de sécurité Deep Copy à partir de la Console de gestion**

1. **Connectez-vous à la Console de gestion.**
   La console de gestion est normalement disponible à un URL tel que http://127.0.0.1:9990/. Ajuster cette valeur pour qu'elle corresponde à vos besoins.

2. **Domaine géré : sélectionner le profil qui convient.**
   Dans un domaine géré, le sous-système de sécurité est configuré par profil, ou vous pouvez activer ou désactiver le mode de sécurité Depp Copy pour chacun, de manière indépendante.

   Pour sélectionner un profil, cliquer sur **`Profiles`** en haut et à droite de l'affichage console, puis sélectionner le profil que vous souhaitez modifier à partir de la case de sélection **`Profile`** en haut et à gauche.

3. **Ouvrir le menu de configuration `Security Subsystem`.**
   Étendre l'item de menu **`Security`** à droite de la console de gestion, puis cliquer sur le lien **`Security Subsystem`**.

4. **Modifier `deep-copy-subject-mode`.**
   Cliquer sur le bouton **`Edit`**. Cochez la case qui se trouve à côté de **`Deep Copy Subjects:`** pour activer le mode Copier Sujet (copy subject).

**Activer Deep Copy Subject Mode par la Console CLI**

Si vous préférez utiliser le Management CLI pour activer cette option, utiliser une des commandes suivantes.

**Exemple 10.2. Domaine géré**

```
/profile=full/subsystem=security:write-attribute(name=deep-copy-subject-mode,value=TRUE)
```

**Exemple 10.3. Serveur autonome**

```
/subsystem=security:write-attribute(name=deep-copy-subject-mode,value=TRUE)
```

Report a bug

## 10.6. DOMAINES DE SÉCURITÉ

### 10.6.1. Les domaines de sécurité

Les domaines de sécurité font partie du sous-système de sécurité JBoss Enterprise Application Platform. La configuration de la sécurité est désormais gérée de façon centralisée, ou par le contrôleur de domaine d'un domaine géré ou par le serveur autonome.

Un domaine de sécurité se compose de configurations d'authentification, d'autorisation, de mappage de sécurité et d'audit. Il met en place la sécurité déclarative *Java Authentication and Authorization Service (JAAS)*.

L'authentification est impliquée dans la vérification de l'identité d'un utilisateur. Dans la terminologie de la sécurité, cet utilisateur est appelé un *principal*. Bien que l'authentification et l'autorisation soient différentes, de nombreux modules d'authentification intégrés gèrent également l'autorisation.

Une *authorization* est une police de sécurité, qui contient des informations sur les actions qui sont autorisées ou interdites. Dans la teminologie de sécurité, on parle de «rôle».

*Security mapping* se rapporte à la possibilité d'ajouter, de modifier ou de supprimer des informations d'un principal, rôle ou attribut avant de passer les informations à votre application.

L'Auditing Manager vous permet de configurer les *provider modules* pour contrôler la façon dont les événements de sécurité sont rapportés.

Si vous utilisez des domaines de sécurité, vous pouvez supprimer toutes les configurations de sécurité spécifiques de votre application proprement dite. Cela permet de modifier les paramètres de sécurité de façon centralisée. Un scénario courant qui bénéficie de ce type de structure de configuration est le processus de déplacement des applications entre les environnements de test et de production.

Report a bug

### 10.6.2. Picketbox

Picketbox est le cadre de sécurité de base qui fournit l'authentification, l'autorisation, la vérification et des fonctions de mappage à des applications Java exécutant dans JBoss Enterprise Application Platform. Il fournit les fonctions suivantes, dans un cadre unique avec une configuration simple :

- Section 10.6.3, « Authentification »
- Section 10.6.5, « L'autorisation » et contrôle d'accès
- Section 10.6.7, « Security Auditing »
- Section 10.6.9, « Security Mapping » des principaux, rôles et attributs

Report a bug

### 10.6.3. Authentification

L'authentification consiste à identifier un sujet et à vérifier l'authenticité de l'identification. Le mécanisme d'authentification le plus commun est une combinaison de nom d'utilisateur et de mot de passe. D'autres mécanismes d'authentification communs utilisent des clés partagées, des smart cards (cartes à puce) ou des empreintes digitales. Le résultat d'une authentification à succès s'appelle un Principal, en termes de sécurité déclarative Java Enterprise Edition.

JBoss Enterprise Application Platform utilise un système pouvant se connecter à des modules d'authentification en vue de flexibilité et d'intégration avec les systèmes d'authentification que vous

utilisez dans votre organisation. Chaque domaine de sécurité contient un ou plusieurs modules d'authentification configurés. Chaque module comprend des paramètres de configuration supplémentaires pour personnaliser son comportement. Le plus simple consiste à configurer le sous-système d'authentification dans la console de gestion web.

L'authentification n'est pas le même chose que l'autorisation, même si elles sont souvent liées. Bon nombre des modules d'authentification intégrés peuvent également gérer des autorisations.

Report a bug

### 10.6.4. Configurer l'authentification dans un Domane de sécurité

Pour configurer les paramètres d'authentification d'un domaine de sécurité, connectez-vous dans la console de gestion et suivre la procédure suivante :

**Procédure 10.2. Configurer l'authentification dans un Domane de sécurité**

1. **Ouvrir l'affichage détaillé du domaine de sécurité**
   Cliquer sur l'étiquette **`Profiles`** en haut et à droite de la console de gestion. Dans un domaine géré, sélectionner le profile à modifier à partir de la case de sélection **`Profile`** en haut et à gauche de l'affichge du profil. Cliquer sur l'item de menu **`Security`** sur la gauche, et cliquer sur **`Security Domains`** à partir du menu déroulant. Cliquer sur le lien **`View`** pour obtenir le domaine de sécurité que vous souhaitez modifier.

2. **Naviguer dans la configuration du sous-système d'authentification.**
   Cliquer sur l'étiquette **`Authentification`** en haut de l'affichage s'il n'a pas déjà été sélectionné.

   La zone de configuration est divisée en deux : **`Login Modules`** et **`Details`**. Le module de connexion est l'unité de base de configuration. Un domaine de sécurité peut inclure plusieurs polices d'autorisation, et chacune peut inclure plusieurs attributs et options.

3. **Ajouter un module de connexion**
   Cliquer sur le bouton **`Add`** pour ajouter un module de police d'authentification JAAS. Remplir les informations pour votre module. Le **`Code`** est le nom de classe de votre module. Les **`Flags`** contrôlent la façon dont le module est lié à d'autres modules de polices d'authentification du même domaine de sécurité.

   **Explication des indicateurs**

   La spécification Java Enterprise Edition 6 fournit l'explication suivante sur les marqueurs des modules de sécurité. La liste suivante provient de http://docs.oracle.com/javase/6/docs/technotes/guides/security/jaas/JAASRefGuide.html#Appendix Consulter ce document pour obtenir des informations plus détaillées.

| Marqueur | Détails |
|---|---|
| Requis | Le LoginModule est requis pour la réussite. En cas de réussite ou d'échec, l'autorisation continuera son chemin dans la liste du LoginModule. |

| Marqueur | Détails |
| --- | --- |
| Pré-requis | Le LoginModule est requis pour la réussite. En cas de succès, l'authentification continue vers le bas de la liste LoginModule. En cas d'échec, le contrôle est renvoyé immédiatement à l'application (l'authentification ne continue pas son chemin le long de la liste LoginModule). |
| Suffisant | Le LoginModule n'est pas nécessaire pour aboutir. S'il ne réussit pas, le contrôle retourne immédiatement à l'application (l'authentification ne se déroule pas vers le bas de la liste LoginModule). S'il échoue, l'authentification se poursuit vers le bas de la liste LoginModule. |
| Option | Le LoginModule n'est pas requis pour la réussite. En cas de réussite ou d'échec, l'autorisation continuera son chemin dans la liste du LoginModule. |

Une fois que vous aurez ajouté votre module, vous pourrez modifier son **Code** ou ses **Flags** (indicateurs) en cliquant sur le bouton **Edit** dans la section **Details** de l'écran. Veillez à ce que l'onglet **Attributes** soit bien sélectionné.

4. **Option: ajouter, éditer, ou supprimer un module**
   Si vous avez besoin d'ajouter des options à votre module, cliquer sur leurs entrées dans la liste **Login Modules**, et sélectionner l'onglet **Module Options** dans la section **Details** qui se trouve en bas de la page. Cliquer sur le bouton **Add**, et fournir la clé et la valeur de l'option. Pour la modifier, cliquer sur la clé et modifier. Utiliser le bouton **Remove** pour supprimer l'option.

**Résultat**

Votre module d'authentification est ajouté au domaine de sécurité, et est rendu disponible immédiatement auprès des applications qui utilisent le domaine de sécurité.

**L'option de module `jboss.security.security_domain`**

Par défaut, chaque module de connexion défini dans un domaine de sécurité a l'option de module **`jboss.security.security_domain`** ajoutée automatiquement. Cette option cause des problèmes avec les modules de connexion, qui vérifient que seules les options connues soient définies. Le module de connexion Kerberos d'IBM **`com.ibm.security.auth.module.Krb5LoginModule`** est l'une d'entre elles.

Vous pouvez désactiver le comportement d'ajout de cette option de module, en définissant la propriété de système à **`true`** en démarrant la plate-forme JBoss Enterprise Application Platform. Ajouter ce qui suit pour démarrer les paramètres.

```
-Djboss.security.disable.secdomain.option=true
```

Vous pourrez également définir cette propriété par la Console de gestion. Dans un serveur autonome, vous pouvez définir les propriétés de système dans la section **Profile** de configuration. Dans un domaine géré, vous pouvez définir les propriétés du système pour chaque groupe de serveur.

Report a bug

### 10.6.5. L'autorisation

L'autorisation est un mécanisme d'octroi ou de refus de permission d'accéder à une ressource, basé sur l'identité. Ce mécanisme est implémenté sous forme de rôles de sécurité déclarative, qui peuvent être donnés à des principaux.

JBoss Enterprise Application Platform utilise un système modulaire pour configurer l'autorisation. Chaque domaine de sécurité peut contenir une ou plusieurs stratégies d'autorisation. Chaque stratégie possède un module de base qui définit son comportement. Il est configuré via les attributs et indicateurs spécifiques. La façon la plus simple consiste à configurer le sous-système de l'autorisation à l'aide de la console de gestion basée web.

L'authentification est différente de l'autorisation, et a lieu, en général, après l'authentification. La plupart des modules d'authentification gèrent également l'autorisation.

Report a bug

### 10.6.6. Configurer l'autorisation pour un domaine de sécurité

Pour configurer les paramètres d'un domaine de sécurité, connectez-vous à la console de gestion et suivre la procédure suivante :

**Procédure 10.3. Configurer l'autorisation pour un domaine de sécurité**

1. **Ouvrir l'affichage détaillé du domaine de sécurité**
   Cliquer sur l'étiquette **`Profiles`** en haut et à droite de la console de gestion. Dans un domaine géré, sélectionner le profil à modifier à partir de la case de sélection **`Profile`** en haut et à gauche de l'affichage du profil. Cliquer sur l'item de menu **`Security`** sur la gauche, et cliquer sur **`Security Domains`** à partir du menu déroulant. Cliquer sur le lien **`View`** pour obtenir le domaine de sécurité que vous souhaitez modifier.

2. **Naviguer dans la configuration du sous-système d'autorisation.**
   Cliquer sur l'étiquette **`Authorization`** en haut de l'affichage sil elle n'a pas déjà été sélectionnée.

   La zone de configuration est divisée en deux : **`Policies`** et **`Details`**. Le module de connexion est l'unité de base de configuration. Un domaine de sécurité peut inclure plusieurs polices d'autorisation, et chacune d'elle peut inclure plusieurs attributs ou options.

3. **Ajouter une police.**
   Cliquer sur le bouton **`Add`** pour ajouter un module de police d'autorisation JAAS. Remplir les informations pour votre module. Le **`Code`** est le nom de classe de votre module. Les **`Flags`** contrôlent la façon dont le module est lié à d'autres modules de polices d'autorisation du même domaine de sécurité.

   **Explication des indicateurs**

   La spécification Java Enterprise Edition 6 fournit l'explication suivante sur les marqueurs des modules de sécurité. La liste suivante provient de http://docs.oracle.com/javase/6/docs/technotes/guides/security/jaas/JAASRefGuide.html#Appendix Consulter ce document pour obtenir des informations plus détaillées.

| Marqueur | Détails |
| --- | --- |
| Requis | Le LoginModule est requis pour la réussite. En cas de réussite ou d'échec, l'autorisation continuera son chemin dans la liste du LoginModule. |
| Pré-requis | Le LoginModule est requis pour la réussite. En cas de succès, l'autorisation continue vers le bas de la liste LoginModule. En cas d'échec, le contrôle est renvoyé immédiatement à l'application (l'autorisation ne continue pas son chemin le long de la liste LoginModule). |
| Suffisant | Le LoginModule n'est pas nécessaire pour aboutir. S'il ne réussit pas, le contrôle retourne immédiatement à l'application (l'autorisation ne se déroule pas vers le bas de la liste LoginModule). S'il échoue, l'authentification se poursuit vers le bas de la liste LoginModule. |
| Option | Le LoginModule n'est pas requis pour la réussite. En cas de réussite ou d'échec, l'autorisation continuera son chemin dans la liste du LoginModule. |

Une fois que vous aurez ajouté votre module, vous pourrez modifier son **`Code`** ou ses **`Flags`** (indicateurs) en cliquant sur le bouton **`Edit`** dans la section **`Details`** de l'écran. Veillez à ce que l'onglet **`Attributes`** soit bien sélectionné.

4. **Option: ajouter, éditer, ou supprimer un module**
   Si vous avez besoin d'ajouter des options à votre module, cliquer sur leurs entrées dans la liste **`Login Modules`**, et sélectionner l'onglet **`Module Options`** dans la section **`Details`** qui se trouve en bas de la page. Cliquer sur le bouton **`Add`**, et fournir la clé et la valeur de l'option. Pour la modifier, cliquer sur la clé et modifier. Utiliser le bouton **`Remove`** pour supprimer l'option.

**Résultat**

Votre module de police de sécurité est ajouté au domaine de sécurité, et est rendu disponible immédiatement auprès des applications qui utilisent le domaine de sécurité.

Report a bug

### 10.6.7. Security Auditing

Security Auditing se réfère au déclenchement d'événements, comme écrire un blog en réponse à un événement qui a lieu dans le sous-système de sécurité. Les mécanismes de sécurité sont configurés dans le cadre du domaine de sécurité, avec les informations d'authentification, d'autorisation ou de mappage de sécurité.

Auditing utilise *des modules de fournisseur*. Vous pouvez en utiliser un existant ou bien créer le vôtre.

Report a bug

### 10.6.8. Configurer Security Auditing

Pour configurer les paramètres de Security Auditing, connectez-vous à la console de gestion et suivre la procédure suivante :

**Procédure 10.4. Configurer Security Auditing pour un domaine de sécurité**

1. **Ouvrir l'affichage détaillé du domaine de sécurité**
   Cliquer sur l'étiquette **Profiles** en haut et à droite de la console de gestion. Dans un domaine autonome, l'onglet est intitulé **Profile**. Dans un domaine géré, sélectionner le profil à modifier à partir de la case de sélection **Profile** en haut et à gauche de l'affichage du profil. Cliquer sur l'item de menu **Security** sur la gauche, et cliquer sur **Security Domains** à partir du menu déroulant. Cliquer sur le lien **View** pour obtenir le domaine de sécurité que vous souhaitez modifier.

2. **Naviguer dans la configuration du sous-système Auditing.**
   Cliquer sur l'étiquette **Audit** en haut de l'affichage si elle n'a pas déjà été sélectionnée.

   La zone de configuration est divisée en deux : **Provider Modules** et **Details**. Le module de fournisseur est l'unité de base de configuration. Un domaine de sécurité peut inclure plusieurs polices d'autorisation, et chacune peut inclure plusieurs attributs et options.

3. **Ajouter un module de fournisseur.**
   Cliquer sur le bouton **Add** pour ajouter un module de fournisseur. Remplir la section **Code** avec le nom de classe du module du fournisseur.

   Une fois que vous aurez ajouté votre module, vous pourrez modifier son **Code** en cliquant sur le bouton **Edit** dans la section **Details** de l'écran. Veillez à ce que l'onglet **Attributes** soit bien sélectionné.

4. **Vérifier que le module fonctionne**
   Le but d'un module d'auditing est d'offrir un moyen de surveiller les événements dans le sous-système de sécurité. Ce monitoring peut être réalisé sous la forme d'écriture dans un fichier de journalisation, des notifications email ou autre mécanisme d'audit mesurable.

   Par exemple, JBoss Enterprise Application Server inclut le module **LogAuditProvider** par défaut. S'il est activé par les étapes décrites ci-dessus, ce module d'audit écrira des notifications de sécurité dans un fichier **audit.log** qui se situe dans le sous-dossier **log** du répertoire ***EAP_HOME***.

   Pour vérifier si les étapes ci-dessus fonctionnent dans le contexte **LogAuditProvider**, procédez à une action qui risque de déclencher une notification, puis vérifier le fichier de journalisation de l'audit.

   Pour obtenir une liste complète des Modules Security Auditing Provider, voir : Section 11.4, « Modules de fournisseurs d'auditing de sécurité inclus »

5. **Option: ajouter, éditer, ou supprimer un module**
   Si vous avez besoin d'ajouter des options à votre module, cliquer sur leurs entrées dans la liste **Modules**, et sélectionner **Module Options**, dans la section **Details** qui se trouve en bas de la page. Cliquer sur le bouton **Add**, et fournir la clé et la valeur de l'option. Pour modifier une option existante, supprimez-là en cliquant sur l'étiquette **Remove**, et ajoutez-la à nouveau grâce aux options qui conviennent en cliquant le bouton **Add**.

**Résultat**

Votre module d'auditing de sécurité est ajouté au domaine de sécurité, et est rendu disponible immédiatement auprès des applications qui utilisent le domaine de sécurité.

Report a bug

### 10.6.9. Security Mapping

Le mappage de sécurité vous permet de combiner l'authentification et l'autorisation d'informations après que l'authentification ou l'autorisation aient eu lieu, mais avant que l'information ait été passée à votre application. Un exemple qui l'illustre est l'utilisation d'un certificat X509 pour l'authentification, puis la conversion du principal du certificat en nom logique que votre application puisse afficher.

Vous pouvez mapper vos principaux (authentification), rôles (autorisation), ou identifiants (attributs qui ne sont ni principaux, ni rôles).

Le mappage de rôles est utilisé pour ajouter, remplacer ou supprimer des rôles du sujet après l'authentification.

Le mappage du principal est utilisé pour modifier un principal après l'authentification.

Le mappage d'attributs est utilisé pour convertir des attributs d'un système externe à utiliser par votre application, et vice versa.

Report a bug

### 10.6.10. Configurer le Security Mapping dans un domaine de sécurité

Pour configurer les paramétres de Sécurity Mapping, connectez-vous à la console de gestion et suivre la procédure suivante :

**Procédure 10.5. Configurer le Mappage de sécurité pour un Domaine de sécurité**

1. **Ouvrir l'affichage détaillé du domaine de sécurité.**
   Cliquer sur l'étiquette **`Profiles`** en haut et à droite de la console de gestion. Dans un domaine autonome, l'onglet est intitulé **`Profile`**. Dans un domaine géré, sélectionner le profile à modifier à partir de la case de sélection **`Profile`** en haut et à gauche de l'affichage du profil. Cliquer sur l'item de menu **`Security`** sur la gauche, et cliquer sur **`Security Domains`** à partir du menu déroulant. Cliquer sur le lien **`View`** pour obtenir le domaine de sécurité que vous souhaitez modifier.

2. **Naviguer dans la configuration du sous-système de Mapping**
   Cliquer sur l'étiquette **`Mapping`** en haut de l'affichage si elle n'a pas déjà été sélectionnée.

   La zone de configuration est divisée en deux : **`Modules`** et **`Details`**. Le module de connexion est l'unité de base de configuration. Un domaine de sécurité peut inclure plusieurs polices de mapping, et chacune peut inclure plusieurs attributs et options.

3. **Ajouter un module.**
   Cliquer sur le bouton **`Add`** pour ajouter un module de police de mapping. Remplir les informations pour votre module. Le **`Code`** est le nom de classe du module. Le champ **`Type`** se rapporte au type de mapping effectué par ce module. Les valeurs autorisées sont les suivantes : principal, rôle, attribut ou identifiant.

Une fois que vous aurez ajouté votre module, vous pourrez modifier son **Code** ou ses **Type** en cliquant sur le bouton **Edit** dans la section **Details** de l'écran. Veillez à ce que l'onglet **Attributes** soit bien sélectionné.

4. **Option: ajouter, éditer, ou supprimer un module**
   Si vous avez besoin d'ajouter des options à votre module, cliquer sur leurs entrées dans la liste **Modules**, et sélectionner l'onglet **Module Options** dans la section **Details**. Cliquer sur le bouton **Add**, et fournir la clé et la valeur de l'option. Pour modifier une option existante, supprimez-là en cliquant sur l'étiquette **Remove**, et ajouter la nouvelle valeur. Utiliser le bouton **Remove** pour supprimer une option.

**Résultat**

Votre module de mappage de sécurité est ajouté au domaine de sécurité, et est rendu disponible immédiatement auprès des applications qui utilisent le domaine de sécurité.

Report a bug

### 10.6.11. Utiliser un domaine de sécurité dans votre application

**Aperçu**

Pour utiliser un domaine de sécurité dans votre application, vous devez tout d'abord configurer le domaine dans le fichier de configuration du serveur ou dans le fichier de descripteur de l'application. Ensuite, vous devez ajouter les annotations requises à l'EJB qui l'utilise. Cette rubrique décrit les étapes requises pour utiliser un domaine de sécurité dans votre application.

**Procédure 10.6. Configurer votre application pour qu'elle puisse utiliser un Domaine de sécurité**

1. **Définir le Domaine de sécurité**
   Vous pouvez définir le domaine de sécurité soit dans le fichier de configuration du serveur, soit dans le fichier du descripteur de l'application.

   - **Configurer le domaine de sécurité dans le fichier de configuration du serveur**
     Le domaine de la sécurité est configuré dans le sous-système de **sécurité** du fichier de configuration du serveur. Si l'instance de JBoss Enterprise Application Platform s'exécute dans un domaine géré, il s'agira du fichier **domain/configuration/domain.xml**. Si l'instance de JBoss Enterprise Application Platform s'exécute comme un serveur autonome, ce sera le fichier **standalone/configuration/standalone.xml**.

     Les domaines de sécurité **other**, **jboss-web-policy**, et **jboss-ejb-policy** sont fournis par défaut dans JBoss Enterprise Application Platform 6. L'exemple XML suivant a été copié à partir du sous-système de **sécurité** dans le fichier de configuration du serveur.

```
<subsystem xmlns="urn:jboss:domain:security:1.2">
    <security-domains>
        <security-domain name="other" cache-type="default">
            <authentication>
                <login-module code="Remoting" flag="optional">
                    <module-option name="password-stacking"
value="useFirstPass"/>
                </login-module>
                <login-module code="RealmDirect"
flag="required">
                    <module-option name="password-stacking"
value="useFirstPass"/>
```

```
                </login-module>
            </authentication>
        </security-domain>
        <security-domain name="jboss-web-policy" cache-
type="default">
            <authorization>
                <policy-module code="Delegating"
flag="required"/>
            </authorization>
        </security-domain>
        <security-domain name="jboss-ejb-policy" cache-
type="default">
            <authorization>
                <policy-module code="Delegating"
flag="required"/>
            </authorization>
        </security-domain>
    </security-domains>
</subsystem>
```

Vous pouvez configurer des domaines de sécurité supplémentaires selon les besoins par la Console de gestion ou par le Management CLI.

- **Configurer le domaine de sécurité dans le fichier de descripteur de l'application.**
  Le domaine de sécurité est spécifié dans l'élément enfant **`<security-domain>`** de l'élément **`<jboss-web>`** du fichier **`WEB-INF/jboss-web.xml`** de l'application. L'exemple suivant configure un domaine de sécurité nommé **`my-domain`**.

  ```
  <jboss-web>
      <security-domain>my-domain</security-domain>
  </jboss-web>
  ```

  Il s'agit d'une des configurations que vous pouvez indiquer dans le descripteur **`WEB-INF/jboss-web.xml`**.

2. **Ajouter l'annotation requise à l'EJB.**
   Vous pouvez configurer la sécurité dans EJB par les annotations **`@SecurityDomain`** et **`@RolesAllowed`**. L'exemple de code EJB suivant limite l'accès au domaine de sécurité **`other`** aux utilisateurs ayant pour rôle **`guest`** (invité).

   ```
   package example.ejb3;

   import java.security.Principal;

   import javax.annotation.Resource;
   import javax.annotation.security.RolesAllowed;
   import javax.ejb.SessionContext;
   import javax.ejb.Stateless;

   import org.jboss.ejb3.annotation.SecurityDomain;

   /**
   ```

```
 * Simple secured EJB using EJB security annotations
 * Allow access to "other" security domain by users in a "guest"
role.
 */
@Stateless
@RolesAllowed({ "guest" })
@SecurityDomain("other")
public class SecuredEJB {

   // Inject the Session Context
   @Resource
   private SessionContext ctx;

   /**
    * Secured EJB method using security annotations
    */
   public String getSecurityInfo() {
      // Session context injected using the resource annotation
      Principal principal = ctx.getCallerPrincipal();
      return principal.toString();
   }
}
```

Pour obtenir des exemples de code supplémentaires, voir **`ejb-security`** Quickstart dans le package JBoss Enterprise Application Platform 6 Quickstarts disponible à partir du Portail Clients Red Hat.

Report a bug

### 10.6.12. Java Authorization Contract for Containers (JACC)

#### 10.6.12.1. Java Authorization Contract for Containers (JACC)

Java Authorization Contract for Containers (JACC) est une norme qui définit un contrat entre les conteneurs et les fournisseurs de services d'autorisation, et qui se traduit par l'implémentation de fournisseurs qui seront utilisés par des conteneurs. Il a été défini dans JSR-115, qui se trouve sur le site Web de Java Community Process http://jcp.org/en/jsr/detail?id=115. A fait partie de la spécification Java Enterprise Edition (Java EE) depuis la version 1.3 de Java EE.

JBoss Enterprise Application Platform implémente un support pour JACC dans la fonctionnalité de sécurité du sous-système de sécurité.

Report a bug

#### 10.6.12.2. Configurer la sécurité JACC (Java Authorization Contract for Containers)

Pour configurer JACC (Java Authorization Contract for Containers), vous devez configurer votre domaine de sécurité avec le module qui convient, puis modifiez votre fichier **`jboss-web.xml`** pour y inclure les paramètres qu'il faut.

**Ajouter JACC au Domaine de sécurité**

Pour ajouter JACC au domaine de sécurité, ajouter la police d'autorisation **`JACC`** à la pile d'autorisations du domaine de sécurité, avec l'indicateur **`required`**. Voici un exemple de domaine de sécurité avec support JACC. Cependant, le domaine de sécurité est configuré dans la Console de gestion ou

Management CLI, plutôt que directement dans le code XML.

```
<security-domain name="jacc" cache-type="default">
    <authentication>
        <login-module code="UsersRoles" flag="required">
        </login-module>
    </authentication>
    <authorization>
        <policy-module code="JACC" flag="required"/>
    </authorization>
</security-domain>
```

#### Configurer une application web qui utilise JACC

Le fichier **`jboss-web.xml`** se trouve dans **`META-INF/`** ou dans le répertoire **`WEB-INF/`** de votre déploiement, et contient des ajouts ou remplacements de configuration spécifique JBoss pour le conteneur web. Pour utiliser votre domaine de sécurité activé-JACC, vous devrez inclure l'élément **`<security-domain>`**, et aussi définir l'élément **`<use-jboss-authorization>`** à **`true`**. L'application suivante est configurée correctement pour pouvoir utiliser le domaine de sécurité JACC ci-dessus.

```
<jboss-web>
    <security-domain>jacc</security-domain>
    <use-jboss-authorization>true</use-jboss-authorization>
</jboss-web>
```

#### Configurer une application EJB pour utiliser JACC

La façon de configurer les EJB pour qu'ils utilisent un domaine de sécurité et pour qu'ils utilisent JACC diffère selon de la méthode utilisée pour les applications Web. Pour un EJB, vous pouvez déclarer des *method permissions* (permissions de méthode) sur une méthode ou sur un groupe de méthodes, dans le descripteur **`ejb-jar.xml`**. Dans l'élément **`<ejb-jar>`**, chaque élément **`<method-permission>`** dépendant contient des informations sur les rôles JACC. Voir l'exemple de configuration pour plus d'informations. La classe **`EJBMethodPermission`** fait partie de l'API Java Enterprise Edition 6, et est documentée dans http://docs.oracle.com/javaee/6/api/javax/security/jacc/EJBMethodPermission.html.

**Exemple 10.4. Exemple de Permissions de méthode JACC dans un EJB**

```
<ejb-jar>
  <method-permission>
    <description>The employee and temp-employee roles may access any
method of the EmployeeService bean </description>
    <role-name>employee</role-name>
    <role-name>temp-employee</role-name>
    <method>
      <ejb-name>EmployeeService</ejb-name>
      <method-name>*</method-name>
    </method>
  </method-permission>
</ejb-jar>
```

Vous pouvez également contraindre les mécanismes d'authentification et d'autorisation d'un EJB à l'aide d'un domaine de sécurité, comme vous pouvez le faire pour une application web. Les domaines de sécurité sont déclarés dans le descripteur **jboss-ejb3.xml** qui se trouve dans l'élément dépendant **<security>**. En plus du domaine de sécurité, vous pouvez également spécifier le*run-as principal*, qui change le principal que l'EJB exécute.

**Exemple 10.5. Exemple de déclaration de domaine de sécurité dans un EJB**

```
<security>
  <ejb-name>*</ejb-name>
  <security-domain>myDomain</s:security-domain>
  <run-as-principal>myPrincipal</s:run-as-principal>
</s:security>
```

Report a bug

### 10.6.13. Java Authentication SPI for Containers (JASPI)

#### 10.6.13.1. Sécurité Java Authentication SPI pour Conteneurs (JASPI)

Java Application SPI pour Conteneurs (JASPI or JASPIC) est une interface enfichable pour applications JSR-196 du Java Community Process. Consulter http://www.jcp.org/en/jsr/detail?id=196 pour obtenir des informations sur la spécification.

Report a bug

#### 10.6.13.2. Configuration de la Sécurité Java Authentication SPI pour Conteneurs (JASPI)

Pour s'authentifier auprès d'un fournisseur JASPI, ajouter un élément **<authentication-jaspi>** à votre domaine de sécurité. La configuration est similaire à celle d'un module d'authentification standard, mais les éléments de module de login sont inclus dans l'élément **<login-module-stack>**. La structure de configuration est la suivante :

**Exemple 10.6. Structure de l'élément `authentication-jaspi`**

```
<authentication-jaspi>
 <login-module-stack name="...">
   <login-module code="..." flag="...">
     <module-option name="..." value="..."/>
   </login-module>
 </login-module-stack>
 <auth-module code="..." login-module-stack-ref="...">
   <module-option name="..." value="..."/>
 </auth-module>
</authentication-jaspi>
```

Le module de connexion est lui-même configuré de la même façon que le module d'authentification standard.

Comme la console de gestion basée web n'expose pas la configuration des modules d'authentification JASPI, vous devez stopper la plateforme JBoss Enterprise Application Platform complètement avant d'ajouter la configuration directement dans le fichier ***EAP_HOME*/domain/configuration/domain.xml** ou dans le fichier ***EAP_HOME*/standalone/configuration/standalone.xml**.

Report a bug

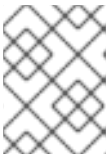
## 10.7. MANAGEMENT INTERFACE SECURITY

### 10.7.1. Configuration Sécurité Utilisateur par défaut

#### Introduction

Toutes les interfaces de gestion de JBoss Enterprise Application Platform 6 sont sécurisées par défaut. Cette sécurité existe sous deux formes :

- Les interfaces locales sont sécurisées par un contrat SASL entre des clients locaux et le serveur auquel ils se connectent. Ce mécanisme de sécurité repose sur la capacité du client à accéder au système de fichiers local. C'est parce que l'accès au système de fichiers local permettrait au client d'ajouter un utilisateur ou encore de modifier la configuration pour déjouer les autres mécanismes de sécurité. Cela est conforme au principe selon lequel si l'accès physique au système de fichiers est réussi, les autres mécanismes de sécurité sont superflus. Le mécanisme passe par quatre étapes :

  **NOTE**

  L'accès HTTP est considéré comme éloigné, même si vous vous connectez à l'hôte local par HTTP.

  1. Le client envoie un message au serveur incluant une requête pour authentifier le mécanisme SASL local.
  2. Le serveur génère un token unique, qu'il écrit dans un fichier unique, et envoie un message au client avec le chemin d'accès fichier complet.
  3. Le client lit le token dans le fichier et l'envoie au serveur, pour vérifier qu'il a l'accès local au système de fichiers.
  4. Le serveur vérifie le token et efface le fichier.

- Les clients distants, y compris HTTP, utilisent une sécurité basée domaine. Le domaine par défaut qui comprend les autorisations pour configurer la plate-forme JBoss EAP 6 à distance en utilisant les interfaces de gestion est **ManagementRealm**. Un script est fourni pour vous permettre d'ajouter des utilisateurs à ce domaine (ou à des domaines que vous créez). Pour plus d'informations sur la façon d'ajouter des utilisateurs, voir le chapitre Getting Started (Guide de démarrage) de la plateforme JBoss EAP 6. Pour chaque utilisateur, le nom d'utilisateur, un mot de passe haché et le domaine sont stockés dans un fichier.

  **Serveur autonome**

  ```
  JPP_HOME/standalone/configuration/mgmt-users.properties
  ```

-

Même si les contenus de **mgmt-users.properties** sont masqués, le fichier doit toujours être considéré comme un fichier sensible. Il est recommandé qu'il soit défini sur le mode de fichier **600**, qui ne donne aucun accès autre que l'accès en lecture et écriture au propriétaire du fichier.

Report a bug

### 10.7.2. Aperçu Général de la Configuration de l'Interface de Gestion avancée

La configuration de l'interface de Gestion ***EAP_HOME*/domain/configuration/host.xml** or ***EAP_HOME*/standalone/configuration/standalone.xml** contrôle laquelle de vos interfaces réseau lie le processus contrôleur hôte, les types d'interfaces de gestion qui sont disponibles à tous, et quel type de système d'authentification est utilisé pour authentifier les utilisateurs sur chaque interface. Cette rubrique explique comment configurer les interfaces de gestion en fonction de votre environnement.

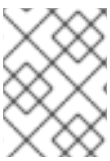
Le sous-système de Gestion est formé d'un élément **<management>** avec plusieurs attributs configurables, et les trois éléments enfants configurables suivants. Les domaines de sécurité et les connexions sortantes sont tout d'abord définis, puis appliqués aux interfaces de gestion en tant qu'attributs.

- **<security-realms>**
- **<outbound-connections>**
- **<management-interfaces>**

#### Domaines de sécurité

Le domaine de sécurité est chargé de l'authentification et de permettre aux utilisateurs autorisés d'administrer JBoss Enterprise Application Platform via l'API de gestion, le Management CLI ou la Console de gestion basée-web.

Il existe deux domaines de sécurité basés fichier différents qui existent dans l'installation par défaut : **ManagementRealm** et **ApplicationRealm**. Chacun de ces domaines de sécurité utilise un fichier **-users.properties** pour stocker les utilisateurs et les mots de passe de hachage, et **-roles.properties** pour stocker les correspondances entre les utilisateurs et les rôles. Un support est également inclus pour le domaine de sécurité activé-LDAP.



**NOTE**

Les domaines de sécurité peuvent également être utilisés pour vos propres applications. Les domaines de sécurité dont il s'agit sont particuliers aux interfaces de gestion.

#### Les connexions de sortie

Certains domaines de sécurité se connectent à des interfaces externes, comme un serveur LDAP. Une connexion sortante définit la manière d'établir cette connexion. Un type de connexion prédéfinie, la connexion **ldap-connection**, définit tous les attributs obligatoires et facultatifs pour se connecter au serveur LDAP et vérifier les informations d'identification.

#### Interfaces de gestion

Une interface de gestion comprend des propriétés sur la façon de connecter et configurer JBoss Enterprise Application Platform. Ces informations comprennent l'interface réseau nommée, le port, le

domaine de sécurité et autres informations configurables sur l'interface. Deux interfaces sont incluses dans une installation par défaut :

- **`http-interface`** est la configuration de la console de gestion basée-web.
- **`native-interface`** est la configuration pour la Management CLi en ligne de commande et l'API de gestion Rest-like.

Chacun des trois principaux éléments configurables du sous-système de gestion de l'hôte sont étroitement liés. Un domaine de sécurité fait référence à une connexion sortante et une interface de gestion à un domaine de sécurité.

Report a bug

### 10.7.3. LDAP

Lightweight Directory Access Protocol (LDAP) est un protocole pour le stockage et la distribution d'informations en provenance de répertoires à travers un réseau. Ces informations de répertoires incluent des informations sur les utilisateurs, les périphériques physiques, les rôles d'accès et de restrictions et autres informations.

Certaines de implémentations communes de LDAP incluent OpenLDAP, Microsoft Active Directory, IBM Tivoli Directory Server, Oracle Internet Directory, et autres.

La plateforme JBoss Enterprise Application Platform comprend plusieurs modules d'authentification et d'autorisation qui vous permettent d'utiliser un serveur LDAP comme autorité d'authentification et d'autorisation pour vos applications Web et EJB.

Report a bug

### 10.7.4. Utiliser LDAP pour vous authentifier auprès des interfaces de Gestion

Pour utiliser un serveur de répertoire LDAP comme source d'authentification pour la Console de gestion, le Management CLI ou l'API de gestion, vous devez effectuer les procédures suivantes :

1. Créer une connexion sortante au serveur LDAP.
2. Créer un domaine de sécurité activé-LDAP.
3. Référencer le nouveau domaine de sécurité dans l'interface de Gestion.

#### Créer une Connexion sortante au serveur LDAP

La connexion sortante LDAP autorise les attributs suivants :

**Tableau 10.1. Attributs d'une Connexion sortante LDAP**

| Attribut | Requis | Description |
|---|---|---|
| Le nom | oui | Le nom qui sert à identifier cette connexion. Ce nom est utilisé dans la définition du domaine de sécurité. |
| url | oui | Ĺ'adresse URL du serveur de répertoires. |

| Attribut | Requis | Description |
|---|---|---|
| search-dn | oui | Le nom unique (DN) de l'utilisateur autorisé à effectuer des recherches. |
| search-credentials | oui | Le mot de passe de l'utilisateur autorisé à effectuer des recherches. |
| initial-context-factory | non | L'usine de contexte initiale à utiliser quand on établit une connexion. Valeur par défaut **`com.sun.jndi.ldap.LdapCtxFactory`**. |

**Exemple 10.7. Ajouter une connexion sortante LDAP**

Cette exemple ajoute une connexion sortante par le jeu de propriétés suivant :

- Search DN: **`cn=search,dc=acme,dc=com`**
- Search Credential: **`myPass`**
- URL: **`ldap://127.0.0.1:389`**

```
/host=master/core-service=management/ldap-
connection=ldap_connection/:add(search-
credential=myPass,url=ldap://127.0.0.1:389,search-
dn="cn=search,dc=acme,dc=com")
```

**Exemple 10.8. XML représentant un connexion sortante LDAP**

```
<outbound-connections>
   <ldap name="ldap_connection" url="ldap://127.0.0.1:389" search-
dn="cn=search,dc=acme,dc=com" search-credential="myPass" />
</outboundconnections>
```

### Créer un domaine de sécurité activé-LDAP

Les Interfaces de gestion peuvent authentifier sur le serveur LDAP au lieu des domaines de sécurité basés propriété-fichier et configurés par défaut. L'authentificateur LDAP fonctionne en établissant tout d'abord une connexion au serveur de répertoires distant. Il effectue ensuite une recherche en utilisant le nom d'utilisateur que l'utilisateur a transmis au système d'authentification, afin de trouver le nom unique complet (DN) du dossier LDAP. Une nouvelle connexion est alors établie, utilisant le DN de l'utilisateur comme informations d'identification et mot de passe fournis par l'utilisateur. Si cette authentification au serveur LDAP réussit, le DN est considéré comme valide.

Le domaine de sécurité LDAP utilise les éléments et attributs de configuration suivants pour pouvoir effectuer ses fonctions.

**connection**

Le nom de la connexion définie dans **<outbound-connections>** à utiliser pour se connecter au répertoire LDAP.

**base-dn**

Le nom unique (DN) du contexte pour commencer à chercher l'utilisateur.

**recursive**

Indique si la recherche doit être récursive dans toute l'arborescence de répertoires LDAP, ou si l'on doit rechercher uniquement le contexte spécifié. La valeur par défaut est **false**.

**user-dn**

Attribut de l'utilisateur qui détient le nom unique (DN). Utilisé par la suite pour tester l'authentification. Valeur par défaut **dn**.

**Soit username-filter ou advanced-filter, comme élément enfant.**

Le **username-filter** utilise un attribut unique nommé **attribute**, dont la valeur correspond au nom de l'attribut LDAP qui contient le nom d'utilisateur, comme **userName** ou **sambaAccountName**.

Le **advanced-filter** prend un attribut unique nommé **filter**, qui contient une recherche de filtre en syntaxe LDAP standard. Veillez à bien échapper les caractères **&** en commutant à **& amp;**. Un exemple de filtre est :

```
(&(sAMAccountName={0})(memberOf=cn=admin,cn=users,dc=acme,dc=com))
```

Après avoir échappé un caractère esperluette, le filtre apparaîtra ainsi :

```
(&amp;(sAMAccountName={0})(memberOf=cn=admin,cn=users,dc=acme,dc=com))
```

**Exemple 10.9. XML représentant un Domaine de sécurité activé-LDAP**

Cet exemple utilise les paramètres suivants :

- connection - **ldap_connection**
- base-dn - **cn=users,dc=acme,dc=com**.
- username-filter - **attribute="sambaAccountName"**

```
<security-realm name="TestRealm">
   <authentication>
      <ldap connection="ldap_connection" base-
dn="cn=users,dc=acme,dc=com">
         <username-filter attribute="sambaAccountName" />
      </ldap>
  </authentication>
```

```
</security-realm>
```

**AVERTISSEMENT**

Il est important de veiller à ne pas autoriser les mots de passe LDAP. À moins que vous désiriez particulièrement les avoir dans votre environnement, ils représentent un problème de sécurité sérieux.

EAP 6.1 inclut un correctif pour CVE-2012-5629, qui définit l'option allowEmptyPasswords des modules de connexion LDAP à false si l'option n'est pas déjà configurée. Pour les versions plus anciennes, cette option devra être configurée manuellement.

**Exemple 10.10. Ajout d'un Domaine de sécurité LDAP**

La commande ci-dessous ajoute un domaine de sécurité et définit ses attributs pour un serveur autonome.

```
/host=master/core-service=management/security-
realm=ldap_security_realm/authentication=ldap:add(base-
dn="DC=mycompany,DC=org", recursive=true, username-
attribute="MyAccountName", connection="ldap_connection")
```

**Appliquer le Nouveau domaine de sécurité à l'Interface de gestion**

Après avoir créé un domaine de sécurité, vous devez le référencer dans la configuration de votre interface de gestion. L'interface de gestion utilisera le domaine de sécurité pour l'authentification HTTP digest.

**Exemple 10.11. Ajouter le Domaine de sécurité à l'interface HTTP**

Une fois que la configuration est en place, et que vous aurez démarré à nouveau le contrôleur d'hôtes, la Console de gestion basée-web utilisera LDAP pour authentifier ses utilisateurs.

```
/host=master/core-service=management/management-interface=http-
interface/:write-attribute(name=security-realm,value=TestRealm)
```

Report a bug

### 10.7.5. Disable the HTTP Management Interface

Dans un domaine géré, vous avez seulement besoin d'un accès à l'interface HTTP sur le contrôleur de domaine, plutôt que sur des serveurs membres de domaine. En outre, sur un serveur de production, vous pouvez finalement décider de désactiver la Console de gestion basée-web.

**NOTE**

Les autres clients, tels que JBoss Operations Network, opèrent également par l'interface HTTP. Si vous souhaitez utiliser ces services ou tout simplement désactiver la gestion de la Console elle-même, vous pouvez définir l'attribut **`console-enabled-attribute`** de l'interface HTTP à **`false`**, au lieu de désactiver l'interface complètement.

```
/host=master/core-service=management/management-interface=http-
interface/:write-attribute(name=console-enabled,value=false)
```

Pour désactiver l'interface HTTP, ce qui désactive également l'accès à la Console de gestion basée-web, vous pouvez finalement supprimer l'interface HTTP.

La commande JBoss CLI suivante vous permettra de lire le contenu actuel de votre interface HTTP, si vous décidez de l'ajouter à nouveau.

**Exemple 10.12. Lire la configuration de l'interface HTTP**

```
/host=master/core-service=management/management-interface=http-
interface/:read-resource(recursive=true,proxies=false,include-
runtime=false,include-defaults=true)
{
    "outcome" => "success",
    "result" => {
        "console-enabled" => true,
        "interface" => "management",
        "port" => expression "${jboss.management.http.port:9990}",
        "secure-port" => undefined,
        "security-realm" => "ManagementRealm"
    }
}
```

Pour supprimer l'interface HTTP, lancer la commande suivante :

**Exemple 10.13. Supprimer l'interface HTTP**

```
/host=master/core-service=management/management-interface=http-
interface/:remove
```

Pour activer l'accès à nouveau, lancer la commande suivante pour recréer l'Interface HTTP avec les valeurs par défaut.

**Exemple 10.14. Recréer l'Interface HTTP**

```
/host=master/core-service=management/management-interface=http-
interface/:write-attribute(name=console-enabled,value=true)
```

```
/host=master/core-service=management/management-interface=http-
interface/:write-attribute(name=interface,value=management)
```

```
/host=master/core-service=management/management-interface=http-
interface/:write-
attribute(name=port,value=${jboss.management.http.port:9990})
```

```
/host=master/core-service=management/management-interface=http-
interface/:write-attribute(name=security-realm,value=ManagementRealm)
```

Report a bug

### 10.7.6. Supprimer l'Authentification silencieuse du Domaine de sécurité par défaut.

**Résumé**

L'installation par défaut de JBoss Enterprise Application Platform 6 contient une méthode d'authentification silencieuse pour un utilisateur de Management LCI local. Cela donne à l'utilisateur local la possibilité d'accéder au Management CLI sans authentification du nom d'utilisateur ou du mot de passe. Cette fonctionnalité est activée dans un but pratique et pour faciliter l'exécution des scripts de Management CLI sans exiger l'authentification des utilisateurs locaux.Cette méthode est considérée comme une caractéristique utile étant donné que l'accès à la configuration locale donne généralement aussi à l'utilisateur la possibilité d'ajouter ses propres détails utilisateur ou sinon la possibilité de désactiver les contrôles de sécurité

La commodité de l'authentification silencieuse des utilisateurs locaux peut être désactivée quand un plus grand contrôle de sécurité est requis. Ceci peut être réalisé en supprimant l'élément **`local`** au sein de la section **`security-realm`** du fichier de configuration. S'applique à la fois pour **`standalone.xml`** pour une instance de serveur autonome, ou pour **`host.xml`** pour un domaine géré. Vous ne devriez envisager de supprimer que l'élément **`local`** si vous comprenez l'impact que ceci pourrait avoir sur la configuration de votre serveur particulier.

La meilleure méthode pour désactiver l'authentification silencieuse est d'utiliser le Management CLI, qui retire l'élément **`local`** visible dans l'exemple suivant.

**Exemple 10.15. Exemple d'élément `local` dans `security-realm`**

```
<security-realms>
    <security-realm name="ManagementRealm">
        <authentication>
            <local default-user="$local"/>
            <properties path="mgmt-users.properties" relative-
to="jboss.server.config.dir"/>
        </authentication>
    </security-realm>
    <security-realm name="ApplicationRealm">
        <authentication>
            <local default-user="$local" allowed-users="*"/>
            <properties path="application-users.properties" relative-
to="jboss.server.config.dir"/>
        </authentication>
        <authorization>
```

```
            <properties path="application-roles.properties" relative-
to="jboss.server.config.dir"/>
        </authorization>
    </security-realm>
</security-realms>
```

**Procédure 10.7. Supprimer l'Authentification silencieuse du Domaine de sécurité par défaut.**

- **Supprimer l'authentification silencieuse par le Management CLI**
  Supprimer l'élément **`local`** du Domaine de gestion et du Domaine d'applications comme requis.

  a. Supprimer l'élément **`local`** du Domaine de gestion.

  - **Pour les serveurs autonomes**

    ```
    /core-service=management/security-
    realm=ManagementRealm/authentication=local:remove
    ```

  - **Pour les domaines gérés**

    ```
    /host=HOST_NAME/core-service=management/security-
    realm=ManagementRealm/authentication=local:remove
    ```

  b. Supprimer l'élément **`local`** du Domaine d'applications.

  - **Pour les serveurs autonomes**

    ```
    /core-service=management/security-
    realm=ApplicationRealm/authentication=local:remove
    ```

  - **Pour les domaines gérés**

    ```
    /host=HOST_NAME/core-service=management/security-
    realm=ApplicationRealm/authentication=local:remove
    ```

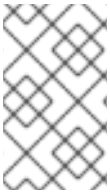
**Résultat**

L'authentification silencieuse est retirée du **`ManagementRealm`** et de l' **`ApplicationRealm`**.

Report a bug

### 10.7.7. Désactiver l'accès à distance du Sous-système JMX

Une connectivité à distance JMX vous permet de déclencher JDK et les opérations de gestion d'applications. Afin de garantir une installation, désactivez cette fonction. Vous pouvez faire cela soit en enlevant la configuration de la connexion à distance, ou en enlevant le sous-système JMX entièrement. Les commandes du JBoss CLI référencent le profil par défaut dans une configuration de domaine géré. Pour modifier un profil différent, modifiez la partie de la commande **`/profil=default`**. Pour un serveur autonome, retirez complètement cette partie de la commande.



**NOTE**

Dans un domaine géré, le connecteur distant est retiré du sous-système JMX par défaut. Cette commande est fournie pour votre information, au cas où vous souhaiteriez l'ajouter en cours de développement.

**Exemple 10.16. Supprimer le Connecteur distant du Sous-système JMX**

```
/profile=default/subsystem=jmx/remoting-connector=jmx/:remove
```

**Exemple 10.17. Supprimer le Sous-système JMX**

Exécuter cette commande pour chaque profil que vous utilisez, si vous utilisez un domaine géré.

```
/profile=default/subsystem=jmx/:remove
```

Report a bug

### 10.7.8. Configurer les Domaines de sécurité pour les Interfaces de gestion

Les interfaces de gestion utilisent des domaines de sécurité pour contrôler l'authentification et l'accès aux mécanismes de configuration de JBoss Enterprise Application Platform. Cette section vous montre comment lire et configurer les domaines de sécurité. Ces commandes utilisent le Management CLI.

**Lire une configuration de domaine de sécurité**

Cet exemple montre la configuration par défaut du domaine de sécurité du **`ManagementRealm`**. Utilise un fichier **`mgmt-users.properties`** qui contient ses informations de configuration.

**Exemple 10.18. ManagementRealm par défaut**

```
 /host=master/core-service=management/security-
realm=ManagementRealm/:read-
resource(recursive=true,proxies=false,include-runtime=false,include-
defaults=true)
{
    "outcome" => "success",
    "result" => {
        "authorization" => undefined,
        "server-identity" => undefined,
        "authentication" => {"properties" => {
            "path" => "mgmt-users.properties",
            "plain-text" => false,
            "relative-to" => "jboss.domain.config.dir"
        }}
    }
}
```

**Rédiger un domaine de sécurité**

Les commandes suivantes créent un nouveau domaine de sécurité nommé **`TestRealm`** et définissent le nom et le répertoire du fichier de propriétés qui conviennent.

**Exemple 10.19. Rédaction d'un domaine de sécurité**

```
/host=master/core-service=management/security-
realm=TestRealm/:add/host=master/core-service=management/security-
realm=TestRealm/authentication=properties/:add(path=TestUsers.properties
, relative-to=jboss.domain.config.dir)
```

**Ajouter le Domaine de sécurité à l'interface de gestion**

Après avoir ajouté un domaine de sécurité, donner son nom comme référence à l'interface de gestion.

**Exemple 10.20. Ajouter un Domaine de sécurité à une interface de gestion**

```
host=master/core-service=management/management-interface=http-
interface/:write-attribute(name=security-realm,value=TestRealm)
```

Report a bug

## 10.8. SÉCURITÉ DE RÉSEAU

### 10.8.1. Sécuriser les interfaces de gestion

**Résumé**

Dans un environnement de test, il est de pratique courante de faire fonctionner JBoss Enterprise Application Platform 6 sans couche de sécurité sur les interfaces de gestion, composées de la Console de gestion, du Management CLI ou autre implémentation de l'API. Cela permet des changements rapides de développement et de configuration.

De plus, un mode silencieux d'authentification est présent par défaut, permettant à un client local sur une machine hôte de se connecter au Management CLI sans exiger un nom d'utilisateur ou un mot de passe. Ce comportement est pratique pour les utilisateurs locaux et les scripts de Management CLI, mais peut être désactivé si nécessaire. La procédure est décrite dans la rubrique .

Quand vous commencez à tester ou à préparer votre environnement pour aller en production, il est vitalement important de sécuriser les interfaces de gestion par l'une des méthodes suivantes au moins:

- Section 10.8.2, « Indiquer les interfaces réseau que la plateforme JBoss EAP utilise »
- Section 10.8.3, « Configurer les pare-feux de réseau pour qu'ils soient opérationnels dans JBoss Enterprise Application Platform 6 »

Report a bug

### 10.8.2. Indiquer les interfaces réseau que la plateforme JBoss EAP utilise

**Aperçu**

Isoler les services afin qu'ils soient accessibles uniquement aux clients qui en ont besoin augmente la sécurité de votre réseau. JBoss Enterprise Application Platform inclut deux interfaces dans sa configuration par défaut, qui se lient toutes les deux à l'adresse IP **`127.0.0.1`** ou **`localhost`**, par défaut. Une des interfaces est appelée **`management`** et est utilisée par les consoles de CLI et API de gestion. L'autre est appelée **`public`** et est utilisée pour déployer des applications. Ces interfaces ne sont pas spéciales, ni importantes, mais sont fournies comme point de départ.

L'interface **`management`** utilise les ports 9990 et 9999 par défaut, et l'interface **`public`** utilise le port 8080, ou le port 8443 avec HTTPS.

Vous pouvez changer l'adresse IP de l'interface de gestion, de l'interface publique ou bien les deux à la fois.

**AVERTISSEMENT**

Si vous exposez les interfaces de gestion à d'autres interfaces de réseau non accessibles par les hôtes distants, vérifiez les implications au niveau de la sécurité. Le plus souvent, il n'est pas conseillé de donner un accès à distance aux interfaces de gestion.

1. **Stopper JBoss Enterprise Application Platform.**
   Arrêter JBoss Enterprise Application Platform en interrompant votre système d'exploitation de manière appropriée. Si vous exécutez Enterprise Application Platform comme une application de premier plan, vous devrez appuyer sur **`Ctrl+C`** .

2. **Démarrer à nouveau JBoss Enterprise Application Platform, en indiquant l'adresse de liaison.**
   Utilisez la ligne de commande **`-b`** pour démarrer JBoss Enterprise Application Platform sur une interface particulière.

   **Exemple 10.21. Indiquer l'interface publique.**

   ```
   EAP_HOME/bin/domain.sh -b 10.1.1.1
   ```

   **Exemple 10.22. Indiquer l'interface de gestion.**

   ```
   EAP_HOME/bin/domain.sh -bmanagement=10.1.1.1
   ```

   **Exemple 10.23. Indiquer des adresses différentes pour chaque interface.**

   ```
   EAP_HOME/bin/domain.sh -bmanagement=127.0.0.1 -b 10.1.1.1
   ```

   **Exemple 10.24. Lier l'interface publique à toutes les interfaces de réseau.**

```
EAP_HOME/bin/domain.sh -b 0.0.0.0
```

Il est possible de modifier votre fichier de configuration XML directement, pour changer les adresses de liaison par défaut. Toutefois, si vous faites cela, vous ne serez plus en mesure d'utiliser la commande **-b** pour spécifier une adresse IP en cours d'exécution, donc ce n'est pas recommandé. Si vous décidiez de le faire, n'oubliez pas d'arrêter JBoss Enterprise Application Platform complètement avant d'éditer le fichier XML.

Report a bug

### 10.8.3. Configurer les pare-feux de réseau pour qu'ils soient opérationnels dans JBoss Enterprise Application Platform 6

**Résumé**

La plupart des environnements de production utilisent des pare-feux pour la stratégie globale de sécurité réseau. Si vous avez besoin de plusieurs instances de serveurs pour communiquer les uns avec les autres ou avec des services externes tels que des serveurs web ou des bases de données, votre pare-feu doit en tenir compte. Un pare-feu bien géré ouvre seulement les ports qui sont utiles à l'opération et limite l'accès aux ports à des adresses IP spécifiques, des sous-réseaux et des protocoles réseau.

Une discussion plus complète sur les pare-feux est au delà du dessein de cette documentation.

**Prérequis**

- Détermine les ports que vous avez besoin d'ouvrir. Voir Section 10.8.4, « Ports de réseau utilisés par la plateforme JBoss EAP 6 » pour déterminer la liste des ports pour votre situation.
- Vous devez avoir une bonne compréhension de vos logiciels de pare-feux. Cette procédure utilise la commande **system-config-firewall** de Red Hat Enterprise Linux 6. Microsoft Windows Server inclut un pare-feu intégré, et plusieurs solutions de pare-feux de tierce partie existent pour chaque plate-forme.

**Hypothèses**

Cette procédure configure un pare-feu dans un environnement qui comprend les hypothèses suivantes :

- Le système d'exploitation est Red Hat Enterprise Linux 6
- La plate-forme JBoss EAP 6 exécute sur l'hôte **10.1.1.2**. En option, le serveur peut posséder son propre pare-feu.
- Le serveur du pare-feu de réseau exécute sur l'hôte **10.1.1.1** sur l'interface **eth0**, et possède une interface externe **eth1**.
- Le traffic de réseau devra être redirigé vers le port 5445 (port utilisé par JMS) renvoyé sur EAP. Le traffic ne doit pas pouvoir transiter par le pare-feu du réseau.

**Procédure 10.8. Gérer les pare-feux de réseau pour qu'ils soient opérationnels dans JBoss Enterprise Application Platform 6**

1. **Connectez-vous à la Console de gestion.**
   Connectez-vous à la Console de gestion. Par défaut, exécute sur http://localhost:9990/console/.

2. **Déterminer les liaisons de socket utilisées par le groupe de liaisons de socket.**
Cliquer sur l'étiquette **`Profiles`** qui se trouve en haut et à droite de la Console de gestion. Sur la gauche de l'écran, vous verrez une série de menus. Le titre en bas du menu est **`General Configuration`** (Configuration générale). Cliquer sur **`Socket Binding Groups`** sous ce titre. L'écran **`Socket Binding Declarations`** apparaîtra. Pour commencer, vous verrez le groupe **`standard-sockets`**. Vous pourrez choisir un autre groupe en le sélectionnant de la case mixte à droite.

**NOTE**

Si vous utilisez un serveur autonome, il ne possédera qu'un seul groupe de liaisons de socket.

La liste de noms de sockets et des ports apparaît, avec six valeurs par page. Vous pourrez naviguer entre les pages grâce à la flèche de navigation en dessous du tableau.

3. **Déterminer les ports que vous souhaitez ouvrir.**
Suivant la fonction d'un port particulier, et suivant les besoins de votre environnement, certains ports devront sans doute être disponibles en dépit du pare-feu. Si vous n'êtes pas certain du but de la liaison de socket, voir Section 10.8.4, « Ports de réseau utilisés par la plateforme JBoss EAP 6 » pour obtenir une liste des liaisons de socket par défaut, et leur but.

4. **Configurer votre pare-feu pour rediriger le trafic réseau vers la plateforme JBoss EAP 6.**
Procédez à ces étapes de configuration de votre pare-feu de réseau pour permettre au trafic de se diriger vers le port désiré.

   a. Connectez-vous au pare-feu de votre machine, et accéder à cette commande, en tant qu'utilisateur root.

   b. Saisir la commande **`system-config-firewall`** pour lancer l'utilitaire de configuration du pare-feu. Un GUI ou Utilitaire de ligne de commande opérera, selon la façon dont vous êtes connecté au système de pare-feu. Cette tâche assume que vous êtes connecté via SSH et que vous utilisez l'interface de ligne de commande.

   c. Utiliser la clé **TAB** de votre clavier pour naviguer vers le bouton **`Customize`**, puis appuyer sur la clé **ENTER**. L'écran**`Trusted Services`** apparaîtra.

   d. Ne changez aucune valeur, mais utilisez la clé **TAB** pour naviguer vers le bouton **`Forward`**, puis, appuyer sur **ENTER** pour aller vers le prochain écran. L'écran **`Other Ports`** apparaîtra.

   e. Utiliser la clé **TAB** pour naviguer vers le bouton **<Add>**, puis appuyer sur la clé **ENTER**. L'écran **`Port and Protocol`** apparaîtra.

   f. Saisissez **5445** dans le champ **`Port / Port Range`**, puis utilisez la clé **TAB** pour vous rendre dans le champ **`Protocol`**, puis saisissez **`tcp`**. Utilisez la clé **TAB** pour naviguer vers le bouton **OK**, puis appuyez sur **ENTER**.

   g. Utilisez la clé **TAB** pour naviguer vers le bouton **`Forward`** jusqu'à ce que vous atteigniez **`Port Forwarding`**.

   h. Utiliser la clé **TAB** pour naviguer vers le bouton **<Add>**, puis appuyer sur la clé **ENTER**.

   i. Remplir les valeurs suivantes pour définir la redirection de port vers port 5445.

- Interface source: eth1
- Protocol: tcp
- Port / Port Range: 5445
- Destination IP address: 10.1.1.2
- Port / Port Range: 5445

Utiliser la clé **TAB** pour naviguer vers le bouton **OK**, puis appuyer sur la clé **ENTER**.

j. Utiliser la clé **TAB** pour naviguer vers le bouton **Close**, puis appuyer sur la clé **ENTER**.

k. Utiliser la clé **TAB** pour naviguer vers le bouton **OK**, puis appuyer sur **ENTER**. Pour appliquer les changements, lire la notice d'avertissement, puis appuyer sur **Yes**.

5. **Configurer un pare-feu sur votre hôte de plateforme JBoss EAP 6.**
Certaines organisations choisissent de configurer un pare-feu sur le serveur JBoss EAP 6 lui-même et de fermer tous les ports qui ne sont pas utiles à son fonctionnement. Consulter Section 10.8.4, « Ports de réseau utilisés par la plateforme JBoss EAP 6 » pour déterminer quels ports ouvrir, puis fermer le reste. La configuration par défaut de Red Hat Enterprise Linux 6 ferme tous les ports sauf 22 (utilisé pour Secure Shell (SSH) et 5353 (utilisé pour la multi-diffusion DNS). Si vous configurez les ports, assurez-vous que vous avez un accès physique à votre serveur pour ne pas, par inadvertance, vous verrouiller vous-même.

**Résultat**

Votre pare-feu est configuré pour renvoyer le trafic vers votre serveur JBoss EAP 6 interne, de la façon dont vous avez spécifié dans la configuration de votre pare-feu. Si vous avez choisi d'activer un pare-feu sur votre serveur JBoss Enterprise Application Platform 6, tous les ports seront fermés sauf ceux nécessaires à l'exécution de vos applications.

Report a bug

## 10.8.4. Ports de réseau utilisés par la plateforme JBoss EAP 6

Les ports de réseau utilisés par la configuration par défaut de la plateforme JBoss EAP 6 dépendent de plusieurs facteurs:

- Le fait que vos groupes de serveurs utilisent le groupe de liaisons de sockets par défaut , ou un groupe de liaisons de sockets personnalisé.
- Des exigences de vos déploiements individuels.

**NOTE**

Un décalage de port numérique peut être configuré pour atténuer les conflits de ports lorsque vous exécutez plusieurs serveurs sur un même serveur physique. Si votre serveur utilise un décalage de port numérique, ajouter le décalage au numéro de port par défaut pour le groupe de liaisons de socket de son groupe de serveurs. Par exemple, si le port HTTP du groupe de liaisons de socket est 8080 et si votre serveur utilise un décalage de port de 100, son port HTTP sera 8180.

À moins d'instruction particulière, les ports utilisent le protocole TCP.

**Groupes de liaison de socket par défaut**

- `full-ha-sockets`
- `full-sockets`
- `ha-sockets`
- `standard-sockets`

**Tableau 10.2. Référence aux Groupes de liaison de socket par défaut**

| Nom | Port | Port Multidiffusion | Description | full-ha-sockets | full-sockets | ha-socket | standard-socket |
|---|---|---|---|---|---|---|---|
| `ajp` | 8009 | | Protocole Apache JServ. Utilisé pour le clustering HTTP et pour l'équilibrage des charges. | Oui | Oui | Oui | Oui |
| `http` | 8080 | | Le port par défaut des applications déployées. | Oui | Oui | Oui | Oui |
| `https` | 8443 | | Connexion cryptée-SSL entre les applications déployées et les clients. | Oui | Oui | Oui | Oui |
| `jacorb` | 3528 | | Services CORBA pour les transactions JTS et autres services dépendants-ORB. | Oui | Oui | Non | Non |
| `jacorb-ssl` | 3529 | | Services CORBA cryptés-SSL. | Oui | Oui | Non | Non |
| `jgroups-diagnostics` | | 7500 | Multicast. Utilisé pour la découverte de paires dans les groupements HA. | Oui | Non | Oui | Non |
| `jgroups-mping` | | 45700 | Multicast. Utilisé pour découvrir l'appartenance de groupe d'orgine dans un cluster HA. | Oui | Non | Oui | Non |

| Nom | Port | Port Multidiffusion | Description | full-ha-sockets | full-sockets | ha-socket | standard-socket |
|---|---|---|---|---|---|---|---|
| **jgroups-tcp** | 7600 | | Découverte de paires unicast dans les groupements HA avec TCP. | Oui | Non | Oui | Non |
| **jgroups-tcp-fd** | 57600 | | Utilisé pour la détection des échecs en TCP. | Oui | Non | Oui | Non |
| **jgroups-udp** | 55200 | 45688 | Découverte de paires unicast dans les groupements HA avec TCP. | Oui | Non | Oui | Non |
| **jgroups-udp-fd** | 54200 | | Utilisé pour la détection des échecs par UDP. | Oui | Non | Oui | Non |
| **messaging** | 5445 | | Service JMS. | Oui | Oui | Non | Non |
| **messaging-group** | | | Référencé par la difffusion HornetQ JMS et les groupes Discovery | Oui | Oui | Non | Non |
| **messaging-throughput** | 5455 | | Utilisé par JMS à distance. | Oui | Oui | Non | Non |
| **mod_cluster** | | 23364 | Port multicast de communication entre l'équilibreur de charge HTTP et JBoss Enterprise Application Platform. | Oui | Non | Oui | Non |
| **osgi-http** | 8090 | | Utilisé par les composants internes qui utilisent le sous-système OSGi. | Oui | Oui | Oui | Oui |
| **remoting** | 4447 | | Utilisé pour l'invocation EJB. | Oui | Oui | Oui | Oui |

| Nom | Port | Port Multidiffusion | Description | full-ha-sockets | full-sockets | ha-socket | standard-socket |
|---|---|---|---|---|---|---|---|
| `txn-recovery-environment` | 4712 | | Gestionnaire de recouvrement des transactions JTA. | Oui | Oui | Oui | Oui |
| `txn-status-manager` | 4713 | | Gestionnaire des transactions JTA / JTS. | Oui | Oui | Oui | Oui |

**Ports de gestion**

En plus des groupes de liaisons de socket, chaque contrôleur d'hôte ouvre deux ports supplémentaires pour la gestion:

- 9990 - Le port de Console de gestion
- 9999 - Le port utilisé par la Console de gestion et par le Management API

Report a bug

## 10.9. JAVA SECURITY MANAGER

### 10.9.1. Java Security Manager

**Java Security Manager**

> Le gestionnaire de sécurité Java est une classe qui gère la limite extérieure de la sandbox Java Virtual Machine (JVM), contrôlant ainsi comment le code qui est exécuté dans la JVM peut interagir avec les ressources à l'extérieur de la machine virtuelle Java. Lorsque le gestionnaire de sécurité Java est activé, l'API Java vérifie avec le gestionnaire de sécurité pour approbation avant d'exécuter une vaste gamme d'opérations potentiellement dangereuses.

Le Java Security Manager utilise une police de sécurité pour déterminer si une action est permise ou refusée.

Report a bug

### 10.9.2. Exécuter JBoss Enterprise Application Platform dans le Java Security Manager (gestionnaire de sécurité Java)

Pour spécifier une police de gestionnaire de sécurité Java, vous devez modifier les options Java transmises à l'instance de serveur ou de domaine lors du processus d'amorçage. Pour cette raison, vous ne pouvez passer les paramètres en option aux scripts **`domain.sh`** or **`standalone.sh`**. La procédure suivante va vous guider à travers les étapes de configuration de votre instance pour exécuter avec la police du gestionnaire de sécurité Java.

**Prérequis**

- Avant de suivre cette procédure, vous devrez rédiger une politique de sécurité, en utilisant la commande **`policytool`** comprise dans votre Java Development Kit (JDK). Cette procédure assume que votre police se trouve à ***`EAP_HOME`*`/bin/server.policy`**.
- Le domaine ou le serveur autonome doivent être tout à fait arrêtés avant d'éditer un fichier de configuration quelconque.

Procéder à la procédure suivante pour chaque hôte physique ou instance de votre domaine, si vous avez des membres de domaine éparpillés dans des systèmes multiples.

**Procédure 10.9. Modifier les Fichiers de configuration**

1. **Ouvrir le fichier de configuration.**
   Ouvrir le fichier de configuration pour le modifier. Ce fichier se trouve dans un de ces emplacements, suivant que vous utilisiez un domaine géré ou un serveur autonome. Il ne s'agit pas du fichier exécutable utilisé pour démarrer le serveur ou le domaine.
   - **Domaine géré**
     ***`EAP_HOME`*`/bin/domain.conf`**
   - **Serveur autonome**
     ***`EAP_HOME`*`/bin/standalone.conf`**
2. **Ajouter les options Java en fin de fichier.**
   Ajouter la ligne suivante sous forme de nouvelle ligne en fin de fichier. Vous pourrez modifier la valeur de **`-Djava.security.policy`** pour spécifier l'emplacement exact de votre police de sécurité. Doit être contenu sur une ligne seulement, sans interruption. Vous pouvez modifier **`-Djava.security.debug`** pour journaliser des information, en indiquant leur niveau de débogage. La plus verbeuse est **`failure,access,policy`**.

   ```
   JAVA_OPTS="$JAVA_OPTS -Djava.security.manager -
   Djboss.home.dir=$PWD/.. -Djava.security.policy==$PWD/server.policy -
   Djava.security.debug=failure"
   ```

3. **Démarrer le domaine ou le serveur.**
   Démarrer le domaine ou le serveur en tant que normal.

Report a bug

### 10.9.3. About Java Security Manager Policies

**Security Policy**

> Un jeu d'autorisations définies pour les différentes classes du code. Le gestionnaire de sécurité Java examine les requêtes en provenance des applications en fonction de la politique de sécurité. Si une action est autorisée par la politique, le gestionnaire de sécurité permettra que l'action ait lieu. Si l'action n'est pas autorisée par la police, le gestionnaire de sécurité refusera cette action. La stratégie de sécurité peut définir des autorisations basées sur l'emplacement du code ou sur la signature du code.

Le Java Security Manager et la police de sécurité utilisés sont configurés à l'aide des options Java Virtual Machine **`java.security.manager`** et **`java.security.policy`**.

Report a bug

### 10.9.4. Écrire une police pour le Java Security Manager

#### Introduction

Il y a une application nommée **`policytool`** dans la plupart des distributions JDK et JRE, ayant pour but la modification ou la création de polices de sécurité pour le Java Security Manager. Vous trouverez des informations sur **`policytool`** dans http://docs.oracle.com/javase/6/docs/technotes/tools/.

#### Informations de base

Une police de sécurité consiste en les éléments de configuration suivants :

**CodeBase**

L'emplacement de l'URL (à l'exclusion des informations sur l'hôte ou le domaine) d'où viennent les codes. Ce paramètre est en option.

**SignedBy**

L'alias est utilisé dans le fichier de clés pour référencer le signataire dont la clé privée a été utilisée pour signer le code. Cela peut être une valeur unique ou une liste séparée par des virgules. Ce paramètre est facultatif. Si omis, la présence ou l'absence de signature n'a aucun impact sur le gestionnaire de sécurité Java.

**Principaux**

Une liste de paires de principal_type/principal_name, qui doivent se trouver dans l'ensemble Principal du thread en cours d'exécution. L'entrée Principal est facultative. S'il est omis, il signifie « n'importe quel principal »

**Permissions**

Une permission est l'accès qui est accordé au code. De nombreuses autorisations sont fournies dans le cadre de la spécification Java Enterprise Edition 6 (Java EE 6). Ce document couvre uniquement les autorisations supplémentaires qui sont fournies par JBoss Enterprise Application Platform.

**Procédure 10.10. Définir une police pour le Java Security Manager**

1. **Démarrer `policytool`.**
   Démarrer l'outil **`policytool`** d'une des façons suivantes.

   - **Red Hat Enterprise Linux**
     À partir de votre GUI ou invite de commande, exécuter **`/usr/bin/policytool`**.

   - **Microsoft Windows Server**
     Exécuter **`policytool.exe`** à partir du menu de Démarrage (Start) ou à partir de **`bin\`** de votre installation Java. L'emplacement peut varier.

2. **Créer une nouvelle police.**
   Pour créer une nouvelle police, sélectionner **`Add Policy Entry`**. Ajouter les paramètres dont vous aurez besoin, et cliquer sur **`Done`**.

3. **Modifier une police existante**
   Sélectionner la police à partir d'une liste de polices existantes, et sélectionner le bouton **`Edit Policy Entry`**. Modifier les paramètres suivant les besoins.

4. **Supprimer une police existante.**
   Sélectionner la police à partir d'une liste de polices existantes, et sélectionner le bouton **`Delete Policy Entry`**.

**Permission spécifique à JBoss Enterprise Application Platform**

**org.jboss.security.SecurityAssociation.getPrincipalInfo**

Donne accès aux méthodes **`org.jboss.security.SecurityAssociationgetPrincipal()`** et **`getCredential()`**. Le risque encouru avec cette permission est que l'on peut voir le thread de l'appelant et ses détails d'authentification.

**org.jboss.security.SecurityAssociation.getSubject**

Donne accès à la méthode **`org.jboss.security.SecurityAssociationgetSubject()`**.

**org.jboss.security.SecurityAssociation.setPrincipalInfo**

Donne accès aux méthodes **`org.jboss.security.SecurityAssociationsetPrincipal()`**, **`setCredential()`**, **`setSubject()`**, **`pushSubjectContext()`**, et **`popSubjectContext()`**. Le risque encouru avec cette permission est que l'on peut voir le thread de l'appelant et ses détails d'authentification.

**org.jboss.security.SecurityAssociation.setServer**

Donne accès aux méthodes **`org.jboss.security.SecurityAssociationsetPrincipal()`**. Le risque encouru avec cette permission est que l'on peut activer ou désactiver le stockage multi-thread de l'appelant principal et ses détails d'authentification.

**org.jboss.security.SecurityAssociation.setRunAsRole**

Donne accès aux méthodes **`org.jboss.security.SecurityAssociationpushRunAsRole`**, **`popRunAsRole`**, **`pushRunAsIdentity`**, et **`popRunAsIdentity`**. Le risque encouru avec cette permission est que l'on peut voir le thread de l'appelant et ses détails d'authentification.

**org.jboss.security.SecurityAssociation.accessContextInfo**

Donne accès aux méthodes **`org.jboss.security.SecurityAssociationaccessContextInfo`**, et **`accessContextInfo`**. Cela vous permet les actions set et get pour définir et obtenir les informations de sécurité.

**org.jboss.naming.JndiPermission**

Fournit des permissions spéciales pour les fichiers et répertoires d'un chemin d'accès JNDI spécifié, ou bien de façon récursive pour tous les fichiers et sous-répertoires. Une JndiPermission consiste en un nom de chemin et un ensemble de permissions valides liées au fichier ou répertoire.

Les permissions disponibles sont les suivantes :

- bind
- rebind
- unbind
- lookup

- list
- listBindings
- createSubcontext
- all

Les noms de chemin d'accès se terminant par `/*` indiquent que les permissions indiquées s'appliquent à tous les fichiers et répertoires de nom du chemin. Les noms de chemin se terminant par `/-` indiquent des permissions récursives vers tous les fichiers et sous-directoires du nom du chemin. Les noms de chemins consistants avec le token <<ALL BINDINGS>> correspondent à n'importe quel fichier du répertoire.

**org.jboss.security.srp.SRPPermission**

Une classe de permissions personnalisées pour protéger l'accès à des informations sensibles SRP comme la clé de session privée et la clé privée. Cette autorisation n'a pas toutes les actions définies. La cible de **`getSessionKey`** donne accès à la clé de session privée qui résulte de la négociation SRP. L'accès à cette clé permet de chiffrer et de déchiffrer les messages qui ont été chiffrés avec la clé de session.

**org.hibernate.secure.HibernatePermission**

Cette classe de permissions fournit des permissions de base pour sécuriser les sessions Hibernate. La cible de cette propriété est le nom de l'entité. Les actions disponibles sont les suivantes :

- insérer
- supprimer
- update
- lecture
- * (all)

**org.jboss.metadata.spi.stack.MetaDataStackPermission**

Fournit une classe de permission personnalisée pour contrôler la façon dont les appelants interagissent avec la pile de métadonnées. Les permissions disponibles sont les suivantes :

- modifer
- push (vers la pile)
- pop (de la pile)
- peek (dans la pile)
- * (all)

**org.jboss.config.spi.ConfigurationPermission**

Sécurise la mise en place des propriétés de configuration. Définit uniquement les noms des cibles, mais aucune action. Les cibles pour cette propriété incluent :

- <property name> (la propriété que ce code a la permission de définir)

- * (all properties)

**org.jboss.kernel.KernelPermission**

Sécurise l'accès à la configuration du noyau. Définit uniquement les noms des cibles, mais aucune action. Les cibles pour cette propriété incluent :

- access (à la configuration du noyau)
- configure (implique accès)
- * (all)

**org.jboss.kernel.plugins.util.KernelLocatorPermission**

Sécurise l'accès au noyau. Définit uniquement les noms des cibles, mais aucune action. Les cibles pour cette propriété incluent :

- noyau
- * (all)

Report a bug

### 10.9.5. Débogage des polices du gestionnaire de sécurité

Vous pouvez activer les informations de débogage pour vous aider à résoudre les problèmes liés aux polices de sécurité. L'option **`java.security.debug`** configure le niveau des informations liées à la sécurité qui ont été reportées. La commande **`java -Djava.security.debug=help`** vous produira l'assistance avec l'ensemble complet des options de débogage. Définir le niveau de débogage à **`all`** est utile quand on résout un échec lié à la sécurité dont la cause est inconnue, mais en général, cela produit trop d'informations. Une valeur par défaut utile et raisonnable est **`access:failure`**.

**Procédure 10.11. Activer le débogage général**

- **Cette procédure permettra un bon niveau de sécurité général pour les informations de débogage liées à la sécurité.**
  Ajouter la ligne suivante au fichier de configuration du serveur.
  - Si l'instance de JBoss Enterprise Application Platform exécute sur une domaine géré, la ligne sera ajoutée au fichier **`bin/domain.conf`** de Linux ou au fichier **`bin/domain.conf.bat`** de Windows.
  - Si l'instance de JBoss Enterprise Application Platform exécute sur une domaine autonome, la ligne sera ajoutée au fichier **`bin/standalone.conf`** de Linux ou au fichier **`bin/standalone.conf.bat`** de Windows.

**`Linux`**

```
JAVA_OPTS="$JAVA_OPTS -Djava.security.debug=access:failure"
```

**`Windows`**

```
JAVA_OPTS="%JAVA_OPTS% -Djava.security.debug=access:failure"
```

**Résultat**

Un niveau général d'informations de débogage lié à la sécurité a été activé.

Report a bug

## 10.10. SÉCURITÉ DES APPLICATIONS

### 10.10.1. Activer/Désactiver un remplacement de propriété basé descripteur

**Résumé**

Le contrôle précis du remplacement de propriété de descripteur a été introduit dans **`jboss-as-ee_1_1.xsd`**. Cette tâche couvre les étapes requises pour configurer le remplacement de propriété basé descripteur.

**Prérequis**

- Section 2.1.1, « Démarrer JBoss Enterprise Application Platform 6 »
- Section 3.5.2, « Lancement du Management CLI »

Les indicateurs de remplacement de propriété basé descripteur ont les valeurs booléennes suivantes :

- Si défini à **`true`**, les remplacements de propriété sont activés.
- Si défini à **`false`**, les remplacements de propriété sont désactivés.

**Procédure 10.12. `jboss-descriptor-property-replacement`**

**`jboss-descriptor-property-replacement`** est utilisé pour activer ou désactiver le remplacement de propriété dans les descripteurs suivants :

- **`jboss-ejb3.xml`**
- **`jboss-app.xml`**
- **`jboss-web.xml`**
- **`*-jms.xml`**
- **`*-ds.xml`**

La valeur par défaut de **`jboss-descriptor-property-replacement`** est **`true`**.

1. Avec le Management CLI, exécuter la commande suivante pour déterminer la valeur de **`jboss-descriptor-property-replacement`**:

```
/subsystem=ee:read-attribute(name="jboss-descriptor-property-replacement")
```

2. Exécuter la commande suivante pour configurer le comportement :

```
/subsystem=ee:write-attribute(name="jboss-descriptor-property-replacement",value=VALUE)
```

**Procédure 10.13. `spec-descriptor-property-replacement`**

**`spec-descriptor-property-replacement`** est utilisé pour activer ou désactiver le remplacement de propriété dans les descripteurs suivants :

- **`ejb-jar.xml`**
- **`persistence.xml`**

La valeur par défaut de **`spec-descriptor-property-replacement`** est **`false`**.

1. Avec le Management CLI, exécuter la commande suivante pour confirmer la valeur de **`spec-descriptor-property-replacement`**:

```
/subsystem=ee:read-attribute(name="spec-descriptor-property-replacement")
```

2. Exécuter la commande suivante pour configurer le comportement :

```
/subsystem=ee:write-attribute(name="spec-descriptor-property-replacement",value=VALUE)
```

**Résultat**

Les indicateurs de remplacement de propriété basé descripteur ont bien été configurés.

Report a bug

## 10.11. ENCODAGE SSL

### 10.11.1. Implémentation du cryptage SSL pour le serveur de JBoss Enterprise Application Platform.

**Introduction**

De nombreuses applications web requièrent une connexion cryptée-SSL entre les clients et le serveur, connue sous le nom de connexion **HTTPS**. Vous pouvez utiliser cette procédure pour activer **HTTPS** sur votre serveur ou groupe de serveurs.

**Prérequis**

- Vous aurez besoin d'un ensemble de clés de cryptage SSL et d'un certificat de cryptage SSL. Vous pourrez vous les procurer par l'intermédiaire d'une autorité de signature de certificats. Pour générer les clés de cryptage par les utilitaires de Red Hat Enterprise Linux, voir Section 10.11.2, « Générer une clé de cryptage SSL et un certificat ».
- Vous devrez connaitre les informations suivantes sur votre environnement et sur votre installation :
  - Le nom complet du répertoire et le chemin d'accès à vos fichiers de certificats
  - Le mot de passe de cryptage pour vos clés de cryptage.

- Vous devrez exécuter le Management CLI et le connecter à votre contrôleur de domaine ou à votre serveur autonome.

**NOTE**

Cette procédure utilise des commandes appropriées à la configuration de JBoss Enterprise Application Platform, qui utilise un domaine géré. Si vous utilisez un domaine autonome, modifier les commandes de Management CLI en supprimant **`/profile=default`** du début d'une commande de Management CLI.

**Procédure 10.14. Configurer le JBoss Web Server pour qu'il puisse utiliser HTTPS**

1. **Ajouter un nouveau connecteur HTTPS.**
   Exécuter la commande de Management CLI suivante, en changeant le profil comme il se doit. Cela va créer un nouveau connecteur sécurisé, nommé **`HTTPS`**, qui utilise le protocole **`https`**, la liaison de socket **`https`** (ayant comme valeur par défaut **`8443`**), et qui est définie pour être sécurisée.

   **Exemple 10.25. Commande de Management CLI**

   ```
   /profile=default/subsystem=web/connector=HTTPS/:add(socket-
   binding=https,scheme=https,protocol=HTTP/1.1,secure=true)
   ```

2. **Configurer le certificat de cryptage SSL et les clés.**
   Exécutez les commandes CLI suivantes pour configurer votre certificat SSL, en remplaçant vos propres valeurs par celles de l'exemple. Cet exemple suppose que le keystore est copié dans le répertoire de configuration du serveur, qui est ***`EAP_HOME`*`/domain/configuration/`** pour un domaine géré.

   **Exemple 10.26. Commande de Management CLI**

   ```
   /profile=default/subsystem=web/connector=HTTPS/ssl=configuration:a
   dd(name=https,certificate-key-
   file=${jboss.server.config.dir}/keystore.jks,password=SECRET, key-
   alias=KEY_ALIAS)
   ```

   Pour obtenir une liste complète des paramètres que vous pouvez définir pour les propriétés SSL du connecteur, voir Section 10.11.3, « Référence de connecteur SSL ».

3. **Déployer une application.**
   Déployer une application dans un groupe de serveurs qui utilise le profil que vous avez configuré. Si vous utilisez un serveur autonome, déployer une application sur votre serveur. Les demandes HTTP en sa direction utilisent la nouvelle connexion cryptée-SSL.

Report a bug

### 10.11.2. Générer une clé de cryptage SSL et un certificat

Pour utiliser une connexion chiffrée SSL HTTP (HTTPS), ainsi que d'autres types de communication cryptée-SSL, vous avez besoin d'un certificat de chiffrement signé. Vous pouvez acheter un certificat d'une autorité de certification (AC), ou vous pouvez utiliser un certificat auto-signé. Les certificats auto-

signés ne sont pas considérés dignes de confiance par de nombreux tiers, mais conviennent à des fins de test internes.

Cette procédure vous permet de créer un certificat auto-signé lié à des utilitaires disponibles dans Red Hat Enterprise Linux.

**Prérequis**

- La commande **keytool** doit être disponible. Elle est fournie par le Java Development Kit. Chercher le chemin du fichier. Dans Red Hat Enterprise Linux, OpenJDK installe cette commande à l'emplacement suivant **/usr/bin/keytool**.
- Comprendre la syntaxe et les paramètres de la commande **keytool**. Cette procédure utilise des instructions extrêmement génériques, car des discussions plus sophistiquées sur les spécificités des certificats SSL ou sur la commande **keytool** sont hors de portée de cette documentation.

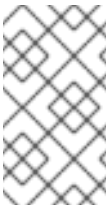
**Procédure 10.15. Générer une clé de cryptage SSL et un certificat**

1. **Générer un keystore avec des clés privées et des clés publiques.**
   Exécuter la commande suivante pour générer un keystore nommé **server.keystore** ayant comme alias **jboss** dans votre répertoire actuel.

   ```
   keytool -genkey -alias jboss -keyalg RSA -keystore server.keystore -
   storepass mykeystorepass --dname
   "CN=jsmith,OU=Engineering,O=mycompany.com,L=Raleigh,S=NC,C=US"
   ```

   Le tableau suivant décrit les paramètres utilisés avec la commande «keytool».

| Paramètre | Description |
|---|---|
| ***-genkey*** | La commande **keytool** qui génére une paire de clés contenant une clé publique et une clé privée. |
| ***-alias*** | L'alias est pour le keystore. Cette valeur est arbitraire, mais l'alias **jboss** est la valeur par défaut utilisée par le serveur JBoss Web. |
| ***-keyalg*** | L'algorithme de création de paires de clés. Dans ce cas, c'est **RSA**. |
| ***-keystore*** | Le nom et l'emplacement du fichier keystore. L'emplacement par défaut est le répertoire en cours. Le nom que vous choisissez est arbitraire. Dans ce cas, il s'agit du fichier nommé **server.keystore**. |

| Paramètre | Description |
| --- | --- |
| ***-storepass*** | Ce mot de passe est utilisé pour authentifier le keystore, et pour que la clé puisse être lue. Le mot de passe doit contenir au moins 6 caractères de long et doit être fourni quand on accède au keystore. Dans un tel cas, on utilise **`mykeystorepass`**. Si vous omettez ce paramètre, on vous demandera de le saisir quand vous exécuterez la commande. |
| ***-keypass*** | Il s'agit du mot de passe pour la clé.<br>  <br>**NOTE**<br>À cause d'une limitation d'implémentation, il doit correspondre à celui du mot de passe du store. |
| ***--dname*** | Une chaîne avec des guillemets qui décrit le nom distinct de la clé, comme par exemple : "CN=jsmith,OU=Engineering,O=mycompany.com,L=Raleigh,C=US". Cette chaîne est une compilation des composants suivants :<br>◦ **CN** - Le nom commun ou le nom d'hôte. Si le nom d'hôte est "jsmith.mycompany.com", le **CN** sera "jsmith".<br>◦ **OU** - L'unité organisationnelle, par exemple "Engineering"<br>◦ **O** - Le nom de l'organisation, par exemple "mycompany.com".<br>◦ **L** - La localité, par exemple "Raleigh" ou "London"<br>◦ **S** - L'état ou la province, par exemple "NC". Ce paramètres est optionnel.<br>◦ **C** - Les 2 lettres d'un code pays, par exemple "US" ou "UK", |

Quand vous exécuterez la commande ci-dessus, on vous demandera les informations suivantes :

- Si vous n'utilisiez pas le paramètre ***-storepass*** sur la ligne de commande, on vous demandera de saisir le mot de passe du keystore. Saisir le nouveau mot de passe à la seconde invite.

- Si vous n'utilisiez pas le paramètre **-keypass** sur la ligne de commande, on vous demandera de saisir le mot de passe de la clé. Appuyez sur **Enter** pour le définir à la même valeur que celle du mot de passe du keystore.

Quand la commande s'achèvera, le fichier **server.keystore** contiendra la clé unique avec l'alias **jboss**.

2. **Vérifier la clé.**
Vérifier que la clé fonctionne correctement en utilisant la commande suivante.

```
keytool -list -keystore server.keystore
```

On vous demande le mot de passe du keystore. Les contenus du keystore sont affichés (dans ce cas, il s'agit d'une simple clé nommée **jboss**). Notez le type de la clé **jboss**, qui est **keyEntry**. Cela indique que le keystore contient à la fois une entrée publique et une entrée privée pour cette clé.

3. **Créer une demande de signature de certificat.**
Exécutez la commande suivante pour générer une demande de signature de certificat en utilisant la clé publique du keystore que vous avez créée dans la 1ère étape.

```
keytool -certreq -keyalg RSA -alias jboss -keystore server.keystore
-file certreq.csr
```

On vous demandera le mot de passe pour pouvoir authentifier le keystore. La commande **keytool** crée alors une nouvelle demande de signature de certificat nommée **certreq.csr** dans le répertoire en cours d'utilisation.

4. **Tester le certificat nouvellement généré.**
Tester les contenus du certificat avec la commande suivante :

```
openssl req -in certreq.csr -noout -text
```

Les détails du certificat apparaissent.

5. **En option: soumettre votre certificat à une autorité de certification (AC).**
Une Autorité de Certification (AC) athentifie votre certificat pour qu'il soit considéré de confiance par des clients de tierce partie. L'AC vous a produit un certificat signé, et en option, vous a peut être fourni un ou plusieurs certificats intermédiaires.

6. **Option: exporter un certificat auto-signé du keystore.**
Si vous n'en avez besoin que dans un but de test ou en interne, vous pourrez utiliser un certificat auto-signé. Vous pourrez en exporter un, créé dans la première étape, en provenance du keystore, comme suit :

```
keytool -export -alias jboss -keystore server.keystore -file
server.crt
```

On vous demande un mot de passe pour pouvoir s'authentifier au keystore. Un certificat auto-signé, intitulé **server.crt**, a été créé dans le répertoire en cours d'utilisation.

7. **Importer le certificat signé avec tout certificat intermédiaire.**
Importer chaque certificat, dans l'ordre dans lequel l'AC vous le demande. Pour chaque AC que vous importez, remplacer **intermediate.ca** ou **server.crt** par le nom du fichier. Si vos

certificats ne sont pas fournis dans des fichiers séparés, créer un fichier séparé pour chaque certificat, et coller leur contenu dans le fichier.

**NOTE**

Votre certificat signé et les clés de ce certificat sont des ressources de valeur. Soyez vigilant sur la façon donc vous les transportez entre les serveurs.

```
keytool -import -keystore server.keystore -alias intermediateCA -
file intermediate.ca
```

```
keytool -import -alias jboss -keystore server.keystore -file
server.crt
```

8. **Testez que vos certificats soient bien importés avec succès.**
Exécuter la commande suivante, et saisir le mot de passe de keystore quand on vous le demandera. Les contenus de votre keystore sont affichés, et les certificats font partie de la liste.

```
keytool -list -keystore server.keystore
```

**Résultat**

Votre certificat signé est maintenant inclus dans votre keystore et est prêt à l'utilisation pour crypter les connexions SSL, y compris les communications au serveur web HTTPS.

Report a bug

### 10.11.3. Référence de connecteur SSL

Les connecteurs JBoss Web peuvent inclure les attributs de configuration SSL suivants. Les commandes CLI fournies sont conçues pour un domaine géré à l'aide du profil **`par défaut`**. Changer le nom du profil à celui que vous souhaitez configurer, pour un domaine géré, ou omettre la portion **`/profile=default`** de la commande, pour un serveur autonome.

**Tableau 10.3. Attributs de connecteurs SSL**

| Attribut | Description | Commande CLI |
|---|---|---|
| Nom | Le nom d'affichage du connecteur SSL | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=name,value=https)` |

| Attribut | Description | Commande CLI |
| --- | --- | --- |
| verify-client | Définir à **`true`** pour obtenir une chaîne de certificat valide de la part d'un client avant d'accepter une connexion. Définir à **`want`** si vous voulez que la pile SSL demande un certificat de client, mais ne pas l'échouer si celui-ci n'est pas présenté. Définir à **`false`** (la valeur par défaut) si vous ne souhaitez pas demander de chaîne de certificat, à moins que le client ne demande une ressource protégée par une contrainte de sécurité qui utilise l'authentification de **`CLIENT-CERT`**. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=verify-client,value=want)` |
| verify-depth | Le nombre maximal d'émetteurs de certificats intermédiaires vérifiés avant de décider que les clients n'ont pas de certificat valide. La valeur par défaut est **`10`**. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=verify-depth,value=10)` |
| certificate-key-file | Le chemin d'accès complet et nom de fichier du keystore où le certificat de serveur signé est stocké. Avec le cryptage JSSE, ce fichier de certificat sera le seul, tandis que OpenSSL utilise plusieurs fichiers. La valeur par défaut est le fichier **`.keystore`** dans le répertoire home de l'utilisateur exécutant JBoss Enterprise Application Platform. Si votre **`keystoreType`** n'utilise pas de fichier, définissez le paramètre en chaîne vide | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=certificate-key-file,value=../domain/configuration/server.keystore)` |
| certificate-file | Si vous utilisez un cryptage OpenSSL, définir la valeur de ce paramètre au chemin d'accès du fichier qui contient le certificat de serveur. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=certificate-file,value=server.crt)` |

| Attribut | Description | Commande CLI |
| --- | --- | --- |
| mot de passe | Le mot de passe pour le trustore et le keystore à la fois. La valeur par défaut est **changeit**, donc vous devrez le changer pour qu'il corresponde au mot de passe de votre keystore si vous souhaitez que votre configuration fonctionne. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=password,value=changeit)` |
| protocol | La version de protocole SSL à utiliser. Les valeurs prises en charge comprennent **SLv2**, **SSLv3**, **TLSv1**, **SSLv2+SSLv3**, et **ALL**. La valeur par défaut est **ALL**. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=protocol,value=ALL)` |
| cipher-suite | Une liste séparée par des virgules des algorithmes de cryptage autorisés. La valeur par défaut JVM pour JSSE contient des algorithmes de chiffrement faibles, qui ne doivent pas être utilisés. L'exemple répertorie uniquement les deux algorithmes de chiffrement possibles, mais des exemples concrets en utiliseront probablement plus. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=cipher-suite, value="TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA")` |
| key-alias | L'alias utilisé pour le certificat de serveur dans le keystore. La valeur par défaut est **jboss**. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=key-alias,value=jboss)` |
| truststore-type | Le type de truststore. Il existe différents types de keystores, dont **PKCS12** et le **JKS** standard de Java. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=truststore-type,value=jks)` |

| Attribut | Description | Commande CLI |
| --- | --- | --- |
| keystore-type | Le type de keystore. Il existe différents types de keystores, dont **PKCS12** et le **JKS** standard de Java. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=keystore-type,value=jks)` |
| ca-certificate-file | Le fichier contenant les certificats CA. Il s'agit du **trustStoreFile**, dans le cas de JSSE, et il utilise le même mot de passe que le keystore. Le fichier **ca-certificate-file** est utilisé pour valider les certificats de clients. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=certificate-file,value=ca.crt)` |
| ca-certificate-password | Le mot de passe de certificat est **ca-certificate-file**. Dans l'exemple ci-dessous, remplacer le mot de passe par votre propre mot de passe masqué. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=ca-certificate-password,value=`*`MASKED_PASSWORD`*`)` |
| ca-revocation-url | Un fichier ou URL qui contient la liste de révocations. Se réfère au **crlFile** pour JSSE ou au **SSLCARevocationFile** pour SSL. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=ca-revocation-url,value=ca.crl)` |
| session-cache-size | La taille du cache SSLSession. La valeur par défaut est **0**, qui désactive la session cache. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=session-cache-size,value=100)` |

| Attribut | Description | Commande CLI |
|---|---|---|
| session-timeout | Le nombre de secondes avant qu'une SSLSession n'expire. La valeur par défaut est **86400** secondes, ce qui correspond à 24 heures. | `/profile=default/subsystem=web/connector=HTTPS/ssl=configuration/:write-attribute(name=session-timeout,value=43200)` |

Report a bug

## 10.12. L'ARCHIVAGE SÉCURITÉS DES MOTS DE PASSE POUR LES STRINGS DÉLICATS

### 10.12.1. Sécurisation des chaînes sensibles des fichiers en texte clair

Les applications web et autres déploiements incluent souvent des fichiers en texte clair, comme des descripteurs de déploiement XML, qui comprennent des informations sensibles telles que les mots de passe et autres chaînes sensibles. JBoss Enterprise Application Platform inclut un mécanisme d'archivage sécurisé de mots de passe qui vous permet de crypter les chaînes sensibles et de les stocker dans un keystore chiffré. Le mécanisme d'archivage sécurisé parvient à décrypter les chaînes à utiliser dans les domaines de sécurité, ou autres systèmes de vérification. Ceci fournit une couche supplémentaire de sécurité. Le mécanisme s'appuie sur les outils qui sont inclus dans toutes les implémentations de Java Development Kit (JDK) prises en charge.

Report a bug

### 10.12.2. Créer un Keystore Java pour stocker des strings sensibles

**Prérequis**

- La commande **keytool** doit être disponible. Elle est fournie par le Java Runtime Environment (JRE). Chercher le chemin du fichier. Se trouve à l'emplacement suivant **/usr/bin/keytool** dans Red Hat Enterprise Linux.

**Procédure 10.16. Installation du Java Keystore**

1. **Créer un répertoire pour stocker votre keystore et autres informations cryptées.**
   Créer un répertoire qui contiendra votre keystore et autres informations pertinentes. Le reste de cette procédure assume que le répertoire est **/home/*USER*/vault/**.

2. **Déterminer les paramètres à utiliser avec keytool.**
   Déterminer les paramètres suivants :

   **alias**

   L'alias est un identificateur unique pour l'archivage sécurisé ou autres données stockées dans le keystore. L'alias dans l'exemple de commande à la fin de cette procédure est **vault** (archivage sécurisé). Les alias sont insensibles à la casse.

**keyalg**

L'algorithme à utiliser pour le cryptage. La valeur par défaut est **DSA**. Dans cette procédure, l'exemple utilise **RSA**. Consultez la documentation de votre JRE et de votre système d'exploitation pour étudier vos possibilités.

**keysize**

La taille d'une clé de cryptage impacte sur la difficulté de décrypter au seul moyen de la force brutale. La taille par défaut des clés est de 1024. Doit être entre 512 et 1024 et un multiple de 64. Dans cette procédure, l'exemple utilise **1024**.

**keystore**

Le keystore est une base de données qui contient des informations chiffrées et des informations sur la façon de déchiffrer. Si vous ne spécifiez pas de keystore, le keystore par défaut à utiliser est un fichier appelé **.keystore** dans votre répertoire personnel. La première fois que vous ajoutez des données dans un keystore, il sera créé. L'exemple de cette procédure utilise le keystore **vault.keystore**.

La commande du **keystore** a plusieurs options. Consulter la documentation de votre JRE ou de votre système d'exploitation pour obtenir plus d'informations.

3. **Détermine les réponses aux questions que la commande keystore vous demandera.**
Le **keystore** a besoin des informations suivantes pour remplir le keytore :

**Mot de passe du keystore**

Lorsque vous créez un keystore, vous devez définir un mot de passe. Pour pouvoir travailler dans keystore dans le futur, vous devez fournir le mot de passe. Créer un mot de passe dont vous vous souviendrez. Le keystore est sécurisé par son mot de passe et par la sécurité du système d'exploitation et du système de fichiers où il se trouve.

**Mot de passe Clé (en option)**

En plus du mot de passe du keystore, vous pouvez indiquer un mot de passe pour chaque clé contenue. Pour utiliser une clé, le mot de passe doit être donné à chaque utilisation. Normalement, cette fonction n'est pas utilisée.

**Prénom et nom de famille**

Cela et le reste de l'information dans la liste aide à identifier la clé de façon unique et à la placer dans une hiérarchie par rapport aux autres clés. N'a pas nécessairement besoin d'être un nom dutout, mais doit être composé de deux mots et doit être unique à une clé. L'exemple dans cette procédure utilise **Accounting Administrator**. En termes de répertoires, cela devient le *common name* (nom commmun) du certificat.

**Unité organisationnelle**

Il s'agit d'un mot unique d'identification qui utilise le certificat. Il se peut que ce soit l'application ou l'unité commerciale. L'exemple de cette procédure utilise **enterprise_application_platform**. Normalement, tous les keystores utilisés par un groupe ou une application utilisent la même unité organisationnelle.

**Organisation**

Il s'agit normalement d'une représentation de votre nom d'organisation en un seul mot. Demeure constant à travers tous les certificats qui sont utilisés par une organisation. Cet exemple utilise **`MyOrganization`**.

**Ville ou municipalité**

Votre ville.

**État ou province**

Votre état ou province, ou l'équivalent pour votre localité.

**Pays**

Le code pays en deux lettres.

Ces informations vont créer ensemble une hiérarchie de vos keystores et certificats, qui garantira qu'ils utilisent une structure de nommage consistante, et unique.

4. **Exécuter la commande `keytool`, fournissant les informations que vous avez collectées.**

**Exemple 10.27. Exemple d'entrée et de sortie de la commande `keystore`**

```
$ keytool -genkey -alias vault -keyalg RSA -keysize 1024 -keystore
/home/USER/vault/vault.keystore
Enter keystore password: vault22
Re-enter new password:vault22
What is your first and last name?
  [Unknown]:  Accounting Administrator
What is the name of your organizational unit?
  [Unknown]:  AccountingServices
What is the name of your organization?
  [Unknown]:  MyOrganization
What is the name of your City or Locality?
  [Unknown]:  Raleigh
What is the name of your State or Province?
  [Unknown]:  NC
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=Accounting Administrator, OU=AccountingServices,
O=MyOrganization, L=Raleigh, ST=NC, C=US correct?
  [no]:  yes

Enter key password for <vault>
        (RETURN if same as keystore password):
```

**Résultat**

Un fichier nommé **`vault.keystore`** est créé dans le répertoire **`/home/USER/vault/`**. Il stocke une clé simple, nommée **`vault`**, qui sera utilisée pour stocker des strings cryptés, comme des mots de passe, pour la plateforme JBoss Enterprise Application Platform.

Report a bug

### 10.12.3. Masquer le mot de passe du keystore et Initialiser le mot de passe de l'archivage de sécurité

**Prérequis**

- Section 10.12.2, « Créer un Keystore Java pour stocker des strings sensibles »
- L'application ***EAP_HOME*/bin/vault.sh** doit pouvoir être accessible via l'interface de ligne de commande.

1. **Exécuter la commande `vault.sh`.**
   Exécuter ***EAP_HOME*/bin/vault.sh**. Démarrer une nouvelle session interactive en tapant **`0`**.

2. **Saisir le nom du répertoire où les fichiers cryptés seront stockés.**
   Ce répertoire doit être raisonnablement sécurisé, mais JBoss Enterprise Application Platform doit pouvoir y accéder. Si vous suivez Section 10.12.2, « Créer un Keystore Java pour stocker des strings sensibles », votre keystore sera dans un répertoire nommé **`vault/`** dans votre répertoire de base (home). Cet exemple utilise le répertoire **`/home/`*`USER`*`/vault/`**.

   **NOTE**

   N'oubliez pas d'inclure la barre oblique finale dans le nom du répertoire. Soit `/` ou `\`, selon votre système d'exploitation.

3. **Saisir le nom de votre keystore.**
   Saisir le nom complet vers le fichier de keystore. Cet exemple utilise **`/home/`*`USER`*`/vault/vault.keystore`**.

4. **Crypter le mot de passe du keystore.**
   Les étapes suivantes vous servent à crypter le mot de passe du keystore, afin que vous puissiez l'utiliser dans les applications et les fichiers de configuration en toute sécurité

   a. **Saisir le mot de passe du keystore.**
      Quand vous y serez invité, saisir le mot de passe du keystore.

   b. **Saisir une valeur salt.**
      Entrez une valeur salt de 8 caractères. La valeur salt, ainsi que le nombre d'itérations (ci-dessous), sont utilisés pour créer la valeur de hachage

   c. **Saisir le nombre d'itérations.**
      Saisir un nombre pour le nombre d'itérations.

   d. **Notez les informations de mot de passe masqué.**
      Le mot de passe masqué, salt et le nombre d'itérations sont imprimés en sortie standard. Prenez en note dans un endroit sûr. Un attaquant pourrait les utiliser pour déchiffrer le mot de passe.

   e. **Saisir un alias pour l'archivage de sécurité.**
      Quand on vous y invite, saisir un alias pour l'archivage de sécurité. Si vous suivez Section 10.12.2, « Créer un Keystore Java pour stocker des strings sensibles » pour créer votre archivage de sécurité, l'alias sera **`vault`**.

5. **Sortir de la console interactive.**
   Saisir le mot **`exit`** pour sortir de la console interactive.

**Résultat**

Votre mot de passe de keystore est masqué afin de pouvoir être utilisé dans les fichiers de configuration et déploiement. De plus, votre archivage de sécurité est complètement configuré et prêt à l'utilisation.

Report a bug

### 10.12.4. Configurer JBoss Enterprise Application Platform pour qu'il utilise l'archivage sécurisé des mots de passe

**Aperçu**

Avant de masquer les mots de passe et d'autres attributs sensibles dans les fichiers de configuration, vous devez sensibiliser JBoss Enterprise Application Platform à l'archivage sécurisé des mots de passe qui les stocke et les déchiffre. Actuellement, cela vous oblige à arrêter Enterprise Application Platform et à modifier la configuration directement.

**Prérequis**

- Section 10.12.2, « Créer un Keystore Java pour stocker des strings sensibles »
- Section 10.12.3, « Masquer le mot de passe du keystore et Initialiser le mot de passe de l'archivage de sécurité »

**Procédure 10.17. Assigner un mot de passe d'archivage sécurisé.**

1. **Déterminer les valeurs qui conviennent pour la commande.**
   Déterminer les valeurs pour les paramètres suivants, qui sont déterminés par les commandes utilisées pour créer le keystore lui-même. Pour obtenir des informations sur la façon de créer un keystore, voir les sujets suivants : Section 10.12.2, « Créer un Keystore Java pour stocker des strings sensibles » et Section 10.12.3, « Masquer le mot de passe du keystore et Initialiser le mot de passe de l'archivage de sécurité ».

| Paramètre | Description |
|---|---|
| KEYSTORE_URL | Le chemin d'accès ou URI du fichier keystore, qui s'appelle normalement `vault.keystore` |
| KEYSTORE_PASSWORD | Le mot de passe utilisé pour accéder au keystore. Cette valeur devrait être masquée. |
| KEYSTORE_ALIAS | Le nom du keystore. |
| SALT | Le salt utilisé pour crypter et décrypter les valeurs de keystore. |
| ITERATION_COUNT | Le nombre de fois que l'algorithme de chiffrement est exécuté. |
| ENC_FILE_DIR | Le chemin d'accès au répertoire à partir duquel les commandes de keystore sont exécutées. Normalement, le répertoire contient les mots de passe sécurisés. |

| Paramètre | Description |
| --- | --- |
| hôte (domaine géré uniquement) | Le nom de l'hôte que vous configurez |

2. **Utiliser le Management CLI pour activer les mots de passe sécurisés.**
   Exécutez une des commandes suivantes, selon que vous utilisiez un domaine géré ou une configuration de serveur autonome. Substituez les valeurs de la commande par celles de la première étape de cette procédure.

   - **Domaine géré**

     ```
     /host=YOUR_HOST/core-service=vault:add(vault-options=
     [("KEYSTORE_URL" => "PATH_TO_KEYSTORE"), ("KEYSTORE_PASSWORD" =>
     "MASKED_PASSWORD"), ("KEYSTORE_ALIAS" => "ALIAS"), ("SALT" =>
     "SALT"),("ITERATION_COUNT" => "ITERATION_COUNT"), ("ENC_FILE_DIR"
     => "ENC_FILE_DIR")])
     ```

   - **Serveur autonome**

     ```
     /core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
     "PATH_TO_KEYSTORE"), ("KEYSTORE_PASSWORD" => "MASKED_PASSWORD"),
     ("KEYSTORE_ALIAS" => "ALIAS"), ("SALT" => "SALT"),
     ("ITERATION_COUNT" => "ITERATION_COUNT"), ("ENC_FILE_DIR" =>
     "ENC_FILE_DIR")])
     ```

   Ce qui suit est un exemple de la commande avec des valeurs hypothétiques :

   ```
   /core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
   "/home/user/vault/vault.keystore"), ("KEYSTORE_PASSWORD" => "MASK-
   3y28rCZlcKR"), ("KEYSTORE_ALIAS" => "vault"), ("SALT" =>
   "12438567"),("ITERATION_COUNT" => "50"), ("ENC_FILE_DIR" =>
   "/home/user/vault/")])
   ```

**Résultat**

JBoss Enterprise Application Platform est configurée pour décrypter les strings masqués par l'intermédiaire de l'archivage sécurisé de mots de passe. Pour ajouter des strings à l'archivage sécurisé, et les utiliser dans votre configuration, voir la section suivante : Section 10.12.5, « Stocker et Résoudre des strings sensibles cryptés du Keystore Java. ».

Report a bug

## 10.12.5. Stocker et Résoudre des strings sensibles cryptés du Keystore Java.

**Résumé**

En comptant les mots de passe et les autres strings sensibles, les fichiers de configuration en texte brut ne sont pas sécurisés. JBoss Enterprise Application Platform inclut la capacité à stocker et à utiliser les valeurs masquées dans les fichiers de configuration, et d'utiliser ces valeurs masquées dans les fichiers de configuration.

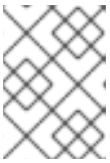
**Prérequis**

- Section 10.12.2, « Créer un Keystore Java pour stocker des strings sensibles »
- Section 10.12.3, « Masquer le mot de passe du keystore et Initialiser le mot de passe de l'archivage de sécurité »
- Section 10.12.4, « Configurer JBoss Enterprise Application Platform pour qu'il utilise l'archivage sécurisé des mots de passe »
- L'application ***EAP_HOME*/bin/vault.sh** doit pouvoir être accessible via l'interface de ligne de commande.

**Procédure 10.18. Installation du Java Keystore**

1. **Exécuter la commande `vault.sh`.**
   Exécuter ***EAP_HOME*/bin/vault.sh**. Démarrer une nouvelle session interactive en tapant **`0`**.

2. **Saisir le nom du répertoire où les fichiers cryptés seront stockés.**
   Si vous suivez Section 10.12.2, « Créer un Keystore Java pour stocker des strings sensibles », votre keystore sera dans un répertoire nommé **`vault/`** de votre répertoire de base. Dans la plupart des cas, il est logique de stocker toutes vos informations cryptées au même endroit dans le keystore. Cet exemple utilise le répertoire **`/home/`*USER*`/vault/`**.

   **NOTE**

   N'oubliez pas d'inclure la barre oblique finale dans le nom du répertoire. Soit `/` ou `\`, selon votre système d'exploitation.

3. **Saisir le nom de votre keystore.**
   Saisir le nom complet vers le fichier de keystore. Cet exemple utilise **`/home/`*USER*`/vault/vault.keystore`**.

4. **Saisir le mot de passe du keystore, le nom de l'archivage sécurisé, salt et le nombre d'itérations.**
   Quand vous y êtes invité, saisir le mot de passe du keystore, le nom de l'archivage sécurisé, salt et le nombre d'itérations.

5. **Sélectionner l'option de stockage d'un mot de passe.**
   Sélectionner l'option **`0`** de stockage d'un mot de passe ou autre string sensible.

6. **Saisir la valeur.**
   Une fois que vous y êtes invité, saisir la valeur deux fois. Si les valeurs ne correspondent pas, vous serez invité à essayer à nouveau.

7. **Saisir le bloc d'archivage sécurisé.**
   Saisir le bloc d'archivage sécurisé, qui est un conteneur pour les attributs qui ont trait à la même ressource. Un exemple de nom d'attribut serait **`ds_ExampleDS`**. Cela fera partie de la référence à la chaîne cryptée, dans votre source de données ou autre définition de service.

8. **Saisir le nom de l'attribut.**
   Saisir le nom de l'attribut que vous stockez. Exemple de nom d'attribut **`password`**.

   **Résultat**

   Un message comme celui qui suit montre que l'attribut a été sauvegardé.

   ■

```
Valeur de l'attribut pour (ds_ExampleDS, password) sauvegardé
```

9. **Notez les informations pour ce string crypté.**
   Un message s'affiche sur la sortie standard, montrant le bloc d'archivage sécurisé, le nom de l'attribut, la clé partagée et des conseils sur l'utilisation du string dans votre configuration. Prendre note de ces informations dans un emplacement sécurisé. Voici un exemple de sortie.

```
*****************************************
Vault Block:ds_ExampleDS
Attribute Name:password
Shared
Key:N2NhZDYzOTMtNWE0OS00ZGQ0LWE4MmEtMWNlMDMyNDdmNmI2TElORV9CUkVBS3Zh
dWx0
Configuration should be done as follows:
VAULT::ds_ExampleDS::password::N2NhZDYzOTMtNWE0OS00ZGQ0LWE4MmEtMWNlM
DMyNDdmNmI2TElORV9CUkVBS3ZhdWx0
*****************************************
```

10. **Utiliser le string crypté dans votre configuration.**
    Utiliser le string de l'étape de configuration précédente, à la place du string en texte brut. Une source de données utilisant le mot de passe crypté ci-dessus, est montrée ci-dessous.

```
...
  <subsystem xmlns="urn:jboss:domain:datasources:1.0">
    <datasources>
      <datasource jndi-name="java:jboss/datasources/ExampleDS"
enabled="true" use-java-context="true" pool-name="H2DS">
        <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-
1</connection-url>
        <driver>h2</driver>
        <pool></pool>
        <security>
          <user-name>sa</user-name>

<password>${VAULT::ds_ExampleDS::password::N2NhZDYzOTMtNWE0OS00ZGQ0L
WE4MmEtMWNlMDMyNDdmNmI2TElORV9CUkVBS3ZhdWx0}</password>
        </security>
      </datasource>
      <drivers>
         <driver name="h2" module="com.h2database.h2">
            <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-
datasource-class>
         </driver>
      </drivers>
    </datasources>
  </subsystem>
...
```

Vous pouvez utiliser un string crypté n'importe où dans votre fichier de configuration autonome ou de domaine pour lequel les expressions sont autorisées.

**NOTE**

Pour vérifier si les expressions sont utilisées dans un sous-système particulier, exécuter la commande CLI suivante sur ce sous-système :

```
/host=master/core-service=management/security-
realm=TestRealm:read-resource-description(recursive=true)
```

À partir du résultat de cette commande, chercher la valeur du paramètre **`expressions-allowed`**. Si 'true', vous pourrez utiliser des expressions dans la configuration de ce sous-système particulier.

Une fois que vous aurez mis votre string dans le keystore, utiliser la syntaxe suivante pour remplacer tout string en texte claire par un texte crypté.

```
${VAULT::<replaceable>VAULT_BLOCK</replaceable>::
<replaceable>ATTRIBUTE_NAME</replaceable>::
<replaceable>ENCRYPTED_VALUE</replaceable>}
```

Voici un exemple de valeur réelle, où le bloc d'archivage sécurisé est **`ds_ExampleDS`** et l'attribut est **`password`**.

```
<password>${VAULT::ds_ExampleDS::password::N2NhZDYzOTMtNWE0OS00ZGQ0L
WE4MmEtMWNlMDMyNDdmNmI2TElORV9CUkVBS3ZhdWx0}</password>
```

Report a bug

### 10.12.6. Stocker et Résoudre des strings sensibles de vos Applications

#### Aperçu

Les éléments de configuration de la plate-forme JBoss Application Enterprise prennent en charge la capacité de régler les chaînes cryptés par rapport aux valeurs stockées dans Java Keystore, via le mécanisme Security Vault. Vous pouvez ajouter le support pour cette fonctionnalité à vos propres applications.

Tout d'abord, ajoutez le mot de passe dans votre Security Vault. En second lieu, remplacer le mot de passe de texte clair par celui qui est stocké dans le Security Vault. Vous pouvez utiliser cette méthode pour obscurcir les strings sensibles de votre application.

#### Prérequis

Avant d'effectuer cette procédure, assurez-vous que le répertoire pour stocker vos fichiers dans le Security Vault existe bien. Qu'importe où vous les placez, tant que l'utilisateur qui exécute JBoss Enterprise Application Platform dispose de l'autorisation de lire et écrire des fichiers. Cet exemple situe le répertoire **`vault/`** dans le répertoire **`/home/`*`USER`*`/vault/`**. Le Security Vault lui-même correspond à un fichier nommé **`vault.keystore`** qui se trouve dans le répertoire **`vault/`**.

**Exemple 10.28. Ajout du String Mot de passe au Security Vault**

Ajouter le string au Security Vault par la commande ***`EAP_HOME`*`/bin/vault.sh`**. La série de commandes et réponses est incluse dans la session suivante. Les valeurs saisies par l'utilisateur apparaîtront clairement. Certaines sorties seront supprimées pour le formatage. Dans Microsoft

Windows, le nom de la commande est **vault.bat**. Notez que dans Microsoft Windows, le chemins d'acccès au fichier utilisent le caractère `\` comme séparateur de répertoire, et non pas le caractère `/`.

```
[user@host bin]$ ./vault.sh
*********************************
****  JBoss Vault ********
*********************************
Please enter a Digit::   0: Start Interactive Session  1: Remove
Interactive Session  2: Exit
0
Starting an interactive session
Enter directory to store encrypted files:/home/user/vault/
Enter Keystore URL:/home/user/vault/vault.keystore
Enter Keystore password: ...
Enter Keystore password again: ...
Values match
Enter 8 character salt:12345678
Enter iteration count as a number (Eg: 44):25

Enter Keystore Alias:vault
Vault is initialized and ready for use
Handshake with Vault complete
Please enter a Digit::   0: Store a password  1: Check whether password
exists  2: Exit
0
Task:  Store a password
Please enter attribute value: sa
Please enter attribute value again: sa
Values match
Enter Vault Block:DS
Enter Attribute Name:thePass
Attribute Value for (DS, thePass) saved

Please make note of the following:
********************************************
Vault Block:DS
Attribute Name:thePass
Shared
Key:OWY5M2I5NzctYzdkOS00MmZhLWExZGYtNjczM2U5ZGUyOWIxTElORV9CUkVBS3ZhdWx0
Configuration should be done as follows:
VAULT::DS::thePass::OWY5M2I5NzctYzdkOS00MmZhLWExZGYtNjczM2U5ZGUyOWIxTElO
RV9CUkVBS3ZhdWx0
********************************************

Please enter a Digit::   0: Store a password  1: Check whether password
exists  2: Exit
2
```

La chaîne qui sera ajoutée au code Java est la dernière valeur des sortie, la ligne commençant par **VAULT**.

Le servlet suivant utilise la chaîne voûtée au lieu d'un mot de passe de texte clair. La version en texte clair est commentée afin que vous puissiez voir la différence.

**Exemple 10.29. Servlet qui utilise un mot de passe Vaulted**

```
package vaulterror.web;

import java.io.IOException;
import java.io.Writer;

import javax.annotation.Resource;
import javax.annotation.sql.DataSourceDefinition;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;


/*@DataSourceDefinition(
        name = "java:jboss/datasources/LoginDS",
        user = "sa",
        password = "sa",
        className = "org.h2.jdbcx.JdbcDataSource",
        url = "jdbc:h2:tcp://localhost/mem:test"
)*/
@DataSourceDefinition(
        name = "java:jboss/datasources/LoginDS",
        user = "sa",
        password =
"VAULT::DS::thePass::OWY5M2I5NzctYzdkOS00MmZhLWExZGYtNjczM2U5ZGUyOWIxTEl
ORV9CUkVBS3ZhdWx0",
        className = "org.h2.jdbcx.JdbcDataSource",
        url = "jdbc:h2:tcp://localhost/mem:test"
)
@WebServlet(name = "MyTestServlet", urlPatterns = { "/my/" },
loadOnStartup = 1)
public class MyTestServlet  extends HttpServlet {

    private static final long serialVersionUID = 1L;


    @Resource(lookup = "java:jboss/datasources/LoginDS")
    private DataSource ds;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        Writer writer = resp.getWriter();
        writer.write((ds != null) + "");
    }
}
```

Votre servlet est maintenant capable de résoudre le string Vaulted.

Report a bug

# 10.13. ENCODAGE SE CONFORMANT À FIPS 140-2

## 10.13.1. Conformité FIPS 140-2

FIPS (Federal Information Processing Standard) 140-2 (FIPS 140-2) est un standard de sécurité informatique gouvernemental US pour l'accréditation des modules informatiques cryptographiques. Le standard FIPS 140-2 est souvent un pré-requis pour les systèmes informatiques des agences gouvernementales et pour le secteur commercial privé.

JBoss Enterprise Application Platform 6 utilise le chiffrement de modules externes et peut être configuré pour pouvoir utiliser un module de chiffrement compatible FIPS 140-2.

Report a bug

## 10.13.2. Mots de passe conformes FIPS 140-2

Un mot de passe conforme FIPS doit avoir les caractéristiques suivantes :

1. Une longueur minimale de sept (7) caractères.
2. Il doit inclure des caractères d'au moins trois (3) classes de caractères suivantes :
    - chiffres ASCII,
    - minuscules ASCII,
    - Majuscules ASCII,
    - non alphanumériques ASCII, et
    - non-ASCII.

Si le premier caractère du mot de passe est en lettre majuscule ASCII, il ne comptera pas comme une lettre majuscule ASCII pour la restriction 2.

Si le dernier caractère du mot de passe est un chiffre ASCII, il ne comptera pas comme chiffre ASCII pour la restriction 2.

Report a bug

## 10.13.3. Active la Cryptography FIPS 140-2 pour SSL dans Red Hat Enterprise Linux 6

Cette tâche décrit comment configurer le conteneur web (JBoss Web) de JBoss Enterprise Application Platform 6 pour que la cryptographie soit conforme à FIPS 140-2 pour SSL. Cette tâche ne couvre que les étapes spécifiques à Red Hat Enterprise Linux 6.

Cette tâche utilise la bibliothèque Mozilla NSS en mode FIPS pour cette fonctionnalité.

**Prérequis**

- Red Hat Enterprise Linux 6 doit déjà être configuré pour être configuré en conformité avec FIPS 140-2. Voir https://access.redhat.com/knowledge/solutions/137833.

**Procédure 10.19. Voir Conformité Cryptographie FIPS 140-2 pour SSL**

1. **Créer la base de données**
   Créer la base de données NSS dans un répertoire qui appartienne à l'utilisateur **`jboss`**.

   ```
   $ mkdir -p  /usr/share/jboss-as/nssdb
   $ chown jboss /usr/share/jboss-as/nssdb
   $ modutil -create -dbdir /usr/share/jboss-as/nssdb
   ```

2. **Créer un fichier de configuration NSS**
   Créer un nouveau fichier texte ayant comme nom **`nss_pkcsll_fips.cfg`** dans le répertoire **`/usr/share/jboss-as`** avec le contenu suivant :

   ```
   name = nss-fips
   nssLibraryDirectory=/usr/lib64
   nssSecmodDirectory=/usr/share/jboss-as/nssdb
   nssModule = fips
   ```

   Le fichier de configuration NSS doit spécifier :

   - un nom,
   - le répertoire où se trouve la bibliothèque, et
   - le répertoire où la base de données NSS a été créée selon l'étape 1.

   Si vous n'êtes pas sur une version 64bit de Red Hat Enterprise Linux 6, alors définir **`nssLibraryDirectory`** à **`/usr/lib`** à la place de **`/usr/lib64`**.

3. **Activer le fournisseur SunPKCS11**
   Modifier le fichier de configuration **`java.security`** de votre JRE (**`$JAVA_HOME/jre/lib/security/java.security`**) et ajouter la ligne suivante :

   ```
   security.provider.1=sun.security.pkcs11.SunPKCS11  /usr/share/jboss-
   as/nss_pkcsll_fips.cfg
   ```

   Notez que le fichier de configuration spécifié sur cette ligne est le fichier créé à l'étape 2.

   Toute autre ligne **`security.provider.`*`X`*** de ce fichier devra posséder la valeur *X* +1 pour que la priorité soit donnée à ce fournisseur.

4. **Activer le mode FIPS pour la bibliothèque NSS**
   Exécutez la commande **`modutil`** comme indiqué pour activer le mode FIPS :

   ```
   modutil -fips true -dbdir /usr/share/jboss-as/nssdb
   ```

   Notez que le répertoire indiqué ici est le répertoire créé à l'étape 1.

   Vous aurez sans doute une erreur de bibliothèque à ce niveau, ce qui vous oblige à régénérer les signatures de bibliothèques sur certains des objets NSS partagés.

5. **Modifier le mot de passe du token FIPS**
   Définir le mot de passe du token FIPS par la commande suivante, en remplaçant *PASSWORD* par le mot de passe qui convient :

   ```
   Modutil -changepw "PASSWORD" -dbdir /usr/share/jboss-as/nssdb
   ```

–

Le mot de passe utilisé pour le token FIPS doit être un mot de passe conforme FIPS.

6. **Créer le certificat grâce aux outils NSS**
   Saisir la commande suivante pour créer un certificat par les outils NSS.

```
certutil -S -k rsa -n jbossweb  -t "u,u,u" -x -s "CN=localhost,
OU=MYOU, O=MYORG, L=MYCITY, ST=MYSTATE, C=MY" -d /usr/share/jboss-
as/nssdb
```

7. **Configurer le connecteur HTTPS pour utiliser le keystore PKCS11**
   Ajouter un connecteur HTTPS par la commande suivante dans JBoss CLI :

```
/subsystem=web/connector=https/:add(socket-
binding=https,scheme=https,protocol=HTTP/1.1,secure=true)
```

   Puis, ajouter la configuration SSL par la commande suivante, en remplaçant PASSWORD par le mot de passe conforme FIPS de l'étape 4.

```
/subsystem=web/connector=https/ssl=configuration:add(name=https,pass
word=PASSWORD,keystore-type=PCKS11,
cipher-
suite="SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_DHE_RSA_WITH_3DES_EDE_CBC_S
HA,
TLS_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,
TLS_DHE_DSS_WITH_AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_SHA,
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_ECDSA_WITH_AES_128_CBC
_SHA,
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_3DES_EDE_CB
C_SHA,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_CB
C_SHA,
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
,
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SH
A,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SH
A,
TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_anon_WITH_AES_128_CBC_S
HA,
TLS_ECDH_anon_WITH_AES_256_CBC_SHA")
```

8. **Vérifier**
   Vérifier que la JVM puisse lire la clé privée du keystore PKCS11 en exécutant la commande suivante :

```
keytool -list -storetype pkcs11
```

**Exemple 10.30. Configuration XML du connecteur HTTPS avec conformité FIPS 140-2**

```
<connector name="https" protocol="HTTP/1.1" scheme="https" socket-
binding="https" secure="true">
  <ssl name="https" password="****"
```

```
      cipher-
suite="SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA,
         TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
         TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,

TLS_DHE_DSS_WITH_AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_SHA,

TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
,

TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SH
A,

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SH
A,

TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_RSA_WITH_AES_128_CBC_SHA,

TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA,

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,

TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_anon_WITH_AES_128_CBC_SHA,
         TLS_ECDH_anon_WITH_AES_256_CBC_SHA"
      keystore-type="PKCS11"/>
</connector>
```

Notez que l'attribut **`cipher-suite`** a des sauts de ligne insérés pour faciliter la lecture.

Report a bug

# CHAPITRE 11. RÉFÉRENCE ADMINISTRATION SÉCURITÉ

## 11.1. MODULES D'AUTHENTIFICATION INCLUS

Les modules d'authentification suivants sont inclus dans JBoss Enterprise Application Platform. Certains d'entre eux gèrent l'autorisation ainsi que l'authentification. Ceux-ci incluent généralement le mot **`Role`** dans le nom de **`Code`**

Quand vous configurez ces modules, utiliser la valeur **`Code`** ou le nom complet (package) pour vous référer au module.

**Modules d'authentification**

- Tableau 11.40, « **`UsersRoles`** »
- Modules d'authentification personnalisés

**Tableau 11.1. `Client`**

| | |
|---|---|
| Code | **`Client`** |
| Class | **`org.jboss.security.ClientLoginModule`** |
| Description | Ce module de connexion est conçu pour établir l'identité de l'appelant et les informations d'identification lorsque JBoss Enterprise Application Platform agit en tant que client. Il ne doit jamais servir sous forme de domaine de sécurité pour l'authentification serveur. |

**Tableau 11.2. Options de Modules Client**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`multi-threaded`** | **`true`** ou **`false`** | **`false`** | Définir la valeur sur true si chaque thread possède son propre stockage d'informations d'identification et principal. La valeur false pour indiquer que tous les threads de la machine virtuelle partagent la même identité et informations d'identification. |
| **`password-stacking`** | **`useFirstPass`** ou **`false`** | **`false`** | Définir la valeur sur useFirstPass pour indiquer que ce module de connexion doive rechercher les informations stockées dans le LoginContext à utiliser comme identité. Cette option peut être utilisée lorsque vous empilez les autres modules de connexion avec. |
| **`>restore-login-identity`** | **`true`** ou **`false`** | **`false`** | Définir sur true si l'identité et les informations d'identification du début le la méthode login() doivent être restaurées une fois que la méthode logout() est invoquée. |

**Tableau 11.3. `Certificate`**

| | |
|---|---|
| Code | **`Certificate`** |
| Class | **`org.jboss.security.auth.spi.BaseCertLoginModule`** |
| Description | Ce module de connexion est conçu pour authentifier les utilisateurs basés sur des **`X509 Certificates`**. Un cas d'utilisation pour ceci est l'authentification **`CLIENT-CERT`** d'une application web. |

**Tableau 11.4. Options de module `Certificate`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`securityDomain`** | chaîne | aucun | Nom du domaine de sécurité qui possède la configuration JSSE du truststore qui contient les certificats approuvés. |
| **`verifier`** | Class | aucun | Le nom de classe du org.jboss.security.auth.certs.X509CertificateVerifier à utiliser pour vérifier le certificat de connexion. |

**Tableau 11.5. `CertificateUsers`**

| | |
|---|---|
| Code | **`CertificateUsers`** |
| Class | **`org.jboss.security.auth.spi.UsersRolesLoginModule`** |
| Description | Utilise deux ressources de propriété. La première mappe les noms d'utilisateur aux mots de passe, et la seconde mappe les noms d'utilisateurs aux rôles. |

**Tableau 11.6. Options de module `CertificateUsers`**

| Option | Type | Par défaut | Description |
|---|---|---|---|

| Option | Type | Par défaut | Description |
| --- | --- | --- | --- |
| **`unauthenticatedIdentity`** | Une chaîne | aucun | Définit le nom principal devant être affecté aux demandes qui ne contiennent aucune information d'authentification. Cela peut permettre à des servlets non protégés d'appeler des méthodes sur EJB ne nécessitant pas un rôle spécifique. Un tel principal ne possédera aucun rôle associé et ne ne pourra accéder qu'aux méthodes EJB ou EJB non sécurisées associées à la contrainte de **`permission non contrôlée`**. |
| **`password-stacking`** | **`useFirstPass`** ou **`false`** | **`false`** | Définir la valeur sur **`useFirstPass`** pour indiquer que ce module de connexion doive rechercher les informations stockées dans le **`LoginContext`** à utiliser comme identité. Cette option peut être utilisée lorsque vous empilez les autres modules de connexion avec celui-ci. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **hashAlgorithm** | Une chaîne | aucun | Le nom de l'algorithme **java.security.MessageDigest** à utiliser pour hacher le mot de passe. Il n'y a pas de valeur par défaut pour cette option, donc vous devez la définir explicitement pour permettre le hachage. Quand hashAlgorithm est spécifié, le mot de passe en texte clair obtenu de **CallbackHandler** sera haché avant de passer à **UsernamePasswordLoginModule.validatePassword** comme argument d'inputPassword. Le **expectedPassword** stocké dans le fichier **users.properties** doit être haché de façon similaire. Voir http://docs.oracle.com/javase/6/docs/api/java/security/MessageDigest.html pour obtenir des informations sur **java.security.MessageDigest** |
| **hashEncoding** | **base64** ou **hex** | **base64** | Le format du string du mot de passe haché, si **hashAlgorithm** est défini également. |
| **hashCharSet** | Une chaîne | Le codage par défaut défini dans l'environnement du conteneur. | Le codage utilisé pour convertir le mot de passe de texte en clair en un tableau d'octets. |
| **usersProperties** | Le chemin d'accès complet et le nom d'une ressource ou d'un fichier de propriétés | **users.properties** | Le fichier qui contient les mappages entre les utilisateurs et les mots de passe. Chaque propriété du fichier a le format **username=password**. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **rolesProperties** | Le chemin d'accès complet et le nom d'une ressource ou d'un fichier de propriétés | **roles.properties** | Le fichier qui contient les mappages entre les utilisateurs et les rôles. Chaque propriété du fichier a le format **username=role1,role2,...,roleN** |
| **ignorePasswordCase** | **true** ou **false** | **false** | Si la comparaison de mot de passe doit ignorer la casse. Ceci est utile pour le codage de mots de passe hachés quand le mot de passe haché n'est pas significatif. |
| **principalClass** | Un nom de classe complet. | aucun | Une classe d'implémentation de **Principal**contenant un constructeur qui prend l'argument du String comme nom de principal. |
| **roleGroupSeparator** | Un seul caractère | **.** (un seul point) | Le caractère utilisé pour séparer le nom d'utilisateur du nom de groupe de rôle dans le fichier **rolesGroup**. |
| **defaultUsersProperties** | chaîne | **defaultUsers.properties** | Nom de la ressource ou du fichier où se replier si le fichier usersProperties est introuvable. |
| **defaultRolesProperties** | chaîne | **defaultRoles.properties** | Nom de la ressource ou du fichier où se replier si le fichier **rolesProperties** est introuvable. |
| **hashUserPassword** | **true** ou **false** | **true** | Indique si on doit hacher le mot de passe entré par l'utilisateur, lorsqu'**hashAlgorithm** est indiqué. La valeur par défaut est **true**. |

| Option | Type | Par défaut | Description |
| --- | --- | --- | --- |
| **hashStorePassword** | **true** ou **false** | **true** | Indique le mot de passe de store retourné quand le **getUsersPassword()** doit être haché, quand **hashAlgorithm** est spécifié. |
| **digestCallback** | Un nom de classe complet. | aucun | Le nom de classe de l'implémentation **org.jboss.crypto.digest.DigestCallback** qui inclut le contenu pre/post digest comme les valeurs salt. Utilisé uniquement si l'**hashAlgorithm** est spécifié. |
| **storeDigestCallback** | Un nom de classe complet. | aucun | Le nom de classe de l'implémentation **org.jboss.crypto.digest.DigestCallback** qui inclut le contenu pre/post digest comme les valeurs salt pour hacher le mot de passe du store. Utilisé uniquement si l'**hashStoreAlgorithm** est sur true et si **hashAlgorithm** est spécifié. |
| **callback.option.*STRING*** | Divers | aucun | Toutes les options ayant comme préfixe **callback.option.** sont passées à la méthode **DigestCallback.init(Map)**. Le nom d'utilisateur entré est toujours passé par l'option **javax.security.auth.login.name**, et le mot de passe input/store password est passé par l'option **javax.security.auth.login.password** au **digestCallback** ou au **storeDigestCallback**. |

**Tableau 11.7. `CertificateRoles`**

| | |
|---|---|
| Code | **`CertificateRoles`** |
| Class | **`org.jboss.security.auth.spi.CertRolesLoginModule`** |
| Description | Ce module de connexion étend le module de connexion de certificat pour ajouter des fonctions de mappage de rôle à partir d'un fichier de propriétés. Il prend les mêmes options que le module de connexion du certificat et ajoute les options suivantes. |

**Tableau 11.8. Options de module `CertificateRoles`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`rolesProperties`** | Une chaîne | **`roles.properties`** | Le nom de la ressource ou du fichier contenant les rôles à attribuer à chaque utilisateur. Le fichier de propriétés de rôle doit être dans sous format nom d'utilisateur=role1,role2, où le nom d'utilisateur est le nom unique du certificat, sans signe égal ou espace blanc. L'exemple suivant est dans le bon format :<br>`CN\=unit-tests-client,\ OU\=Red\ Hat\ Inc.,\ O\=Red\ Hat\ Inc.,\ ST\=North\ Carolina,\ C\=US=JBossAdmin` |
| **`defaultRolesProperties`** | Une chaîne | **`defaultRoles.properties`** | Nom de la ressource ou du fichier où se replier si le fichier **`rolesProperties`** est introuvable. |
| **`roleGroupSeparator`** | Un seul caractère | `.` (un seul point) | Le caractère à utiliser comme séparateur de groupe rôle dans le fichier de **`roleProperties`** |

**Tableau 11.9. `Database`**

| | |
|---|---|
| Code | **Database** |
| Class | **org.jboss.security.auth.spi.DatabaseServerLoginModule** |
| Description | Un module de connexion basé-JDBC supportant le mappage de rôle et l'authentification. Basé sur deux tables logiques, avec les définitions suivantes.<br>• **Principals: PrincipalID (text), Password (text)**<br>• **Roles: PrincipalID (text), Role (text), RoleGroup (text)** |

**Tableau 11.10. `Database` Module Options**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **dsJndiName** | Ressource JNDI | aucun | Le nom de la ressource JNDI qui store les informations d'authentification. Cette option est requise. |
| **principalsQuery** | Énoncé SQL | **select Password from Principals where PrincipalID=?** | La requête SQL préparée pour obtenir les informations sur le principal. |
| **rolesQuery** | Énoncé SQL | **select Role, RoleGroup from Roles where PrincipalID=?** | Enoncé SQL préparé en vue d'exécuter pour mapper les rôles. Doit être équivalent à **select Role, RoleGroup from Roles where PrincipalID=?**, avec Role comme nom de rôle et la valeur de la colonne RoleGroup doit toujours être **Roles**. avec un **R** majuscule. |

**Tableau 11.11. `DatabaseCertificate`**

| | |
|---|---|
| Code | **DatabaseCertificate** |
| Class | **org.jboss.security.auth.spi.DatabaseCertLoginModule** |

| | |
|---|---|
| Description | Ce module de connexion étend le module de connexion de certificat pour ajouter des fonctions de mappage de rôle d'une table de bases de données. Il possède les mêmes options mais aussi ces options supplémentaires : |

**Tableau 11.12. `DatabaseCertificate` Module Options**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`dsJndiName`** | Ressource JNDI | | Le nom de la ressource JNDI qui store les informations d'authentification. Cette option est requise. |
| **`rolesQuery`** | Énoncé SQL | **`select Role,RoleGroup from Roles where PrincipalID=?`** | Enoncé SQL préparé en vue d'exécuter pour mapper les rôles. Doit être équivalent à **`select Role, RoleGroup from Roles where PrincipalID=?`**, avec Role comme nom de rôle et la valeur de la colonne RoleGroup doit toujours être **`Roles`**. avec un **R** majuscule. |
| **`suspendResume`** | **`true`** ou **`false`** | **`true`** | Indique si une transaction JTA existante doit être suspendue pendant les opérations de base de données. |

**Tableau 11.13. `Identity`**

| | |
|---|---|
| Code | **`Identity`** |
| Class | **`org.jboss.security.auth.spi.IdentityLoginModule`** |
| Description | Associe le principal spécifié dans les options de module avec n'importe quel sujet authentifié dans le module. Le type de classe de principal utilisée est **`org.jboss.security.SimplePrincipal.`**. Si aucune option de principal n'est spécifiée, on utilisera un principal ayant pour nom **`guest`**. |

**Tableau 11.14. Options de module `Identity`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **principal** | Une chaîne | **guest** | Le nom à utiliser pour le principal. |
| **roles** | Une liste de strings séparée par des virgules | aucun | Une liste de rôles séparée par des virgules qui sera assignée au sujet. |

**Tableau 11.15. Ldap**

| | |
|---|---|
| Code | **Ldap** |
| Class | **org.jboss.security.auth.spi.LdapLoginModule** |
| Description | Authentifie sur un serveur LDAP, lorsque le nom d'utilisateur et le mot de passe sont stockés dans un serveur LDAP qui est accessible à l'aide d'un fournisseur LDAP JNDI. Bon nombre des options ne sont pas requises, car elles sont déterminées par le fournisseur LDAP ou l'environnement. |

**Tableau 11.16. Options de module Ldap**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **java.naming.factory.initial** | class name | **com.sun.jndi.ldap.LdapCtxFactory** | Nom de classe de l'implémentation **InitialContextFactory**. |
| **java.naming.provider.url** | **ldap://** URL | aucun | URL pour le serveur LDAP |
| **java.naming.security.authentication** | **none**, **simple**, ou le nom d'un mécanisme SASL | **simple** | Le niveau de sécurité à utiliser pour la liaison avec le serveur LDAP. |
| **java.naming.security.protocol** | Protocole de transport | Si non spécifié, déterminé par le fournisseur. | Le protocole de transport à utiliser pour l'accès sécurisé, comme SSL. |
| **java.naming.security.principal** | Une chaîne | aucun | Le nom du principal permettant d'authentifier l'appelant vers le service. Il est construit à partir des autres propriétés décrites ci-dessous. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **java.naming.security.credentials** | Un type d'information d'identification | aucun | Le type d'information d'identification utilisée par le schéma d'authentification. Exemples : mot de passe haché, mot de passe de texte clair, clé ou certificat. Si cette propriété n'est pas spécifiée, le comportement est déterminé par le fournisseur de service. |
| **principalDNPrefix** | Une chaîne | aucun | Préfixe ajouté au nom d'utilisateur pour former le DN de l'utilisateur. Vous pouvez demander à l'utilisateur un nom d'utilisateur et créer le nom de domaine DN complet en utilisant **principalDNPrefix** et **principalDNSuffix**. |
| **principalDNSuffix** | chaîne | | Suffixe ajouté au nom d'utilisateur pour former le DN de l'utilisateur. Vous pouvez demander à l'utilisateur un nom d'utilisateur et créer le nom de domaine DN complet en utilisant **principalDNPrefix** et **principalDNSuffix**. |
| **useObjectCredential** | **true** ou **false** | false | Si les informations d'identification doivent être obtenues sous forme d'objet opaque à l'aide du type de Callback **org.jboss.security.auth.callback.ObjectCallback** plutôt que comme mot de passe char[] utilisant un JAAS PasswordCallback. Cela permet de passer les informations d'identification non-char[] au serveur Ldap. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **rolesCtxDN** | DN complet | aucun | Le DN complet pour le contexte à chercher pour les rôles d'utilisateur. |
| **userRolesCtxDNAttributeName** | Attribut | aucun | L'attribut dans l'objet utilisateur qui contient le nom unique DN pour que le contexte rechercher des rôles d'utilisateur. Cela diffère de **rolesCtxDN** dans ce contexte de recherche car les rôles d'un utilisateur peuvent être uniques pour chaque utilisateur. |
| **rolesAttributeID** | Attribut | **roles** | Nom de l'attribut qui contient les rôles d'utilisateur. |
| **rolesAttributeIsDN** | **true** ou **false** | **false** | Indique si le **roleAttributeID** contient le nom de domaine complet d'un objet de rôle. Si false, le nom de rôle est tiré de la valeur de l'attribut **roleNameAttributeId** du nom de contexte. Certains schémas de répertoire, tel que Active Directory, requièrent que cet attribut soit défini à **true**. |
| **rolesNameAttributeID** | Attribut | **group** | Nom de l'attribut du contexte de **roleCtxDN** qui contient le nom de rôle. Si la propriété **roleAttributeIsDN** est définie sur **true**, cette propriété sera utilisée pour rechercher l'attribut de nom de l'objet rôle. |
| **uidAttributeID** | Attribut | **uid** | Nom de l'attribut qui contient le **UserRolesAttributeDN** correspondant à l'ID d'utilisateur. Utilisé pour localiser les rôles d'utilisateur. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **matchOnUserDN** | **true** ou **false** | **false** | Si la recherche de rôles d'utilisateur doit correspondre au DN de l'utilisateur entièrement distinct ou au nom d'utilisateur uniquement. Si **true**, l'userDN complet sera utilisé comme valeur de correspondance. Si **false**, seul le nom d'utilisateur sera utilisé comme valeur de correspondance pour l'attribut **UserRolesAttributeDN**. |
| **allowEmptyPasswords** | **true** ou **false** | **true** | Indique si on doit autoriser les mots de passe vides. La plupart des serveurs LDAP traitent les mots de passe vides comme des tentatives de connexion anonymes. Pour rejeter les mots de passe vides, définir à false. |

**Tableau 11.17. LdapExtended**

| | |
|---|---|
| Code | **LdapExtended** |
| Class | **org.jboss.security.auth.spi.LdapExtLoginModule** |

| | |
|---|---|
| Description | Une autre implémentation de module de connexion LDAP qui utilise les recherches pour localiser l'utilisateur de liaisons et rôles associés. La requête de rôles suit récursivement les noms de domaines DN pour naviguer dans une structure de rôles hiérarchique. Il utilise les mêmes options **`java.naming`** que le module Ldap, et utilise les options suivantes à la place des autres options du module Ldap.<br><br>L'authentification a lieu en 2 étapes :<br><br>1. Une première liaison au serveur LDAP est faite par les options **`bindCredential`**. **`bindDN`** est l'utilisateur ayant la possibilité de rechercher les arborescences baseCtxDN et **`rolesCtxDN`** pour l'utilisateur et les rôles. Le DN pour authentifier l'utilisateur est interrogé en utilisant le filtre spécifié par l'attribut **`baseFilter`**.<br><br>2. Le DN de l'utilisateur qui en résulte est authentifié par la liaison au serveur LDAP en utilisant le DN de l'utilisateur comme environnement de **`InitialLdapContext`** **`Context.SECURITY_PRINCIPAL`**. La propriété **`Context.SECURITY_CREDENTIALS`** est définie avec le mot de passe String obtenu par le gestionnaire de rappel. |

**Tableau 11.18. Options de module `LdapExtended`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`baseCtxDN`** | DN complet | aucun | Le DN fixe de contexte de niveau supérieur pour commencer la recherche utilisateur. |
| **`bindDN`** | DN complet | aucun | Le DN utilisé pour la liaison au serveur LDAP pour les requêtes d'utilisateurs et de rôles. Ce nom unique a besoin d'autorisations de lecture et de recherche pour les valeurs **`baseCtxDN`** et **`rolesCtxDN`**. |
| **`bindCredential`** | Un string, parfois crypté | aucun | Le mot de passe pour le **`bindDN`**. Peut être crypté si le **`jaasSecurityDomain`** est spécifié. |

| Option | Type | Par défaut | Description |
| --- | --- | --- | --- |
| **jaasSecurityDomain** | JMX ObjectName | aucun | L' **ObjectName** JMX du **JaasSecurityDomain** à utiliser pour décrypter les **bindCredential**. La forme cryptée du mot de passe correspond au format retourné par la méthode **JaasSecurityDomain.encrypt64(byte[])**. |
| **baseFilter** | String de filtre LDAP | aucun | Un filtre de recherche permettant de localiser le contexte de l'utilisateur à authentifier. L'entrée username ou userDN obtenue à partir du rappel de module de connexion est substitué dans le filtre à chaque fois qu'une expression **{0}** est utilisée. Un exemple de filtre de recherche est **(uid={0})**. |
| **rolesCtxDN** | DN complet | aucun | Le DN fixe du contexte pour rechercher des rôles d'utilisateur. Ce n'est pas le DN où les rôles se trouvent, mais le DN où les objets contenant les rôles d'utilisateur se trouvent. Par exemple, dans un serveur Active Directory de Microsoft, c'est le DN où le compte d'utilisateur se trouve. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **roleFilter** | String de filtre LDAP | | Un filtre de recherche permettant de localiser les rôles associés à l'utilisateur authentifié. L'entrée user ou userDN obtenue à partir du rappel de module de connexion est substitué dans le filtre à chaque fois qu'une expression **{0}** est utilisée. L'utilisateurDN (userDN) authentifié sera substitué dans le filtre à chaque fois qu'un **{1}** est utilisé. Un exemple de filtre de recherche qui correspond au nom d'utilisateur d'entrée serait **(member={0})**. Un autre correspondant au userDN authentifié pourrait être **(member={1})**. |
| **roleAttributeIsDN** | **true** ou **false** | **false** | Indique si le **roleAttributeID** contient le nom de domaine complet d'un objet de rôle. Si false, le nom de rôle est tiré de la valeur de l'attribut **roleNameAttributeId** du nom de contexte. Certains schémas de répertoire, tel que Active Directory, requièrent que cet attribut soit défini à **true**. |
| **defaultRole** | Un nom de rôle. | aucun | Un rôle inclus pour tous les utilisateurs authentifiés |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **parseRoleNameFromDN** | **true** ou **false** | **false** | Un indicateur qui signale si le DN retourné par une requête contient le roleNameAttributeID. Si la valeur est **true**, le DN est vérifié pour le roleNameATtributeID. Si la valeur est **false**, le DN n'est pas coché pour le roleNameAttributeID. Cet indicateur peut améliorer les performances des requêtes LDAP. |
| **parseUsername** | **true** ou **false** | **false** | Un indicateur qui signale si le DN doit être vérifié niveau nom d'utilisateur. Si la valeur est **true**, le DN est vérifié niveau nom d'utilisateur. Si la valeur est **false**, le DN n'est pas vérifié au niveau nom d'utilisateur. Cet option peut à la fois être utilisée pour usernameBeginString et usernameEndString. |
| **usernameBeginString** | une chaîne | aucun | Définit le string qui doit être supprimé au début du DN pour révéler le nom d'utilisateur. Cette option est utilisée avec **usernameEndString**. |
| **usernameEndString** | une chaîne | aucun | Définit le string qui doit être supprimé à la fin du DN pour révéler le nom d'utilisateur. Cette option est utilisée avec **userBeginString**. |
| **roleNameAttributeID** | Attribut | **group** | Nom de l'attribut du contexte de **roleCtxDN** qui contient le nom de rôle. Si la propriété **roleAttributeIsDN** est définie sur **true**, cette propriété sera utilisée pour rechercher l'attribut de nom de l'objet rôle. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **distinguishedNameAttribute** | Attribut | **distinguishedName** | Le nom de l'attribut dans l'entrée de l'utilisateur qui contient le nom unique de l'utilisateur. Ceci peut être nécessaire si le DN de l'utilisateur lui-même contient des caractères spéciaux (barre oblique inverse par exemple) qui empêchent le mappage de l'utilisateur. Si l'attribut n'existe pas, le DN de l'entrée sera utilisé. |
| **roleRecursion** | Un entier relatif | **0** | Le nombre de niveaux de récursivité de la recherche de rôle dans un contexte de correspondance donné. Désactiver la récursivité en attribuant le paramètre **0**. |
| **searchTimeLimit** | Un entier relatif | **10000** (10 seconds) | Timeout en millisecondes pour les recherches utilisateur/rôle. |
| **searchScope** | Un parmi : **OBJECT_SCOPE, ONELEVEL_SCOPE, SUBTREE_SCOPE** | **SUBTREE_SCOPE** | Étendue à utiliser. |
| **allowEmptyPasswords** | **true** ou **false** | **true** | Indique si on doit autoriser les mots de passe vides. La plupart des serveurs LDAP traitent les mots de passe vides comme des tentatives de connexion anonymes. Pour rejeter les mots de passe vides, définir à false. |

**Tableau 11.19. `RoleMapping`**

| | |
|---|---|
| Code | **RoleMapping** |
| Class | **org.jboss.security.auth.spi.RoleMappingLoginModule** |

| | |
|---|---|
| Description | Mappe un rôle qui est le résultat final du processus d'authentification de manière déclarative. Ce module doit être marqué comme étant **optionnel** quand vous y ajoutez le domaine de sécurité. |

**Tableau 11.20. Options de module RoleMapping**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **rolesProperties** | Le chemin d'accès complet et le nom d'une ressource ou d'un fichier de propriétés | **roles.properties** | Nom de la ressource ou du fichier de propriétés qui mappe les rôles aux rôles de remplacement. Le format est **original_role=role1,role2,role3** |
| **replaceRole** | **true** ou **false** | **false** | Indique si on doit ajouter les rôles en cours, ou les remplacer par les rôles mappés. Sont remplacés si définis sur **true**. |

**Tableau 11.21. RunAs**

| | |
|---|---|
| Code | **RunAs** |
| Class | **Class: org.jboss.security.auth.spi.RunAsLoginModule** |
| Description | Un module d'assistance qui pousse un rôle **run as** dans la pile pour la durée de la phase d'authentification de la connexion, et qui extrait le rôle **run as** de la pile soit dans la phase commit ou abort. Ce module de connexion fournit un rôle pour les autres modules de connexion qui doivent accéder aux ressources sécurisées afin d'effectuer leur authentification, par exemple un module de connexion qui accède à un EJB sécurisé. **RunAsLoginModule** doit être configuré avant que les modules de connexion qui ont besoin d'une rôle **run as** soient mis en place. |

**Tableau 11.22. Options RunAs**

| Option | Type | Par défaut | Description |
|---|---|---|---|

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **roleName** | Un nom de rôle. | **nobody** | Le nom de rôle à utiliser comme rôle **run as** pendant la phase de connexion. |

**Tableau 11.23. Simple**

| | |
|---|---|
| Code | **Simple** |
| Class | **org.jboss.security.auth.spi.SimpleServerLoginModule** |
| Description | Module d'installation rapide de sécurité pour les tests. Implémente ce simple algorithme :<br>• Si le mot de passe est null, authentifie l'utilisateur et assigne une identité **guest** et un rôle **guest**.<br>• Sinon, quand le mot de passe est égal à l'utilisateur, assigne une identité égale au nom d'utilisateur et aux deux rôles **admin** et **guest**.<br>• Sinon, l'authentification échoue. |

### Simple Module Options

Le module **Simple** n'a pas d'options.

**Tableau 11.24. ConfiguredIdentity**

| | |
|---|---|
| Code | **ConfiguredIdentity** |
| Class | **org.picketbox.datasource.security.ConfiguredIdentityLoginModule** |
| Description | Associe le principal spécifié dans les options du module avec n'importe quel sujet authentifié dans le module. Le type de classe du Principal classe est **org.jboss.security.SimplePrincipal**. |

**Tableau 11.25. ConfiguredIdentity Module Options**

| Option | Type | Par défaut | Description |
|---|---|---|---|

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`principal`** | Nom d'un principal. | **`guest`** | Le principal qui sera associé avec n'importe quel sujet authentifié dans le module. |

**Tableau 11.26. `SecureIdentity`**

| | |
|---|---|
| Code | **`SecureIdentity`** |
| Class | **`org.picketbox.datasource.security.SecureIdentityLoginModule`** |
| Description | Ce module est fourni à des fins d'héritage. Il permet de crypter un mot de passe et ensuite d'utiliser le mot de passe crypté avec un principal statique. Si votre application utilise **`SecureIdentity`**, envisager plutôt d'utiliser un mécanisme d'archivage sécurisé de mot de passe. |

**Tableau 11.27. Options de module `SecureIdentity`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`username`** | chaîne | aucun | Le nom d'utilisateur pour l'authentification. |
| **`password`** | string encodé | aucun | Le mot de passe à utiliser pour l'authentification. Pour crypter le mot de passe, utilisez le module directement dans la ligne de commande.<br>`java org.picketbox.datasource.security.SecureIdentityLoginModule` *`password_to_encrypt`*<br>Coller le résultat de cette commande dans le champ d'option de module. |
| **`managedConnectionFactoryName`** | Une ressource JCA | aucun | Le nom de la fabrique de connexions JCA à utiliser pour votre source de données. |

**Tableau 11.28. `PropertiesUsers`**

| | |
|---|---|
| Code | **`PropertiesUsers`** |
| Class | **`org.jboss.security.auth.spi.PropertiesUsersLoginModule`** |
| Description | Utilise un fichier de propriétés pour stocker les noms d'utilisateur et mots de passe pour l'authentification. Aucune autorisation (correspondance de rôle) n'est fournie. Ce module convient seulement pour les tests. |

**Tableau 11.29. Options de module `PropertiesUsers`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`properties`** | Le chemin d'accès complet et le nom d'une ressource ou d'un fichier de propriétés Java. | aucun | Le fichier de propriétés contenant les noms d'utilisateur et mots de passe en texte clair à utiliser pour l'authentification. |

**Tableau 11.30. `SimpleUsers`**

| | |
|---|---|
| Code | **`SimpleUsers`** |
| Class | **`org.jboss.security.auth.spi.SimpleUsersLoginModule`** |
| Description | Ce module de connexion stocke le nom d'utilisateur et le mot de passe en texte clair dans un fichier de propriétés de Java. Il est inclus pour les essais uniquement et n'est pas approprié pour un environnement de production. |

**Tableau 11.31. Options de module `SimpleUsers`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`username`** | chaîne | aucun | Le nom d'utilisateur pour l'authentification. |
| **`password`** | chaîne | aucun | Le nom d'utilisateur en texte clair pour l'authentification. |

**Tableau 11.32. `LdapUsers`**

| | |
|---|---|
| Code | **`LdapUsers`** |

| | |
|---|---|
| Class | **`org.jboss.security.auth.spi.LdapUsersLoginModule`** |
| Description | Le module **`LdapUsers`** est remplacé par les modules **`ExtendedLDAP`** et **`AdvancedLdap`**. |

**Tableau 11.33. `Kerberos`**

| | |
|---|---|
| Code | **`Kerberos`** |
| Class | **`com.sun.security.auth.module.Krb5LoginModule`** |
| Description | Effectue l'authentification de connexion Kerberos avec GSSAPI. Ce module fait partie de la structure de sécurité de l'API fourni par Sun Microsystems. Vous trouverez des informations à ce sujet dans http://docs.oracle.com/javase/1.4.2/docs/guide/security/jaas/spec/com/sun/security/auth/module/Krb5LoginModule.html. Ce module devra être mis en correspondance avec un autre module qui gère les mappages d'authentification et de rôles. |

**Tableau 11.34. Options de module `Kerberos`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`storekey`** | **`true`** ou **`false`** | false | Indique si on doit ajouter **`KerberosKey`** dans les informations d'authentification privées du sujet. |
| **`doNotPrompt`** | **`true`** ou **`false`** | false | Si défini à **`true`**, l'utilisateur n'aura pas besoin de mot de passe. |
| **`useTicketCache`** | Valeur booléenne de **`true`** ou **`false`** . | false | Si défini à **`true`**, le GTG sera obtenu à partir du cache du ticket. Si défini sur **`false`**, le cache du ticket ne sera pas utilisé. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`ticketcache`** | Un fichier ou une ressource qui représente un cache de ticket Kerberos. | La valeur par défaut dépend du système d'exploitation que vous utilisez.<br>• Red Hat Enterprise Linux / Solaris: **`/tmp/krb5cc_`*`uid`***, utilisant la valeur UID du système d'exploitation.<br>• Microsoft Windows Server: utilise l'API LSA (Local Security Authority) pour trouver le ticketcache. | Emplacement du ticketcache. |
| **`useKeyTab`** | **`true`** ou **`false`** | false | Indique si on doit obtenir la clé du principal à partir d'un keytab. |
| **`keytab`** | Un fichier ou une ressource représentant un onglet de clé Kerberos. | l'emplacement du fichier de configuration Kerberos du système d'exploitation, ou **`/home/`*`user`*`/krb5.keytab`** | Emplacement du fichier keytab. |
| **`principal`** | Une chaîne | aucun | Le nom du principal. Cela peut être un simple nom d'utilisateur ou un nom de service tel que **`host/testserver.acme.com`**. Utiliser cela au lieu d'obtenir le principal d'un fichier keytab, ou lorsque le fichier keytab contient plus d'un principal. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **useFirstPass** | **true** ou **false** | false | Indique si on doit extraire le nom d'utilisateur et le mot de passe de l'état partagé du module à l'aide de **javax.security.auth.login.name** et de **javax.security.auth.login.password** comme clés. Si l'authentification échoue, il n'y aura pas de nouvelle tentative. |
| **tryFirstPass** | **true** ou **false** | false | Identique à **useFirstPass**, mais si l'authentification échoue, le module utilise le **CallbackHandler** pour récupérer un nouveau nom d'utilisateur et mot de passe. Si la seconde authentification échoue, l'échec sera signalé à l'application appelante. |
| **storePass** | **true** ou **false** | false | Indique si on doit stocker le nom d'utilisateur et le mot de passe de l'état partagé du module. N'a pas lieu si les clés existent déjà dans l'état partagé, ou si l'authentification a échoué. |
| **clearPass** | **true** ou **false** | false | Définir à **true** pour supprimer le nom de l'utilisateur et le mot de passe de l'état partagé une fois que les deux phases d'authentification sont terminées. |

**Tableau 11.35. SPNEGOUsers**

| | |
|---|---|
| Code | **SPNEGOUsers** |
| Class | **org.jboss.security.negotiation.spnego.SPNEGOLoginModule** |

| | |
|---|---|
| Description | Effectue l'authentification de connexion SPNEGO vers un serveur Microsoft Active Directory ou autre environnement qui supporte SPNEGO. SPNEGO peut également transporter les informations d'identification de Kerberos. Ce module a besoin de fonctionner en parallèle à un autre module qui gère l'authentification et le mappage des rôles. |

**Tableau 11.36. Options de module `SPNEGO`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`storeKey`** | **`true`** ou **`false`** | **`false`** | Indique si on doit stocker la clé ou non. |
| **`useKeyTab`** | **`true`** ou **`false`** | **`false`** | Indique si on doit utiliser un keytab. |
| **`principal`** | String représentant un principal d'authentification. | aucun | Le nom du principal pour l'authentification. |
| **`keyTab`** | Un fichier ou une ressource représentant un keytab. | **`none`** | L'emplacement d'un keytab. |
| **`doNotPrompt`** | **`true`** ou **`false`** | **`false`** | Si on doit demander un mot de passe. |
| **`debug`** | **`true`** ou **`false`** | **`false`** | Indique si on doit enregistrer plus de messages verbeux pour pouvoir déboguer. |

**Tableau 11.37. `AdvancedLdap`**

| | |
|---|---|
| Code | **`AdvancedLdap`** |
| Class | **`org.jboss.security.negotiation.AdvancedLdapLoginModule`** |
| Description | Module qui fournit davantage de fonctionnalité, comme SASL et l'utilisation d'un domaine de sécurité JAAS. |

**Tableau 11.38. Options de module `AdvancedLdap`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **bindAuthentication** | chaîne | aucun | Le type d'authentification SASL à utiliser pour se lier au serveur de répertoires. |
| **jassSecurityDomain** | **string** | aucun | Le nom du domaine de sécurité JAAS à utiliser. |
| **java.naming.provider.url** | **string** | aucun | L'URI du serveur de répertoires. |
| **baseCtxDN** | DN (Nom Distinctif) complet | aucun | Le nom distinctif à utiliser comme base pour les recherches. |
| **baseFilter** | Chaîne représentant un filtre de recherche LDAP | aucun | Le filtre à utiliser pour réduire les résultats des recherches. |
| **roleAttributeID** | Chaîne représentant un attribut LDAP | aucun | L'attribut LDAP qui contient les noms des rôles d'autorisation. |
| **roleAttributeIsDN** | **true** ou **false** | **false** | Indique si l'attribut de rôle est un Nom Distinctif (DN). |
| **roleNameAttributeID** | Chaîne représentant un attribut LDAP | aucun | L'attribut contenu dans **RoleAttributeId** qui contient lui-même l'attribut de rôle. |
| **recurseRoles** | **true** ou **false** | **false** | Indique si on doit chercher récursivement des rôles dans **RoleAttributeId**. |

**Tableau 11.39. AdvancedADLdap**

| | |
|---|---|
| Code | **AdvancedADLdap** |
| Class | **org.jboss.security.negotiation.AdvancedADLoginModule** |
| Description | Ce module étend le module de connexion **AdvancedLdap**, et ajoute des paramètres supplémentaires utiles au répertoire Microsoft Active DIrectory. |

**Tableau 11.40. UsersRoles**

| | |
|---|---|
| Code | **`UsersRoles`** |
| Class | **`org.jboss.security.auth.spi.UsersRolesLoginModul`** |
| Description | Un module de connexion supportant des utilisateurs multiples et des rôles utilisateur stockés dans deux fichiers de propriétés différents. |

**Tableau 11.41. Options de module `UsersRoles`**

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **`usersProperties`** | Chemin d'accès à un fichier ou à une ressource. | **`users.properties`** | Fichier ou ressource qui contient les mappages d'utilisateur aux mots de passe. Le format du fichier est ***`user=hashed-password`*** |
| **`rolesProperties`** | Chemin d'accès à un fichier ou à une ressource. | **`roles.properties`** | Fichier ou ressource qui contient les mappages d'utilisateur aux rôles. Le format du fichier est **`username=role1,role2,role3`** |
| **`password-stacking`** | **`useFirstPass`** ou **`false`** | **`false`** | La valeur de **`useFirstPass`** indique si ce module de connexion doit tout d'abord rechercher les informations stockées dans le **`LoginContext`** à utiliser comme identité. Cette option peut être utilisée lorsque vous empilez les autres modules de connexion avec celui-ci. |

| Option | Type | Par défaut | Description |
|---|---|---|---|
| **hashAlgorithm** | Chaîne représentant un algorithme de hachage de mot de passe. | **none** | Le nom de l'algorithme **java.security.MessageDigest** à utiliser pour hacher le mot de passe. Il n'y a pas de valeur par défaut pour cette option, donc vous devez la définir explicitement pour permettre le hachage. Quand **hashAlgorithm** est spécifié, le mot de passe en texte clair obtenu de **CallbackHandler** isera haché avant d'être passé à **UsernamePasswordLoginModule.validatePassword** comme argument **inputPassword**. Le mot de passe stocké dans le fichier **users.properties** doit être haché de façon similaire. |
| **hashEncoding** | **base64** ou **hex** | **base64** | Le format du string du mot de passe haché, si hashAlgorithm est défini également. |
| **hashCharset** | Une chaîne | Le codage par défaut défini dans l'environnement runtime du conteneur. | Le codage utilisé pour convertir le mot de passe de texte en clair en un tableau d'octets. |
| **unauthenticatedIdentity** | Un nom de principal | aucun | Définit le nom principal affecté aux demandes qui ne contiennent aucune information d'authentification. Cela peut permettre à des servlets non protégés d'appeler des méthodes sur EJB ne nécessitant pas un rôle spécifique. Un tel principal ne possédera aucun rôle associé et ne ne pourra accéder qu'aux méthodes EJB ou EJB non sécurisées associées à la contrainte de **permission non contrôlée**. |

### Modules d'authentification personnalisés

Les modules d'authentification sont des implémentations de **`org.jboss.security.LoginModule`**. Consulter la documentation de l'API pour obtenir plus d'informations sur la façon de créer un module d'authentification personnalisé.

Report a bug

## 11.2. MODULES D'AUTORISATION INCLUS

Les modules suivants procurent des services d'autorisation.

| Code | Class |
|---|---|
| DenyAll | org.jboss.security.authorization.modules.AllDenyAuthorizationModule |
| PermitAll | org.jboss.security.authorization.modules.AllPermitAuthorizationModule |
| Delegating | org.jboss.security.authorization.modules.DelegatingAuthorizationModule |
| Web | org.jboss.security.authorization.modules.WebAuthorizationModule |
| JACC | org.jboss.security.authorization.modules.JACCAuthorizationModule |

Report a bug

## 11.3. MODULES DE SÉCURITÉ INCLUS

Les rôles de mappage de sécurité suivants sont fournis dans JBoss Enterprise Application Platform.

| Code | Class |
|---|---|
| PropertiesRoles | org.jboss.security.mapping.providers.role.PropertiesRolesMappingProvider |
| SimpleRoles | org.jboss.security.mapping.providers.role.SimpleRolesMappingProvider |
| DeploymentRoles | org.jboss.security.mapping.providers.DeploymentRolesMappingProvider |
| DatabaseRoles | org.jboss.security.mapping.providers.role.DatabaseRolesMappingProvider |

| Code | Class |
| --- | --- |
| LdapRoles | org.jboss.security.mapping.providers.role.LdapRolesMappingProvider |

Report a bug

## 11.4. MODULES DE FOURNISSEURS D'AUDITING DE SÉCURITÉ INCLUS

La plateforme Enterprise Application Platform fournit un fournisseur d'auditing de sécurité.

| Code | Class |
| --- | --- |
| LogAuditProvider | org.jboss.security.audit.providers.LogAuditProvider |

Report a bug

# CHAPITRE 12. CONFIGURATION DE SOUS-SYSTÈME

## 12.1. APERÇU CONFIGURATION SOUS-SYSTÈME

### Introduction

JBoss Enterprise Application Platform 6 utilise une configuration simplifiée, avec un fichier de configuration par domaine ou par serveur autonome. Dans un domaine autonome, un fichier distinct existe pour chaque contrôleur hôte également. Les modifications apportées à la configuration persistent automatiquement, donc le XML ne doit pas être édité manuellement. La configuration est scannée et automatiquement remplacée par l'API de gestion. La ligne de commande du Management CLI et la Console de gestion basée-web permettent de configurer chaque aspect de JBoss Enterprise Application Platform.

JBoss Enterprise Application Platform 6 repose sur le concept de chargement de classes modulaire. Chaque API ou service fourni par la plateforme est implémenté comme un module, qui est chargé et déchargé à la demande. La plupart des modules incluent un élément configurable appelé un sous-système. Les informations de configuration du sous-système sont stockées dans le fichier de configuration unifiée ***EAP_HOME*/domain/configuration/domain.xml** pour un domaine géré ou ***EAP_HOME*/standalone/configuration/standalone.xml** pour un serveur autonome. La plupart des sous-systèmes incluent les détails de configuration configurés par l'intermédiaire de descripteurs de déploiements dans les versions précédentes de JBoss Enterprise Application Platform.

### Schémas de configuration du sous-système

Chaque configuration de sous-système est définie dans un schéma XML. Les schémas de configuration se trouvent dans le répertoire ***EAP_HOME*/docs/schema/** de votre installation.

Les sous-systèmes suivants sont connus comme *sous-systèmes simples*, parce qu'ils n'ont pas d'attributs ou éléments configurables. Ils sont généralement répertoriés en haut du fichier de configuration.

### Sous-systèmes simples

- **ee**– implémentation Java EE 6 API
- **ejb**– sous-système Enterprise JavaBeans (EJB)
- **jaxrs**– l'API JAX-RS API, fourni par RESTeasy.
- **sar**– le sous-système qui supporte Service Archives.
- **threads**– le sous-système qui supporte le traitement des threads.
- **weld**– l'API Contexts and Dependency Injection, fourni par Weld.

Report a bug

# CHAPITRE 13. LE SOUS-SYSTÈME DE JOURNALISATION

## 13.1. INTRODUCTION

### 13.1.1. Logging (Journalisation)

JBoss Enterprise Application Platform 6 fournit des fonctionnalités de journalisation hautement configurable pour son propre usage et pour utilisation par des applications déployées. Le sous-système d'enregistrement est basé sur JBoss LogManager et prend en charge plusieurs frameworks de journalisation d'applications de tierce partie en plus du JBoss Logging.

Le sous-système de journalisation est configuré à l'aide d'un système de catégories de journaux et des gestionnaires de journaux. Les catégories de journalisation définissent quels messages capturer et les gestionnaires de journaux définissent comment procéder avec ces messages (écriture sur disque, envoyer à la console, etc.).

Le Profils de journalisation est une fonctionnalité ajoutée dans la version 6.1.0 qui permet à des configurations de journalisation possédant un nom unique d'être créées et assignées à des applications indépendamment de toute autre configuration de journalisation. La configuration des profils de journalisation est presque identique pour le sous-système de journalisation principal.

Toutes ces configurations peuvent être effectuées dans la Console de gestion avec le CLI.

Report a bug

### 13.1.2. Frameworks de Logging (journalisation) d'applications pris en charge par JBoss LogManager

JBoss LogManager prend en charge les frameworks de journalisation suivants :

- JBoss Logging - inclus avec JBoss Enterprise Application Platform 6
- Apache Commons Logging - http://commons.apache.org/logging/
- Simple Logging Facade for Java (SLF4J) - http://www.slf4j.org/
- Apache log4j - http://logging.apache.org/log4j/1.2/
- Java SE Logging (java.util.logging) - http://download.oracle.com/javase/6/docs/api/java/util/logging/package-summary.html

Report a bug

### 13.1.3. Configuration du journal d'amorçage

La journalisation d'amorçage enregistre des événements qui ont lieu quand le serveur démarre (ou "est amorcé").

Le journal d'amorçage peut être configuré en modifiant le fichier **`logging.properties`**. Ce fichier est un fichier standard de propriétés Java qui peut être modifié à l'aide d'un éditeur de texte. Chaque ligne du fichier possède le format **`property=value`**.

Selon que vous exécutez Jboss Enterprise Application Platform dans un domaine géré ou dans un serveur autonome, votre fichier de configuration **`logging.properties`** se trouvera soit dans

***EAP_HOME*/domain/configuration/logging.properties** ou ***EAP_HOME*/standalone/configuration/logging.properties**.

Report a bug

### 13.1.4. Emplacements de Fichiers de journalisation par défaut

Il s'agit des fichiers de journalisation qui ont été créés pour les configurations de journalisation par défaut. La configuration par défaut écrit des fichiers de journalisation du serveur à l'aide de handlers de journaux périodiques.

**Tableau 13.1. Les Fichiers de journalisation par défaut d'un Serveur autonome**

| Fichier journal | Description |
|---|---|
| ***EAP_HOME*/standalone/log/boot.log** | Le journal d'amorçage du serveur. Contient les messages de journalisation liés au démarrage du serveur. |
| ***EAP_HOME*/standalone/log/server.log** | Le journal du serveur. Contient les messages de journalisation après que le serveur ait été lancé. |

**Tableau 13.2. Fichiers de journalisation par défaut d'un domaine géré**

| Fichier journal | Description |
|---|---|
| ***EAP_HOME*/domain/log/host-controller/boot.log** | Journal d'amorçage du contrôleur hôte. Contient les messages de journalisation liés au démarrage du contrôleur hôte. |
| ***EAP_HOME*/domain/log/process-controller/boot.log** | Journal d'amorçage du contrôleur de processus. Contient les messages de journalisation liés au démarrage du contrôleur de processus. |
| ***EAP_HOME*/domain/servers/*SERVERNAME*/log/boot.log** | Journal d'amorçage du serveur désigné. Contient les messages de journalisation liés au démarrage du serveur désigné. |
| ***EAP_HOME*/domain/servers/*SERVERNAME*/log/server.log** | Le journal du serveur désigné. Contient tous les messages de journalisation de ce serveur après que le serveur ait été lancé. |

Report a bug

### 13.1.5. A propos des Niveaux de journalisation

Les niveaux de journalisation sont des ensembles ordonnés de valeurs énumérées qui indiquent la nature et la sévérité d'un message de journalisation. Le niveau d'un message de journalisation donné est indiqué par le développeur par des méthodes qui conviennent dans un framework de journalisation particulier pour envoyer le message.

JBoss Enterprise Application Platform 6 accepte tous les niveaux de journalisation utilisés par les frameworks de journalisation de l'application prise en charge. Les six niveaux de journalisation les plus utilisés sont (dans l'ordre croissant) : **TRACE**, **DEBOG**, **INFO**, **ATTENTION**, **ERREUR** et **FATAL**.

Les niveaux de journal sont utilisés par des catégories et gestionnaires de journalisation pour limiter les messages dont ils sont responsables. Chaque niveau de journal possède une valeur numérique qui indique son ordre par rapport à d'autres niveaux de journal. Les catégories et gestionnaires de journalisation correspondent à un certain niveau de journal, et ils traitent les messages de journalisation du même niveau ou d'un niveau supérieur uniquement. Par exemple, un gestionnaire de journalisation du niveau **ATTENTION** enregistrera uniquement les messages des niveaux **ATTENTION**, **ERREUR** et **FATAL**.

Report a bug

### 13.1.6. Niveaux de journalisation pris en charge

**Tableau 13.3. Niveaux de journalisation pris en charge**

| Niveau de Journalisation | Valeur | Description |
|---|---|---|
| FINESSE MAX | 300 | - |
| PLUS FIN | 400 | - |
| TRACE | 400 | Utiliser pour des messages qui fournissent des informations détaillées sur l'état d'exécution d'une application. Les messages de journalisation **TRACE** sont habituellement seulement capturés lors du débogage d'une application. |
| DEBOG | 500 | Utiliser pour des messages qui indiquent des demandes individuelles de progrès ou des activités d'une application. Les messages de journalisation **DEBUG** sont habituellement seulement capturés lors du débogage d'une application. |
| FINESSE | 500 | - |
| CONFIG | 700 | - |
| INFO | 800 | Utiliser pour des messages qui indiquent la progression globale de l'application. Souvent utilisé pour le démarrage de l'application, la fermeture et autres événements majeurs de cycle de vie. |
| ATTENTION | 900 | Utiliser pour indiquer une situation qui n'est pas en erreur, mais n'est pas considéré comme idéale. Peut indiquer des circonstances qui peuvent entraîner des erreurs dans le futur. |
| ATTENTION | 900 | - |
| ERREUR | 1000 | Utiliser pour indiquer une erreur qui s'est produite et qui puisse empêcher l'activité actuelle ou la demande de se remplir, mais qui n'empêchera pas l'application d'exécuter. |
| SÉVÈRE | 1000 | - |

| Niveau de Journalisation | Valeur | Description |
|---|---|---|
| FATAL | 1100 | Utiliser pour indiquer les événements qui pourraient entraîner des défaillances de services critiques ou la fermeture de l'application, ou qui pourraient entraîner la fermeture de la plateforme JBoss Enterprise Application Platform 6. |

Report a bug

### 13.1.7. Catégories de journalisation

Les catégories de journalisation définissent les messages de journalisation à acquérir et un ou plusieurs gestionnaires de journalisation qui traitent les messages.

Les messages de journalisation à capturer sont définis par leur package Java d'origine et leurs niveau de journalisation. Les messages de classes de ce package et de niveau de journalisation ou niveau inférieur sont capturés par la catégorie de journalisation et envoyés aux gestionnaires de journal spécifiés.

Les catégories ont la possibilité d'utiliser les gestionnaires de journalisation du Root Logger au lieu de leurs propres gestionnaires.

Report a bug

### 13.1.8. Root Logger

Le Root Logger capture tous les messages de journalisation qui sont envoyés au serveur (à un niveau indiqué) et qui ne sont pas capturés par une catégorie de journalisation particulière. Ces messages sont alors envoyés à un ou à plusieurs gestionnaires de journalisation.

Par défaut, le Root Logger est configuré pour utiliser une console et un gestionnaire de journalisation périodique. Le gestionnaire de journalisation périodique est configuré pour écrire sur le ficher **`server.log`**. On prénomme parfois ce fichier: journal du serveur (server log).

Report a bug

### 13.1.9. Log Handlers

Les gestionnaires de journaux définissent la façon dont les messages de journalisation sont enregistrés dans JBoss Enterprise Application Platform. Il existe six types de gestionnaires de journalisation configurables : **`Console`**, **`File`**, **`Periodic`**, **`Size`**, **`Async`** et **`Custom`**.

Report a bug

### 13.1.10. Types de gestionnaires de journalisation

**Console**

Les gestionnaires de journaux de console écrivent des messages du journalisation soit dans le système d'exploitation hôte (stdout) ou dans le flux d'erreurs standard (stderr). Ces messages sont affichés lorsque JBoss Enterprise Application Platform 6 est exécuté à partir d'une invite de ligne de

commande. Les messages d'un gestionnaire de journal de Console ne sont pas enregistrés à moins que le système d'exploitation ne soit spécifiquement configuré pour capturer stdout ou stderr.

**Fichier**

Les gestionnaires de journaux de fichiers sont les gestionnaires de journalisation les plus simples, qui écrivent les messages de journalisation dans un fichier spécifique.

**Périodique**

Les gestionnaires de journaux périodiques écrivent des messages de journalisation dans un fichier nommé jusqu'à ce qu'une certaine durée se soit écoulée. Une fois que cette période a expiré, le fichier est nommé à nouveau en rajoutant l'horodatage et le gestionnaire continue d'écrire dans un fichier de journalisation nouvellement créé avec le nom d'origine.

**Taille**

Les gestionnaires de journaux de Taille écrivent les messages de journalisation dans un fichier jusqu'à ce que le fichier atteigne une taille spécifiée. Lorsque le fichier atteint une taille donnée, il est renommé avec un préfixe numérique et le gestionnaire continue d'écrire dans un fichier journal récemment créé avec le nom d'origine. Chaque gestionnaire de journaux de Taille doit spécifier le nombre maximal de fichiers contenus de cette façon.

**Async**

Les gestionnaires de journaux async sont des gestionnaires de journaux wrapper qui fournissent un comportement asynchrone pour un ou plusieurs autres gestionnaires de journaux. Ils sont utiles pour les gestionnaires de journaux qui pourraient avoir une latence élevée ou autres problèmes de performances comme l'écriture d'un fichier journal à un système de fichiers réseau.

**Personnalisé**

Les gestionnaires d'informations personnalisées vous permettent de configurer de nouveaux types de gestionnaires de journaux mis en place. Un gestionnaire personnalisé doit être implémenté comme classe Java qui s'étend **`java.util.logging.Handler`** et qui doit être contenue dans un module.

Report a bug

### 13.1.11. Log Formatters

Un formateur de journalisation (log formatter) est une propriété de configuration d'un gestionnaire de journalisation qui détermine l'apparence des messages de journalisation. Il s'agit d'un string qui utilise une syntaxe basée sur la classe **`java.util.Formatter`**.

Ainsi, le string du formateur de journalisation de la configuration par défaut, **`%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n`** , crée des messages de journalisation qui ressemblent à ceci :

```
15:53:26,546 INFO  [org.jboss.as] (Controller Boot Thread) JBAS015951:
Admin console listening on http://127.0.0.1:9990
```

Report a bug

### 13.1.12. Syntaxe de Formateur de journaux

**Tableau 13.4. Syntaxe de Formateur de journaux**

| Symbole | Description |
|---|---|
| **%c** | La catégorie de l'événement de journalisation |
| **%p** | Le niveau de saisie de la journalisation (info/déboggage/etc) |
| **%P** | Le niveau localisé de la saisie de journalisation |
| **%d** | Les date/heure (yyyy-MM-dd HH:mm:ss,SSS form) |
| **%r** | L'heure relative (en millisecondes depuis l'initialisation de la journalisation) |
| **%z** | Le réseau horaire |
| **%k** | Une clé de ressource de journalisation (utilisée pour la localisation de messages de journalisation) |
| **%m** | Le message de journalisation (avec trace d'exception) |
| **%s** | Le simple message de journalisation (sans trace d'exception) |
| **%e** | Exception stack trace (sans informations sur les modules étendus) |
| **%E** | Exception stack trace (avec informations sur les modules étendus) |
| **%t** | Le nom du thread en cours |
| **%n** | Un caractère de nouvelle ligne |
| **%C** | La classe du code appelant la méthode de journalisation (lente) |
| **%F** | Le nom de fichier de la classe appelant la méthode de journalisation (lente) |
| **%l** | L'emplacement d'origine du code appelant la méthode de journalisation (lente) |
| **%L** | Le numéro de ligne du code appelant la méthode de journalisation (lente) |
| **%M** | La méthode du code appelant la méthode de journalisation (lente) |
| **%x** | Log4J Nested Diagnostic Context |
| **%X** | Log4J Message Diagnostic Context |
| **%%** | Un pourcentage (caractère d'échappement) |

Report a bug

## 13.2. CONFIGURER LA JOURNALISATION PAR LA CONSOLE DE GESTION

La console de gestion fournit une interface graphique utilisateur pour la configuration du root logger, des log handlers et des catégories de journalisation.Vous pourrez trouver la configuration du logger dans la console de gestion en suivant les étapes suivantes :

Vous pourrez accéder à cette configuration en suivant les étapes suivantes :

1. Connectez-vous à la console de gestion
2. Naviguez dans la configuration du sous-système de logging. Cette étape varie suivant qu'il s'agisse de serveurs qui s'exécutent sur les serveurs autonomes ou des serveurs exécutant dans un domaine géré.
    - **Serveur autonome**
      Cliquer sur Profil, étendre Core dans la panneau Profile, puis cliquer sur Logging.
    - **Domaine géré**
      Cliquer sur Profil, sélectionner le Profil à modifier, étendre Core, puis cliquer sur Logging.

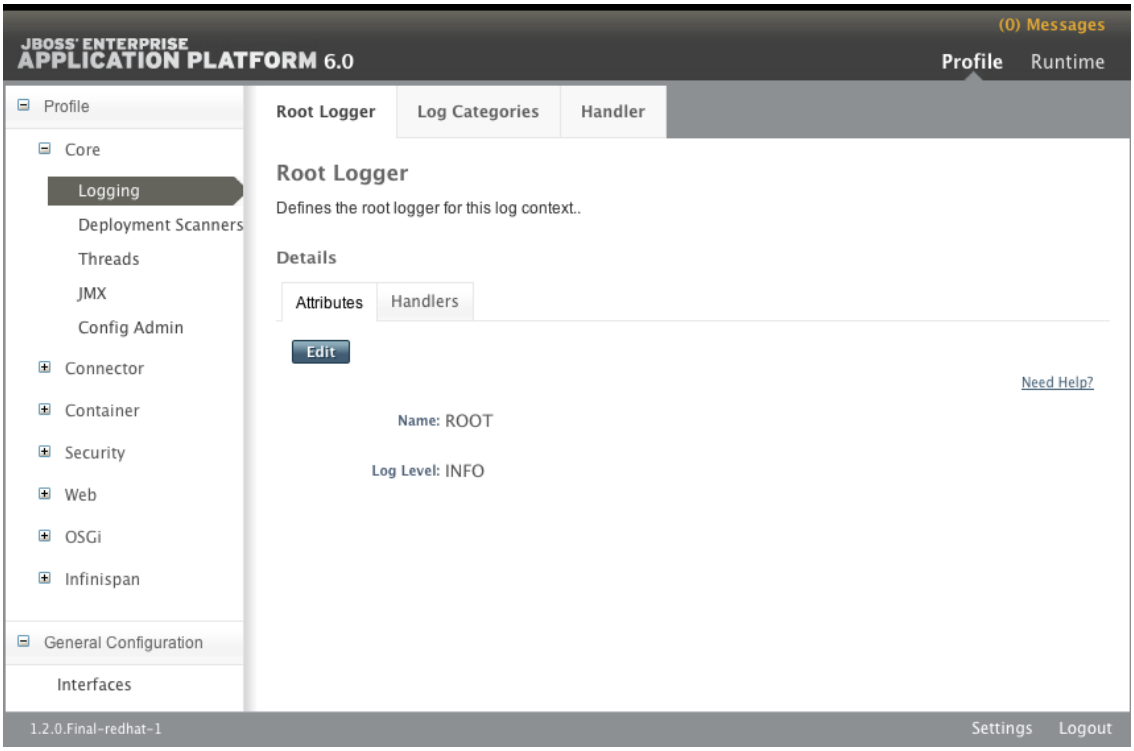



**Figure 13.1. Configuration Logging avec la console de gestion**

Les tâches à effectuer pour configurer le root logger sont les suivantes :

- Modifier le niveau de journalisation.
- Ajouter et supprimer des log handlers.

Les tâches à effectuer pour configurer les catégories de journalisation sont les suivantes :

- Ajouter et supprimer les catégories de journalisation.
- Modifier les propriétés de catégories.

- Ajouter et supprimer les log handlers d'une catégorie.

Les tâches principales à effectuer pour configurer les log handlers sont les suivantes :

- Ajouter de nouveaux handlers.
- Configurer les handlers.

Les six log handlers (y compris le handler personnalisé) peuvent être configurés dans la console de gestion.

Report a bug

## 13.3. CONFIGURATION DE LOGGING DANS LE CLI

### 13.3.1. Configurer le Root Logger par le CLI

La configuration du Root Logger peut s'afficher ou être modifiée par le CLI.

Les tâches principales à effectuer pour configurer le Root Logger sont les suivantes :

- Ajouter des Log Handler au Root Logger.
- Afficher la configuration du Root Logger.
- Modifier le niveau de journalisation.
- Supprimer des Log Handler du Root Logger.

**IMPORTANT**

Pour la configuration du Root Logger dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

**Ajouter un Log Handler au Root Logger**

Utiliser l'opération **`root-logger-assign-handler`** avec la syntaxe suivante *HANDLER* dans le nom du gestionnaire de journalisation (Log Hangler) à ajouter.

```
 /subsystem=logging/root-logger=ROOT:root-logger-assign-
handler(name="HANDLER")
```

Le gestionnaire de journalisation doit avoir déjà été créé avant qu'il puisse être ajouté au Root Logger.

**Exemple 13.1. Root Logger root-logger-assign-handler operation**

```
[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:root-logger-assign-handler(name="AccountsNFSAsync")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Afficher le contenu de la configuration du Root Logger**

Utiliser l'opération **`read-resource`** avec la syntaxe suivante.

```
/subsystem=logging/root-logger=ROOT:read-resource
```

**Exemple 13.2. Opération Root Logger «read-resource»**

```
[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:read-resource
{
   "outcome" => "success",
   "result" => {
      "filter" => {"match" => "names"},
      "handlers" => [
            "CONSOLE",
            "FILE"
      ],
      "level" => "INFO"
   }
}
```

**Définir le niveau de journalisation du Root Logger**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante avec *LEVEL* indiquant les niveaux de journalisation pris en charge.

```
/subsystem=logging/root-logger=ROOT:write-attribute(name="level",
value="LEVEL")
```

**Exemple 13.3. L'opération «write-attribute» du Root Logger pour définir le niveau de journalisation**

```
[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:write-attribute(name="level", value="DEBUG")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Supprimer un Log Handler du Root Logger**

Utiliser **`root-logger-unassign-handler`** avec la syntaxe suivante, avec *HANDLER* comme nom du gestionnaire de journalisation à supprimer.

```
/subsystem=logging/root-logger=ROOT:root-logger-unassign-
handler(name="HANDLER")
```

**Exemple 13.4. Supprimer un Log Handler**

```
[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:root-logger-unassign-handler(name="AccountsNFSAsync")
{"outcome" => "success"}
```

```
[standalone@localhost:9999 /]
```

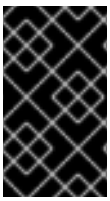
Report a bug

### 13.3.2. Configurer une Catégorie dans l'interface CLI

Les catégories de journalisation peuvent être ajoutées, supprimées et modifiées dans le CLI.

Les tâches principales à effectuer pour configurer les catégories de journalisation sont les suivantes :

- Ajouter une nouvelle catégorie de journalisation :
- Afficher la configuration d'une catégorie de journalisation.
- Définir un niveau de journalisation.
- Ajouter des Log Handlers à une catégorie de journalisation.
- Supprimer des Log Handlers d'une catégorie de journalisation.
- Supprimer une catégorie de journalisation.



**IMPORTANT**

Pour la configuration d'une catégorie de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

**Ajouter une catégorie de journalisation**

Utiliser l'opération **add** avec la syntaxe suivante. Remplacer *CATEGORY* par la catégorie à ajouter.

```
/subsystem=logging/logger=CATEGORY:add
```

**Exemple 13.5. Ajouter une nouvelle catégorie de journalisation**

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:add
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Afficher une configuration de catégorie de journalisation**

Utiliser l'opération **`read-resource`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie.

```
/subsystem=logging/logger=CATEGORY:read-resource
```

**Exemple 13.6. Opération de lecture de ressource de la catégorie de journalisation**

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=org.apache.tomcat.util.modeler:read-
resource
{
    "outcome" => "success",
    "result" => {
        "filter" => undefined,
        "handlers" => undefined,
        "level" => "WARN",
        "use-parent-handlers" => true
    }
}
[standalone@localhost:9999 /]
```

**Définir le niveau de journalisation.**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie de journalisation et *LEVEL* par le niveau de journalisation à définir.

```
 /subsystem=logging/logger=CATEGORY:write-attribute(name="level",
value="LEVEL")
```

**Exemple 13.7. Définir un niveau de journalisation**

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:write-
attribute(name="level", value="DEBUG")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir la Catégorie de journalisation pour utiliser le Log Handler du Root Logger.**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie de journalisation. Remplacer *BOOLEAN* par true pour cette catégorie de journalisation pour utiliser les handlers du Root Logger. Remplacer le par false s'il doit utiliser ses propres handlers.

```
 /subsystem=logging/logger=CATEGORY:write-attribute(name="use-parent-
handlers", value="BOOLEAN")
```

**Exemple 13.8. Configurer use-parent-handlers**

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:write-
attribute(name="use-parent-handlers", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Ajouter un Log Handler à une catégorie de journalisation.**

Utiliser l'opération **`assign-handler`** avec la syntaxe suivante. Remplacer *CATEGORY* par la catégorie à ajouter et *HANDLER* par le nom du handler à ajouter.

```
/subsystem=logging/logger=CATEGORY:assign-handler(name="HANDLER")
```

Le Log Handler doit déjà avoir été créé avant de l'ajouter au Root Handler.

**Exemple 13.9. Ajouter un Log Handler**

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:assign-
handler(name="AccountsNFSAsync")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

#### Supprimer un Log Handler d'une catégorie de journalisation

Utiliser l'opération **`unassign-handler`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie à *HANDLER* par le nom du Log Handler à supprimer.

```
/subsystem=logging/logger=CATEGORY:unassign-handler(name="HANDLER")
```

**Exemple 13.10. Supprimer un Log Handler**

```
[standalone@localhost:9999 /] /subsystem=logging/root-
logger=ROOT:root-logger-unassign-handler(name="AccountsNFSAsync")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

#### Supprimer une catégorie

Utiliser l'opération **`remove`** avec la syntaxe suivante. Remplacer *CATEGORY* par le nom de la catégorie à supprimer.

```
/subsystem=logging/logger=CATEGORY:remove
```

**Exemple 13.11. Supprimer une catégorie de journalisation**

```
[standalone@localhost:9999 /]
/subsystem=logging/logger=com.company.accounts.rec:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

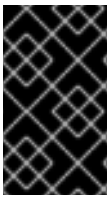
Report a bug

### 13.3.3. Configurer un Log Handler de console dans le CLI

Les Log Handlers de console peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un Log Handler de console :

- Ajouter un nouveau Log Handler de console.
- Afficher la configuration d'un Log Handler de console.
- Définir le niveau de journalisation du handler.
- Définir la cible de la sortie du handler.
- Définir la codification utilisée pour la sortie du handler.
- Définir le formateur utilisé pour la sortie du handler.
- Définir si le handler utilise autoflush ou non.
- Supprimer un handler de journalisation de console.



**IMPORTANT**

Pour la configuration d'un handler de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

**Ajouter un Log Handler de console**

Utiliser l'opération **add** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du gestionnaire de journalisation (Log Hangler) de la console à ajouter.

```
/subsystem=logging/console-handler=HANDLER:add
```

**Exemple 13.12. Ajouter un Log Handler de console**

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:add
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Afficher une configuration de Log Handler de journalisation de la console**

Utiliser l'opération **`read-resource`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de la console.

```
/subsystem=logging/console-handler=HANDLER:read-resource
```

**Exemple 13.13. Afficher une configuration de Log Handler de journalisation de la console**

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=CONSOLE:read-resource
{
    "outcome" => "success",
    "result" => {
        "autoflush" => true,
```

```
        "encoding" => undefined,
        "filter" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => "INFO",
        "target" => "System.out"
    }
}
[standalone@localhost:9999 /]
```

**Définir le Niveau de journalisation**

Utiliser l'opération **`change-log-level`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de la console et *LEVEL* par le niveau de journalisation à définir.

```
 /subsystem=logging/console-handler=HANDLER:change-log-
level(level="LEVEL")
```

**Exemple 13.14. Définir le Niveau de journalisation**

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:change-log-level(level="TRACE")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir la cible**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de la console et *TARGET* par **`System.err`** ou **`System.out`** pour le flux Erreurs système ou le flux Standard out.

```
 /subsystem=logging/console-handler=HANDLER:write-
attribute(name="target", value="TARGET")
```

**Exemple 13.15. Définir la cible**

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="target",
value="System.err")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir le Codage**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de la console et *ENCODING* par le nom de codification du caractère qui convient.

```
 /subsystem=logging/console-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

**Exemple 13.16. Définir le Codage**

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

#### Définir Formateur

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de la console et *FORMAT* par le string de formateur requis.

```
 /subsystem=logging/console-handler=HANDLER:write-
attribute(name="formatter", value="FORMAT")
```

**Exemple 13.17. Définir Formateur**

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

#### Définir Auto Flush

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de la console et *BOOLEAN* par **`true`** si le handler doit écrite sa sortie immédiatement.

```
 /subsystem=logging/console-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

**Exemple 13.18. Définir Auto Flush**

```
[standalone@localhost:9999 /] /subsystem=logging/console-
handler=ERRORCONSOLE:write-attribute(name="autoflush", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

#### Supprimer un Log Handler de console

Utiliser l'opération **`remove`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de la console à supprimer.

```
 /subsystem=logging/console-handler=HANDLER:remove
```

**Exemple 13.19. Supprimer un Log Handler de console**

```
[standalone@localhost:9999 /] /subsystem=logging/console-
```

```
handler=ERRORCONSOLE:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Report a bug

### 13.3.4. Configurer un Log Handler de fichiers dans le CLI

Les Log Handlers de fichiers peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un Log Handler de fichiers :

- Ajouter un nouveau Log Handler de fichiers.
- Afficher la configuration d'un Log Handler de fichiers
- Définir le niveau de journalisation du handler.
- Définir le comportement d'ajout du handler.
- Définir si le handler utilise autoflush ou non.
- Définir la codification utilisée pour la sortie du handler.
- Indiquer le fichier dans lequel le Log Handler écrit.
- Définir le formateur utilisé pour la sortie du handler.
- Supprimer un Log Handler de fichiers.

**IMPORTANT**

Pour la configuration d'un handler de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

**Ajouter un Log Handler de fichiers**

Utiliser l'opération **add** avec la syntaxe suivante. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

```
 /subsystem=logging/file-handler=HANDLER:add(file={"path"=>"PATH",
"relative-to"=>"DIR"})
```

**Exemple 13.20. Ajouter un Log Handler de fichiers**

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:add(file={"path"=>"accounts.log", "relative-
to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Afficher une configuration de Log Handler de fichiers**

Utiliser l'opération **`read-resource`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de fichiers.

```
/subsystem=logging/file-handler=HANDLER:read-resource
```

**Exemple 13.21. Utilisation de l'opération read-resource**

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:read-resource
{
    "outcome" => "success",
    "result" => {
        "append" => true,
        "autoflush" => true,
        "encoding" => undefined,
        "file" => {
            "path" => "accounts.log",
            "relative-to" => "jboss.server.log.dir"
        },
        "filter" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => undefined
    }
}
[standalone@localhost:9999 /]
```

**Définir Niveau de journalisation**

Utiliser l'opération **`change-log-level`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de fichiers et *LEVEL* par le niveau de journalisation à définir.

```
/subsystem=logging/file-handler=HANDLER:change-log-level(level="LEVEL")
```

**Exemple 13.22. Changer le niveau de journalisation**

```
/subsystem=logging/file-handler=accounts_log:change-log-
level(level="DEBUG")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir le comportement d'ajout**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de fichiers. Remplacer *BOOLÉEN* par false si vous souhaitez qu'un nouveau fichier de journalisation soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer *BOOLÉEN* par **`true`** si le serveur d'applications doit continuer à utiliser le même fichier.

```
 /subsystem=logging/file-handler=HANDLER:write-attribute(name="append",
value="BOOLEAN")
```

**Exemple 13.23. Changer la propriété d'ajout**

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="append", value="true")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /]
```

JBoss Enterprise Application Platform 6 doit démarrer à nouveau pour prendre effet.

#### Définir Auto Flush

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de fichiers et *BOOLEAN* par **`true`** si le handler doit écrite sa sortie immédiatement.

```
 /subsystem=logging/file-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

**Exemple 13.24. Changer la propriété autoflush**

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="autoflush", value="false")
{
    "outcome" => "success",
    "response-headers" => {"process-state" => "reload-required"}
}
[standalone@localhost:9999 /]
```

JBoss Enterprise Application Platform 6 doit démarrer à nouveau pour prendre effet.

#### Définir le Codage

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de fichiers et *ENCODING* par le nom de codification du caractère qui convient.

```
 /subsystem=logging/file-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

**Exemple 13.25. Définir le Codage**

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}
```

```
[standalone@localhost:9999 /]
```

**Changer le fichier dans lequel le Log Handler écrit.**

Utiliser l'opération **`change-file`** avec la syntaxe suivante. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

```
 /subsystem=logging/file-handler=HANDLER:change-file(file=
{"path"=>"PATH", "relative-to"=>"DIR"})
```

**Exemple 13.26. Changer le fichier dans lequel le Log Handler écrit.**

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:change-file(file={"path"=>"accounts-debug.log",
"relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir Formateur**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de fichiers et *FORMAT* par le string de formateur requis.

```
 /subsystem=logging/file-handler=HANDLER:write-
attribute(name="formatter", value="FORMAT")
```

**Exemple 13.27. Définir Formateur**

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts-log:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Supprimer un Log Handler de fichiers.**

Utiliser l'opération **`remove`** avec la syntaxe suivante. Remplacer *HANDLER* par le nom du Log Handler de fichiers à supprimer.

```
 /subsystem=logging/file-handler=HANDLER:remove
```

**Exemple 13.28. Supprimer un Log Handler de fichiers.**

```
[standalone@localhost:9999 /] /subsystem=logging/file-
handler=accounts_log:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Un Log Handler ne peut être supprimé que s'il ne peut pas être référencé par une catégorie de journalisation ou par un Log Handler async.

Report a bug

### 13.3.5. Configurer un Log Handler périodique dans le CLI

Les Log Handlers périodiques peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches principales qui vous reviendront pour configurer un Log Handler périodique :

- Ajouter un nouveau Log Handler périodique.
- Afficher la configuration d'un Log Handler périodique
- Définir le niveau de journalisation du handler.
- Définir le comportement d'ajout du handler.
- Définir si le handler utilise autoflush ou non.
- Définir la codification utilisée pour la sortie du handler.
- Indiquer le fichier dans lequel le Log Handler écrit.
- Définir le formateur utilisé pour la sortie du handler.
- Définir le suffixe pour les journaux en rotation
- Supprimer un Log Handler périodique

Chacune de ces actions sont décrites ci-dessous.

**IMPORTANT**

Pour la configuration d'un handler de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

**Ajouter un nouveau Log Handler périodique en rotation.**

Utiliser l'opération **add** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:add(file=
{"path"=>"PATH", "relative-to"=>"DIR"}, suffix="SUFFIX")
```

Remplacer *HANDLER* par le nom du Log Handler. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin. Remplacer *SUFFIX* par le suffixe de rotation de fichiers à utiliser.

**Exemple 13.29. Créer un nouvel handler**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:add(file={"path"=>"daily-debug.log",
```

```
"relative-to"=>"jboss.server.log.dir"}, suffix=".yyyy.MM.dd")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Afficher une configuration de Log Handler de fichiers en rotation périodique**

Utiliser l'opération **`read-resource`** avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:read-resource
```

Remplacer *HANDLER* par le nom du Log Handler.

**Exemple 13.30. Utiliser l'opération read-resource**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:read-resource
{
    "outcome" => "success",
    "result" => {
        "append" => true,
        "autoflush" => true,
        "encoding" => undefined,
        "file" => {
            "path" => "daily-debug.log",
            "relative-to" => "jboss.server.log.dir"
        },
        "filter" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => undefined
    }
}
[standalone@localhost:9999 /]
```

**Définir Niveau de journalisation**

Utiliser l'opération **`change-log-level`** avec la syntaxe suivante.

```
/subsystem=logging/periodic-rotating-file-handler=HANDLER:change-log-
level(level="LEVEL")
```

Remplacer *HANDLER* par le nom du Log Handler périodique et *LEVEL* par le niveau de journalisation à définir.

**Exemple 13.31. Définir le niveau de journalisation**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:change-log-level(level="DEBUG")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir le comportement d'ajout**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-handler=HANDLER:write-
attribute(name="append", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du Log Handler périodique. Remplacer *BOOLÉEN* par **`false`** si vous souhaitez qu'un nouveau fichier de journalisation soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer *BOOLÉEN* par **`true`** si le serveur d'applications doit continuer à utiliser le même fichier.

JBoss Enterprise Application Platform 6 doit démarrer à nouveau pour prendre effet.

**Exemple 13.32. Définir le comportement d'ajout**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="append", value="true")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /]
```

**Définir Auto Flush**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du Log Handler périodique et *BOOLEAN* par **`true`** si le handler doit écrite sa sortie immédiatement.

JBoss Enterprise Application Platform 6 doit démarrer à nouveau pour prendre effet.

**Exemple 13.33. Définir le comportement Auto Flush**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="autoflush",
value="false")
{
    "outcome" => "success",
    "response-headers" => {"process-state" => "reload-required"}
}
[standalone@localhost:9999 /]
```

**Définir le Codage**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

Remplacer *HANDLER* par le nom du Log Handler périodique et *ENCODING* par le nom de codification du caractère qui convient.

**Exemple 13.34. Définir le Codage**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="encoding", value="utf-
8")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Changer le fichier dans lequel le Log Handler écrit.**

Utiliser l'opération **`change-file`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:change-
file(file={"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *HANDLER* par le nom du Log Handler périodique. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

**Exemple 13.35. Changer le fichier dans lequel le Log Handler écrit.**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:change-file(file={"path"=>"daily-
debug.log", "relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir Formateur**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="formatter", value="FORMAT")
```

Remplacer *HANDLER* par le nom du Log Handler périodique et *FORMAT* par le string de formateur à définir.

**Exemple 13.36. Définir Formateur**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n")
```

```
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir le suffixe pour les journaux en rotation**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:write-
attribute(name="suffix", value="SUFFIX")
```

Remplacer *HANDLER* par le nom du Log Handler et *SUFFIX* par le string de suffixe à définir.

**Exemple 13.37.**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:write-attribute(name="suffix", value=".yyyy-
MM-dd-HH")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Supprimer un Log Handler périodique**

Utiliser l'opération **`remove`** avec la syntaxe suivante.

```
 /subsystem=logging/periodic-rotating-file-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du Log Handler périodique.

**Exemple 13.38. Supprimer un Log Handler périodique**

```
[standalone@localhost:9999 /] /subsystem=logging/periodic-rotating-
file-handler=HOURLY_DEBUG:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Report a bug

### 13.3.6. Configurer un Log Handler Taille dans le CLI

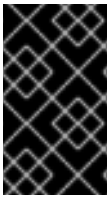
Les Log Handlers Taille de fichiers en rotation peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Voici les tâches qui vous reviendront pour configurer un Log Handler Taille de fichiers en rotations :

- Ajouter un nouveau Log Handler
- Afficher la configuration du Log Handler
- Définir le niveau de journalisation du handler.

- Définir le comportement d'ajout du handler.
- Définir si le handler utilise autoflush ou non.
- Définir la codification utilisée pour la sortie du handler.
- Indiquer le fichier dans lequel le Log Handler écrit.
- Définir le formateur utilisé pour la sortie du handler.
- Définir la taille maximum de chaque fichier de journalisation
- Définir le nombre maximum de journaux de sauvegarde à conserver
- Supprimer un Log Handler.

Chacune de ces actions est décrite ci-dessous.



**IMPORTANT**

Pour la configuration d'un handler de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

**Ajouter un nouveau Log Handler**

Utiliser l'opération **`add`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:add(file=
{"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *HANDLER* par le nom du Log Handler. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

**Exemple 13.39. Ajouter un nouveau Log Handler**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:add(file={"path"=>"accounts_trace.log",
"relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Afficher la configuration du Log Handler**

Utiliser l'opération **`read-resource`** avec la syntaxe suivante.

```
/subsystem=logging/size-rotating-file-handler=HANDLER:read-resource
```

Remplacer *HANDLER* par le nom du Log Handler.

**Exemple 13.40. Afficher la configuration du Log Handler**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:read-resource
{
    "outcome" => "success",
    "result" => {
        "append" => true,
        "autoflush" => true,
        "encoding" => undefined,
        "file" => {
            "path" => "accounts_trace.log",
            "relative-to" => "jboss.server.log.dir"
        },
        "filter" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => undefined,
        "max-backup-index" => 1,
        "rotate-size" => "2m"
    }
}
[standalone@localhost:9999 /]
```

**Définir le niveau de journalisation. du handler**

Utiliser l'opération **`change-log-level`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:change-log-
level(level="LEVEL")
```

Remplacer *HANDLER* par le nom du Log Handler et *LEVEL* par le niveau de journalisation à définir.

**Exemple 13.41. Définir le niveau de journalisation. du handler**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:change-log-level(level="TRACE")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir le comportement d'ajout du handler**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="append", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du Log Handler. Remplacer *BOOLÉEN* par **`false`** si vous souhaitez qu'un nouveau fichier de journalisation soit créé à chaque fois qu'un serveur d'applications est lancé. Remplacer *BOOLÉEN* par **`true`** si le serveur d'applications doit continuer à utiliser le même fichier.

JBoss Enterprise Application Platform 6 doit démarrer à nouveau pour prendre effet.

**Exemple 13.42. Définir le comportement d'ajout du handler**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="append", value="true")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /]
```

#### Définir si le handler utilise autoflush ou non

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="autoflush", value="BOOLEAN")
```

Remplacer *HANDLER* par le nom du Log Handler et *BOOLEAN* par **`true`** si le handler doit écrite sa sortie immédiatement.

**Exemple 13.43. Définir si le handler utilise autoflush ou non**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="autoflush", value="true")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

#### Définir la codification utilisée pour la sortie du handler

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="encoding", value="ENCODING")
```

Remplacer *HANDLER* par le nom du Log Handler et *ENCODING* par le nom de codification du caractère qui convient.

**Exemple 13.44. Définir la codification utilisée pour la sortie du handler**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="encoding", value="utf-8")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

#### Indiquer le fichier dans lequel le Log Handler écrit

Utiliser l'opération **`change-file`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:change-file(file=
{"path"=>"PATH", "relative-to"=>"DIR"})
```

Remplacer *HANDLER* par le nom du Log Handler. Remplacer *PATH* par le nom du fichier dans lequel le log est écrit. Remplacer *DIR* par le nom du répertoire dans lequel le fichier se trouve. La valeur *DIR* peut correspondre à une variable de chemin.

**Exemple 13.45. Indiquer le fichier dans lequel le Log Handler écrit**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:change-file(file=
{"path"=>"accounts_trace.log", "relative-
to"=>"jboss.server.log.dir"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir le formateur utilisé pour la sortie du handler.**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="formatter", value="FORMATTER")
```

Remplacer *HANDLER* par le nom du Log Handler et *FORMAT* par le string de formateur à définir.

**Exemple 13.46. Définir le formateur utilisé pour la sortie du handler.**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="formatter",
value="%d{HH:mm:ss,SSS} %-5p (%c) [%t] %s%E%n")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir la taille maximum de chaque fichier de journalisation**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="rotate-size", value="SIZE")
```

Remplacer *HANDLER* par le nom du Log Handler et *SIZE* par la taille de fichier maximum.

**Exemple 13.47. Définir la taille maximum de chaque fichier de journalisation**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="rotate-size",
value="50m")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir le nombre maximum de journaux de sauvegarde à conserver**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:write-
attribute(name="max-backup-index", value="NUMBER")
```

Remplacer *HANDLER* par le nom du Log Handler et *NUMBER* par le nombre de fichiers de journalisation à conserver.

**Exemple 13.48. Définir le nombre maximum de journaux de sauvegarde à conserver**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:write-attribute(name="max-backup-index",
value="5")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Supprimer un Log Handler**

Utiliser l'opération **`remove`** avec la syntaxe suivante.

```
 /subsystem=logging/size-rotating-file-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du Log Handler.

**Exemple 13.49. Supprimer un Log Handler**

```
[standalone@localhost:9999 /] /subsystem=logging/size-rotating-file-
handler=ACCOUNTS_TRACE:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Report a bug

### 13.3.7. Configurer un Log Handler Async dans le CLI

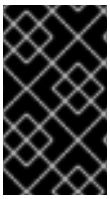
Les Log Handlers async peuvent être ajoutés, supprimés ou modifiés dans le CLI.

Les tâches que vous allez effectuer pour configurer un Log Handler async sont les suivantes :

- Ajouter un nouveau Log Handler async
- Afficher la configuration d'un Log Handler async
- Modifier le niveau de journalisation
- Définir la longueur de la file d'attente
- Définir l'action de dépassement

- Ajouter les sous-handlers
- Supprimer les sous-handlers
- Supprimer un sous-handler async

Chacune de ces tâches sont décrites ci-dessous.



**IMPORTANT**

Pour la configuration d'un handler de journalisation dans un profil de journalisation, la racine (root) du chemin de configuration est **`/subsystem=logging/logging-profile=NAME/`** au lieu de **`/subsystem=logging/`**.

**Ajouter un nouveau Log Handler async**

Utiliser l'opération **`add`** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:add(queue-length="LENGTH")
```

Remplacer *HANDLER* par le nom du Log Handler et *LENGTH* par la valeur du nombre maximum de requêtes de journalisation pouvant tenir dans une file d'attente.

**Exemple 13.50.**

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:add(queue-length="10")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Afficher la configuration d'un Log Handler async**

Utiliser l'opération **`read-resource`** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:read-resource
```

Remplacer *HANDLER* par le nom du Log Handler.

**Exemple 13.51.**

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:read-resource
{
    "outcome" => "success",
    "result" => {
        "encoding" => undefined,
        "filter" => undefined,
        "formatter" => "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n",
        "level" => undefined,
        "overflow-action" => "BLOCK",
        "queue-length" => "50",
        "subhandlers" => undefined
```

```
    }
}
[standalone@localhost:9999 /]
```

**Modifier le niveau de journalisation**

Utiliser l'opération **`change-log-level`** avec la syntaxe suivante.

```
 /subsystem=logging/async-handler=HANDLER:change-log-
level(level="LEVEL")
```

Remplacer *HANDLER* par le nom du Log Handler et *LEVEL* par le niveau de journalisation à définir.

**Exemple 13.52.**

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:change-log-level(level="INFO")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Définir la longueur de la file d'attente**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/async-handler=HANDLER:write-attribute(name="queue-
length", value="LENGTH")
```

Remplacer *HANDLER* par le nom du Log Handler et *LENGTH* par la valeur du nombre maximum de requêtes de journalisation pouvant tenir dans une file d'attente.

JBoss Enterprise Application Platform 6 doit démarrer à nouveau pour prendre effet.

**Exemple 13.53.**

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:write-attribute(name="queue-length", value="150")
{
    "outcome" => "success",
    "response-headers" => {
        "operation-requires-reload" => true,
        "process-state" => "reload-required"
    }
}
[standalone@localhost:9999 /]
```

**Définir l'action de dépassement**

Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
 /subsystem=logging/async-handler=HANDLER:write-
attribute(name="overflow-action", value="ACTION")
```

Remplacer *HANDLER* par le nom du Log Handler et *ACTION* par *DISCARD* ou *BLOCK*.

**Exemple 13.54.**

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:write-attribute(name="overflow-action",
value="DISCARD")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Ajouter les sous-handlers**

Utiliser l'opération **`assign-subhandler`** avec la syntaxe suivante.

```
 /subsystem=logging/async-handler=HANDLER:assign-
subhandler(name="SUBHANDLER")
```

Remplacer *HANDLER* par le nom du Log Handler et *SUBHANDLER* par le nom du Log Handler qui doit être ajouté comme sous-handler de ce handler asynch.

**Exemple 13.55.**

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:assign-subhandler(name="NFS_FILE")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Supprimer les sous-handlers**

Utiliser l'opération **`unassign-subhandler`** avec la syntaxe suivante.

```
 /subsystem=logging/async-handler=HANDLER:unassign-
subhandler(name="SUBHANDLER")
```

Remplacer *HANDLER* par le nom du Log Handler et *SUBHANDLER* par le nom du Log Handler qui doit être supprimé.

**Exemple 13.56.**

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:unassign-subhandler(name="NFS_FILE")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Supprimer un sous-handler async**

Utiliser l'opération **`remove`** avec la syntaxe suivante.

```
/subsystem=logging/async-handler=HANDLER:remove
```

Remplacer *HANDLER* par le nom du Log Handler.

**Exemple 13.57.**

```
[standalone@localhost:9999 /] /subsystem=logging/async-
handler=NFS_LOGS:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Report a bug

## 13.4. PROFILS DE JOURNALISATION

### 13.4.1. Profils de journalisation

**IMPORTANT**

Les Profils de journalisation ne sont disponibles qu'en version 6.1.0 ou supérieure

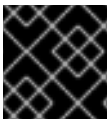
Les Profils de journalisation sont des ensembles de configuration de journalisation indépendants qui peuvent être assignés à des applications déployées. Un profil de journalisation peut définir les handlers, les catégories et un root logger à la manière du sous-système de journalisation standard, mais ne peut par recommander la configuration à d'autres profils ou au sous-système de journalisation principal. La conception des profils de journalisation émule le sous-système de journalisation au niveau de l'aisance de configuration.

Les profils de journalisation permettent aux administrateurs de créer une configuration de journalisation spécifique à une ou à plusieurs applications sans affecter une autre configuration de journalisation. Comme chaque profil est défini dans une configuration serveur, cela implique que la configuration de journalisation peut être modifiée sans exiger que les applications affectées ne soient re-déployées.

Chaque profil de journalisation peut avoir la configuration suivante :

- Un nom unique. Ceci est requis.
- N'importe quel nombre de Log Handlers.
- N'importe quelle catégorie log.
- Un root logger maximum.

Une application peut spécifier un profil de journalisation à utiliser dans son fichier MANIFEST.MF, en utilisant l'attribut Logging-profile.



**IMPORTANT**

Les profils de journalisation ne peuvent pas être configurés par la console de gestion.

Report a bug

### 13.4.2. Créer un nouveau Profil de journalisation par le CLI

On peut créer un nouveau profil de journalisation par la commande CLI suivante, en remplaçant *NAME* par le nom de profil qui convient :

```
/subsystem=logging/logging-profile=NAME:add
```

Cela va créer un nouveau profil vide auquel les handlers, catégories et root logger peuvent être ajoutés.

Report a bug

### 13.4.3. Créer un Profil de journalisation par le CLI

Un profil de journalisation peut être configuré par les log handlers, catégories et root logger en utilisant exactement la même syntaxe que lorsque l'on utilise le sous-système de journalisation principal.

Il existe uniquement deux différences entre configurer le sous-système de journalisation principal et le profil de journalisation :

1. Le chemin de configuration root est **`/subsystem=logging/logging-profile=NAME`**
2. Un profil de journalisation ne peut pas contenir d'autres profils de journalisation.

Référez vous à la tâche de gestion de journalisation qui convient :

- Section 13.3.1, « Configurer le Root Logger par le CLI »
- Section 13.3.2, « Configurer une Catégorie dans l'interface CLI »
- Section 13.3.3, « Configurer un Log Handler de console dans le CLI »
- Section 13.3.4, « Configurer un Log Handler de fichiers dans le CLI »
- Section 13.3.5, « Configurer un Log Handler périodique dans le CLI »
- Section 13.3.6, « Configurer un Log Handler Taille dans le CLI »
- Section 13.3.7, « Configurer un Log Handler Async dans le CLI »

**Exemple 13.58. Créer et Configurer un Profil de journalisation**

Créer un profil de journalisation et ajouter une catégorie et un Log Handler de fichiers.

1. Créer le profil :

```
/subsystem=logging/logging-profile=accounts-app-profile:add
```

2. Créer gestionnaire de fichiers

```
/subsystem=logging/logging-profile=accounts-app-profile/file-
handler=ejb-trace-file:add(file={path=>"ejb-trace.log", "relative-
to"=>"jboss.server.log.dir"})
```

```
/subsystem=logging/logging-profile=accounts-app-profile/file-
handler=ejb-trace-file:change-log-level(level="DEBUG")
```

3. Créer une catégorie de logger

```
/subsystem=logging/logging-profile=accounts-app-
profile/logger=com.company.accounts.ejbs:add(level=TRACE)
```

4. Assigner un gestionnaire de fichiers à une catégorie

```
/subsystem=logging/logging-profile=accounts-app-
profile/logger=com.company.accounts.ejbs:assign-handler(name="ejb-
trace-file")
```

Report a bug

### 13.4.4. Spécifier un Profil de journalisation dans une application

Une application spécifie le profil de journalisation à utiliser dans son fichier **`MANIFEST.MF`**.

**Pré-requis :**

1. Vous devez connaître le nom du profil de journalisation qui a été défini sur le serveur pour cette application. Demandez à votre administrateur de systèmes le nom du profil à utiliser.

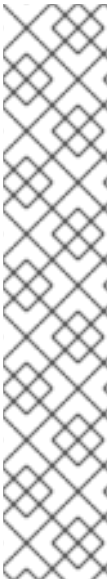
**Procédure 13.1. Ajouter une configuration de Profil de journalisation à une application**

- **Modifier `MANIFEST.MF`**
  Si votre application ne possède pas de fichier **`MANIFEST.MF`** : créez-en un avec le contenu suivant, en remplaçant *NAME* par le nom de profil qui convient.

```
Manifest-Version: 1.0
      Logging-Profile: NAME
```

  Si votre application contient déjà un fichier **`MANIFEST.MF`** : ajouter la ligne suivante, en remplaçant *NAME* par le nom du profil qui convient.

```
Logging-Profile: NAME
```



**NOTE**

Si vous utilisez Maven et **maven-war-plugin**, vous pourrez mettre votre fichier MANIFEST.MF dans **src/main/resources/META-INF/** et ajouter la configuration suivante à votre fichier **pom.xml**.

```
<plugin>
      <artifactId>maven-war-plugin</artifactId>
      <configuration>
        <archive>
          <manifestFile>src/main/resources/META-
INF/MANIFEST.MF</manifestFile>
        </archive>
      </configuration>
   </plugin>
```

Quand l'application sera déployée, elle utilisera la configuration qui se trouve dans le profil de journalisation spécifié pour ses messages de journalisation.

Report a bug

### 13.4.5. Exemple de Configuration de Profil de journalisation

Cet exemple montre la configuration du profil de journalisation et l'application qui en fait usage. Cela comprend la session CLI affichée, la configuration XML qui est générée et le fichier **MANIFEST.MF** de l'application.

L'exemple de profil de journalisation a les caractéristiques suivantes :

- Le Nom est **accounts-app-profile**.
- La Catégorie de journalisation est **com.company.accounts.ejbs**.
- Le niveau de journalisations est **TRACE**.
- Le Log Handler est un gestionnaire de fichiers qui utilise **ejb-trace.log**.

**Exemple 13.59. Session CLI**

```
localhost:bin user$ ./jboss-cli.sh -c
[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile:add
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile/file-handler=ejb-trace-file:add(file=
{path=>"ejb-trace.log", "relative-to"=>"jboss.server.log.dir"})
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile/file-handler=ejb-trace-file:change-log-
level(level="DEBUG")
{"outcome" => "success"}
```

```
[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-
profile/logger=com.company.accounts.ejbs:add(level=TRACE)
{"outcome" => "success"}

[standalone@localhost:9999 /] /subsystem=logging/logging-
profile=accounts-app-profile/logger=com.company.accounts.ejbs:assign-
handler(name="ejb-trace-file")
{"outcome" => "success"}

[standalone@localhost:9999 /]
```

**Exemple 13.60. Configuration XML**

```
<logging-profiles>
   <logging-profile name="accounts-app-profile">
      <file-handler name="ejb-trace-file">
         <level name="DEBUG"/>
         <file relative-to="jboss.server.log.dir" path="ejb-
trace.log"/>
      </file-handler>
      <logger category="com.company.accounts.ejbs">
         <level name="TRACE"/>
         <handlers>
            <handler name="ejb-trace-file"/>
         </handlers>
      </logger>
   </logging-profile>
</logging-profiles>
```

**Exemple 13.61. Fichier Application MANIFEST.MF**

```
Manifest-Version: 1.0
Logging-Profile: accounts-app-profile
```

Report a bug

## 13.5. PROPRIÉTÉS DE LA CONFIGURATION DE JOURNALISATION

### 13.5.1. Propriétés Root Logger

**Tableau 13.5. Propriétés Root Logger**

| Property | Datatype | Description |
|---|---|---|
| niveau | String | Le niveau maximum de messages log que le root logger souhaite enregistrer. |

| Property | Datatype | Description |
|---|---|---|
| handlers | String[] | Une liste des log handlers utilisés par le root logger. |
| filter-spec | String | Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : `not(match("JBAS.*"))` |

Report a bug

### 13.5.2. Propriétés de catégorie de journalisation

**Tableau 13.6. Propriétés de catégorie de journalisation**

| Propriété | Datatype | Description |
|---|---|---|
| niveau | String | Le niveau maximum de messages log que le root logger souhaite enregistrer. |
| handlers | String[] | Une liste des log handlers utilisés par le root logger. |
| use-parent-handlers | Booléen | Si défini sur true, cette catégorie utilisera les Log handlers du Root logger en plus des handlers assignés. |
| catégorie | String | La catégorie de journalisation à partir de laquelle les messages de journalisation seront capturés. |
| filter-spec | String | Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : `not(match("JBAS.*"))` |

Report a bug

### 13.5.3. Propriétés de Log Handlers de console

**Tableau 13.7. Propriétés de Log Handlers de console**

| Property | Datatype | Description |
|---|---|---|
| level | String | Le niveau maximum de messages de journalisation que le Log Handler enregistre. |
| encoding | String | Définir la codification utilisée pour la sortie. |
| formatter | String | Le formateur de journalisation utilisé par ce Log Handler. |
| target | String | Le flux de sortie du système vers lequel la sortie du Log Handler se dirige. Peut correspondre à System.err ou System.out pour le flux d'erreurs système ou standard out respectivement. |

| Property | Datatype | Description |
|---|---|---|
| autoflush | Boolean | Si défini sur true, les messages de journalisation seront envoyés vers la cible des handlers immédiatement après la réception. |
| name | String | L'identifiant unique de ce Log Handler. |
| activé | Boolean | Si défini à **`true`**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **`false`**, le gestionnaire sera ignoré lors du traitement des messages de journalisation. |
| filter-spec | String | Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond à aucun modèle : **`not(match("JBAS.*"))`** |

Report a bug

### 13.5.4. Propriétés de Log Handlers de fichiers

**Tableau 13.8. Propriétés de Log Handlers de fichiers**

| Property | Datatype | Description |
|---|---|---|
| level | String | Le niveau maximum de messages de journalisation que le Log Handler enregistre. |
| encoding | String | Définir la codification utilisée pour la sortie. |
| formatter | String | Le formateur de journalisation utilisé par ce Log Handler. |
| append | Boolean | Si la valeur true alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false, un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications à **`append`** nécessitent un redémarrage du serveur pour qu'elles soient prises en compte. |
| autoflush | Boolean | Si définis sur true, les messages de journalisation seront envoyés au fichier assigné aux handlers dès réception. Les changements à **`autoflush`** nécessitent un redémarrage de serveur pour pouvoir prendre effet. |
| name | String | L'identifiant unique de ce Log Handler. |
| fichier | Objet | L'objet qui représente le fichier dans lequel la sortie de ce Log Handler est écrite. Il contient deux propriétés de configuration, **`relative-to`** et **`path`**. |

| Property | Datatype | Description |
|---|---|---|
| relative-to | String | C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès de fichier JBoss Enterprise Application Platform 6 peuvent être indiquées ici. La variable **`jboss.server.log.dir`** pointe vers le répertoire **`log/`** du serveur. |
| path | String | C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrit les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété **`relative-to`** pour déterminer le chemin d'accès complet. |
| activé | Boolean | Si défini à **`true`**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **`false`**, le gestionnaire sera ignoré lors du traitement des messages de journalisation. |
| filter-spec | String | Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **`not(match("JBAS.*"))`** |

Report a bug

### 13.5.5. Propriétés de Log Handlers périodiques

**Tableau 13.9. Propriétés de Log Handlers périodiques**

| Property | Datatype | Description |
|---|---|---|
| append | Boolean | Si la valeur true alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false, un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications à «append» (ajout) nécessitent un redémarrage du serveur pour qu'elles soient prises en compte. |
| autoflush | Boolean | Si définis sur true, les messages de journalisation seront envoyés au fichier assigné aux handlers dès réception. Les changements à «autoflush» nécessitent un redémarrage de serveur pour pouvoir prendre effet. |
| encoding | String | Définir la codification utilisée pour la sortie. |
| formatter | String | Le formateur de journalisation utilisé par ce Log Handler. |
| level | String | Le niveau maximum de messages de journalisation que le Log Handler enregistre. |
| name | String | L'identifiant unique de ce Log Handler. |
| fichier | Object | L'objet qui représente le fichier dans lequel la sortie de ce Log Handler est écrite. Il contient deux propriétés de configuration, **`relative-to`** et **`path`**. |

| Property | Datatype | Description |
|---|---|---|
| relative-to | String | C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès peuvent être indiquées ici. La variable **`jboss.server.log.dir`** pointe vers le répertoire **`log/`** du serveur. |
| path | String | C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrit les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété **`relative-to`** pour déterminer le chemin d'accès complet. |
| suffix | String | Cette chaîne est ajoutée au nom de fichier des journaux en rotation et sert à déterminer la fréquence de rotation. Le format du suffixe est un point (.) suivi d'une date de chaîne qui est analysable par la classe **`Ijava.text.SimpleDateFormat`**. Le journal est mis en rotation sur la base de la plus petite unité de temps définie par le suffixe. Par exemple, le suffixe **`.yyyy-MM-dd`** se traduira par rotation quotidienne du log ou journal.<br><br>Voir http://docs.oracle.com/javase/6/docs/api/index.html?java/text/SimpleDateFormat.html |
| activé | Boolean | Si défini à **`true`**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **`false`**, le gestionnaire sera ignoré lors du traitement des messages de journalisation. |
| filter-spec | String | Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **`not(match("JBAS.*"))`** |

Report a bug

## 13.5.6. Propriétés de Log Handlers de Taille

**Tableau 13.10. Propriétés de Log Handlers de Taille**

| Property | Datatype | Description |
|---|---|---|
| append | Boolean | Si la valeur true alors tous les messages rédigés par ce gestionnaire seront ajoutés au fichier si celui-ci existe déjà. Si la valeur est false, un nouveau fichier sera créé chaque fois que le serveur d'applications est lancé. Les modifications à «append» (ajout) nécessitent un redémarrage du serveur pour qu'elles soient prises en compte. |
| autoflush | Boolean | Si définis sur true, les messages de journalisation seront envoyés au fichier assigné aux handlers dès réception. Les changements à «append» (ajout) nécessitent un redémarrage de serveur pour pouvoir prendre effet. |
| encoding | String | Définir la codification utilisée pour la sortie. |

| Property | Datatype | Description |
|---|---|---|
| formatter | String | Le formateur de journalisation utilisé par ce Log Handler. |
| level | String | Le niveau maximum de messages de journalisation que le Log Handler enregistre. |
| name | String | L'identifiant unique de ce Log Handler. |
| file | Objet | L'objet qui représente le fichier dans lequel la sortie de ce Log Handler est écrite. Il contient deux propriétés de configuration, **`relative-to`** et **`path`**. |
| relative-to | String | C'est une propriété de l'objet fichier qui correspond au répertoire où le fichier journal est écrit. Les variables de chemin d'accès peuvent être indiquées ici. La variable **`jboss.server.log.dir`** pointe vers le répertoire **`log/`** du serveur. |
| path | String | C'est une propriété de l'objet fichier qui correspond au nom du fichier où seront écrit les messages du journal. C'est un nom de chemin d'accès relatif qui est ajouté à la valeur de la propriété **`relative-to`** pour déterminer le chemin d'accès complet. |
| rotate-size | Integer | La taille maximale que le fichier journal peut atteindre avant qu'il soit mis en rotation. Un seul caractère ajouté au nombre indique les unités de taille: **`b`** pour bytes, **`k`** pour kilobytes, **`m`** pour megabytes, **`g`** pour gigabytes. Par ex. **`50m`** pour 50 megabytes. |
| max-backup-index | Integer | Le nombre maximum de journaux en rotation conservés. Quand ce nombre est atteint, le journal le plus ancien est utilisé à nouveau. |
| activé | Boolean | Si défini à **`true`**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **`false`**, le gestionnaire sera ignoré lors du traitement des messages de journalisation. |
| filter-spec | String | Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **`not(match("JBAS.*"))`** |

Report a bug

### 13.5.7. Propriétés de Log Handlers Async

**Tableau 13.11. Propriétés de Log Handlers Async**

| Property | Datatype | Description |
|---|---|---|
| level | String | Le niveau maximum de messages de journalisation que le Log Handler enregistre. |
| name | String | L'identifiant unique de ce Log Handler. |

| Property | Datatype | Description |
|---|---|---|
| queue-length | Integer | Nombre maximal de messages de journalisation gardés par le handler en attendant que les sub-handlers répondent. |
| overflow-action | String | La façon dont ce handler répond quand sa file d'attente est dépassée. Peut être défini sur **BLOCK** ou **DISCARD**. **BLOCK** fait patienter l'application de journalisation jusqu'à ce qu'il y ait suffisamment d'espace disponible dans la file d'attente. C'est le même comportement qu'avec un handler non-async. **DISCARD** permet à l'application de journalisation de continuer, mais le message de journalisation sera effacé. |
| subhandlers | String[] | Il s'agit de la liste de Log Handlers à laquelle cet handler async passe ses messages log. |
| enabled | Booléen | Si défini à **true**, le gestionnaire sera activé et fonctionnera normalement. Si défini sur **false**, le gestionnaire sera ignoré lors du traitement des messages de journalisation. |
| filter-spec | String | Expression qui définit un filtre. L'expression suivante définit un filtre qui ne correspond pas à un modèle : **not(match("JBAS.*"))** |

Report a bug

## 13.6. EXEMPLE DE CONFIGURATION XML DE LOGGING

### 13.6.1. Échantillon de Configuration XML pour Root Logger

```
<subsystem xmlns="urn:jboss:domain:logging:1.2">

   <root-logger>
      <level name="INFO"/>
      <handlers>
         <handler name="CONSOLE"/>
         <handler name="FILE"/>
      </handlers>
   </root-logger>

</subsystem>
```

Report a bug

### 13.6.2. Échantillon de Configuration XML pour une Catégorie de journalisation

```
<subsystem xmlns="urn:jboss:domain:logging:1.2">

   <logger category="com.company.accounts.rec">
      <handlers>
```

```
            <handler name="accounts-rec"/>
         </handlers>
      </logger>

   </subsystem>
```

Report a bug

### 13.6.3. Échantillon de Configuration XML pour un Log Handler de console

```
<subsystem xmlns="urn:jboss:domain:logging:1.2">

   <console-handler name="CONSOLE">
      <level name="INFO"/>
      <formatter>
         <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t)
%s%E%n"/>
      </formatter>
   </console-handler>

</subsystem>
```

Report a bug

### 13.6.4. Échantillon de Configuration XML pour un Gestionnaire de journalisation ou Log Handler de fichiers

```
<file-handler name="accounts-rec-trail" autoflush="true">
    <level name="INFO"/>
    <file relative-to="jboss.server.log.dir" path="accounts-rec-
trail.log"/>
    <append value="true"/>
</file-handler>
```

Report a bug

### 13.6.5. Échantillon de Configuration XML pour un Log Handler périodique

```
<periodic-rotating-file-handler name="FILE">
   <formatter>
      <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t)
%s%E%n"/>
   </formatter>
   <file relative-to="jboss.server.log.dir" path="server.log"/>
   <suffix value=".yyyy-MM-dd"/>
   <append value="true"/>
</periodic-rotating-file-handler>
```

Report a bug

### 13.6.6. Échantillon de Configuration XML pour un Log Handler de Taille

```
<size-rotating-file-handler name="accounts_debug" autoflush="false">
```

```
    <level name="DEBUG"/>
    <file relative-to="jboss.server.log.dir" path="accounts-debug.log"/>
    <rotate-size value="500k"/>
    <max-backup-index value="5"/>
    <append value="true"/>
</size-rotating-file-handler>
```

Report a bug

### 13.6.7. Échantillon de Configuration XML pour un Log Handler Async

```
<async-handler name="Async_NFS_handlers">
    <level name="INFO"/>
    <queue-length value="512"/>
    <overflow-action value="block"/>
    <subhandlers>
        <handler name="FILE"/>
        <handler name="accounts-record"/>
    </subhandlers>
</async-handler>
```

Report a bug

# CHAPITRE 14. JVM

## 14.1. JVM

### 14.1.1. Paramètres de configuration de JVM

Les paramètres de configuration de Machines virtuelles Java (JVM) varient selon les instances de domaine géré et les instances de serveur autonome. Dans un domaine géré, les paramètres de JVM sont déclarés dans les fichiers de configuration **`host.xml`** et **`domain.xml`**, et ils sont déterminés par les composants de contrôleur de domaine chargés de lancer et d'arrêter le processus du serveur. Dans une instance de serveur autonome, les processus de démarrage de serveur peuvent passer des paramètres de ligne de commande au démarrage. Ceux-ci peuvent être déclarés depuis la ligne de commande ou via l'écran de **`Propriétés système`** dans la Console de gestion.

**Domaine géré**

Une caractéristique importante du domaine géré est la possibilité de définir des paramètres de la JVM à plusieurs niveaux. Vous pouvez configurer les paramètres de JVM personnalisés au niveau de l'hôte, par groupe de serveurs, ou par instance de serveur. Les éléments enfants plus spécialisés remplacent la configuration parent, permettant la déclaration des configurations de serveur spécifique sans nécessiter d'exclusions au niveau groupe ou hôte. Cela permet également la configuration parent d'être héritée par les autres niveaux jusqu'à ce que les paramètres soient déclarés dans les fichiers de configuration ou transmis pendant le runtime.

**Exemple 14.1. Les paramètres de configuration JVM du fichier de configuration du domaine**

L'exemple suivant montre une déclaration JVM pour un groupe de serveurs dans le fichier de configuration **`domain.xml`**.

```
<server-groups>
       <server-group name="main-server-group" profile="default">
           <jvm name="default">
               <heap size="64m" max-size="512m"/>
           </jvm>
           <socket-binding-group ref="standard-sockets"/>
       </server-group>
       <server-group name="other-server-group" profile="default">
           <jvm name="default">
               <heap size="64m" max-size="512m"/>
           </jvm>
           <socket-binding-group ref="standard-sockets"/>
       </server-group>
</server-groups>
```

Dans cette instance, un groupe de serveurs appelé **`main-server-group`** déclare une taille de segment de 64 mégaoctets et une taille de segment maximale de 512 méga-octets. N'importe quel serveur qui appartient à ce groupe héritera de ces paramètres. Vous pouvez modifier ces paramètres pour le groupe dans son ensemble, par hôte ou serveur individuel.

**Exemple 14.2. Les paramètres de configuration du domaine dans le fichier de configuration de l'hôte**

L'exemple suivant montre une déclaration JVM pour un groupe de serveurs dans le fichier de configuration **host.xml**.

```
<servers>
       <server name="server-one" group="main-server-group" auto-
start="true">
           <jvm name="default"/>
       </server>
       <server name="server-two" group="main-server-group" auto-
start="true">
           <jvm name="default">
               <heap size="64m" max-size="256m"/>
           </jvm>
           <socket-binding-group ref="standard-sockets" port-
offset="150"/>
       </server>
       <server name="server-three" group="other-server-group" auto-
start="false">
           <socket-binding-group ref="standard-sockets" port-
offset="250"/>
       </server>
</servers>
```

Dans ce cas, un serveur appelé **server-two** appartient au groupe de serveurs nommé **main-server-group**, qui hérite les paramètres du groupe de JVM **par défaut**. Dans l'exemple précédent, la taille du segment principal de **main-server-group** a été fixée à 512 méga-octets. En déclarant une taille de segment basse de 256 mégaoctets, **server-two** peut substituer les paramètres de **domain.xml** pour ajuster les performances comme vous le souhaitez.

#### Paramètres de configuration de serveur autonome en cours d'exécution

Les paramètres de JVM pour des instances de serveurs autonomes peuvent être déclarés pendant l'exécution en définissant la variable d'environnement **JAVA_OPTS** avant de démarrer le serveur. Un exemple de définition de la variable d'environnement **JAVA_OPTS** en ligne de commande Linux est :

```
[user@host bin]$ export JAVA_OPTS="-Xmx1024M"
```

La même configuration peut être utilisée dans un environnement Microsoft Windows, comme suit :

```
C:\> set JAVA_OPTS="Xmx1024M"
```

Alternativement, les paramètres de configuration JVM peuvent être ajoutés au fichier **standalone.conf** qui se trouve dans le dossier ***EAP_HOME*/bin**, contenant des exemples d'options à passer à la JVM.

Report a bug

### 14.1.2. Afficher le statut JVM dans la Console de gestion

#### Prérequis

- Section 2.1.2, « Démarrez JBoss EAP 6 comme un serveur autonome »

- Section 2.1.3, « Démarrer JBoss Enterprise Application Platform 6 en tant que domaine géré »
- Section 3.4.2, « Connectez-vous à la Console de management »

Le statut de la Machine virtuelle Java (JVM) peut être affichée dans la Console de gestion pour le serveur autonome ou un domaine géré. La console affiche l'utilisation de segments, leur non utilisation, et l'usage de threads du serveur en mégaoctets. Malgré que les statistiques ne soient pas affichés en temps réel, vous pouvez actualiser l'affichage de la console pour donner un aperçu à jour des ressources de la machine virtuelle Java.

Le statut de la JVM affiche les valeurs suivantes.

**Tableau 14.1. Attributs de Statut JVM**

| Type | Description |
|---|---|
| Max | Le montant de mémoire maximal en octets pouvant être utilisés pour la gestion de la mémoire. |
| Utilisé | Le montant de mémoire utilisé en méga octets. |
| Validé | Le montant de mémoire en octets alloué à l'utilisation de la machine virtuelle Java. |
| Init | Le montant de mémoire en octets que la machine virtuelle Java a demandé au départ au système d'exploitation pour la gestion de la mémoire. |

**Procédure 14.1. Afficher le statut JVM dans la Console de gestion**

- **Afficher le statut de la JVM**
  Vous pouvez afficher le statut de la JVM dans l'instance de serveur autonome ou dans un domaine géré.
  - **Affichage du statut de la JVM pour une instance de serveur autonome**
    Sélectionner **JVM Status** du menu **Server Status** sure l'écran de **Runtime**.

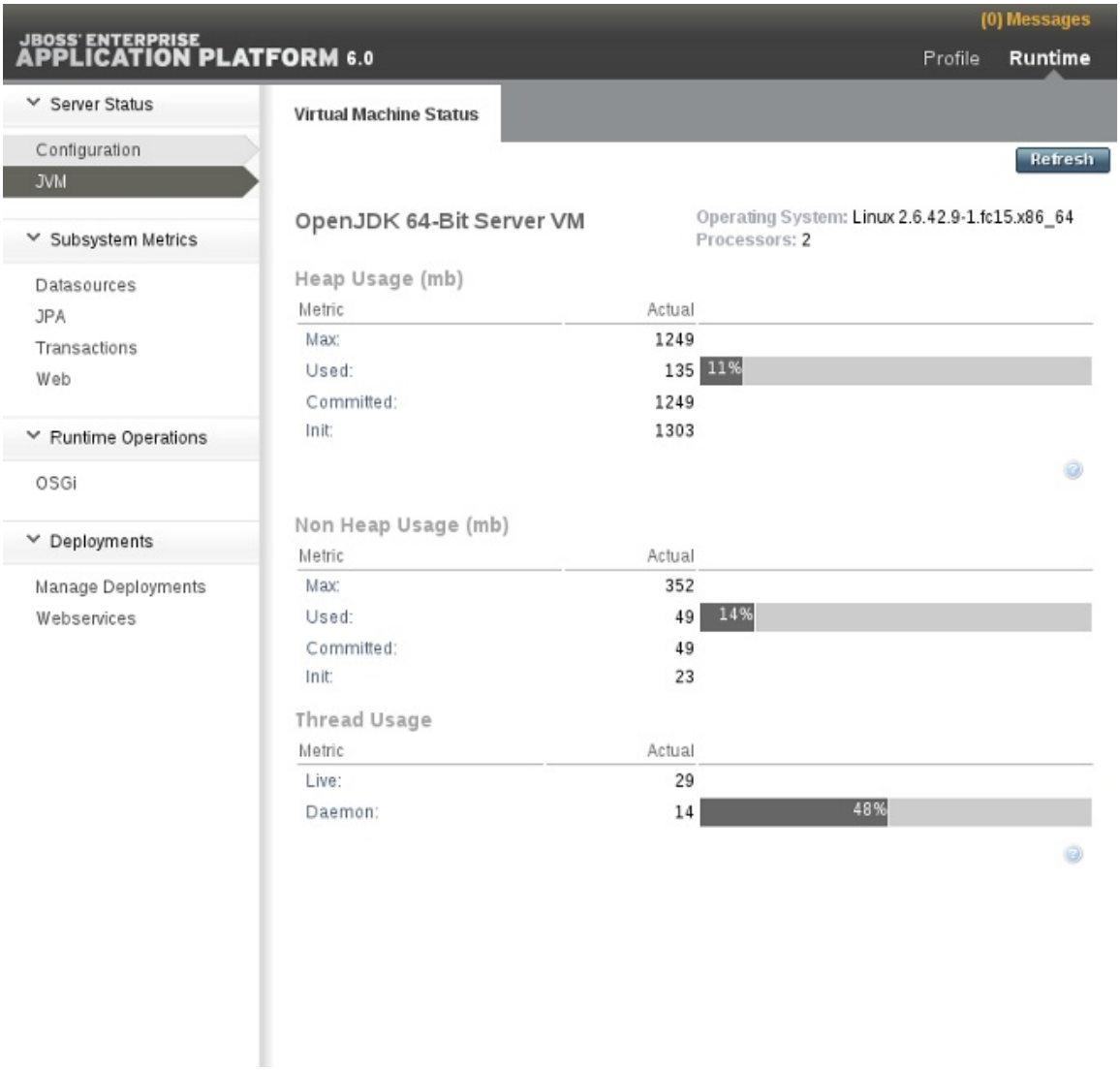



**Figure 14.1. Statut de la JVM pour une instance de serveur autonome**

- **Afficher le statut de la JVM d'un domaine géré**
  Sélectionner le statut de la JVM à partir du menu Statut de domaine sur l'écran Runtime.

- Le domaine géré peut rendre visibles toutes les instances de serveur dans le groupe de serveurs, mais ne vous permettra d'afficher qu'un seul serveur à la fois en sélectionnant dans le menu serveur. Pour afficher le statut des autres serveurs dans votre groupe de serveurs, cliquez sur le menu déroulant en haut à gauche de l'écran pour sélectionner à partir de l'hôte et des serveurs affichés dans votre groupe et cliquez sur le bouton **Done** pour charger les résultats.

### Résultat

Le statut des paramètres de configuration de la JVM de l'instance de serveur est affiché.

Report a bug

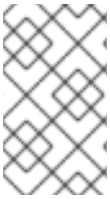
# CHAPITRE 15. SOUS-SYSTÈME WEB

## 15.1. CONFIGURER LE SOUS-SYSTÈME WEB

Vous pouvez configurer la plupart des aspects du sous-système Web à l'aide de la Console de gestion sur le web ou le Management CLI de ligne de commande. Chaque paramètre est expliqué dans l'ordre dans lequel il apparaît dans la Console de gestion, et les commandes de Management CLI de gestion sont également fournies.

#### Afficher le Sous-système basé Web par la Console de gestion

Pour configurer le sous-système de Web à l'aide de la Console de gestion sur le web, cliquez sur l'onglet **Profile(s)** en haut à droite. Pour un domaine géré, sélectionnez le profil de serveur que vous souhaitez configurer dans la boîte de sélection de **Profil** en haut à gauche. Développez le menu de **Subsystems**, puis le menu **Web**. Chaque partie configurable du sous-système Web est montré.



**NOTE**

Le composant **mod_cluster** n'est disponible que si votre profil est **ha** ou **full-ha**, dans un domaine géré, ou si vous démarrez votre serveur autonome avec le profil **standalone-ha**. La configuration **mod_cluster** est abordée dans .

#### Configurer le Conteneur JSP, les connecteurs HTTP, et les serveurs HTTP virtuels

Pour configurer le conteneur JSP, les connecteurs HTTP et des serveurs virtuels HTTP, cliquez sur l'entrée de menu **Servlet/HTTP**. Cliquez sur le bouton **Edit** pour modifier une valeur. Cliquez sur le bouton **Advanced** pour afficher les options avancées. Les options sont expliquées ci-dessous. Les options pour les serveurs virtuels et les connecteurs HTTP figurent dans des tableaux distincts.

**Tableau 15.1. Options de configuration Servlet/HTTP**

| Option | Description | Commande CLI |
|---|---|---|
| Disabled? | Si sur **true**, désactive le conteneur Java ServerPages (JSP). Valeur par défaut **false**. Utile si vous n'utilisez pas les pages Java ServerPages (JSPs). | `/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=disabled,value=false)` |
| Development? | Si sur true, active Development Mode, qui produit davantage d'informations verbeuses de débogage. Valeur par défaut **false**. | `/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=development,value=false)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Keep Generated? | Cliquer sur **Advanced** pour voir cette option, si elle est cachée. Si sur **true** garde les servlets générés. Activé par défaut. | `/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=keep-generated,value=true)` |
| Check Interval? | Cliquer sur **Advanced** pour voir cette option, si elle est cachée. Valeur en secondes qui détermine la fréquence des vérifications de mises à jour JSP par un processus en arrière-plan. La valeur par défaut est **0**. | `/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=check-interval,value=0)` |
| Display Source? | Cliquer sur **Advanced** pour voir cette option, si elle est cachée. Si sur **true**, le fragment de source JSP est affiché quand une erreur d'exécution a lieu. La valeur par défaut est **true**. | `/profile=full-ha/subsystem=web/configuration=jsp-configuration/:write-attribute(name=display-source-fragment,value=true)` |

Les connecteurs AJP and HTTP utilisent **mod_cluster**, **mod_jk**, **mod_proxy**, **ISAPI**, et **NSAPI** pour l'équilibrage des charges et pour le clustering HA. Pour configurer un connecteur, sélectionner l'onglet **Connectors** et cliquer sur **Add**. Pour supprimer un connecteur, le sélectionner et cliquer sur **Remove**. Pour modifier un connecteur, le sélectionner et cliquer sur **Edit**.

Quand vous créez un nouveau connecteur par le Management CLI, ses options sont définies aussitôt, comme dans la commande suivante :

**Exemple 15.1. Créer un Nouveau connecteur**

```
/profile=full-ha/subsystem=web/connector=ajp/:add(socket-binding=ajp,scheme=http,protocol=AJP/1.3,secure=false,name=ajp,max-post-size=2097152,enabled=true,enable-lookups=false,redirect-port=8433,max-save-post-size=4096)
```

**Tableau 15.2. Options de connecteur**

| Option | Description | Commande CLI |
| --- | --- | --- |
| Nom | Un nom unique de connecteur, à but d'affichage. | `/profile=full-ha/subsystem=web/connector=ajp/:read-attribute(name=name)` |
| Liaisons de sockets | La liaison de socket nommée à laquelle le connecteur doit se lier. La liaison de socket est un mappage entre un nom de socket et un port réseau. Les liaisons de socket sont configurées pour chaque serveur autonome, ou par l'intermédiaire de groupes de liaison de socket dans un domaine géré. Un groupe de liaisons de sockets est appliqué à un groupe de serveurs. | `/profile=full-ha/subsystem=web/connector=ajp/:write-attribute(name=socket-binding,value=ajp)` |
| Schéma | Le schéma de connecteur web, comme HTTP ou HTTPS. | `/profile=full-ha/subsystem=web/connector=ajp/:write-attribute(name=scheme,value=http)` |
| Protocole | Le protocole de connecteur web à utiliser, comme AJP ou HTTP. | `/profile=full-ha/subsystem=web/connector=ajp/:write-attribute(name=protocol,value=AJP/1.3)` |
| Activé | Indique si le connecteur web connecté est activé. | `/profile=full-ha/subsystem=web/connector=ajp/:write-attribute(name=enabled,value=true)` |

Pour configurer des serveurs virtuels. cliquer sur l'onglet **`Virtual Servers`**. Utiliser le bouton **`Add`** pour ajouter un nouveau serveur virtuel. Pour modifier ou supprimer un serveur virtuel, le sélectionner et cliquer le bouton**`Edit`** ou **`Remove`**.

Quand vous ajoutez un nouveau serveur virtuel par le Management CLI, toutes les options requises sont définies en même temps, comme par la commande suivante.

**Exemple 15.2. Ajouter un nouveau serveur virtuel**

```
/profile=full-ha/subsystem=web/virtual-server=default-host/:add(enable-welcome-root=true,default-web-module=ROOT.war,alias=["localhost","example.com"],name=default-host)
```

**Tableau 15.3. Options de serveurs virtuels**

| Option | Description | Commande CLI |
|---|---|---|
| Nom | Nom unique de serveur virtuel, à but d'affichage. | `/profile=full-ha/subsystem=web/virtual-server=default-host/:write-attribute(name=name,value=default-host)` |
| Alias | Une liste de noms d'hôtes qui doivent correspondre à ce serveur virtuel. Dans la Console de management, utiliser un nom d'hôte par ligne. | `/profile=full-ha/subsystem=web/virtual-server=default-host/:write-attribute(name=alias,value=["localhost","example.com"])` |
| Module par défaut | Le module dont l'application web doit être déployée au noeud racine de ce serveur virtuel et qui sera affiché quand aucun répertoire n'est donné par la requête HTTP. | `/profile=full-ha/subsystem=web/virtual-server=default-host/:write-attribute(name=default-web-module,value=ROOT.war)` |

### Configurer les Options de services web

Pour configurer les options de services web, cliquer sur **`Web Services`**. Les options sont expliquées dans le tableau ci-dessous.

**Tableau 15.4. Options de configuration des services web**

| Option | Description | Commande CLI |
|---|---|---|

| Option | Description | Commande CLI |
| --- | --- | --- |
| Modifier l'adresse WSDL | Indique si l'adresse WSDL peut être modifiée par les applications. Valeur par défaut **`true`**. | `/profile=full-ha/subsystem=webservices/:write-attribute(name=modify-wsdl-address,value=true)` |
| Hôte WSDL | Le contrat WSDL d'un Service Web JAX-WS inclut un élément <soap:address> qui pointe vers l'emplacement du point de terminaison. Si la valeur de <soap:address> est un URL valide, elle n'est pas remplacée à moins que **`modify-wsdl-adress`** soit défini a la valeur **`true`**. Si la valeur de <soap:address> n'est pas un URL valide, elle est remplacée en utilisant les valeurs **`wsdl-host`** et **`wsdl-port`** ou **`wsdl-secure-port`**. Si **`wsdl-host`** est défini sur **`jbossws.undefined.host`**, l'adresse hôte de l'auteur de la demande est utilisée lorsque <soap:address> est réécrite. Par défaut, **`${jboss.bind.address:127.0.0.1}`**, qui utilise **`127.0.0.1`** si aucune adresse de liaison est spécifiée lors du démarrage de JBoss Enterprise Application Platform. | `/profile=full-ha/subsystem=webservices/:write-attribute(name=wsdl-host,value=127.0.0.1)` |
| Port WSDL | Le port non-sécurisé utilisé pour écrire à nouveau l'adresse SOAP. Si définie sur **`0`** (défaut), le port sera identifié en demandant la liste des connecteurs installés. | `/profile=full-ha/subsystem=webservices/:write-attribute(name=wsdl-port,value=80)` |
| Port sécurisé WSDL | Le port sécurisé utilisé pour écrire à nouveau l'adresse SOAP. Si définie sur **`0`** (défaut), le port sera identifié en demandant la liste des connecteurs installés. | `/profile=full-ha/subsystem=webservices/:write-attribute(name=wsdl-secure-port,value=443)` |

Report a bug

## 15.2. REMPLACER L'APPLICATION WEB WELCOME PAR DÉFAUT

JBoss Enterprise Application Platform 6 inclut l'application Welcome, qui s'affiche quand on ouvre l'URL du serveur sur le port 8080. Vous pouvez remplacer cette application par votre propre application web, en suivant la procédure suivante.

**Procédure 15.1. Remplacer l'application web Welcome par défaut par votre propre application web**

1. **Désactiver l'application Welcome**
   Utiliser le script de Management CLI ***EAP_HOME*/bin/jboss-cli.sh** pour exécuter la commande suivante. Vous aurez sans doute besoin de modifier un profil de domaine géré, ou retirer une portion de la commande **/profile=default** du serveur autonome.

   ```
   /profile=default/subsystem=web/virtual-server=default-host:write-
   attribute(name=enable-welcome-root,value=false)
   ```

2. **Configurer votre application web par le contexte root.**
   Afin de configurer votre application web, pour qu'elle utilise (/) comme adresse URL, modifier son fichier **jboss-web.xml**, qui se trouve dans le répertoire **META-INF/** ou **WEB-INF/**. Remplacer sa directive **<context-root>** par une autre qui ressemble à ce qui suit.

   ```
   <jboss-web>
       <context-root>/</context-root>
   </jboss-web>
   ```

3. **Déployer votre application.**
   Déployer votre application pour le groupe de serveurs ou le serveur que vous avez modifié, lors de la première étape. L'application est maintenant disponible sur **http://*SERVER_URL*:*PORT*/**.

Report a bug

# CHAPITRE 16. HTTP CLUSTERING ET ÉQUILIBRAGE DES CHARGES

## 16.1. INTRODUCTION

### 16.1.1. Clusters de Haute disponibilité et Clusters d'équilibrage des charges

Le terme *Clustering* se réfère à l'utilisation de ressources multiples, comme des serveurs, comme s'ils constituaient une seule entité. Les deux principaux types de clustering sont *Load balancing (LB)* et *High-availability (HA)*. Dans un groupement LB, toutes les ressources exécutent en même temps, et une couche de gestion se charge de répartir la charge de travail entre eux.

Dans le clustering HA, une ressource exécute, et une autre est prête à prendre position quand la première est rendue disponible. Le but du clustering HA est de réduire les chances de panne niveau matériel, logiciels ou réseau.

JBoss Enterprise Application Platform supporte le clustering à plusieurs niveaux. Certains sous-systèmes que l'on puisse rendre disponibles sont les suivants :

- Les instances du serveur d'applications
- Le serveur web, qu'il s'agisse de serveur JBoss Web, Apache HTTPD, Microsoft IIS, Oracle iPlanet Web Server, ou Apache Tomcat
- Les EJB avec, ou sans état
- Les services JNDI
- Mécanismes Single Sign On (SSO)
- Cache distribué
- Sessions HTTP
- Les services JMS et les MDB (Messages Driven Beans)

Report a bug

### 16.1.2. Composants pouvant bénéficier de la haute disponibilité (HA)

La haute disponibilité (HA) tombe dans un certain nombre de larges catégories de JBoss Enterprise Application Platform.

**Le Conteneur**

Plusieurs instances de JBoss Enterprise Application Server (exécutant en tant que serveur autonome) ou les membres d'un groupe de serveurs (exécutant en tant que domaine géré) peuvent être configurés pour être hautement disponibles. Cela signifie que si une instance ou un membre est arrêté ou disparaît du groupement, sa charge de travail sera déplacée vers un père. La charge de travail peut être gérée de manière à fournir une fonctionnalité d'équilibrage de la charge, afin que les serveurs ou les groupes de serveurs avec des ressources plus ou moins supérieures puissent prendre une part plus importante de la charge de travail, ou qu'une capacité supplémentaire puisse être ajoutée pendant les périodes de forte charge.

**Le serveur web**

Le serveur web lui-même peut être groupé pour HA, à l'aide d'un des mécanismes d'équilibrage de charge compatible. Le plus souple est le connecteur **`mod_cluster`**, qui est intégré dans le conteneur de JBoss Enterprise Application Platform. Les autres possibilités incluent les connecteurs Apache **`mod_jk`** ou **`mod_proxy`**, ou les connecteurs ISAPI et NSAPI.

**L'application**

Les application déployées peuvent être redues HA (High Availability) à cause de la spécification Java Enterprise Edition 6 (Java EE 6). Les EJB de session avec ou sans état peuvent être clusterisées, de façon à ce que si le nœud impliqué dans le travail disparaît, un autre nœud prendra sa place, et dans le cas de beans de session avec état, préserveront l'état.

Report a bug

### 16.1.3. Connecteurs HTTP - Aperçu général

JBoss Enterprise Application Platform a la possibilité d'utiliser des mécanismes d'équilibrage de charge et de haute disponibilité intégrés à des serveurs web HTTPD externes, tels que Apache Web Server, IIS de Microsoft et Oracle iPlanet. La plate-forme JBoss Enterprise Application Platform communique avec le serveur web externe à l'aide d'un connecteur HTTP. Ces connecteurs HTTP sont configurées dans le sous-système de web de JBoss Enterprise Application Platform.

Les serveurs HTTPD incluent des modules informatiques qui contrôlent la façon dont les requêtes HTTP sont routées vers les nœuds de worker de JBoss Enterprise Application Platform. Chacun de ces modules varie dans la façon dont il fonctionne et comment il est configuré. Les modules sont configurés pour équilibrer les charges de travail entre plusieurs nœuds de serveur de JBoss Enterprise Application Platform, pour déplacer des charges de travail vers d'autres serveurs dans le cas d'échec, ou les deux. Ces deux capacités sont appelées *L'équilibrage de charge* et *Haute disponibilité (HA)*.

JBoss Enterprise Application Platform supporte un certain nombre de connecteurs HTTP. Celui que vous choisirez dépendra du HTTPD auquel vous allez vous connecter et de l'autre fonctionnalité dont vous aurez besoin.

Le tableau ci-dessous liste les différences entre les différents connecteurs HTTP compatibles avec JBoss Enterprise Application Platform. Pour obtenir les dernières informations sur les configurations prises en charge pour les connecteurs HTTP, voir https://access.redhat.com/site/articles/111663.

**Tableau 16.1. Caractéristiques et contraintes des connecteurs HTTP**

| Connecteur | Web server | Systèmes d'exploitation pris en charge | Protocoles pris en charge | S'adapte au statut de déploiement | Prend en charge une sticky session |
|---|---|---|---|---|---|

| Connecteur | Web server | Systèmes d'exploitation pris en charge | Protocoles pris en charge | S'adapte au statut de déploiement | Prend en charge une sticky session |
|---|---|---|---|---|---|
| mod_cluster | JBoss Enterprise Web Server HTTPD, Native HTTPD (Red Hat Enterprise Linux only) | Red Hat Enterprise Linux, Microsoft Windows Server, Oracle Solaris | HTTP, HTTPS, AJP | Oui. Détecte le déploiement et l'annulation du déploiement d'applications et décide dynamiquement s'il faut diriger les demandes clients vers un serveur basé sur la question de savoir si l'application est déployée sur ce serveur. | Oui |
| mod_jk | JBoss Enterprise Web Server HTTPD, Native HTTPD (Red Hat Enterprise Linux only) | Red Hat Enterprise Linux, Microsoft Windows Server, Oracle Solaris | AJP | Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application. | Oui |
| mod_proxy | JBoss Enterprise Web Server HTTPD | Red Hat Enterprise Linux, Microsoft Windows Server, Oracle Solaris | HTTP, HTTPS, AJP | Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application. | Oui |
| ISAPI | Microsoft IIS | Microsoft Windows Server | AJP | Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application. | Oui |
| NSAPI | Oracle iPlanet Web Server | Oracle Solaris | AJP | Non. Redirige les demandes des clients vers le conteneur tant que le conteneur est disponible, quel que soit le statut de l'application. | Oui |

**Pour en savoir plus sur les connecteurs HTTP**

- Section 16.5.1, « Le connecteur HTTP `mod_cluster` »
- Section 16.6.1, « Le connecteur Apache mod_HTTP »

- Section 16.7.1, « Le connecteur Apache mod_proxy HTTP »
- Section 16.8.1, « Internet Server API (ISAPI) HTTP Connector »
- Section 16.9.1, « Netscape Server API (NSAPI) HTTP Connector »

JBoss Enterprise Application Platform supporte les configurations disponibles ici : https://access.redhat.com/site/articles/111663.

Report a bug

### 16.1.4. Noeud de worker

**Noeud de connecteur HTTP**

Un *worker node*, connu sous le simple nom de *node*, est un serveur JBoss Enterprise Application Platform qui accepte des requêtes d'un ou plusieurs serveurs HTTPD faisant face au client. JBoss Enterprise Application Platform peut accepter des requêtes de son propre HTTPD, c-a-d le HTTPD fourni avec JBoss Enterprise Web Server, Apache HTTPD, Microsoft IIS, ou Oracle iPlanet Web Server (ancien Netscape Web Server).

Pour avoir un aperçu des connecteurs HTTP pris en charge par JBoss EAP et sur la façon de les configurer, voir Section 16.1.3, « Connecteurs HTTP - Aperçu général ».

**Noeud de cluster**

Un nœud de cluster est un membre d'un groupement de serveurs. Un tel cluster peut être en équilibrage de charge, en haute disponibilité, ou les deux. Dans un cluster d'équilibrage de charge, un gestionnaire central distribue également la charge de travail parmi ses nœuds, par mesure d'égalité suivant la situation particulière. Dans un cluster de haute disponibilité (HA), certains nœuds travaillent activement, tandis que d'autres sont en attente d'intervenir si un des nœuds actifs quitte le cluster.

Report a bug

## 16.2. CONFIGURATION DE CONNECTEUR

### 16.2.1. Définir

**Résumé**

Les pools de threads de JBoss Enterprise Application Platform peuvent être partagés entre les différents éléments à l'aide du modèle Executor. Ces pools peuvent être partagés non seulement par les différents connecteurs (HTTP), mais aussi par d'autres composants de JBoss Enterprise Application Platform qui prennent en charge le modèle Executor. Obtenir le pool de threads de connecteurs HTTP qui corresponde à vos exigences de performance web actuel est un art délicat et nécessite une étroite surveillance du pool de threads en cours et des exigences de charge web en cours ou anticipées. Dans cette tâche, vous allez apprendre à définir un pool de threads de connecteur HTTP en utilisant le modèle Executor. Vous apprendrez comment définir cela en utilisant à la fois l'interface par ligne de commande et en modifiant le fichier de configuration XML.

**Procédure 16.1. Installation d'un pool de threads pour un connecteur HTTP**

1. **Définir une usine de threads**
   Ouvrir votre fichier de configuration (**`standalone.xml`** si vous modifiez un serveur autonome ou **`domain.xml`** si vous modifiez une configuration basée domaine. Ce fichier se trouve dans le dossier ***`EAP_HOME`*`/standalone/configuration`** ou dans

***EAP_HOME*/domain/configuration**).

Ajouter l'entrée de sous-système suivante, en modifiant les valeurs en fonction de vos besoins de serveur.

```
<subsystem xmlns="urn:jboss:domain:threads:1.0">
    <thread-factory name="http-connector-factory" thread-name-
pattern="HTTP-%t" priority="9" group-name="uq-thread-pool"/>
</subsystem>
```

Si vous préférez utiliser le CLI pour cette tâche, alors exécutez la commande suivante dans une invite de commande CLI :

```
[standalone@localhost:9999 /] ./subsystem=threads/thread-
factory=http-connector-factory:add(thread-name-pattern="HTTP-%t",
priority="9", group-name="uq-thread-pool")
```

2. **Créer un exécuteur**
   Vous pouvez utiliser une des six classes d'exécuteur intégrées pour qu'elle agisse en tant qu'exécuteur pour cette usine : **unbounded-queue-thread-pool**, **bounded-queue-thread-pool**, **blocking-bounded-queue-thread-pool**, **queueless-thread-pool**, **blocking-queueless-thread-pool** et **scheduled-thread-pool**.

   Dans cet exemple, nous utiliserons **unbounded-queue-thread-pool** comme exécuteur. Modifier les valeurs des paramètres ***max-threads*** et ***keepalive-time*** selon les besoins de votre serveur.

```
<unbounded-queue-thread-pool name="uq-thread-pool">
  <thread-factory name="http-connector-factory" />
  <max-threads count="10" />
  <keepalive-time time="30" unit="seconds" />
</unbounded-queue-thread-pool>
```

   Ou bien, si vous préférez utiliser le CLI :

```
[standalone@localhost:9999 /] ./subsystem=threads/unbounded-queue-
thread-pool=uq-thread-pool:add(thread-factory="http-connector-
factory", keepalive-time={time=30, unit="seconds"}, max-threads=30)
```

3. **Forcez le connecteur web HTTP à utiliser ce pool de threads**
   Dans le même fichier de configurations, cherchez l'élément de connecteur HTTP dans le sous-système web et modifiez-le en utilisant le pool de threads défini dans les étapes suivantes.

```
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-
binding="http" executor="uq-thread-pool" />
```

   Si vous préférez utiliser le CLI :

```
[standalone@localhost:9999 /] ./subsystem=web/connector=http:write-
attribute(name=executor, value="uq-thread-pool")
```

4. **Redémarrer le serveur**
   Redémarrer le serveur (autonome ou domaine) pour que les changements puissent prendre effet. Utiliser les commandes CLI suivantes pour confirmer si les changements des étapes ci-dessus ont eu lieu :

```
[standalone@localhost:9999 /] ./subsystem=threads:read-
resource(recursive=true)
{
    "outcome" => "success",
    "result" => {
        "blocking-bounded-queue-thread-pool" => undefined,
        "blocking-queueless-thread-pool" => undefined,
        "bounded-queue-thread-pool" => undefined,
        "queueless-thread-pool" => undefined,
        "scheduled-thread-pool" => undefined,
        "thread-factory" => {"http-connector-factory" => {
            "group-name" => "uq-thread-pool",
            "name" => "http-connector-factory",
            "priority" => 9,
            "thread-name-pattern" => "HTTP-%t"
        }},
        "unbounded-queue-thread-pool" => {"uq-thread-pool" => {
            "keepalive-time" => {
                "time" => 30L,
                "unit" => "SECONDS"
            },
            "max-threads" => 30,
            "name" => "uq-thread-pool",
            "thread-factory" => "http-connector-factory"
        }}
    }
}
[standalone@localhost:9999 /] ./subsystem=web/connector=http:read-
resource(recursive=true)
{
    "outcome" => "success",
    "result" => {
        "configuration" => undefined,
        "enable-lookups" => false,
        "enabled" => true,
        "executor" => "uq-thread-pool",
        "max-connections" => undefined,
        "max-post-size" => 2097152,
        "max-save-post-size" => 4096,
        "name" => "http",
        "protocol" => "HTTP/1.1",
        "proxy-name" => undefined,
        "proxy-port" => undefined,
        "redirect-port" => 443,
        "scheme" => "http",
        "secure" => false,
        "socket-binding" => "http",
        "ssl" => undefined,
        "virtual-server" => undefined
    }
}
```

**Résultat**

Vous avez pu créer une usine de threads et un exécuteur, tout en modifiant votre Connecteur HTTP pour qu'il utilise ce pool de threads.

Report a bug

## 16.3. CONFIGURATION HTTP

### 16.3.1. HTTP Autonome

JBoss Enterprise Application Platform est testé et pris en charge avec Apache HTTPD qui est inclus avec les versions Red Hat Enterprise Linux 6 certifiées. Apache HTTPD est également disponible pour les autres configurations, telles que Microsoft Windows Server. Cependant, comme Apache HTTPD est un produit distinct, produit de la Fondation Apache, il était difficile d'être certain que la version d'Apache HTTPD qu'un client utilisait était compatible avec JBoss Enterprise Application Platform.

Apache HTTPD Autonome est maintenant inclus en tant que téléchargement séparé avec JBoss Enterprise Application Platform 6. Ceci simplifie l'installation et la configuration dans les environnements autres que Red Hat Enterprise Linux, ou sur des systèmes qui déjà ont une configuration HTTPD et qui souhaitent utiliser une instance distincte pour les applications web. Vous pouvez télécharger ce HTTPD en tant que téléchargement distinct dans le Portail de Services Client, qui se trouve dans la liste des déchargements de JBoss Enterprise Application Platform 6 disponibles pour votre plate-forme d'installation.

Report a bug

### 16.3.2. Installer Apache HTTPD inclus avec JBoss Enterprise Application Platform 6

**Prérequis**

- Vous aurez besoin d'un accès administrateur ou root pour compléter cette tâche.

**Procédure 16.2. Installer Apache HTTPD**

1. **Naviguez dans la liste des téléchargements de JBoss Enterprise Application Platform de votre plateforme dans le Portail de Services Clients de Red Hat.**
   Connectez-vous au Portail de service à la clientèle de Red Hat à l'adresse suivante https://access.redhat.com. Grâce aux menus en haut de la page, sélectionner **`Downloads`**, **`JBoss Enterprise Middleware`**,**`Downloads`**. Sélectionner **`Application Platform`** à partir du menu déroulant. Un autre menu déroulant apparaîtra. Sélectionner la version de plate-forme JBoss Enterprise Application Platform pour voir les téléchargements disponibles pour cette version.

2. **Sélectionner le binaire HTTPD de la liste.**
   Cherchez le binaire HTTPD pour votre système d'exploitation et votre architecture. Cliquer sur le lien **`Download`**. Un fichier ZIP qui contient la distribution HTTPD se télécharge dans votre ordinateur.

3. **Extraire le ZIP dans le système où le binaire HTTPD exécutera.**
   Extraire le fichier ZIP sur votre serveur web préféré, à un endroit de votre choix. Il est souvent judicieux de le placer dans le répertoire où vous avez installé la plate-forme JBoss Enterprise Application Platform, communément appelé *EAP_HOME*. Dans un tel cas, votre HTTPD se

situera dans ***EAP_HOME*/httpd/**. Vous pourrez utiliser dès maintenant cet emplacement pour *HTTPD_HOME*, comme c'est le cas dans les autres documentations JBoss Enterprise Application Platform.

4. **Confuguration du HTTPD.**
   Configurer le démon HTTPD pour répondre aux besoins de votre organisation. Vous pouvez utiliser la documentation disponible auprès de la fondation Apache à http://httpd.apache.org/ pour vous guider.

5. **Démarrer le HTTPD.**
   Démarrer le HTTPD par la commande suivante :

   ```
   EAP_HOME/sbin/apachectl start
   ```

6. **Stopper le HTTPD.**
   Pour stopper le HTTP, lancer la commande suivante :

   ```
   EAP_HOME/sbin/apachectl stop
   ```

Report a bug

### 16.3.3. Configuration mod_cluster sur httpd

**Résumé**

mod_cluster est un équilibreur de charges basé httpd. Il utilise un réseau de communication pour envoyer des requêtes de httpd vers un groupe de noeuds de serveur d'application. Les dérivatifs suivants peuvent être définis pour configurer mod cluster sur httpd.

**NOTE**

Il n'est nul besoin d'utiliser les directives ProxyPass car mod_cluster configure automatiquement les URL qui doivent être envoyés à JBossWEB.

**Tableau 16.2. Dérivatifs mod_cluster**

| Dérivatif | Description | Valeurs |
|---|---|---|

| Dérivatif | Description | Valeurs |
| --- | --- | --- |
| CreateBalancers | Définit comment les équilibreurs de charge sont créés dans les hôtes virtuels httpd. Cela active les directives comme : **`ProxyPass /balancer://mycluster1/`** . | 0: Créer tous les hôtes virtuels httpd<br><br>1: Ne pas créer d'équilibreurs (vous aurez besoin d'un ProxyPass ou d'un ProxyMatch au moins pour définir les noms des équilibreurs)<br><br>2: Ne créer que le serveur principal<br><br>Par défaut: 2<br><br>  <br>**NOTE**<br><br>Si vous utilisez la valeur 1, n'oubliez pas de configurer l'équilibreur dans la directive ProxyPass, car la valeur par défaut correspond à une session sticky vide. De plus **`nofailover=Off`** et les valeurs reçues via message MCMP CONFIG sont ignorées. |
| UseAlias | Vérifier que l'alias corresponde bien au nom du serveur. | 0: Ignorer les alias<br><br>1: Vérifier les alias<br><br>Par défaut: 0 |
| LbstatusRecalTime | Intervalle d'équilibrage de charge (en secondes) de la logique pour recalculer le statut d'un noeud. | Par défaut: 5 secondes |
| WaitForRemove | Durée en secondes avant qu'un noeud supprimé soit oublié par httpd. | Par défaut: 10 secondes |

| Dérivatif | Description | Valeurs |
|---|---|---|
| ProxyPassMatch/ProxyPass | ProxyPassMatch et ProxyPass sont des directives mod_proxy qui, quand on utilise **!** (à la place de l'url de back-end), évitent un proxy inverse sur le chemin d'accès. Utilisé pour autoriser httpd à fournir des informations statiques comme des images. Ainsi,<br>**ProxyPassMatch ^(/.*\.gif)$ !**<br>L'exemple ci-dessus permet à httpd de servir les fichiers .gif directement. | |

**mod_manager**

Le contexte d'une directive mod_manager est l'hôte virtuel dans tous les cas, sauf contre indication. Le contexte **server config** implique que la directive doit se trouver en dehors d'une configuration de l'hôte virtuel. Si tel n'est pas le cas, un message erreur apparaîtra et httpd ne démarrera pas.

**Tableau 16.3. Dérivatifs mod_manager**

| Dérivatif | Description | Valeurs |
|---|---|---|
| EnableMCPMReceive | Autorise l'hôte virtuel à recevoir MCPM des noeuds. Inclure EnableMCPMReceive dans la configuration httpd pour permettre au mod_cluster de fonctionner. Le sauvegarder dans l'hôte virtuel de configuration advertise. | |
| MemManagerFile | Le nom de base pour les noms que mod_manager utilise pour stocker la configuration, générer des clés pour mémoire partagée ou fichiers verrouillés. Ce doit être un nom de chemin d'accès absolu; les répertoires seront créés si nécessaire. Il est recommandé que ces fichiers soient placés sur un lecteur local et non pas en NFS share.<br>Context: config serveur | **$server_root/logs/** |
| Maxcontext | Le nombre maximum de contextes pris en charge par mod_cluster<br>Context: config serveur | Par défaut: 100 |

| Dérivatif | Description | Valeurs |
| --- | --- | --- |
| Maxnode | Le nombre maximum de noeuds supportés par le mod_cluster. Context: config serveur | Par défaut: 20 |
| Maxhost | Le nom maximum d'hôtes (alias) supportés par mod_cluster. Inclut également le nombre maximum d'équilibreurs de charge. Context: config serveur | 10 |
| Maxsessionid | Le nombre d'ID de session actifs stockés afin de procurer le nombre de sessions actives du gestionnaire mod_cluster-manager. Une session est inactive quand mod_cluster ne reçoit aucune information de la session pendant 5 minutes.<br><br>Context: config serveur<br><br>Ce champ est à but de démonstration et débogage uniquement. | 0: la logique n'est pas activée. |
| MaxMCMPMaxMessSize | Taille maximum des messages MCMP en provenance d'autres directives max | Calculé sur la base d'autres directives max. Min: 1024 |
| ManagerBalancerName | Le nom que l'équilibreur des charges utilise quand JBoss AS/JBossWeb/Tomcat ne fournit pas de nom d'équilibreur. | `mycluster` |
| PersistSlots | Indique à mod_slotmem de persister les noeuds, les alias et les contextes dans des fichiers. Context: config serveur | Off |
| CheckNonce | Contrôle de la vérification de la valeur unique avec le gestionnaire mod_cluster=manager. | on/off<br><br>Par défaut: on - Nonce checked |
| AllowDisplay | Contrôle des affichages supplémentaires sur la page principale du mod_cluster-manager. | on/off<br><br>Par défaut: off - only version is displayed |

| Dérivatif | Description | Valeurs |
|---|---|---|
| AllowCmd | Autorise les commandes qui utilisent l'URL mod_cluster-manager. | on/off<br><br>Par défaut: on - Commands allowed |
| ReduceDisplay | Réduit le montant d'informations affichées sur la page principale de mod_cluster-manager, afin qu'un plus grand nombre de noeuds puissent être affichés sur la page. | on/off<br><br>Default: off - full information is displayed |
| SetHandler mod_cluster-manager | Affiche des informations sur le nœud que mod_cluster voit dans le cluster. L'information comprend des informations génériques et compte aussi le nombre de sessions actives.<br><br>`<Location /mod_cluster-manager>`<br>`SetHandler mod_cluster-manager`<br>`Order deny,allow`<br>`Allow from 127.0.0.1`<br>`</Location>` | on/off<br><br>Default: off |

**NOTE**

Quand on accède à l'emplacement défini dans httpd.conf :

Transferred: Correspond aux données POST envoyées du serveur de back-end.

Connected: Correspond au nombre de requêtes traitées au moment où la page de statuts du mod_cluster a été demandée.

Num_sessions: Correspond au nombre de sessions que le rapport mod_cluster a reporté comme étant inactives (sur lesquelles il y a eu une requête au cours des 5 dernières minutes). Ce champ n'est pas présent quand Maxsessionid est égal à zéro. Ce champ est à but de démonstration et de débogage uniquement.

Report a bug

### 16.3.4. Utiliser un HTTPD externe comme Web frontal pour la plate-forme JBoss EAP

### Aperçu

Pour comprendre les raisons d'utiliser un service HTTPD externe comme le serveur web frontal, et pour connaître les avantages et inconvénients des différents connecteurs HTTP pris en charge par JBoss Enterprise Application Platform, consulter Section 16.1.3, « Connecteurs HTTP - Aperçu général ». Dans certaines situations, vous pouvez utiliser l'HTTPD de votre système d'exploitation. Sinon, vous pouvez utiliser l'HTTPD qui est fourni par le serveur JBoss Enterprise Web.

Une fois que vous aurez décidé quel HTTPD ou quel connecteur HTTP utiliser, voir une des procédures suivantes :

- Section 16.3.2, « Installer Apache HTTPD inclus avec JBoss Enterprise Application Platform 6 »
- Section 16.5.3, « Installer le Module mod cluster dans Apache HTTPD ou dans JBoss Enterprise Web Server HTTPD »
- Section 16.6.3, « Installer le Module_jk_mod dans Apache HTTPD ou dans JBoss Enterprise Web Server HTTPD »
- Section 16.8.2, « Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI Redirector »
- Section 16.9.2, « Configurer le connecteur NSAPI dans Oracle Solaris »
- Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe »

Report a bug

## 16.3.5. Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe

### Aperçu

La plateforme EAP n'a pas besoin de savoir de quel proxy elle accepte les requêtes, uniquement le port et le protocole. Ce n'est pas le cas pour **`mod_cluster`**, qui est davantage lié à la configuration de JBoss Enterprise Application Platform. En revanche, les tâches suivantes fonctionnent pour **`mod_jk`**, **`mod_proxy`**, **`ISAPI`**, et **`NSAPI`**. Substituer les protocoles et les ports que vous aurez besoin de configurer par ceux des exemples.

Pour configurer JBoss Enterprise Application Platform pour **`mod_cluster`**, consulter Section 16.5.5, « Configurer un Worker Node de mod_cluster ».

### Prérequis

- Section 16.7.2, « Installer Mod_proxy HTTP Connector dans Apache HTTPD »
- Vous devrez être connecté au Management CLI ou à la Console de gestion pour effectuer cette tâche. Les étapes précises utilisent le Management CLI, mais la même procédure de base est utilisée dans la Console de gestion.
- Vous aurez besoin d'une liste des protocoles que vous devrez utiliser, que ce soit HTTP, HTTPS ou AJP.

**Procédure 16.3. Modifier la configuration et ajouter les liaisons de socket**

1. **Configurer les propriétés de système `jvmRoute` et `useJK`**

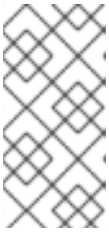
Par défaut, le jvmRoute est sur la même valeur que le nom du serveur. Si vous avez besoin de le personnaliser, vous pouvez utiliser une commande semblable à la suivante. Remplacer ou supprimer la partie de la commande **`/profile=ha`**, en fonction du profil ou si vous utilisiez un serveur autonome. Remplacez la chaîne **`CUSTOM_ROUTE_NAME`** par votre nom jvmRoute personnalisé.

```
[user@localhost:9999 /] /profile=ha/subsystem=web:write-
attribute(name="instance-id",value="CUSTOM_ROUTE_NAME")
```

Activer **`useJK`** en la définissant sur **`true`** grâce à la commande suivante :

```
[user@localhost:9999 /] /system-property=UseJK/:add(value=true)
```

2. **Lister les connecteurs disponibles dans le sous-sytème.**



**NOTE**

Cette étape est nécessaire uniquement si vous n'utilisez pas la configuration **`autonome-ha.xml`** pour un serveur autonome, ou les profils **`ha`** ou **`full-ha`** d'un groupe de serveurs dans un domaine géré. Ces configurations ont déjà tous les connecteurs nécessaires.

Pour qu'un service externe HTTPD puisse se connecter au serveur web de la plate-forme JBoss EAP, le sous-système web doit avoir un connecteur. Chaque protocole a besoin de son propre connecteur, lié à un groupe de sockets.

Pour avoir la liste des connecteurs actuellement disponibles, lancer la commande suivante:

```
[standalone@localhost:9999 /] /subsystem=web:read-children-
names(child-type=connector)
```

S'il n'y a aucune ligne sur le connecteur dont vous avez besoin (HTTP, HTTPS, AJP), vous devrez ajouter un connecteur.

3. **Lire la configuration d'un connecteur.**
Pour voir les détails de configuration d'un connecteur, vous pourrez lire sa configuration. La commande suivante lit la configuration du connecteur AJP. Les autres connecteurs ont des sorties de configuration semblables.

```
[standalone@localhost:9999 /] /subsystem=web/connector=ajp:read-
resource(recursive=true)
{
    "outcome" => "success",
    "result" => {
        "enable-lookups" => false,
        "enabled" => true,
        "max-post-size" => 2097152,
        "max-save-post-size" => 4096,
        "protocol" => "AJP/1.3",
        "redirect-port" => 8443,
        "scheme" => "http",
        "secure" => false,
        "socket-binding" => "ajp",
```

```
        "ssl" => undefined,
        "virtual-server" => undefined
    }
}
```

4. **Ajouter les connecteurs utiles au sous-système web.**
   Pour ajouter un connecteur au sous-système web, il doit y avoir une liaison de sockets. La liaison de sockets est ajoutée au groupe de liaisons de sockets utilisées par votre serveur ou groupe de serveurs. Les étapes suivantes supposent que votre groupe de serveurs est **`server-group-one`** et que votre groupe de liaisons de sockets est **`standard-sockets`**.

   a. **Ajouter une liaison au groupe de liaisons de sockets.**
      Pour ajouter un groupe de liaison de sockets, lancer la commande suivante, en remplaçant le protocole et le port par ceux dont vous avez besoin.

      ```
      [standalone@localhost:9999 /] /socket-binding-group=standard-
      sockets/socket-binding=ajp:add(port=8009)
      ```

   b. **Ajouter la liaison de sockets au sous-système web.**
      Lancer la commande suivante pour ajouter un connecteur au sous-système web, en substituant le nom de liaison de socket et le protocole par ceux que vous avez besoin.

      ```
      [standalone@localhost:9999 /]
      /subsystem=web/connector=ajp:add(socket-binding=ajp,
      protocol="AJP/1.3", enabled=true, scheme="http")
      ```

Report a bug

## 16.4. CLUSTERING

### 16.4.1. Utiliser la Communication TCP pour le sous-système de clusterisation

Par défaut, les nœuds de cluster surveillent leurs statuts respectifs avec le protocole UDP. Certains réseaux ne permettent que TCP. Dans ce cas, vous pouvez ajouter la pile de protocole **`TCPPING`** à votre configuration et l'utiliser comme mécanisme par défaut. Ces options de configuration sont disponibles en ligne de commandes avec le Management CLI.

Le sous-système **`mod_cluster`** utilise également la communication UDP par défaut, et vous pouvez opter pour TCP également.

Voir les deux procédures suivantes pour configurer les sous-systèmes mod_cluster et JGroups pour qu'ils puissent utiliser TCP pour la communication réseau :

- Section 16.4.2, « Configurer le sous-système JGroup pour Utilisation TCP »
- Section 16.4.3, « Désactiver les annonces dans le sous-système **`mod_cluster`**. »

Report a bug

### 16.4.2. Configurer le sous-système JGroup pour Utilisation TCP

Par défaut, le sous-système JGroups communique à l'aide de la multidiffusion UDP. Utilisez la procédure suivante pour configurer le sous-système JGroups pour qu'il puisse utiliser la monodiffusion TCP à la place.

Pour configurer le sous-système **mod_cluster** pour qu'il puisse utiliser TCP également, consulter Section 16.4.3, « Désactiver les annonces dans le sous-système **mod_cluster**. ».

1. **Exécuter le Management CLI**
   Lancer la Management CLI, avec la commande ***EAP_HOME*/bin/jboss-cli.sh** dans Linux ou bien la commande ***EAP_HOME*\bin\jboss-cli.bat** dans le serveur Microsoft Windows. Saisir **connect** pour connecter le contrôleur de domaine sur l'hôte local, ou **connect *IP_ADDRESS*** pour vous connecter à un contrôleur de domaines sur un serveur éloigné.

2. **Modifier le script suivant selon votre environnement.**
   Copiez le script suivant dans un éditeur de texte. Si vous utilisez un profil différent sur un domaine géré, changer le nom du profil. Si vous utilisez un serveur autonome, supprimer la portion **/profile=full-ha** des commandes. Modifiez les propriétés figurant au bas de la commande comme suit. Chacune de ces propriétés est facultative.

   **initial_hosts**

   Une liste des hôtes considérés comme connus, séparés par des virgules sera à votre disposition pour rechercher l'adhésion de départ.

   **port_range**

   Si vous le souhaitez, vous pouvez attribuer une plage de ports. Si vous affectez une plage de ports de 2, et que le port initial est 7600, alors TCPPING tentera de contacter chaque hôte sur les ports 7600-7601. Cette propriété est facultative.

   **timeout**

   Une valeur de timeout facultative, en millisecondes, pour les membres d'un cluster.

   **num_initial_members**

   Le nombre de noeuds avant qu'un cluster soit considéré comme complet. Cette propriété est facultative.

```
cd /profile=full-ha/subsystem=jgroups
./stack=tcpping:add
cd stack=tcpping
./transport=TRANSPORT:add(type=TCP,socket-binding=jgroups-tcp)
:add-protocol(type=TCPPING)
:add-protocol(type=MERGE2)
:add-protocol(type=FD_SOCK,socket-binding=jgroups-tcp-fd)
:add-protocol(type=FD)
:add-protocol(type=VERIFY_SUSPECT)
:add-protocol(type=BARRIER)
:add-protocol(type=pbcast.NAKACK)
:add-protocol(type=UNICAST2)
:add-protocol(type=pbcast.STABLE)
:add-protocol(type=pbcast.GMS)
:add-protocol(type=UFC)
:add-protocol(type=MFC)
:add-protocol(type=FRAG2)
```

```
:add-protocol(type=RSVP)
cd protocol=TCPPING
./property=initial_hosts/:add(value="HostA[7600],HostB[7600]")
./property=port_range/:add(value=0)
./property=timeout/:add(value=3000)
./property=num_initial_members/:add(value=3)
cd ../..
:write-attribute(name=default-stack,value=tcpping)
```

3. **Exécuter le script en mode lot.**

**AVERTISSEMENT**

Les serveurs qui exécutent le profil devront être fermés avant de pouvoir exécuter le fichier de commandes.

Sur invitation du Management CLI, saisir **batch** et appuyer sur la touche **Enter**. L'invite change pour inclure un signe (#) pour indiquer que vous êtes en mode lot. Cela vous permet d'entrer une série de commandes. Si l'une d'entre elles venait à échouer, toute l'opération serait annulée.

Coller le script modifié de l'étape précédente, ajouter une nouvelle ligne supplémentaire à la fin. Saisir **run-batch** pour exécuter le lot. Une fois que toutes les commandes sont exécutées, le message **The batch executed successfully** apparaîtra.

**Résultat**

La pile **TCPPING** est maintenant disponible pour les sous-systèmes JGroups. Si elle est utilisée, le sous-système JGroups utilisera TCP pour toute la communication de réseau. Pour configurer le sous-système **mod_cluster** à utiliser TCP également, consulter Section 16.4.3, « Désactiver les annonces dans le sous-système **mod_cluster**. ».

Report a bug

### 16.4.3. Désactiver les annonces dans le sous-système mod_cluster.

Par défaut, l'équilibreur du sous-système **mod_cluster** utilise l'UDP multidiffusion pour annoncer sa présence aux workers d'arrière-plan. Si vous le souhaiter, vous pouvez désactiver les annonces. Utiliser la procédure suivante pour configurer ce comportement.

**Procédure 16.4.**

1. **Modifier la configuration HTTPD.**
Modifier la configuration HTTPD pour désactiver la publicité serveur et pour utiliser un proxy à la place. La liste de proxys est configurée sur le worker, et contient tous les serveurs HTTPS activés-**mod_cluster** auxquels le worker peut parler.

La configuration de **mod_cluster** pour le serveur HTTPD se trouve généralement dans le répertoire **/etc/httpd/** ou **etc/httpd/** au sein de l'installation de HTTPD, s'il est installé dans un emplacement non standard. Reportez-vous à Section 16.5.3, « Installer le Module mod cluster dans Apache HTTPD ou dans JBoss Enterprise Web Server HTTPD » et Section 16.5.4,

« Configurer les propriétés Server Advertisement de votre HTTPD activé par un cluster » pour plus d'informations sur le fichier lui-même. Ouvrez le fichier contenant l'hôte virtuel qui écoute les requêtes MCPM (à l'aide de la directive **EnableMCPMReceive**) et désactiver le serveur d'annonces en remplaçant la directive **ServerAdvertise** comme suit.

```
ServerAdvertise Off
```

2. **Désactiver les annonces dans le sous-système mod_cluster de JBoss Enterprise Application Platform, et fournir une liste de proxys.**
Vous pouvez désactiver les annonces du sous-système **mod_cluster** et fournir une liste de console de gestion basée web ou le Management CLI de lignes de commande. La liste de proxys est utile car le sous-système **mod_cluster** ne sera pas en mesure de découvrir les proxys automatiquement si les annonces sont désactivées.

   - **Console de gestion**

     1. Si vous utilisez un domaine géré, vous ne pourrez uniquement configurer que **mod_cluster** dans les profils où il est activé, tels que **ha** et **full-ha**.

     2. Connectez-vous à la Console de gestion et sélectionner l'étiquette **Profiles** en haut et à droite de l'écran. Si vous utilisez un domaine géré, sélectionner le profil **ha** ou **full-ha** à partir de la boîte de sélection **Profiles** en haut et à gauche de la page **Profiles**.

     3. Cliquer sur le menu **Subsystems** pour l'étendre. Étendre le sous-menu **Web**, et sélectionner **Modcluster**.

     4. Cliquer sur le bouton **Edit** en haut, pour modifier les options qui s'appliquent à tout le sous-système **mod_cluster**. Modifier la valeur de **Advertise** sur **false**. Utilisez le bouton **Save** pour enregistrer les paramètres.

     5. Cliquer sur l'onglet **Proxies** près du bas de l'écran. Cliquer sur le bouton **Edit** dans la sous-page de **Proxies** et entrez une liste de serveurs proxys. La syntaxe correcte est une liste séparée par des virgules de chaînes **HOSTNAME:PORT**, semblable à la suivante.

        ```
        10.33.144.3:6666,10.33.144.1:6666
        ```

        Cliquer sur le bouton **Save** pour enregistrer les modifications.

   - **Management CLI**
     Les deux commandes de Management CLI suivantes créent la même configuration que les instructions de la Console de gestion ci-dessus. Elles supposent que vous exécutez un domaine géré et que votre groupe de serveurs utilise le profil **full-ha**. Si vous utilisez un profil différent, modifiez son nom dans les commandes. Si vous utilisez un serveur autonome à l'aide du profil **standalone-ha** profil, supprimez la portion **/profile=full-ha** des commandes.

     ```
     /profile=full-ha/subsystem=modcluster/mod-cluster-
     config=configuration/:write-attribute(name=advertise,value=false)

     /profile=full-ha/subsystem=modcluster/mod-cluster-
     config=configuration/:write-attribute(name=proxy-
     list,value="10.33.144.3:6666,10.33.144.1:6666")
     ```

**Résultat**

Le sous-système **mod_cluster** n'annonce plus sa disponibilité.

Report a bug

## 16.5. WEB, CONNECTEURS HTTP, ET HTTP CLUSTERING

### 16.5.1. Le connecteur HTTP mod_cluster

*mod_cluster* est le module qui permet l'équilibrage des charges dans le conteneur de JBoss Web. On l'appelle le *connector*. Le choix de connecteur dépend du conteneur web que vous choisissez d'utiliser avec la plate-forme JBoss Enterprise Application. Pour en savoir plus sur les autres connecteurs, voir ce qui suit :

- Section 16.6.1, « Le connecteur Apache mod_HTTP »
- Section 16.8.1, « Internet Server API (ISAPI) HTTP Connector »
- Section 16.9.1, « Netscape Server API (NSAPI) HTTP Connector »

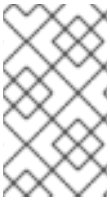
Le mod_cluster possède plusieurs avantages.

- *mod_cluster Management Protocol (MCMP)* est un lien supplémentaire entre les nœuds de serveur d'applications et httpd, utilisé par les nœuds de serveur d'applications pour transmettre des facteurs d'équilibrage des charges côté serveur et des événements de cycle de vie de retour vers le conteneur web via un ensemble personnalisé de méthodes HTTP.
- La configuration dynamique des proxies HTTPS permet à la plateforme JBoss Enterprise Application Platform de s'adapter sur le champ, sans besoin de configuration supplémentaire.
- Les serveurs d'application se chargent des calculs de factorisation d'équilibre des charges.
- mod_cluster offre un contrôle précis du cycle de vie. Chaque serveur transmet tout événement de cycle de vie du contexte d'application web au proxy, l'informant de démarrer ou d'arrêter les demandes de routage dans un contexte donné du serveur. Ceci empêche les utilisateurs finaux de voir les erreurs 404 en raison des ressources non disponibles.
- AJP est optionel. Apache HTTP exige l'utilisation d'AJP, mais mod_cluster peut utiliser HTTP, HTTPS, ou AJP.

Report a bug

### 16.5.2. Configurer le sous-système mod_cluster

Dans la Console de gestion sur le web, les options de **mod_cluster** sont disponibles dans la zone de configuration du sous-système Web. Cliquer sur l'onglet **Profiles** en haut à gauche. Si vous utilisez un domaine géré, sélectionnez le bon profil pour configurer dans la boîte de sélection de **Profile** en haut et à droite. Par défaut, les profils **ha** et **full-ha** ont le sous-système **mod_cluster** activé. Si vous utilisez un serveur autonome, vous devez utiliser le profil **standalone-ha** pour démarrer le serveur. Cliquer sur l'élément **Web** dans le menu de gauche et choisissez **Modcluster** dans le sous-menu. Les options sont expliquées dans les tableaux ci-dessous. La configuration générale est indiquée en premier, suivie de la configuration de sessions, les contextes web, proxy, SSL et réseautage. Chacune d'elles possède son propre onglet dans l'écran de configuration de **Modcluster**



**NOTE**

**`Modcluster`** configuration page is only visible for profiles with the HA Clustering subsystem enabled. These profiles are **`ha`** and **`full-ha`** for a managed domain, or **`standalone-ha`** for a standalone server.

**Tableau 16.4. Options de configuration `mod_cluster`**

| Option | Description | Commande CLI |
|---|---|---|
| Groupe d'équilibrage des charges | Si non nulles, les requêtes devront être envoyées vers un groupe d'équilibrage de charges sur l'équilibreur de charges. Laisser cet espace vide si vous ne souhaitez pas utiliser ces groupes d'équilibrage des charges. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=load-balancing-group,value=myGroup)` |
| Équilibreur | Le nom de l'équilibreur. Doit correspondre à la configuration du proxy HTTPD. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=balancer,value=myBalancer)` |
| Socket Advertise | Le nom de la liaison de sockets à utiliser pour les annonces de cluster. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise-socket,value=modcluster)` |
| Clé Sécurité Advertise | Une chaîne contenant la clé de sécurité à annonce. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise-security-key,value=myKey)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Adresse Groupe Advertise | Adresse UDP sur laquelle écouter les annonces multidiffusion proxy httpd | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise-group-address,value=224.0.1.105)` |
| Port Advertise | Port UDP sur lequel écouter les annonces multidiffusion proxy httpd | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise-port,value=23364)` |
| Fabrique des threads Advertise | La fabrique de threads utilisée pour créer le listener d'annonces d'arrière plan. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise-thread-factory,value=Executors.defaultThreadFactory())` |
| Advertise | Indique si les annonces sont activées. Valeur par défaut **`true`**. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=advertise,value=true)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| JVM Route Factory | La fabrique qui détermine la stratégie qui détermine la route JVM Route d'un noeud, si non indiqué dans **`server.xml`**. La fabrique par défaut commence par consulter la propriété système **`jboss.mod_cluster.jvmRoute`**. Si cette propriété système est pas définie, JVM Route reçoit un UUID. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=jvm-route-factory,value=new SystemPropertyJvmRouteFactory(new UUIDJvmRouteFactory(), "jboss.mod_cluster.jvmRoute"))` |

**Tableau 16.5. Options de configuration de session `mod_cluster`**

| Option | Description | Commande CLI |
| --- | --- | --- |
| Sticky Session | Si vous souhaitez utiliser des sessions pour les demandes. Cela signifie qu'après que le client ait établi une connexion sur un nœud de cluster spécifique, leur transmission ultérieure est routée vers ce même nœud à moins qu'il ne soit plus disponible. La valeur par défaut est **`true`**, qui est le paramètre recommandé. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=sticky-session,value=true)` |
| Sticky Session Force | Si sur **`true`**, la demande ne sera pas redirigée vers un nouveau nœud de cluster si son nœud initial n'est plus disponible. Au lieu de cela, elle échouera. La valeur par défaut est **`false`**. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=sticky-session-force,value=false)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Sticky Session Remove | Supprime les informations de session en cas d'échec. Désactivé par défaut. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=sticky-session-remove,value=false)` |

**Tableau 16.6. Options de configuration de contexte web `mod_cluster`**

| Option | Description | Commande CLI |
| --- | --- | --- |
| Auto Enable Contexts | Si on doit ajouter de nouveaux contextes à **`mod_cluster`** par défaut ou non. La valeur par défaut **`true`**. Si vous modifiez la valeur par défaut et que vous devez activer le contexte manuellement, l'application web peut activer son contexte à l'aide de la méthode MBean **`enable()`**, ou via le gestionnaire **`mod_cluster`**, une application web qui s'exécute sur le serveur proxy HTTPD, sur un hôte virtuel nommé ou le port qui est spécifié dans la configuration de cet HTTPD. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=auto-enable-contexts,value=true)` |
| Excluded Contexts | Une liste séparée par des virgules de contextes que **`mod_cluster`** doit ignorer. Si aucun hôte n'est indiqué, l'hôte est censé être **`localhost`**. **`ROOT`** indique le contexte racine de l'application web. La valeur par défaut est **`ROOT,invoker,jbossws,juddi,console`**. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=excluded-contexts,value="ROOT,invoker,jbossws,juddi,console")` |

**Tableau 16.7. Options de configuration proxy de `mod_cluster`**

| Option | Description | Commande CLI |
| --- | --- | --- |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Proxy URL | Si défini, cette valeur sera ajoutée à l'URL des commandes MCMP. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=proxy-url,value=myhost)` |
| Proxy List | Une liste séparée par des virgules des adresses proxy HTTPD, dans le format **`hostname:port`**. Ceci indique la liste des serveurs proxy avec lesquels le processus de **`mod_cluster`** va tenter de communiquer au départ. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=proxy-list,value="127.0.0.1,127.0.0.2")` |

### Configurer la Communication SSL pour `mod_cluster`

Par défaut, la communication **`mod_cluster`** a lieu sur un lien HTTP crypté. Si vous définissez le schéma du connecteur à **`HTTPS`** (voir Tableau 16.5, « Options de configuration de session **`mod_cluster`** »), les paramètres ci-dessous indiquent à **`mod_cluster`** où trouver les informations pour encoder la connexion.

**Tableau 16.8. Options de configuration SSL de `mod_cluster`**

| Option | Description | Commande CLI |
| --- | --- | --- |
| ssl | Indique si on doit activer SSL. Valeur par défaut **`false`**. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ssl,value=true)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Clé Alias | Clé alias choisie quand le certificat est créé. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=key-alias,value=jboss)` |
| Key Store | L'emplacement où le keystore garde les certificats clients | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=key-store,value=System.getProperty("user.home") + "/.keystore")` |
| Key Store Type | Le type de keystore | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=key-store-type,value=JKS)` |
| Key Store Provider | Le fournisseur de keystore | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=key-store-provider,value=IBMJCE)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Mot de passe | Mot de passe choisi quand le certificat est créé. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=password,value=changeit)` |
| Trust Algorithm | L'algorithme de la fabrique de gestionnaire de confiance | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=trust-algorithm,value=PKIX)` |
| Cert File | L'emplacement du fichier de certificats. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ca-certificate-file,value=${user.home}/jboss.crt)` |
| CRL File | Fichier de la liste de révocation du certificat. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ca-crl-file,value=${user.home}/jboss.crl)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Max Certificate Length | La longueur maximum du certificat contenue dans le trust store. Valeur par défaut 5. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=trust-max-cert-length,value=5)` |
| Key File | L'emplacement du fichier clé du certificat. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=certificate-key-file,value=${user.home}/.keystore)` |
| Cipher Suite | La suite cipher d'encodage autorisée. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=cipher-suite,value=ALL)` |
| Certificate Encoding Algorithms | L'algorithme de la fabrique de gestionnaire de clés. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=encoding-algorithms,value=ALL)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Revocation URL | L'URL de la liste de révocation de l'autorité de certificat | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=ca-revocation-url,value=jboss.crl)` |
| Protocole | Les protocoles SSL activés. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/ssl=configuration/:write-attribute(name=protocol,value=SSLv3)` |

### Configurer les options de réseautage de `mod_cluster`

Les options de réseautage de **`mod_cluster`** contrôlent des comportements de timeout différents pour des types de services variés avec lesquels le service **`mod_cluster`** communique.

**Tableau 16.9. Options de configuration de réseautage de `mod_cluster`**

| Option | Description | Commande CLI |
| --- | --- | --- |
| Node Timeout | Timeout (en secondes) des connexions de proxy vers un nœud. C'est que le temps **`mod_cluster`** attendra la réponse de dorsal avant de retourner l'erreur. Qui correspond au délai d'attente dans la documentation de mod_proxy de travailleur. La valeur **`-1`** n'indique aucun délai d'attente. Notez que **`mod_cluster`** utilise toujours un cping/cpong avant d'adresser une demande et la valeur **`connectiontimeout`** utilisée par **`mod_cluster`** est la valeur de ping. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=node-timeout,value=-1)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Socket Timeout | Nombre de millisecondes pendant lesquelles patienter avant d'obtenir une réponse d'un proxy httpd à des commandes MCMP avant le timeout, et indiquer erreur de proxy. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=socket-timeout,value=20)` |
| Stop Context Timeout | Durée, mesurée dans les unités spécifiées par stopContextTimeoutUnit, pendant laquelle attendre l'arrêt net d'un contexte (fin des demandes en attente pour un contexte distribuable; ou destruction/expiration des sessions actives pour un contexte non distribuable). | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=stop-context-timeout,value=10)` |
| Session Draining Strategy | Indique si on doit drainer les sessions avant de retirer le déploiement d'une application web.<br>**DEFAULT**<br>Sessions de drainage avant qu'une application web retire son déploiement si l'application web n'est pas distribuable.<br>**ALWAYS**<br>Toujours drainer les sessions avant le retrait du déploiement d'une application web, même pour les applications web distribuables.<br>**NEVER**<br>Ne pas drainer les sessions avant le retrait du déploiement d'une application web, même pour les applications web non distribuables. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=session-draining-strategy,value=DEFAULT)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Max Attempts | Nombre de fois qu'un proxy HTTPD va tenter d'envoyer une requête donnée à un worker avant d'abandonner. La valeur minimale est **1**, ce qui signifie essayer une seule fois. La valeur par défaut du **module mod_proxy** est également 1, ce qui signifie qu'aucune nouvelle tentative ne se produit. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=max-attempts,value=1)` |
| Flush Packets | Indique si on doit activer le vidage des paquets dans le serveur HTTPD. Valeur par défaut **false**. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=flush-packets,value=false)` |
| Flush Wait | Durée, en secondes, pendant laquelle on doit attendre le vidage des paquets dans le serveur HTTPD. Une valeur **-1**. A value of **-1** indique une attente indéfinie avant de vider les paquets. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=flush-wait,value=-1)` |
| Ping | Durée, en secondes, pendant laquelle attendre un réponse au ping d'un noeud de cluster. Valeur par défaut **10** secondes. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=ping,value=10)` |
| SMAX | Le nombre de soft connexions inactives maximales (le même que **smax** dans la documentation du module de worker mod_proxy). La valeur maximale dépend de la configuration de thread httpd et peut être **ThreadsPerChild** ou **1**. | `profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=smax,value=ThreadsPerChild)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| TTL | Time To live (en secondes) pour les connexions inactives au dessus de smax, la valeur par défaut est 60<br><br>Quand **nodeTimeout** n'est pas défini, le **Proxy** de la directive **ProxyTimeout** est utilisé. Si **ProxyTimeout** n'est pas défini, alors le **Timeout** sera utilisé. La valeur par défaut est de 300 secondes. **nodeTimeout**, **ProxyTimeout**, et **Timeout** sont définis au niveau socket. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=ttl,value=-1)` |
| Node Timeout | La durée d'attente, en secondes, pour le traitement d'une requête par un processus de travail de serveur HTTPD externe. Par défaut, **-1**, ce qui signifie que mod_cluster attend indéfiniment la requête traitée par le worker HTTPD. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:write-attribute(name=node-timeout,value=-1)` |

### Options de configuration de Load provider de `mod_cluster`

Les options de configuration de **mod_cluster** suivantes ne sont pas disponibles dans la Console de gestion basée web, et ne peuvent être uniquement définies qu'en utilisant le Management CLI en ligne de commandes.

Le processeur de charge simple est utilisé si aucun processeur de charge dynamique n'est pas présent. Il donne à chaque membre du cluster un facteur de charge **1**, et répartit uniformément les travaux sans prendre en compte un algorithme d'équilibrage de charges. Pour l'ajouter, utilisez la commande CLI suivante : **/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/simple-load-provider:add**

Un fournisseur de charge dynamique peut être configuré pour utiliser une variété d'algorithmes, en combinaison, pour déterminer quel nœud de cluster recevra la demande suivante. Le fournisseur de charge dynamique par défaut utilise **busyness** (niveau d'activité) comme facteur déterminant. Vous trouverez ci-dessous la liste des facteurs possibles. Vous pouvez également créer votre propre fournisseur de charge en fonction de votre propre environnement. Les options suivantes du fournisseur de charge dynamique peuvent être modifiées. Notez que vous pouvez avoir plus d'un facteur (métrique) à tout moment. Il vous suffira de les ajouter par l'interface CLI. Cliquez .

```
:add-metric(type=cpu)
```

**Tableau 16.10. Options Load provider dynamque de `mod_cluster`**

| Option | Description | Commande CLI |
| --- | --- | --- |
| Decay | Le facteur par lequel les métriques historiques se désintègrent de façon significative. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/:write-attribute(name=decay,value=2)` |
| Historique | Le nombre d'enregistrements de métriques de charge historique à considérer pour déterminer la charge. | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/:write-attribute(name=history,value=9)` |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Métrique de charge | Le seule métrique de charge inclus dans le fournisseur de charge dynamique de JBoss Enterprise Application Platform 6 est **busyness** (niveau d'activité), qui tente d'envoyer chaque nouvelle requête au worker le moins occupé. Vous pouvez définir la capacité de votre worker (**1** indique une capacité de 100 %) et le poids accordé au métrique de **busyness** global. . | `/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=busyness/:write-attribute(name=capacity,value=1.0)`<br>`/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=busyness/:write-attribute(name=type,value=busyness)`<br>`/profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/dynamic-load-provider=configuration/load-metric=busyness/:write-attribute(name=weight,value=1)` |

**Algorithmes de métriques de charge**

**cpu**

Le métrique de charge cpu utilise la charge CPU moyenne pour déterminer quel nœud de cluster reçoit la charge de travail suivante.

**mem**

Le métrique de charge mem utilise la mémoire native RAM comme facteur de charge. L'utilisation de ce métrique est déconseillée car elle fournit une valeur qui inclut les tampons et le cache. C'est donc toujours un chiffre très faible sur chaque système décent pourvu d'une bonne gestion de mémoire.

**heap**

La métrique de charge de tas utilise l'usage de tas pour déterminer quel cluster reçoit la charge de travail suivante.

**sessions**

Le métrique de charge de session utilise le nombre de sessions actives comme métrique.

**requêtes**

Le métrique de charge de requêtes utilise le nombre de requêtes en provenance des clients pour déterminer quel nœud de cluster reçoit la charge de travail suivante. Par exemple, capacité 1000 signifie que 1000 requêtes/s est considéré comme une « pleine charge ».

**send-traffic**

Le métrique de charge send-traffic (trafic envoyé) utilise le volume de trafic envoyé à partir d'un nœud de worker vers les clients. Par ex. une capacité par défaut de 512 indique que le nœud doit être considéré en pleine charge, si le trafic sortant moyen est 512 KB/s ou supérieur.

**receive-traffic**

Le métrique de charge receive-traffic (réception de trafic) utilise le volume de trafic envoyé vers le nœud de worker en provenance des clients. Par ex. une capacité par défaut de 1024 indique que le nœud doit être considéré en pleine charge, si le trafic entrant moyen est 1024 KB/s ou supérieur.

**busyness**

Ce métrique représente le nombre de threads d'une pool de threads en train de répondre à des requêtes.

**Exemple 16.1. Définir un métrique d'équilibrage de charges**

```
/profile=full-ha/subsystem=modcluster/mod-cluster-
config=configuration/dynamic-load-provider=configuration/load-
metric=cpu/:write-attribute(name="weight",value="3")
```

Report a bug

### 16.5.3. Installer le Module mod cluster dans Apache HTTPD ou dans JBoss Enterprise Web Server HTTPD

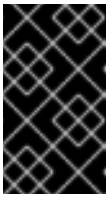
**Prérequis**

- Pour cette tâche, vous devrez utiliser Apache HTTPD installé dans Red Hat Enterprise Linux 6, ou JBoss Enterprise Web Server, ou encore HTTPD autonome comme composant de JBoss EAP 6 à télécharger séparément.
- Si vous avez besoin d'installer Apache HTTPD dans Red Hat Enterprise Linux 6, utiliser les instructions dans *Red Hat Enterprise Linux 6 Deployment Guide*, qui sont disponibles à partir de https://access.redhat.com/site/documentation/.
- Si vous avez besoin d'installer HTTPD autonome en tant que composant téléchargeable de JBoss Boss Enterprise Application Platform 6, consulter Section 16.3.2, « Installer Apache HTTPD inclus avec JBoss Enterprise Application Platform 6 ».

- Si vous avez besoin d'installer le serveur JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide*, qui sont disponibles à partir de https://access.redhat.com/site/documentation/.
- Télécharger le package **`Webserver Connecter Natives`** pour votre système d'exploitation et architecture depuis le portail client de Red Hat à https://access.redhat.com. Ce paquet contient les modules HTTPD mod_cluster binaires précompilés pour votre système d'exploitation. Après avoir extrait l'archive, les modules se trouvent dans le répertoire **`modules/native/lib/httpd/modules/`**. Le répertoire **`etc/`** contient quelques exemples de fichiers de configuration et le répertoire **`share/`** contient une documentation supplémentaire.
- Vous devez être connectés avec des privilèges administratifs (root).

**Procédure 16.5. Installer le Module mod cluster**

1. **Déterminer l'emplacement de votre configuration HTTPD**
   Votre emplacement de configuration HTTPD sera différent selon que vous utilisez Apache HTTPD de Red Hat Enterprise Linux, HTTPD autonome inclus comme composant séparé téléchargeabl dans JBoss Enterprise Application Platform 6 ou HTTPD disponible dans JBoss Enterprise Web Server. C'est l'une des trois options suivantes qui sera mentionnée au cours de cette tâche sous le nom *HTTPD_HOME*.
   - Apache HTTPD - **`/etc/httpd/`**

     **IMPORTANT**

     Malgré que les fichiers supplémentaires habituels de configuration dans **`conf.d/`**, les instructions suivantes ne fonctionneront pas correctement à moins que le répertoire **`HTTPD_HOME/conf/`** ne soit utilisé.
   - JBoss Enterprise Application Platform HTTPD - Cet emplacement est choisi par vous-même sur la base des exigences de votre infrastructure.
   - JBoss Enterprise Web Server HTTPD - ***`EWS_HOME`*`/httpd/`**
2. **Copier les modules dans le répertoire de modules HTTPD.**
   Copier les quatre modules (les fichiers qui se terminent par **`.so`**) à partir du répertoire **`modules/native/lib/httpd/modules/`** de l'archive extraite Webserver Natives vers le répertoire ***`HTTPD_HOME`*`/modules/`**.
3. **Pour JBoss Enterprise Web Server, désactiver le module `mod_proxy_balancer`.**
   Si vous utilisez JBoss Enterprise Web Server, le module **`mod_proxy_balancer`** sera activé par défaut. Il est incompatible avec mod cluster. Pour le désactiver, modifier ***`HTTPD_HOME`*`/conf/httpd.conf`** et décommenter la ligne suivante en mettant le symbole # (hachage) devant la ligne qui charge le module. La ligne apparaîtra sans le commentaire, puis avec, comme ci-dessous.

   ```
   LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
   ```

   ```
   # LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
   ```

   Sauvegarder et fermer le fichier.

4. **Configurer le module mod cluster.**

   a. Ouvrir ***HTTPD_HOME*/conf/httpd.conf** dans un éditeur de texte et ajouter ce qui suit en fin de fichier :

   ```
   # Include mod_cluster's specific configuration file
   Include conf/JBoss_HTTP.conf
   ```

   Sauvegarder et sortir du fichier.

   b. Créer un nouveau fichier nommé ***HTTPD_HOME*/httpd/conf/JBoss_HTTP.conf** et y ajouter ce qui suit.

   ```
   LoadModule slotmem_module modules/mod_slotmem.so
   LoadModule manager_module modules/mod_manager.so
   LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
   LoadModule advertise_module modules/mod_advertise.so
   ```

   Cela oblige Apache HTTPD à charger les modules dont **mod_cluster** a besoin automatiquement pour fonctionner.

5. **Créer un proxy de listener de serveur.**
   Continuer à éditer ***HTTPD_HOME*/httpd/conf/JBoss_HTTP.conf** et ajouter la configuration minimale suivante, en remplaçant les valeurs en lettres majuscules par des valeurs adaptées à votre système.

   ```
   Listen IP_ADDRESS:PORT
   <VirtualHost IP_ADDRESS:PORT>
      <Location />
             Order deny,allow
             Deny from all
             Allow from *.MYDOMAIN.COM
      </Location>

      KeepAliveTimeout 60
      MaxKeepAliveRequests 0
      EnableMCPMReceive On

      ManagerBalancerName mycluster
      ServerAdvertise On

   </VirtualHost>
   ```

   Ces directives créent un nouveau serveur virtuel qui écoute sur le port **IP_ADDRESS:PORT**, permet des connexions de **MYDOMAIN.COM** et se présente comme un équilibreur de charge du nom **mycluster**. Ces directives sont traitées en détail dans la documentation pour le serveur Web Apache Server. Pour en savoir plus sur les directives **ServerAdvertise** et **EnableMCPMReceive** ou les implications des annonces de serveur, consulter Section 16.5.4, « Configurer les propriétés Server Advertisement de votre HTTPD activé par un cluster ».

   Sauvegarder le fichier et sortir.

6. **Redémarrer HTTPD.**

La façon de redémarrer HTTPD dépend de savoir si vous utilisez Apache HTTPD de Red Hat Enterprise Linux ou le HTTDP inclus dans JBoss Enterprise Web Server. Choisir une des deux méthodes ci-dessous.

- **Red Hat Enterprise Linux 6 Apache HTTPD**
  Exécuter la commande suivante :

  ```
  [root@host]# service httpd restart
  ```

- **JBoss Enterprise Web Server HTTPD**
  JBoss Enterprise Web Server exécute à la fois sur Red Hat Enterprise Linux et Microsoft Windows Server. La méthode de redémarrage du HTTPD est différente pour chacun.

  - **Red Hat Enterprise Linux**
    Dans Red Hat Enterprise Linux, JBoss Enterprise Web Server installe son HTTPD en tant que service. Pour redémarrer HTTPD, lancer les deux commandes suivantes :

    ```
    [root@host ~]# service httpd stop[root@host ~]# service httpd start
    ```

  - **Microsoft Windows Server**
    Lancer les commandes suivantes dans une invite de commande avec des privilèges administratifs :

    ```
    C:\> net stop httpdC:\> net start httpd
    ```

**Résultat**

Apache HTTPD est maintenant configuré comme équilibreur de charges, et peut fonctionner avec le sous-système **`mod_cluster`** qui exécute sur JBoss EAP 6. Pour configurer JBoss Enterprise Application Platform pour qu'il soit au fait de mod_cluster, consulter Section 16.5.5, « Configurer un Worker Node de mod_cluster ».

Report a bug

### 16.5.4. Configurer les propriétés Server Advertisement de votre HTTPD activé par un cluster

**Résumé**

Pour obtenir des instructions sur la façon de configurer votre HTTPD pour qu'il intéragisse avec l'équilibreur de charges, consulter Section 16.5.3, « Installer le Module mod cluster dans Apache HTTPD ou dans JBoss Enterprise Web Server HTTPD ». L'élément de configuration *server advertisement* requiert davantage d'explications.

Quand Server Advertisement est inactif, HTTPD envoie des messages qui contiennent l'adresse IP et le numéro de port spécifié dans l'hôte virtuel du mod_cluster. Pour configurer ces valeurs, consulter Section 16.5.3, « Installer le Module mod cluster dans Apache HTTPD ou dans JBoss Enterprise Web Server HTTPD ». Si UDP multicast n'est pas disponible sur votre réseau, ou si vous préférez configurer les worker nodes avec une liste statique de serveurs proxy, vous pouvez désactiver Server Advertisement et configurer les noeuds proxy manuellement. Voir Section 16.5.5, « Configurer un Worker Node de mod_cluster » pour obtenir des informations sur la façon de configurer un worker node.

Vous devrez modifier le **httpd.conf** associé à votre instance d'HTTPD Apache. Il s'agit le plus souvent de **/etc/httpd/conf/httpd.conf** dans Red Hat Enterprise Linux, ou peut-être le répertoire **etc/** de votre instance HTTPD Apache autonome.

**Procédure 16.6. Modifier le fichier httpd.conf et implémenter les changements**

1. **Désactivez le paramètre AdvertiseFrequency, s'il existe.**
   Si vous voyez une ligne qui ressemble à ceci dans votre énoncé **<VirtualHost>**, décommenter cette ligne en ajoutant un signe # devant le premier caractère. La valeur ne devra pas correspondre à **5**.

   ```
   AdvertiseFrequency 5
   ```

2. **Ajouter la directive qui permet de désactiver Server Advertisement.**
   Ajouter la directive suivante dans l'énoncé **<VirtualHost>** afin de désactiver Server Advertisement.

   ```
   ServerAdvertise Off
   ```

3. **Actionner la possibilité de recevoir des messages MCPM.**
   Ajouter la directive suivante pour permettre au serveur HTTPD de recevoir des messages MCPM de la part des worker nodes.

   ```
   EnableMCPMReceive On
   ```

4. **Redémarrer le serveur HTTPD.**
   Redémarrer le serveur HTTPD en lançant une des commandes suivantes, selon que vous utilisez Red Hat Enterprise Linux ou Microsoft Windows Server.

   - **Red Hat Enterprise Linux**

     ```
     [root@host ]# service httpd restart
     ```

   - **Microsoft Windows Server**

     ```
     C:\> net service http
     C:\> net service httpd start
     ```

**Résultat**

Le démon HTTPD n'annonce plus l'adresse IP et le port de votre serveur proxy mod_cluster. Pour l'annoncer, vous devez configurer vos worker nodes pour qu'ils utilisent une adresse statique et un port pour communiquer avec le proxy. Consulter Section 16.5.5, « Configurer un Worker Node de mod_cluster » pour plus de détails.

Report a bug

## 16.5.5. Configurer un Worker Node de mod_cluster

Le master n'est configuré qu'une fois, par l'intermédiaire du sous-système de **mod_cluster**. Pour configurer le sous-système **mod_cluster**, reportez-vous à Section 16.5.2, « Configurer le sous-système **mod_cluster** ». Chaque worker node est configurée séparément, alors répétez cette procédure pour chaque nœud que vous souhaitez ajouter au cluster.

Si vous utilisez un domaine géré, chaque serveur de groupe de serveur est un worker node qui partage une configuration identique. Par conséquent, la configuration s'effectue sur un groupe de serveurs. Dans un serveur autonome, la configuration s'effectue sur une seule instance de JBoss Enterprise Application Platform. Les étapes de configuration sont sinon identiques.

**Configuration d'un worker node**

- Si vous utilisez un serveur autonome, il devra être démarré par le profile **`standalone-ha`**.
- Si vous utilisez un domaine géré, votre groupe de serveurs devra utiliser le profil **ha** ou **`full-ha`**, et le groupe de liaisons de sockets **`ha-sockets`** ou **`full-ha-sockets`**. JBoss Enterprise Application Platform est fournie avec un groupe de serveurs à clusterisation activée, nommé **`other-server-group`** qui remplit ces prérequis.

**NOTE**

Quand vous avez des commandes de Management CLI, celles-ci assument que vous utilisez un domaine géré. Si vous utilisez un serveur autonome, supprimer la portion **`/profile=full-ha`** des commandes.

**Procédure 16.7. Configurer un worker node**

1. **Configurer les interfaces de réseau**
   Les interfaces de réseau ont toutes la valeur **`127.0.0.1`** par défaut. Chaque hôte physique, qui accueille un serveur autonome ou bien un ou plusieurs serveurs au sein d'un groupe de serveurs, a besoin de ses interfaces configurées pour utiliser son adresse IP, que les autres serveurs peuvent apercevoir.

   Pour changer l'adresse IP d'un hôte de JBoss Enterprise Application Platform, vous devrez le fermer et modifier son fichier de configuration directement. C'est parce que l'API de gestion qui actionne la Console de gestion et le Management CLI se fie à une adresse de gestion stable.

   Suivez ces étapes pour changer l'adresse IP sur chaque serveur de votre cluster en votre adresse IP publique de master.

   a. Fermer le serveur complètement.

   b. Modifier soit **`host.xml`**, qui se trouve dans ***`EAP_HOME`*`/domain/configuration/`** pour un domaine géré, ou bien le fichier **`standalone-ha.xml`**, qui se trouve dans ***`EAP_HOME`*`/standalone/configuration/`** pour un serveur autonome.

   c. Chercher l'élément **`<interfaces>`**. Il y a trois interfaces configurées. Ces interfaces sont **`management`**, **`public`**, et **`unsecured`**. Pour chacune d'entre elles, changer la valeur **`127.0.0.1`** à l'adresse IP externe de l'hôte.

   d. Pour les hôtes qui participent à un domaine géré, mais qui ne sont pas master, localiser l'élément **`<host`**. Notez qu'il n'a pas de symbole de fermeture **`>`**, car il contient des attributs. Modifiez la valeur de son nom d'attribut **`master`** par un nom unique, avec un nom différent par esclave. Ce nom servira aussi à l'esclave pour identifier au cluster, donc notez bien ceci.

   e. Pour les hôtes nouvellement configurés qui ont besoin de rejoindre un domaine géré, chercher l'élément **`<domain-controller>`**. Dé-commenter ou supprimer l'élément **`<local />`**, et ajouter la ligne suivante, en changeant l'adresse IP (**`X.X.X.X`**) par l'adresse du contrôleur de domaine. Cette étape ne s'applique pas à un serveur autonome.

```
<remote host="X.X.X.X" port="${jboss.domain.master.port:9999}"
security-realm="ManagementRealm"/>
```

f. Sauvegarder le fichier et sortir.

2. **Configurer l'authentification pour chaque serveur esclave.**
Chaque serveur esclave a besoin d'un nom d'utilisateur et d'un mot de passe créé dans le **`ManagementRealm`** du contrôleur de domaine ou du master autonome. Sur le contrôleur de domaine ou sur le master autonome, exécutez la commande ***`EAP_HOME`*`/add-user.sh`**. Ajouter un utilisateur avec le même nom d'utilisateur comme esclave, au **`ManagementRealm`**. Quand on vous demandera si cet utilisateur doit s'authentifier auprès d'une instance de JBoss AS externe, répondez **`Oui`**. Vous trouverez un exemple de l'entrée et de la sortie de la commande ci-dessous, pour un esclave appelé **`slave1`**, et un mot de passe **`changeme`**.

```
user:bin user$ ./add-user.sh

What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Realm (ManagementRealm) :
Username : slave1
Password : changeme
Re-enter Password : changeme
About to add user 'slave1' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'slave1' to file '/home/user/jboss-eap-
6.0/standalone/configuration/mgmt-users.properties'
Added user 'slave1' to file '/home/user/jboss-eap-
6.0/domain/configuration/mgmt-users.properties'
Is this new user going to be used for one AS process to connect to
another AS process e.g. slave domain controller?
yes/no? yes
To represent the user add the following to the server-identities
definition <secret value="Y2hhbmdlbWU=" />
```

3. **Copier l'élément codé-Base64 `<secret>` à partir de la sortie `add-user.sh`.**
Si vous prévoyez de spécifier un mot de passe codé-Base64 pour l'authentification, copier l'élément **`<secret>`** à partir de la dernière ligne de la sortie **`add-user.sh`** car vous en aurez besoin à l'étape suivante.

4. **Modifier le domaine de sécurité de l'hôte esclave pour la nouvelle authentification.**

   a. Réouvrir le fichier **`host.xml`** ou **`standalone-ha.xml`** de l'hôte esclave.

   b. Chercher l'élément **`<security-realms>`**. C'est là où vous configurez le domaine de sécurité.

   c. Vous pourrez spécifier la valeur secrète d'une des manières suivantes :

      - **Spécifier le mot de passe codé-Base64 dans le fichier de configuration.**

i. Ajouter le bloc suivant de code XML sous la ligne **`<security-realm name="ManagementRealm">`**,

```
<server-identities>
    <secret value="Y2hhbmdlbWU="/>
</server-identities>
```

ii. Remplacer "Y2hhbmdlbWU=" par la valeur secrète retournée de la sortie **`add-user.sh`** lors de l'étape précédente.

- **Configurer l'hôte pour obtenir un mot de passe de l'archivage sécurisé.**

i. Utiliser le script **`vault.sh`** pour générer un mot de passe masqué. Cela va créer un string qui ressemblera à ceci : **`VAULT::secret::password::ODVmYmJjNGMtZDU2ZC00YmNlLWE4ODMtZjQ1NWNmNDU4ZDc1TElORV9CUkVBS3ZhdWx0`**.

Vous pourrez trouver plus d'informations sur l'archivage sécurisé dans Password Vaults de la section Sensitive Strings de ce guide, ici : Section 10.12.1, « Sécurisation des chaînes sensibles des fichiers en texte clair ».

ii. Ajouter le bloc de code XML suivant directement sous la ligne **`<security-realm name="ManagementRealm">`**.

```
<server-identities>
    <secret
value="${VAULT::secret::password::ODVmYmJjNGMtZDU2ZC00YmNlLWE4ODMtZjQ1NWNmNDU4ZDc1TElORV9CUkVBS3ZhdWx0}"/>
</server-identities>
```

Veillez à remplacer la valeur secrète par le mot de passe masqué généré lors de l'étape précédente.

**NOTE**

Quand vous créez un mot de passe dans l'archivage sécurisé, celui-ci devra être spécifié en texte brut, et non pas codé-Base64.

- **Spécifier le mot de passe en tant que propriété système.**

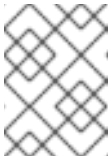
i. Ajouter le bloc de code XML suivant sous la ligne **`<security-realm name="ManagementRealm">`**

```
<server-identities>
    <secret value=${server.identity.password}/>
</server-identities>
```

ii. Quand vous spécifiez le mot de passe en tant que propriété système, vous pouvez configurer l'hôte d'une des manières suivantes :

- Démarrer le serveur en saisissant le mot de passe en texte brut comme argument de ligne de commande, comme par exemple :

```
-Dserver.identity.password=changeme
```



**NOTE**

Le mot de passe doit être saisi en texte brut et sera visible par quiconque lance la commande **`ps -ef`**.

- Mettez le mot de passe dans un fichier de propriétés et passer l'URL du fichier de propriétés sous forme d'argument de ligne de commande.

A. Ajouter la paire clé/valeur à un fichier de propriétés. Par exemple :

```
server.identity.password=changeme
```

B. Démarrer le serveur par les arguments de ligne de commande

```
--properties=URL_TO_PROPERTIES_FILE
```

.

d. Sauvegarder et sortir du fichier.

5. **Redémarrer le serveur.**
L'esclave va maintenant authentifier le master en utilisant son nom d'hôte comme nom d'utilisateur et le string codifié comme son mot de passe.

Report a bug

### 16.5.6. Migration du trafic entre les clusters

**Résumé**

Après avoir créé un nouveau cluster avec JBoss Enterprise Application Platform 6, vous souhaitez sans doute migrer le trafic d'un ancien cluster vers un nouveau dans le cadre d'un processus de mise à niveau. Au cours de cette tâche, vous verrez la stratégie qui peut être utilisée pour migrer ce trafic avec un minimum de temps mort.

**Prérequis**

- Nouvelle installation de cluster: Section 16.5.2, « Configurer le sous-système **`mod_cluster`** » (Nous appelerons ce cluster: Cluster NEW).
- Une ancienne installation de cluster a été rendue obsolète (nous appèlerons ce cluster: CLustr OLD).

**Procédure 16.8. Mise à niveau du processus pour les clusters**

1. Installez votre nouveau cluster en suivant les étapes décrites dans les prérequis.

2. Pour les NOUVEAUX et VIEUX Cluster à la fois, assurez-vous que l'option de configuration **sticky-session** est définie sur **true** (**true** par défaut). L'activation de cette option signifie que toutes les nouvelles demandes présentées à un nœud de cluster dans un de ces clusters continueront d'aller vers ce nœud.

```
/profile=full-ha/subsystem=modcluster/mod-cluster-
config=configuration/:write-attribute(name=sticky-
session,value=true)
```

3. Ajouter les nœuds dans le NOUVEAU Cluster à la configuration de mod_cluster individuellement à l'aide du processus décrit ici: Section 16.5.5, « Configurer un Worker Node de mod_cluster »

4. Configurer l'équilibrage de la charge (mod_cluster) pour arrêter les contextes individuels dans l'ANCIENS Cluster. L'arrêt des contextes (par opposition à leur désactivation) dans l'ANCIEN Cluster permettra aux contextes individuels de s'arrêter gracieusement (et éventuellement à un arrêt total). Les sessions existantes seront toujours servies, mais aucune nouvelle session ne sera dirigée vers ces nœuds. Les contextes arrêtés peuvent prendre plusieurs minutes ou même plusieurs heures pour s'arrêter.

   Vous pouvez utiliser le CLI suivant pour stopper un contexte. Remplacer les valeurs de paramètre par des valeurs adaptées à votre environnement.

```
[standalone@localhost:9999 subsystem=modcluster] :stop-
context(context=/myapp, virtualhost=default-host, waittime=50)
```

**Résultat**

Vous avez réussi à mettre JBoss EAP Cluster à niveau.

Report a bug

## 16.6. APACHE MOD_JK

### 16.6.1. Le connecteur Apache mod_HTTP

Apache **mod_jk** est un connecteur HTTP fourni aux clients qui en ont besoin pour des raisons decompatibilité. Apache **mod_jk** fourni un équilibrage des charges et fait partie des jboss-eap-native-webserver-connectors contenus dans JBoss Web Container. POur les platerformes prises en charge, consulter https://access.redhat.com/site/articles/111663. Le connecteur **mod_jk** est maintenu par Apache, et sa documentation se trouve à l'adresse suivante http://tomcat.apache.org/connectors-doc/.

JBoss Enterprise Application Platform peut accepter des charges de travail en provenance d'un serveur proxy Apache HTTPD. Le serveur proxy accepte les requêtes des clients en provenance des serveurs frontaux web, et passe le travail à des serveurs JBoss Enterprise Application Platform participant. Quand les sessions sticky sont activées, une requête en provenance d'un même client va toujours vers le même serveur Enterprise Application Platform, à moins que celui-ci ne soit pas rendu disponible.

À la différence du JBoss HTTP connector **mod_cluster**, un serveur proxy Apache ne connaît pas le statut des déploiements sur les serveurs ou groupes de serveurs, et ne peut donc pas ajuster les envois et lieux de travail en fonction du statut.

Comme **mod_cluster**, **mod_jk** communique à travers le protocole AJP 1.3.

**NOTE**

**mod_cluster** est un équilibreur de charges plus avancé que **mod_jk**. **mod_cluster** fournit toute la fonctionnalité de **mod_jk** et quelques fonctionnalités supplémentaires. Pour plus d'informations sur **mod_cluster**, consulter Section 16.5.1, « Le connecteur HTTP **mod_cluster** ».

**Prochaine étape : configurer la plateforme JBoss EAP pour qu'elle puisse participer à un groupe d'équilibrage des charges mod_jk**

- Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe »
- Section 16.6.3, « Installer le Module_jk_mod dans Apache HTTPD ou dans JBoss Enterprise Web Server HTTPD »

Report a bug

## 16.6.2. Configurer JBoss Enterprise Application Platform pour qu'il communique avec Apache Mod_jk

**Aperçu**

Le connecteur mod_jk HTTP possède un simple composant, le module **mod_jk.so**, qui est chargé par le démon HTTPD. Ce module reçoit les demandes des clients et les transfère vers le conteneur, en l'occurrence JBoss Enterprise Application Platform. JBoss Enterprise Application Platform doit également être configurée pour accepter ces demandes et envoyer leurs réponses vers le démon HTTPD.

La configuration de HTTPD est couverte dans Section 16.6.3, « Installer le Module_jk_mod dans Apache HTTPD ou dans JBoss Enterprise Web Server HTTPD ».

Pour que JBoss Enterprise Application Platform puisse communiquer avec HTTPD Apache, il doit avoir le connecteur AJP/1.3 HTTPD activé. Ce connecteur sera présent par défaut dans les configurations suivantes :

- Dans un domaine géré, dans les groupes de serveurs qui utilisent les profils **ha** et **full-ha**, et le groupe de liaisons de sockets **ha** ou **full-ha**. Le groupe de serveurs **other-server-group** est configuré correctement dans une installation par défaut.
- Dans un serveur autonome, le profil **standalone-ha** est fourni pour les configurations en cluster. Pour démarrer le serveur autonome avec ce profil, lancer la commande ci-dessous, à partir du répertoire ***EAP_HOME/***.

```
[user@host bin]$ ./bin/standalone.sh --server-config=standalone-ha.xml
```

Report a bug

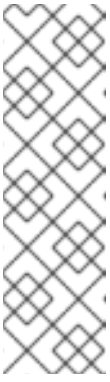
## 16.6.3. Installer le Module_jk_mod dans Apache HTTPD ou dans JBoss Enterprise Web Server HTTPD

**Prérequis**

- Pour cette tâche, vous devrez utiliser Apache HTTPD installé dans un environnement pris en charge ou l'HTTPD installé sur JBoss Enterprise Web Server. Notez que l'HTTPD installé dans JBoss Enterprise Web Server fait partie de la distribution JBoss Enterprise Application Platform.
- Si vous avez besoin d'installer Apache HTTPD, utiliser les instructions dans *Red Hat Enterprise Linux Deployment Guide*disponibles à partir de https://access.redhat.com/site/documentation/.
- Si vous avez besoin d'installer le serveur JBoss Enterprise Web Server, utiliser les instructions dans *JBoss Enterprise Web Server Installation Guide* disponibles à partir de https://access.redhat.com/site/documentation/.
- Si vous utilisez Apache HTTPD, télécharger le package de JBoss Enterprise Application Platform Native Components pour votre plate-forme du portail client de Red Hat à https://access.redhat.com. Ce paquet contient les mod_cluster et mod_jk binaires précompilés pour Red Hat Enterprise Linux. Si vous utilisez JBoss Enterprise Web Server, il comprend déjà le binaire pour mod_jk.
- Vous devez être connectés avec des privilèges administratifs (root).

**Procédure 16.9. Installer le Module mod cluster**

1. **Déterminer l'emplacement de votre configuration HTTPD**
   Votre emplacement de configuration HTTPD sera différente selon que vous utilisiez Apache HTTPD de Red Hat Enterprise Linux, ou le HTTPD disponible dans JBoss Enterprise Web Server. C'est l'une des trois options suivantes qui sera mentionnée au cours de cette tâche sous le nom *HTTPD_HOME*.
   - Apache HTTPD - **`/etc/httpd/`**
   - JBoss Enterprise Web Server HTTPD dans RHEL - ***`EWS_HOME`*`/httpd`**
   - JBoss Enterprise Web Server HTTPD dans Solaris - ***`EWS_HOME`*`/etc/httpd`**
   - JBoss Enterprise Web Server HTTPD dans Windows - ***`EWS_HOME`*`/etc/httpd`**
2. **Configurer le module mod_jk.**
   a. Créer un nouveau fichier nommé ***`HTTPD_HOME`*`/conf.d/mod-jk.conf`** et y ajouter ce qui suit.



**NOTE**

La directive **`JkMount`** indique quels URL Apache doivent aller vers le module mod_jk. Sur la base de la configuration de la directive, mod_jk transfère l'URL reçu aux conteneurs de servlet qui conviennent.

Pour servir le contenu directement, et n'utiliser que l'équilibreur de charges pour les applications Java, le chemin URL doit être **`/application/*`**. Pour utiliser mod_jk en tant qu'équilibreur des charges, utiliser la valeur **`/*`** pour transférer tous les URL au mod_jk.

```
# Load mod_jk module
# Specify the filename of the mod_jk lib
LoadModule jk_module modules/mod_jk.so
```

```
# Where to find workers.properties
JkWorkersFile conf/workers.properties

# Where to put jk logs
JkLogFile logs/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
JkLogStampFormat  "[%a %b %d %H:%M:%S %Y]"

# JkOptions indicates to send SSK KEY SIZE
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount your applications
# The default setting only sends Java application data to mod_jk.
# Use the commented-out line to send all URLs through mod_jk.
# JkMount /* loadbalancer
JkMount /application/* loadbalancer

# Add shared memory.
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
JkMount status
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>
```

Observer les valeurs et vérifier qu'elles conviennent à votre installation. Quand vous serez satisfait, sauvegarder le fichier.

b. **Spécifier une directive JKMountFile**
En plus de la directive JKMount de **`mod-jk.conf`**, vous pourrez spécifier un fichier qui contienne des modèles URL multiples à transférer au mod_jk.

i. Ajouter ce qui suit au fichier ***`HTTPD_HOME`*`/conf/mod-jk.conf`**:

```
# You can use external file for mount points.
# It will be checked for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
JkMountFile conf/uriworkermap.properties
```

ii. Créer un nouveau fichier intitulé ***HTTPD_HOME*/conf/uriworkermap.properties**, avec une ligne pour chaque modèle URL à faire correspondre. L'exemple suivant montre des exemples de syntaxe pour ce fichier.

```
# Simple worker configuration file
/*=loadbalancer
```

c. **Copier le fichier mod_jk.so dans le répertoire de modules HTTPD**

**NOTE**

Utile uniquement si votre HTTPD n'a pas de **mod_jk.so** dans son répertoire **modules/**. Vous pourrez éviter cette étape si vous utilisez le serveur Apache HTTPD inclus, un téléchargement de JBoss Enterprise Application Platform 6.

Extraire le paquet Native Web Server Connectors ZIP. Localiser le fichier **mod_jk.so** soit dans le répertoire ***EAP_HOME*/modules/native/lib/httpd/modules/** ou le répertoire ***EAP_HOME*/modules/native/lib64/httpd/modules/** suivant que votre système d'exploitation est de 32-bit ou de 64-bit.

Copier le fichier dans le répertoire ***HTTPD_HOME*/modules/**.

3. **Configurer les noeuds de worker mod_jk.**

a. Créer un nouveau fichier nommé ***HTTPD_HOME*/conf/workers.properties**. Utiliser l'exemple suivant comme point de départ, et modifier le fichier selon vos besoins.

```
# Define list of workers that will be used
# for mapping requests
worker.list=loadbalancer,status

# Define Node1
# modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=node1.mydomain.com
worker.node1.type=ajp13
worker.node1.ping_mode=A
worker.node1.lbfactor=1

# Define Node2
# modify the host as your host IP or DNS name.
worker.node2.port=8009
worker.node2.host=node2.mydomain.com
worker.node2.type=ajp13
worker.node2.ping_mode=A
worker.node2.lbfactor=1

# Load-balancing behavior
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1
```

```
# Status worker for managing load balancer
worker.status.type=status
```

Pour obtenir une description détaillée de la syntaxe du fichier **`workers.properties`**, et pour obtenir des options de configuration avancées, consulter Section 16.6.4, « Référence de configuration des Apache Mod_jk Workers ».

4. **Redémarrer HTTPD.**
   La façon de redémarrer HTTPD dépend de savoir si vous utilisez Apache HTTPD de Red Hat Enterprise Linux ou le HTTDP inclus dans JBoss Enterprise Web Server. Choisir une des deux méthodes ci-dessous.

   - **Apache HTTPD de Red Hat Enterprise Linux**
     Exécuter la commande suivante :

     ```
     [root@host]# service httpd restart
     ```

   - **JBoss Enterprise Web Server HTTPD**
     JBoss Enterprise Web Server exécute à la fois sur Red Hat Enterprise Linux et Microsoft Windows Server. La méthode de redémarrage du HTTPD est différente pour chacun.

     - **Red Hat Enterprise Linux**
       Dans Red Hat Enterprise Linux, JBoss Enterprise Web Server installe son HTTPD en tant que service. Pour redémarrer HTTPD, lancer les deux commandes suivantes :

       ```
       [root@host ~]# service httpd stop
       [root@host ~]# service httpd start
       ```

     - **Microsoft Windows Server**
       Lancer les commandes suivantes dans une invite de commande avec des privilèges administratifs :

       ```
       C:\> net stop httpd
       C:\> net start httpd
       ```

     - **Solaris**
       Lancer les commandes suivantes dans une invite de commande avec des privilèges administratifs :

       ```
       /opt/jboss-ews-2.0/sbin/apachectl restart
       ```

**Résultat**

Apache HTTPD est maintenant configuré pour pouvoir utiliser l'équilibreur de charges de mod_jk. Pour configurer JBoss Enterprise Application Platform pour qu'il soit au fait de mod_jk, consulter Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe ».

Report a bug

### 16.6.4. Référence de configuration des Apache Mod_jk Workers

Le fichier **`workers.properties`** définit le comportement des noeuds de workers à qui mod_jk passe

les requêtes de clients. Dans Red Hat Enterprise Linux, le fichier se trouve dans **`/etc/httpd/conf/workers.properties`**. Le fichier **`workers.properties`** définit où les différents conteneurs de servlet se trouvent, et la façon dont la charge de travail doit être distribuée parmi eux.

La configuration est divisée en trois sections. La première section traite des propriétés globales, qui s'appliquent à tous les nœuds de worker. La deuxième section contient des paramètres qui s'appliquent à un worker spécifique. La troisième section contient les paramètres qui s'appliquent à un nœud spécifique, équilibré par le worker.

La structure générale d'une propriété est **`worker.WORKER_NAME.DIRECTIVE`**, avec *WORKER_NAME* comme nom unique de worker, et *DIRECTIVE* comme paramètre de configuration à appliquer au worker.

#### Référence de configuration des Apache Mod_jk Workers

Les modèles de noeuds spécifient les paramètres par défaut par noeud. Vous pouvez remplacer le modèle contenu dans le paramètre de nœud lui-même. Vous pouvez voir un exemple de modèle de nœud dans Exemple 16.2, « Exemple de fichier **`workers.properties`** ».

**Tableau 16.11. Propriétés globales**

| Propriété | Description |
|---|---|
| worker.list | La liste des noms de worker utilisés par mod_jk. Ces workers sont prêts à recevoir des requêtes. |

**Tableau 16.12. Propriétés per-worker**

| Propriété | Description |
|---|---|
| type | Le type de worker. Le type par défaut est **`ajp13`**. Autres valeurs possibles **`ajp14`**, **`lb`**, **`status`**.<br><br>Pour plus d'informations sur ces directives, voir la référence de protocole Apache Tomcat Connector AJP à http://tomcat.apache.org/connectors-doc/ajp/ajpv13a.html. |
| balance_workers | Spécifie les nœuds de worker que l'équilibreur de charges doit gérer. Vous pouvez utiliser la directive plusieurs fois pour un même équilibrage de charge. Il se compose d'une liste séparée par des virgules des noms de workers. Ceci est défini par worker, et non pas par nœud. Elle affecte tous les nœuds équilibrés par ce type de worker. |
| sticky_session | Indique si les demandes d'une même session sont toujours acheminées vers le même worker. La valeur par défaut est **`0`**, ce qui signifie que les sticky sessions sont désactivées. Pour activer des sticky sessions, définir à **`1`**. Les sticky sessions doivent habituellement être activées, à moins que toutes vos demandes soient vraiment stateless. Ceci est défini par worker, et non pas par nœud. Affecte tous les nœuds équilibrés par ce type de worker. |

**Tableau 16.13. Propriétés per-node**

| Propriété | Description |
|---|---|

| Propriété | Description |
|---|---|
| host | Le nom d'hôte ou l'adresse IP du worker. Le noeud de worker doit supporter la pile de protocole **ajp**. L avaleur par défaut est **localhost**. |
| Important | Le numéro de port de l'instance de serveur éloigné qui écoute les requêtes de protocoles définis. La valeur par défaut est **8009**, qui correspond au port d'écoute des workers AJP13. La valeur par défaut des workers AJP14 est 8011. |
| ping_mode | Les conditions dans lesquelles les connexions sont interrogées pour le statut du réseau. La sonde utilise un paquet AJP13 vide pour CPing et s'attend à un CPong en réponse. Spécifier les conditions à l'aide d'une combinaison d'indicateurs de la directive. Les indicateurs ne sont pas séparés par une virgule ou un espace blanc. Le ping_mode peut être n'importe quelle combinaison de **C**, **P**, **I**, ou **A**.<br>• **C** - Connect. Sonde la connexion une fois seulement suite à la connexion au serveur. Spécifier le timeout en utilisant la valeur de **connect_timeout**. Sinon, la valeur de **ping_timeout** sera utilisée.<br>• **P** - Prepost. Sonde la connexion avant d'envoyer d'envoyer chaque requête au serveur. Spécifier le timeout en utilisant la directive **prepost_timeout**. Sinon, la valeur **ping_timeout** sera utilisée.<br>• **I** - Interval. Interroge la connexion à un intervalle spécifié par **connection_ping_interval**, si présent. Sinon, utilise la valeur de **ping_timeout**.<br>• **A** - All. Un raccourci pour **CPI**, qui indique que toutes les sondes de connexion sont utilisées. |
| ping_timeout, connect_timeout, prepost_timeout, connection_ping_interval | Les valeurs de timeout pour les paramètres de sonde de connexion ci-dessus. La valeur est spécifiée en millisecondes, et la valeur par défaut pour **ping_timeout** est de 10000. |
| lbfactor | Spécifie le facteur d'équilibrage des charge d'un worker individuel et ne s'applique qu'à un worker membre d'un équilibreur de charge. Ceci est utile pour donner à un serveur plus puissant, une charge de travail supplémentaire. Pour donner à un worker 3 fois la charge de la valeur par défaut, définir cette valeur à **3**: **worker.my_worker.lbfactor=3** |

**Exemple 16.2. Exemple de fichier `workers.properties`**

```
worker.list=node1, node2, node3

worker.balancer1.sticky_sessions=1
worker.balancer1.balance_workers=node1
worker.balancer2.sticky_session=1
worker.balancer2.balance_workers=node2,node3

worker.nodetemplate.type=ajp13
worker.nodetemplate.port=8009
```

```
worker.node1.template=nodetemplate
worker.node1.host=localhost
worker.node1.ping_mode=CI
worker.node1.connection_ping_interval=9000
worker.node1.lbfactor=1

worker.node2.template=nodetemplate
worker.node2.host=192.168.1.1
worker.node2.ping_mode=A

worker.node3.template=nodetemplate
worker.node3.host=192.168.1.2
```

Les détails de configuration de ce document sont limités. Voir la documentation Apache à http://tomcat.apache.org/connectors-doc/ pour obtenir des instructions supplémentaires.

Report a bug

## 16.7. APACHE MOD_PROXY

### 16.7.1. Le connecteur Apache mod_proxy HTTP

Apache offre deux modules différents d'équilibrage de charge et de proxying pour ses démons HTTP : **mod_proxy** et **mod_jk**. Pour en savoir plus sur **mod_jk**, consulter Section 16.6.1, « Le connecteur Apache mod_HTTP ». La plate-forme JBoss Enterprise Application Platform prend en charge l'utilisation de l'un d'entre eux, bien que **mod_cluster**, le connecteur HTTP de JBoss, couple plus étroitement JBoss Enterprise Application Platform et le démon HTTP, et est le connecteur HTTP recommandé. Reportez-vous à Section 16.1.3, « Connecteurs HTTP - Aperçu général » pour une vue d'ensemble des connecteurs HTTP pris en charge, y compris les avantages et les inconvénients.

À la différence de **mod_jk**, **mod_proxy** supporte les connexions via les protocoles HTTP et HTTPS. Chacun d'eux aussi en charge le protocole AJP.

**mod_proxy** peut être configuré en autonome ou en configurations d'équilibrage de charge, et il prend en charge la notion de sticky sessions.

Le module **mod_proxy** nécessite que JBoss Enterprise Application Platform ait le connecteur web HTTP, HTTPS ou AJP configuré. Cela fait partie du sous-système web. Consulter Section 15.1, « Configurer le Sous-système Web » pour obtenir des informations sur la façon de configurer le sous-système web.

Report a bug

### 16.7.2. Installer Mod_proxy HTTP Connector dans Apache HTTPD

**Aperçu**

**mod_proxy** est un module d'équilibrage de charges fourni par Apache. Cette tâche présente une configuration de base. Pour plus d'informations sur la configuration avancée, ou pour plus de détails, reportez-vous à la documentation Apache **mod_proxy** à https://httpd.apache.org/docs/2.2/mod/mod_proxy.html. Pour plus d'informations sur **mod_proxy** d'une perspective JBoss Enterprise Application Platform, consulter Section 16.7.1, « Le connecteur Apache mod_proxy HTTP » et Section 16.1.3, « Connecteurs HTTP - Aperçu général ».

**Prérequis**

- JBoss Enterprise Web Server HTTPD ou Apache HTTPD doivent être installés. Un démon HTTP autonome est fourni séparément dans le portail clients Red Hat à https://access.redhat.com, dans la zone de téléchargement de JBoss Enterprise Application Platform 6. Voir Section 16.3.2, « Installer Apache HTTPD inclus avec JBoss Enterprise Application Platform 6 » pour obtenir des informations sur le démon HTTP si vous souhaitez l'utiliser.
- Les modules **mod_proxy** doivent être installés. Apache HTTPD est généralement livré avec les modules **mod_proxy** déjà inclus. C'est le cas sur Red Hat Enterprise Linux et le démon HTTPD qui vient avec le serveur Web de JBoss Enterprise.
- Vous avez besoin d'être **root** ou de posséder des privilèges administratifs pour modifier la configuration HTTPD.
- Déterminer le répertoire de configuration HTTPD. C'est le répertoire contenant les répertoires **conf/** et **modules/** pour Apache HTTPD. Ceci sera dénommé ***HTTPD_CONF*** pour le reste de cette tâche. Les valeurs typiques sont les suivantes :
  - **/etc/httpd/**
  - ***EWS_HOME*/httpd/**, à partir d'où le serveur JBoss Enterprise Web Server est installé.
- Dans notre exemple, on assume que JBoss Enterprise Application Platform est configuré avec le connecteur web HTTP ou HTTPS. Cela fait partie de la configuration du sous-système web. Voir Section 15.1, « Configurer le Sous-système Web » pour obtenir des informations sur la façon de configurer le sous-système web.

1. **Activer les modules mod_proxy dans le démon HTTP.**
   Recherchez les lignes suivantes dans votre fichier ***HTTPD_CONF*/conf/httpd.conf**. Si elles ne sont pas présentes, ajoutez-les en bas. Si elles sont présentes, mais que les lignes commencent par un caractère de commentaire (#), supprimer le caractère. Enregistrez le fichier par la suite. Habituellement, les modules sont déjà présents et activés.

   ```
   LoadModule proxy_module modules/mod_proxy.so
   LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
   LoadModule proxy_http_module modules/mod_proxy_http.so
   # Uncomment these to proxy FTP or HTTPS
   #LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
   #LoadModule proxy_connect_module modules/mod_proxy_connect.so
   ```

2. **Ajouter un proxy non équilibreur de charges.**
   Ajouter la configuration suivante à votre fichier ***HTTPD_CONF*/conf/httpd.conf**, directement sous une directive **<VirtualHost>** que vous possédez sans doute. Remplacer les valeurs par des valeurs appropriées à votre installation.

   Cet exemple utilise un hôte virtuel. Voir la nouvelle étape pour utiliser la configuration HTTPD par défaut.

   ```
   <VirtualHost *:80>
   # Your domain name
   ServerName Domain_NAME_HERE

   ProxyPreserveHost On
   ```

```
# The IP and port of the JBoss Enterprise Application Platform
# These represent the default values, if your HTTPD is on the same
host
# as your JBoss Enterprise Application Platform managed domain or
server

ProxyPass / http://localhost:8080/
ProxyPassReverse / http://localhost:8080/

# The location of the HTML files, and access control information
DocumentRoot /var/www
<Directory /var/www>
Options -Indexes
Order allow,deny
Allow from all
</Directory>
</VirtualHost>
```

Après avoir appliqué vos changements, sauvegarder le fichier.

3. **Ajouter le proxy d'équilibrage des charges.**
Pour utiliser **mod_proxy** comme équilibreur de charges, et pour envoyer du travail à des serveurs multiples de JBoss Enterprise Application Platform, ajouter la configuration suivante à votre fichier ***HTTPD_CONF*/conf/httpd.conf**.

```
<Proxy balancer://mycluster>

Order deny,allow
Allow from all

# Add each JBoss Enterprise Application Server by IP address and
port.
# If the route values are unique like this, one node will not fail
over to the other.
BalancerMember http://10.16.92.99:8080 route=node1
BalancerMember http://10.16.92.100:8180 route=node2
</Proxy>

<VirtualHost *:80>
 # Your domain name
 ServerName YOUR_DOMAIN_NAME

 ProxyPreserveHost On
 ProxyPass / balancer://mycluster/

 # The location of the HTML files, and access control information
DocumentRoot /var/www
 <Directory /var/www>
  Options -Indexes
  Order allow,deny
  Allow from all
 </Directory>

</VirtualHost>
```

Les exemples ci-dessus communiquent tous par le protocole HTTP. Vous pouvez également utiliser les protocoles AJP ou HTTPS si vous chargez les modules **mod_proxy**. Voir la **mod_proxy** documentation http://httpd.apache.org/docs/2.2/mod/mod_proxy.html pour plus d'informations.

4. **Activer les sticky sessions.**
   *Sticky sessions* signifie que si la demande d'un client va initialement à un nœud spécifique de JBoss Enterprise Application Platform, toutes les demandes futures seront envoyés au même nœud, sauf si le nœud n'est plus disponible. C'est presque toujours le comportement correct.

   Pour activer des sticky sessions du **mod_proxy**, ajoutez le paramètre **stickysession** à l'énoncé **ProxyPass**. Cet exemple montre également d'autres paramètres que vous pouvez utiliser. Reportez-vous à documentation **mod_proxy** Apache à http://httpd.apache.org/docs/2.2/mod/mod_proxy.html pour plus d'informations à leur sujet.

   ```
   ProxyPass /MyApp balancer://mycluster stickysession=JSESSIONID
   lbmethod=bytraffic nofailover=Off
   ```

5. **Redémarrer le HTTPD.**
   Redémarrez le serveur HTTPD pour que les modifications prennent effet.

**Résultat**

Votre HTTPD est configuré pour utiliser le **mod_proxy** pour envoyer des demandes de client aux serveurs ou clusters de JBoss Enterprise Application Platform, en configuration standard ou équilibrage de charge. Pour configurer la plate-forme JBoss Enterprise Application pour répondre à ces demandes, reportez-vous à Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe ».

Report a bug

## 16.8. MICROSOFT ISAPI

### 16.8.1. Internet Server API (ISAPI) HTTP Connector

*Internet Server API (ISAPI)* est le connecteur HTTP du serveur web IIS (Internet Information Services). Vous pourrez utiliser JBoss Enterprise Application Platform comme noeud de worker dans le cluster IIS.

Pour configurer JBoss Enterprise Application Platform pour qu'il participe à un cluster IIS, voir Section 16.8.2, « Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI Redirector ». Pour plus d'informations sur ISAPI, voir http://msdn.microsoft.com/en-us/library/ms524911(v=VS.90).aspx.

Report a bug

### 16.8.2. Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI Redirector

**Prérequis**

- Assurez-vous que vous utilisez un système d'exploitation pris en charge et installez le serveur IIS. Se référer à https://access.redhat.com/site/articles/111663 pour obtenir une liste des configurations prises en charge.

- Télécharger le package JBoss Native Components de Microsoft Windows, à partir du Portail Clients dans https://access.redhat.com. Naviguez dans **Downloads**, puis **JBoss Enterprise Middleware**, puis **Application Platform.**. Choisir soit **i386** ou bien **x86_64**. Décompresser le fichier, qui inclut «ISAPI redirect DLL» dans le répertoire **jboss-eap-6.0/modules/native/sbin/**.

  Extraire le fichier zip de composants natifs et copier son contenu dans le répertoire **sbin** vers une location sur votre serveur. Le reste de cette tâche assume que vous utilisez **C:\connectors**.

**Procédure 16.10. Configurer le IIS Redirector à l'aide du IIS Manager (IIS 7)**

1. Ouvrir le IIS Manager en claiquant sur **Start** → **Run** , puis, saisissez **inetmgr**.
2. Dans le panneau de vue d'aroborescence, développer **IIS 7**.
3. Cliquer à deux fois sur **ISAPI and CGI Registrations** pour l'ouvrir sous forme d'une fenêtre séparée.
4. Dans le panneau **Actions**, cliquer sur **Add**. La fenêtre **Add ISAPI or CGI Restriction** s'ouvrira.
5. Indiquer les valeurs suivantes :
   - **ISAPI or CGI Path**: **c:\connectors\sbin\isapi_redirect.dll**
   - **Description**: **jboss**
   - **Allow extension path to execute**: sélectionner la case à cocher.
6. Cliquer sur **OK** pour fermer la fenêtre **Add ISAPI or CGI Restriction**.
7. **Définir un répertoire virtuel JBoss Native**
   a. Cliquer à droite sur **Default Web Site**, puis cliquer sur **Add Virtual Directory**. La fenêtre **Add Virtual Directory** va s'ouvrir.
   b. Indiquer les valeurs suivantes pour ajouter un répertoire virtuel :
      - **Alias**: **jboss**
      - **Physical Path**: **C:\connectors**
   c. Cliquer sur **OK** pour sauvegarder les valeurs et fermer la fenêtre **Add Virtual Directory**.
8. **Définir un filtre JBoss Native ISAPI Redirect**
   a. Dans le panneau de vue d'arborescence, développer **Sites** → **Default Web Site**.
   b. Cliquer à deux fois sur **Filtres ISAPI**. L'affichage **ISAPI Filters Features** apparaîtra.
   c. Dans le panneau **Actions**, cliquer sur **Add**. La fenêtre **Add ISAPI Filter** s'ouvrira.
   d. Indiquer les valeurs suivantes dans la fenêtre **Add ISAPI Filter**:

- **Filter name**: **jboss**
- **Executable**: **C:\connectors\sbin\isapi_redirect.dll**

e. Cliquer **OK** pour sauvegarder les valeurs et pour fermer la fenêtre **Add ISAPI Filters**.

9. **Activer le handler ISAPI-dll**

a. Cliquer deux fois sur l'élément **IIS 7** qui se trouve sur le panneau d'affichage: **IIS 7 Home Features View** s'ouvrira.

b. Cliquer deux fois sur **Handler Mappings**: **Handler Mappings Features View** s'ouvrira.

c. Dans la liste déroulante modifiable **Group by** sélectionner **State**, les **Handler Mappings** s'affichent dans **Enabled and Disabled Groups**.

d. Trouver **ISAPI-dll**. S'il se trouve dans le groupe **Disabled**, cliquer à droite, et sélectionner **Edit Feature Permissions**.

e. Activer les permissions suivantes :

- Read
- Script
- Execute

f. Cliquer sur **OK** pour sauvegarder les valeurs, et fermer la fenêtre **Edit Feature Permissions**.

**Résultat**

Microsoft IIS est maintenant configuré pour utiliser le re-directeur ISAPI Redirector. Ensuite, Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe », puis Section 16.8.3, « Configurer ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP » ou Section 16.8.4, « Configurer ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP ».

Report a bug

### 16.8.3. Configurer ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP

**Aperçu**

Cette tâche configure un groupe de serveurs de JBoss Enterprise Application Platform pour qu'ils puissent accepter les demandes du redirecteur ISAPI. Il n'inclut pas la configuration d'équilibrage de charge ou de haute disponibilité avec basculement. Si vous avez besoin de ces fonctionnalités, reportez-vous à Section 16.8.4, « Configurer ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP ».

Cette configuration est faîte sur le serveur IIS, et assume que JBoss Enterprise Application Platform est déjà configurée, comme dans Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe ».

**Prérequis**

- Vous aurez besoin d'un accès administrateur pour accéder au serveur IIS
- Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe »
- Section 16.8.2, « Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI Redirector »

**Procédure 16.11. Modifier les fichiers de propriété et Configurer la redirection**

1. **Créer un répertoire pour stocker la journalisation, les fichiers de propriété, et les fichiers de verrouillage.**
   Le reste de cette procédure suppose que vous utilisez le répertoire **`C:\connectors\`** à cet effet. Si vous utilisez un autre répertoire, modifier les instructions en conséquence.

2. **Créer le fichier `isapi_redirect.properties`.**
   Créer un nouveau fichier intitulé **`C:\connectors\isapi_redirect.properties`**. Copier les contenus suivants dans le fichier.

   ```
   # Configuration file for the ISAPI Redirector
   # Extension uri definition
   extension_uri=c:\connectors\isapi_redirect.dll

   # Full path to the log file for the ISAPI Redirector
   log_file=c:\connectors\isapi_redirect.log

   # Log level (debug, info, warn, error or trace)
   # Use debug only testing phase, for production switch to info
   log_level=debug

   # Full path to the workers.properties file
   worker_file=c:\connectors\workers.properties

   # Full path to the uriworkermap.properties file
   worker_mount_file=c:\connectors\uriworkermap.properties

   #Full path to the rewrite.properties file
   rewrite_rule_file=c:\connectors\rewrite.properties
   ```

   Si vous ne souhaitez pas utiliser un fichier **`rewrite.properties`**, dé-commentez la dernière ligne en plaçant un caractère # au début de la ligne. Voir Étape 5 pour plus d'informations.

3. **Créer le fichier `uriworkermap.properties`**
   Le fichier **`uriworkermap.properties`** contient les mappages entre les URL de l'application déployée et quel worker gère leurs demandes vers eux. Le fichier d'exemple suivant illustre la syntaxe du fichier. Placez votre fichier **`uriworkermap.properties`** dans **`C:\connectors\`**.

   ```
   # images and css files for path /status are provided by worker01
   /status=worker01
   /images/*=worker01
   /css/*=worker01

   # Path /web-console is provided by worker02
   # IIS (customized) error page is used for http errors with number
   greater or equal to 400
   ```

```
# css files are provided by worker01
/web-console/*=worker02;use_server_errors=400
/web-console/css/*=worker01

# Example of exclusion from mapping, logo.gif won't be displayed
# !/web-console/images/logo.gif=*

# Requests to /app-01 or /app-01/something will be routed to
worker01
/app-01|/*=worker01

# Requests to /app-02 or /app-02/something will be routed to
worker02
/app-02|/*=worker02
```

4. **Créer le fichier `workers.properties`.**
   Le fichier **`workers.properties`** contient des définitions de mappage entre les étiquettes de workers et les instances de serveur. Le fichier d'exemple suivant illustre la syntaxe du fichier. Placez ce fichier dans le répertoire **`C:\connectors\`**.

```
# An entry that lists all the workers defined
worker.list=worker01, worker02

# Entries that define the host and port associated with these
workers

# First JBoss Enterprise Application Platform server definition,
port 8009 is standard port for AJP in EAP
worker.worker01.host=127.0.0.1
worker.worker01.port=8009
worker.worker01.type=ajp13

# Second JBoss Enterprise Application Platform server definition
worker.worker02.host= 127.0.0.100
worker.worker02.port=8009
worker.worker02.type=ajp13
```

5. **Créer le fichier `rewrite.properties`.**
   Le fichier **`rewrite.properties`** contient des dispositions relatives aux demandes spécifiques de réécriture d'URL simple pour certaines applications. Le chemin d'accès de réécriture est spécifié à l'aide de paires nom / valeur, comme illustré dans l'exemple ci-dessous. Placez ce fichier dans le répertoire **`C:\connectors\`**.

```
#Simple example
# Images are accessible under abc path
/app-01/abc/=/app-01/images/
```

6. **Redémarrer le serveur IIS.**
   Redémarrer votre serveur IIS par les commandes **`net stop`** et **`net start`**.

```
C:\> net stop was /Y
C:\> net start w3svc
```

**Résultat**

Le serveur IIS est configuré pour envoyer des demandes de client à des serveurs spécifiques de JBoss Enterprise Application Platform que vous avez configurés, sur une base spécifique à l'application.

Report a bug

### 16.8.4. Configurer ISAPI Redirector pour qu'il équilibre des requêtes de clients entre des serveurs multiples de la plate-forme JBoss EAP

**Aperçu**

Cette configuration équilibre les requêtes des clients entre les serveurs de JBoss Enterprise Application Platform que vous spécifiez. Si vous préférez envoyer des demandes de client à des serveurs JBoss Enterprise Application Platform spécifiques sur une base «par-déploiement», reportez-vous plutôt à Section 16.8.3, « Configurer ISAPI Redirector pour qu'il envoie des requêtes de clients à la plate-forme JBoss EAP ».

Cette configuration est faîte sur le serveur IIS, et assume que la plateforme JBoss EAP est déjà configurée, selon Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe ».

**Prérequis**

- Vous aurez besoin d'un accès administrateur pour accéder au serveur IIS.
- Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe »
- Section 16.8.2, « Configurer Microsoft IIS pour qu'il puisse utiliser le re-directionneur ISAPI Redirector »

**Procédure 16.12. Équilibrage des requêtes de clients entre des serveurs multiples.**

1. **Créer un répertoire pour stocker la journalisation, les fichiers de propriété, et les fichiers de verrouillage.**
   Le reste de cette procédure suppose que vous utilisez le répertoire **`C:\connectors\`** à cet effet. Si vous utilisez un autre répertoire, modifier les instructions en conséquence.

2. **Créer le fichier `isapi_redirect.properties`.**
   Créer un nouveau fichier intitulé **`C:\connectors\isapi_redirect.properties`**. Copier les contenus suivants dans le fichier.

```
# Configuration file for the ISAPI Redirector
# Extension uri definition
extension_uri=C:\connectors\isapi_redirect.dll

# Full path to the log file for the ISAPI Redirector
log_file==c:\connectors\isapi_redirect.log

# Log level (debug, info, warn, error or trace)
# Use debug only testing phase, for production switch to info
log_level=debug

# Full path to the workers.properties file
worker_file=c:\connectors\workers.properties

# Full path to the uriworkermap.properties file
```

```
worker_mount_file=c:\connectors\uriworkermap.properties

#OPTIONAL: Full path to the rewrite.properties file
rewrite_rule_file=c:\connectors\rewrite.properties
```

Si vous ne souhaitez pas utiliser un fichier **`rewrite.properties`**, dé-commentez la dernière ligne en plaçant un caractère # au début de la ligne. Voir Étape 5 pour plus d'informations.

3. **Créer le fichier `uriworkermap.properties`.**
Le fichier **`uriworkermap.properties`** contient les mappages entre les URL de l'application déployée et quel worker gère les demandes à leur intention. Le fichier exemple suivant illustre la syntaxe du fichier, avec une configuration d'équilibrage de charge. Le caractère générique (**`*`**) envoie toutes les requêtes de divers sous-répertoires d'URL vers l'équilibreur de charges nommé **`router`**. La configuration de l'équilibreur de charges est couverte dans Étape 4.

   Mettez votre fichier **`uriworkermap.properties`** dans **`C:\connectors\`**.

```
# images, css files, path /status and /web-console will be
# provided by nodes defined in the load-balancer called "router"
/css/*=router
/images/*=router
/status=router
/web-console|/*=router

# Example of exclusion from mapping, logo.gif won't be displayed
!/web-console/images/logo.gif=*

# Requests to /app-01 and /app-02 will be routed to nodes defined
# in the load-balancer called "router"
/app-01|/*=router
/app-02|/*=router

# mapping for management console, nodes in cluster can be enabled or
disabled here
/jkmanager|/*=status
```

4. **Créer le fichier `workers.properties`.**
Le fichier **`workers.properties`** contient les définitions de mappage entre les étiquettes de workers et les instances de serveur. Le fichier exemple suivant illustre la syntaxe du fichier. L'équilibrage de charge est configuré vers la fin du fichier, et comprend les workers **`worker01`** et **`worker02`**. Le fichier **`workers.properties`** suit la syntaxe du même fichier que celui utilisé pour la configuration d'Apache mod_jk. Pour plus d'informations sur la syntaxe du fichier **`workers.properties`**, reportez-vous à Section 16.6.4, « Référence de configuration des Apache Mod_jk Workers ».

   Mettez ce fichier dans le répertoire **`C:\connectors\`**.

```
# The advanced router LB worker
worker.list=router,status

# First EAP server definition, port 8009 is standard port for AJP in
EAP
#
# lbfactor defines how much the worker will be used.
```

```
# The higher the number, the more requests are served
# lbfactor is useful when one machine is more powerful
# ping_mode=A – all possible probes will be used to determine that
# connections are still working

worker.worker01.port=8009
worker.worker01.host=127.0.0.1
worker.worker01.type=ajp13
worker.worker01.ping_mode=A
worker.worker01.socket_timeout=10
worker.worker01.lbfactor=3

# Second EAP server definition
worker.worker02.port=8009
worker.worker02.host= 127.0.0.100
worker.worker02.type=ajp13
worker.worker02.ping_mode=A
worker.worker02.socket_timeout=10
worker.worker02.lbfactor=1

# Define the LB worker
worker.router.type=lb
worker.router.balance_workers=worker01,worker02

# Define the status worker for jkmanager
worker.status.type=status
```

5. **Créer le fichier `rewrite.properties`.**
   Le fichier **`rewrite.properties`** contient des dispositions relatives aux demandes spécifiques de réécriture d'URL simple pour certaines applications. Le chemin d'accès de réécriture est spécifié à l'aide de paires nom / valeur, comme illustré dans l'exemple ci-dessous. Placez ce fichier dans le répertoire **`C:\connectors\`**.

   ```
   #Simple example
   # Images are accessible under abc path
   /app-01/abc/=/app-01/images/
   ```

6. **Redémarrer le serveur IIS.**
   Redémarrer votre serveur IIS par les commandes **`net stop`** et **`net start`**.

   ```
   C:\> net stop was /Y
   C:\> net start w3svc
   ```

**Résultat**

Le serveur IIS est configuré pour envoyer des demandes de clients à des serveurs de JBoss Enterprise Application Platform référencés dans le fichier **`workers.properties`**, équilibrant la charge équitablement à travers les serveurs.

Report a bug

## 16.9. ORACLE NSAPI

### 16.9.1. Netscape Server API (NSAPI) HTTP Connector

*Netscape Server API (NSAPI)* est un connecteur HTTP qui permet à la plateforme JBoss EAP de participer en tant que noeud dans le serveur Oracle iPlanet Web Server (anciennement Netscape Web Server). Pour configurer ce connecteur, consulter Section 16.9.4, « Configurer NSAPI en tant que Cluster d'équilibrage des charges ».

Report a bug

### 16.9.2. Configurer le connecteur NSAPI dans Oracle Solaris

**Résumé**

Le connecteur NSAPI est un module qui exécute dans le serveur Oracle iPlanet Web Server.

**Prérequis**

- Votre serveur exécute Oracle Solaris 10 ou supérieur, soit en architecture 32-bit ou 64-bit.
- Oracle iPlanet Web Server 6.1 SP 12 ou 7.0 U8 est installé ou configuré, indépendamment du connecteur NSAPI.
- La plate-forme JBoss Enterprise Application Platform est installée et configurée sur chaque serveur qui servira en tant que noeud de worker. Section 16.3.5, « Configurer JBoss EAP pour que la plate-forme puisse accepter des requêtes en provenance d'HTTPD externe ».
- Le package JBoss Native Components ZIP peut être téléchargé à partir du Portail Clients à https://access.redhat.com.

**Procédure 16.13. Extraire et Installer le connecteur NSAPI**

1. **Extraire le package JBoss Native Components.**
   Le reste de cette procédure assume que le package de Native Components est extrait d'un répertoire nommé **`connectors/`** qui se trouve dans **`/opt/oracle/webserver7/config/`**. Pour le reste de cette procédure, ce répertoire sera identifié comme *IPLANET_CONFIG*. Si votre répertoire de configuration Oracle iPlanet est différent, ou si vous exécutez Oracle iPlanet Web Server 6, modifier la procédure en fonction.

2. **Désactiver les mappages du servlet.**
   Ouvrir le fichier ***`IPLANET_CONFIG`*`/default.web.xml`** et chercher la section avec l'en-tête **`Built In Server Mappings`**. Désactiver les mappages pour les trois servlets suivantes, en les entourant des caractères de commentaire XML (**`<!--`** et **`-->`**).

   - défaut
   - invoker
   - jsp

   L'exemple de configuration suivant montre les mappages désactivés.

```
<!-- ============== Built In Servlet Mappings =============== -->
<!-- The servlet mappings for the built in servlets defined above. -
->
<!-- The mapping for the default servlet -->
<!--servlet-mapping>
 <servlet-name>default</servlet-name>
 <url-pattern>/</url-pattern>
```

```
</servlet-mapping-->
<!-- The mapping for the invoker servlet -->
<!--servlet-mapping>
 <servlet-name>invoker</servlet-name>
 <url-pattern>/servlet/*</url-pattern>
</servlet-mapping-->
<!-- The mapping for the JSP servlet -->
<!--servlet-mapping>
 <servlet-name>jsp</servlet-name>
 <url-pattern>*.jsp</url-pattern>
</servlet-mapping-->
```

Sauvegarder et sortir du fichier.

3. **Configurer iPlanet Web Server pour qu'il puisse charger le module de connecteur NSAPI.** Ajouter les lignes suivantes à la fin de ce fichier ***IPLANET_CONFIG*/magnus.conf**, en modifiant les chemins de fichiers pour qu'ils s'accordent avec votre configuration. Ces lignes définissent l'emplacement du module **nsapi_redirector.so**, ainsi que celle du fichier **workers.properties**, qui liste les worker nodes et leurs propriétés.

```
Init fn="load-modules" funcs="jk_init,jk_service"
shlib="IPLANET_CONFIG/connectors/lib/nsapi_redirector.so"
shlib_flags="(global|now)"
Init fn="jk_init"
worker_file="IPLANET_CONFIG/connectors/workers.properties"
log_level="debug"
log_file="IPLANET_CONFIG/config/connectors/nsapi.log"
shm_file="IPLANET_CONFIG/conf/connectors/jk_shm"
```

La configuration ci-dessus est basée sur une architecture 32-bit. Si vous utilisez 64-bit Solaris, changez le string **lib/nsapi_redirector.so** en **lib64/nsapi_redirector.so**.

Sauvegarder et sortir du fichier.

4. **Configurer le connecteur NSAPI**
Vous pouvez configurer le connecteur NSAPI pour une configuration de base, avec aucun équilibrage des charges, ou une configuration d'équilibrage de des charges. Choisissez l'une des options suivantes, après quoi votre configuration sera terminée

   - Section 16.9.3, « Configurer NSAPI en connecteur de base HTTP »
   - Section 16.9.4, « Configurer NSAPI en tant que Cluster d'équilibrage des charges »

Report a bug

### 16.9.3. Configurer NSAPI en connecteur de base HTTP

#### Aperçu

Cette tâche configure le connecteur NSAPI à rediriger les demandes des clients aux serveurs JBoss Enterprise Application Platform sans aucun équilibrage de charge ou basculement. La redirection se fait sur la base d'un déploiement (est donc basé-URL). Pour une configuration d'équilibrage des charges, consultez Section 16.9.4, « Configurer NSAPI en tant que Cluster d'équilibrage des charges » à la place.

#### Prérequis

- Vous devez compléter Section 16.9.2, « Configurer le connecteur NSAPI dans Oracle Solaris » avant de continuer avec la tâche suivante.

**Procédure 16.14. Installer le connecteur HHP de base**

1. **Définir les chemins URL pour redirection vers les serveurs de la plate-forme JBoss EAP.**

**NOTE**

Dans ***IPLANET_CONFIG*/obj.conf**, les espaces ne sont pas autorisés en début de ligne, sauf quand la ligne est en continuation de la ligne précédente.

Modifier le fichier ***IPLANET_CONFIG*/obj.conf**. Chercher la section qui commence par **<Object name="default">**, et ajouter chaque modèle d'URL à son correspondant, sous le format montré dans le fichier exemple ci-dessous. Le string **jknsapi** fait référence au connecteur HTTP qui sera défini dans la prochaine étape. L'exemple montre comment utiliser les caractère génériques pour la correspondance de modèles.

```
<Object name="default">
[...]
NameTrans fn="assign-name" from="/status" name="jknsapi"
NameTrans fn="assign-name" from="/images(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/css(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/nc(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/jmx-console(|/*)" name="jknsapi"
</Object>
```

2. **Définir le worker qui sert chaque chemin d'accès.**
Continuer à modifier le fichier ***IPLANET_CONFIG*/obj.conf**. Ajouter ce qui suit directement après la balise de fermeture de la section que vous venez de finir d'éditer : **</Object>**.

```
<Object name="jknsapi">
ObjectType fn=force-type type=text/plain
Service fn="jk_service" worker="worker01" path="/status"
Service fn="jk_service" worker="worker02" path="/nc(/*)"
Service fn="jk_service" worker="worker01"
</Object>
```

L'exemple ci-dessus redirige les requêtes vers le chemin URL **/status** et le worker nommé **worker01**, et tous les URL en-dessous **/nc/** vers le worker **worker02**. La troisième ligne indique que tous les URL assignés à l'objet **jknsapi** qui n'ont pas de correspondance avec les lignes précédentes sont servis à **worker01**.

Sauvegarder et sortir du fichier

3. **Définir les workers et leurs attributs.**
Créer un fichier intitulé **workers.properties** dans le répertoire **IPLANET_CONFIG/connectors/**. Coller les commentaires suivants dans le fichier, et les modifier pour qu'il conviennent à votre environnement.

```
# An entry that lists all the workers defined
```

```
worker.list=worker01, worker02

# Entries that define the host and port associated with these
workers
worker.worker01.host=127.0.0.1
worker.worker01.port=8009
worker.worker01.type=ajp13

worker.worker02.host=127.0.0.100
worker.worker02.port=8009
worker.worker02.type=ajp13
```

Le fichier **`workers.properties`** utilise la même syntaxe qu'Apache mod_jk. Pour obtenir plus d'informations sur les options disponibles, voir Section 16.6.4, « Référence de configuration des Apache Mod_jk Workers ».

Sauvegarder et sortir du fichier

4. **Redémarrer le serveur iPlanet Web Server.**
Choisir une des procédures suivantes, suivant que vous souhaitez exécuter iPlanet Web Server 6.1 ou 7.0.

   - **iPlanet Web Server 6.1**

     ```
     IPLANET_CONFIG/../stop
     IPLANET_CONFIG/../start
     ```

   - **iPlanet Web Server 7.0**

     ```
     IPLANET_CONFIG/../bin/stopserv
     IPLANET_CONFIG/../bin/startserv
     ```

**Résultat**

iPlanet Web Server envoie maintenant les requêtes clients aux URL que vous avez configurées vers les déploiements de JBoss Enterprise Application Platform.

Report a bug

### 16.9.4. Configurer NSAPI en tant que Cluster d'équilibrage des charges

**Aperçu**

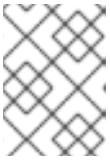
Cette tâche configure le connecteur NSAPI à rediriger les demandes des clients aux serveurs JBoss Enterprise Application Platform dans une configuration d'équilibrage des charges. Pour utiliser NSAPI comme simple connecteur HTTP sans équilibrage des charges, voir Section 16.9.3, « Configurer NSAPI en connecteur de base HTTP ».

**Prérequis**

- Vous devez compléter Section 16.9.2, « Configurer le connecteur NSAPI dans Oracle Solaris » avant de continuer avec la tâche suivante.

**Procédure 16.15. Configurer le connecteur pour l'équilibrage des charges**

1. **Définir les chemins URL pour redirection vers les serveurs de la plate-forme JBoss EAP.**



**NOTE**

Dans ***IPLANET_CONFIG*/obj.conf**, les espaces ne sont pas autorisés en début de ligne, sauf quand la ligne est en continuation de la ligne précédente.

Modifier le fichier ***IPLANET_CONFIG*/obj.conf**. Chercher la section qui commence par **<Object name="default">**, et ajouter chaque modèle d'URL à son correspondant, sous le format montré dans le fichier exemple ci-dessous. Le string **jknsapi** fait référence au connecteur HTTP qui sera défini dans la prochaine étape. L'exemple montre comment utiliser les caractère génériques pour la correspondance de modèles.

```
<Object name="default">
[...]
NameTrans fn="assign-name" from="/status" name="jknsapi"
NameTrans fn="assign-name" from="/images(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/css(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/nc(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/jmx-console(|/*)" name="jknsapi"
NameTrans fn="assign-name" from="/jkmanager/*" name="jknsapi"
</Object>
```

2. **Définir le worker qui sert chaque chemin d'accès.**
Continuer à modifier le fichier ***IPLANET_CONFIG*/obj.conf**. Ajouter ce qui suit directement après la balise de fermeture de la section que vous venez de finir d'éditer dans l'étape précédente (**</Object>**), et modifiez là suivant vos besoins :

```
<Object name="jknsapi">
ObjectType fn=force-type type=text/plain
Service fn="jk_service" worker="status" path="/jkmanager(/*)"
Service fn="jk_service" worker="router"
</Object>
```

Cet objet **jksnapi** définit les noeuds de worker utilisés pour servir chaque chemin relié au mappage **name="jksnapi"** de l'objet **default**. Tout sauf les URL correspondant à **/jkmanager/*** sont redirigés au worker nommé **router**.

3. **Définir les workers et leurs attributs.**
Créer un fichier intitulé **workers.properties** dans le répertoire **IPLANET_CONFIG/conf/connector/**. Coller les commentaires suivants dans le fichier, et les modifier pour qu'il conviennent à votre environnement.

```
# The advanced router LB worker
# A list of each worker
worker.list=router,status

# First JBoss Enterprise Application Platform server
# (worker node) definition.
# Port 8009 is the standard port for AJP
#

worker.worker01.port=8009
worker.worker01.host=127.0.0.1
```

```
worker.worker01.type=ajp13
worker.worker01.ping_mode=A
worker.worker01.socket_timeout=10
worker.worker01.lbfactor=3

# Second JBoss Enterprise Application Platform server
worker.worker02.port=8009
worker.worker02.host=127.0.0.100
worker.worker02.type=ajp13
worker.worker02.ping_mode=A
worker.worker02.socket_timeout=10
worker.worker02.lbfactor=1

# Define the load-balancer called "router"
worker.router.type=lb
worker.router.balance_workers=worker01,worker02

# Define the status worker
worker.status.type=status
```

Le fichier **`workers.properties`** utilise la même syntaxe qu'Apache mod_jk. Pour obtenir plus d'informations sur les options disponibles, voir Section 16.6.4, « Référence de configuration des Apache Mod_jk Workers ».

Sauvegarder et sortir du fichier.

4. **Redémarrer le serveur iPlanet Web Server.**
   Choisir une des procédures suivantes, suivant que vous souhaitez exécuter iPlanet Web Server 6.1 ou 7.0.

   - **iPlanet Web Server 6.1**

     ```
     IPLANET_CONFIG/../stop
     IPLANET_CONFIG/../start
     ```

   - **iPlanet Web Server 7.0**

     ```
     IPLANET_CONFIG/../bin/stopserv
     IPLANET_CONFIG/../bin/startserv
     ```

**Résultat**

iPlanet Web Server redirige les modèles d'URL que vous avez configurés à vos serveurs JBoss Enterprise Application Platform dans une configuration d'équilibrage des charges.

Report a bug

# CHAPITRE 17. MESSAGERIE

## 17.1. INTRODUCTION

### 17.1.1. HornetQ

HornetQ est un système de messagerie multiprotocole, asynchrone développé par Red Hat. HornetQ procure une haute disponibilité (HA) avec basculement automatique des clients pour garantir la fiabilité du message dans le cas d'une panne de serveur. HornetQ prend également en charge des solutions de clustering flexibles avec équilibrage de charge des messages.

Report a bug

### 17.1.2. Java Messaging Service (JMS)

Les systèmes de messagerie vous permettent de coupler de façon informelle des systèmes hétérogènes avec une fiabilité supplémentaire. Les fournisseurs de Service JMS (Java Messaging) utilisent un système de transactions, pour valider ou annuler les modifications atomiquement. Contrairement aux systèmes basés sur un modèle d'échange de données informatisé (RPC, Remote Procedure Call), les systèmes de messagerie utilisent principalement un modèle de passage de messages asynchrone avec aucune relation réelle entre les demandes et les réponses. La plupart des systèmes de messagerie supportent également un mode de requête-réponse, mais ce n'est pas une caractéristique principale des systèmes de messagerie.

Les systèmes de messagerie découplent les expéditeurs des messages des consommateurs de messages. Les expéditeurs et les consommateurs de messages sont complètement indépendantes et ne savent rien l'un de l'autre. Cela vous permet de créer des systèmes flexible et faiblement couplés. Souvent, les grandes entreprises utilisent un système de messagerie pour mettre en place un bus de messages qui couple légèrement des systèmes hétérogènes. Les bus de messages forment souvent la base d'une Enterprise Service Bus (ESB). En utilisant un Message Bus pour découpler des systèmes disparates, on peut permettre au système de croître et de s'adapter plus facilement. Cela permet également plus de souplesse pour ajouter de nouveaux systèmes ou en retirer des "anciens", puisqu'ils n'ont pas de dépendances fragiles les uns avec les autres.

Report a bug

### 17.1.3. Styles de messagerie pris en compte

HornetQ prend en charge les styles de messagerie suivants :

**Modèle de file d'attente de messages**

Le modèle de file d'attente de message consiste à envoyer un message à une file d'attente. Une fois dans la file d'attente, le message est normalement rendu persistant pour en garantir sa livraison. Une fois que le message s'est déplacé dans la file, le système de messagerie le livre à un consommateur de messages. Le consommateur de messages accuse réception de la livraison du message une fois qu'il a été traité.

En messagerie PPP, le modèle de file d'attente de messagerie autorise plusieurs consommateurs pour une même file d'attente, mais chaque message ne peut être reçu que par un seul consommateur.

**Modèle Publish-Subscribe**

Le modèle Publish-Subscribe permet à plusieurs émetteurs d'envoyer des messages vers une seule entité sur le serveur. Cette entité est connue sous le nom de "topic". Chaque topic peut être traité par plusieurs consommateurs, ce que l'on appelle des "abonnements" (ou "subscriptions" en anglais)

Chaque abonnement reçoit une copie des messages envoyés au topic. La différence avec le modèle de file d'attente de messages, c'est que chaque message n'est consommé que par un seul consommateur.

Les abonnements qui sont durables conservent une copie de chaque message envoyé à ce topic jusqu'à ce que l'abonné les consomme. Ces copies sont conservées même en cas d'un redémarrage du serveur. Les abonnements non durables durent le temps de la connexion qui les a créés.

Report a bug

## 17.2. ACCEPTEURS ET CONNECTEURS

HornetQ utilise le concept de connecteurs et d'accepteurs comme un élément clé du système de messagerie.

### Accepteurs et Connecteurs

**`Acceptor`**

Un accepteur définit quels types de connections sont acceptées par le serveur HornetQ.

**`Connector`**

Un connecteur définit comment se connecter au serveur HornetQ, et est utilisé par le client HornetQ.

Il y a deux sortes de connecteurs et d'accepteurs, suivant que le connecteur ou l'accepteur qui correspond sont dans la même JVM ou non.

### Invm et Netty

**`Invm`**

Invm est un acronyme pour Intra Virtual Machine. Peut être utilisé quand le client et le serveur exécutent en même temps dans la même JVM.

**`Netty`**

Le nom d'un porjet JBoss. Doit être utilisé quand le client et le serveur exécutent dans les JVM différentes.

Un client HornetQ doit utiliser un connecteur compatible avec un des accepteurs du serveur. Seulement un connecteur Invm peut se connecter à un accepteur Invm, et seulement un connecteur netty peut se connecter à un accepteur de netty. Les connecteurs et les accepteurs sont configurés sur le serveur dans un **`standalone.xml`** et **`domain.xml`**. Vous pouvez utiliser la Console de gestion ou de la CLI de gestion pour les définir

**Exemple 17.1. Exemple de configuration du connecteur et de l'accepteur par défaut**

```
<connectors>
    <netty-connector name="netty" socket-binding="messaging"/>
```

```
    <netty-connector name="netty-throughput" socket-binding="messaging-
throughput">
        <param key="batch-delay" value="50"/>
    </netty-connector>
    <in-vm-connector name="in-vm" server-id="0"/>
</connectors>
<acceptors>
    <netty-acceptor name="netty" socket-binding="messaging"/>
    <netty-acceptor name="netty-throughput" socket-binding="messaging-
throughput">
        <param key="batch-delay" value="50"/>
        <param key="direct-deliver" value="false"/>
    </netty-acceptor>
    <in-vm-acceptor name="in-vm" server-id="0"/>
</acceptors>
```

La configuration de l'exemple montre également comment l'application JBoss Enterprise Application Platform 6 de HornetQ utilise des liaisons de socket dans la configuration du connecteur et l'accepteur. Cela diffère de la version autonome de HornetQ, qui vous oblige à déclarer les ports et les hôtes spécifiques.

Report a bug

## 17.3. LES PONTS

La fonction des ponts est de consommer des messages à partir d'une file d'attente de source, et de les envoyer vers une adresse cible, qui se trouve normalement sur un serveur HornetQ. Les ponts s'accommodent de connections non fiables, et se reconnectent automatiquement quand les connexions sont rendues disponibles à nouveau. Les ponts HornetQ peuvent être configurés avec des expressions de filtre pour n'envoyer que certains messages.

Report a bug

## 17.4. JNDI (JAVA NAMING AND DIRECTORY INTERFACE)

L'API *Java Naming and Directory Interface (JNDI)* est un API standard Java pour les services de répertoire et de nommage. Il permet aux technologies basées-Java de découvrir ou d'organiser des composants de noms dans un environnement informatique distribué.

Report a bug

## 17.5. TRAVAILLER AVEC DES MESSAGES VOLUMINEUX

HornetQ prend en charge l'utilisation de messages volumineux, même lorsque le client ou le serveur a limité la quantité de mémoire. Les messages volumineux peuvent être traités tels quels, ou comprimés davantage pour un transfert plus efficace.

Report a bug

## 17.6. CONFIGURATION

### 17.6.1. Configurer le Serveur JMS

Pour configurer le JMS Server d'HornetQ, modifier le fichier de configuration du serveur. La configuration du serveur se trouve dans le fichier ***EAP_HOME*/domain/configuration/domain.xml** des serveurs du domaine, ou dans le fichier ***EAP_HOME*/standalone/configuration/standalone.xml** des serveurs autonomes.

L'élément **<subsystem xmlns="urn:jboss:domain:messaging:1.2">** contient toute la configuration JMS. Ajouter toute instance **ConnectionFactory**, **Queue**, ou **Topic** requise pour le JNDI.

1. **Activer le sous-système JMS de la plateforme d'applications JBoss Enterprise.**
   Dans l'élément **<extensions>**, vérifier que la ligne suivante est bien présente et n'est pas dé-commentée :

   ```
   <extension module="org.jboss.as.messaging"/>
   ```

2. **Ajouter le sous-système JMS de base.**
   Si le sous-système de Messagerie n'est pas présent dans votre fichier de configuration, ajoutez-le.

   a. Cherchez le **<profile>** qui correspond à celui que vous utilisez, et chercher sa balise de **<subsystems>**.

   b. Ajouter une nouvelle ligne sous la balise **<subsystems>**. Coller ceci à l'intérieur :

   ```
   <subsystem xmlns="urn:jboss:domain:messaging:1.2">

   </subsystem>
   ```

   Toutes les configurations supplémentaires pourront être ajoutées à la ligne vide ci-dessus.

3. **Ajouter la configuration de base à JMS.**

   ```
   <journal-file-size>102400</journal-file-size>
   <journal-min-files>2</journal-min-files>
   <journal-type>NIO</journal-type>
   <!-- disable messaging persistence -->
   <persistence-enabled>false</persistence-enabled>
   ```

   Personnaliser les valeurs ci-dessus pour qu'elles correspondent à vos besoins.

   **AVERTISSEMENT**

   La valeur de **journal-file-size** doit être plus élevée que celle de la taille du message envoyé au serveur, ou bien le serveur ne pourra pas stocker le message.

4. **Ajouter les instances de fabrique de connexion à HornetQ**

Le client utilise un objet **ConnectionFactory** JMS pour faire des connexions au serveur. Pour ajouter un objet de fabrique de connexion JMS à HornetQ, inclure une simple balise **<jms-connection-factories>** et un élément **<connection-factory>** pour chaque fabrique de connexion comme suit :

```
<subsystem xmlns="urn:jboss:domain:messaging:1.2">
  ...
  <jms-connection-factories>
    <connection-factory name="myConnectionFactory">
      <connectors>
        <connector-ref connector-name="netty"/>
      </connectors>
      <entries>
        <entry name="/ConnectionFactory"/>
      </entries>
    </connection-factory>
  </jms-connection-factories>
  ...
</subsystem>
```

5. **Configurer le connecteur `netty`**
   La fabrique de connexion JMS utilise un connecteur **netty**. Il s'agit d'une référence à un objet de connecteur déployé dans le fichier de configuration du serveur. L'objet de connecteur détermine le transport et les paramètres utilisés pour vous connecter au serveur.

   Pour configurer le connecteur **netty**, inclure les paramètres suivants :

```
<subsystem xmlns="urn:jboss:domain:messaging:1.2">
  ...
  <connectors>
    <netty-connector name="netty" socket-binding="messaging"/>
    <netty-connector name="netty-throughput" socket-
binding="messaging-throughput">
      <param key="batch-delay" value="50"/>
    </netty-connector>
    <in-vm-connector name="in-vm" server-id="0"/>
  </connectors>
  ...
</subsystem>
```

   Le connecteur référence les liaisons de socket de **messaging** et de **messaging-throughput**. La liaison de socket de **messaging** utilise le port 5445, et la liaison de socket **messaging-throughput** utilise le port 5455. Veillez à ce que les liaisons de socket suivantes sont présentes dans l'élément **<socket-binding-groups>** :

```
<socket-binding-groups>
  ...
  <socket-binding-group ... >
    <socket-binding name="messaging" port="5445"/>
    <socket-binding name="messaging-throughput" port="5455"/>
    ...
  </socket-binding-group>
  ...
</socket-binding-groups>
```

6. **Ajouter les instances de file d'attente à HornetQ**
   Il y a quatre façons de configurer les instances de files d'attente (ou destinations JMS) pour HornetQ.

   - Utiliser la Console de gestion

     Pour utiliser la Console de gestion, le serveur devra être démarré sous le mode **`Message-Enabled`**. Vous y parviendrez en utilisant l'option **`-c`** et en forçant l'utilisation du fichier de configuration **`standalone-full.xml`** (pour les serveurs autonomes). Ainsi, en mode autonome, ce qui suit démarrera le serveur en mode activation de message.

     ```
     ./standalone.sh -c standalone-full.xml
     ```

     Une fois que le serveur a démarré, connectez-vous à la Console de gestion et naviguez dans: Profile → Messaging → Destinations → default → View, puis, cliquer sur le bouton Ajouter pour saisir les détails de la destination JMS.

   - Utiliser le Management CLI:

     Tout d'abord, connectez-vous au ManagementLI:

     ```
      bin/jboss-cli.sh --connect
     ```

     Puis, passez au sous-système de messagerie :

     ```
     cd /subsystem=messaging/hornetq-server=default
     ```

     Finalement, exécuter une opération Ajouter, en remplaçant les exemples de valeurs données ci-dessous avec les vôtres :

     ```
     ./jms-queue=testQueue:add(durable=false,entries=
     ["java:jboss/exported/jms/queue/test"])
     ```

   - Créer un fichier de configuration JMS et y ajouter le dossier de déploiements

     Commencer à créer un fichier de configuration JMS: *example-jms.xml*. Ajouter y les entrées suivantes, en remplaçant les valeurs avec les vôtres.

     ```
     <?xml version="1.0" encoding="UTF-8"?>         <messaging-
     deployment xmlns="urn:jboss:messaging-deployment:1.0">
         <hornetq-server>
             <jms-destinations>
                 <jms-queue name="testQueue">
                     <entry name="queue/test"/>
                     <entry
     name="java:jboss/exported/jms/queue/test"/>
                 </jms-queue>
                 <jms-topic name="testTopic">
                     <entry name="topic/test"/>
                     <entry
     name="java:jboss/exported/jms/topic/test"/>
                 </jms-topic>
     ```

```
        </jms-destinations>
    </hornetq-server>
</messaging-deployment>
```

Sauvegardez ce fichier dans le dossier de déploiements et faire un déploiement.

- Ajouter les entrées dans le fichier de configuration de JBOss EAP.

  En utilisant *standalone-full.xml* comme exemple, chercher le sous-système de messagerie dans ce fichier.

```
<subsystem xmlns="urn:jboss:domain:messaging:1.2">
```

  Ajoutez y les entrées suivantes, encore une fois, en remplaçant les valeurs de l'exemple avec les vôtres. Vous devez ajouter ces entrées après la balise de fin </jms-connection-factories> mais avant l'élément </hornetq-server> :

```
<jms-destinations>
        <jms-queue name="testQueue">
            <entry name="queue/test"/>
            <entry name="java:jboss/exported/jms/queue/test"/>
        </jms-queue>
        <jms-topic name="testTopic">
            <entry name="topic/test"/>
            <entry name="java:jboss/exported/jms/topic/test"/>
        </jms-topic>
    </jms-destinations>
```

7. **Procéder à une configuration supplémentaire**
   Si vous avez besoin de davantage de paramètres de configuration, revoir DTD dans ***EAP_HOME*/docs/schema/jboss-messaging_1_2.xsd**.

Report a bug

### 17.6.2. Configurer JNDI pour HornetQ

**AVERTISSEMENT**

Topic 112, Revision 431803 failed validation and is not included in this build.

### 17.6.3. Configuration des paramètres de l'adresse JMS

Le sous-système JMS comprend plusieurs options configurables qui gèrent différents aspects de la transmission des messages, le nombre de tentatives d'envoi, et quand le message devra expirer. Ces options de configuration sont contenues dans l'élément de configuration **<address-settings>**.

Une des caractéristiques des configurations d'adresse est la syntaxe commune pour faire correspondre des adresses diverses, connue également sous le nom de Wildcard (caractères génériques).

### Syntaxe Wildcard

Les adresses en syntaxe wildcard peuvent être utilisées pour faire correspondre plusieurs adresses similaires avec une seule instruction, ce qui est semblable à la façon dont nombreux systèmes utilisent le caractère astérisque (*****) pour faire correspondre plusieurs fichiers ou chaînes avec une seule recherche. Les caractères suivants ont une signification particulière dans un énoncé wildcard.

**Tableau 17.1. Syntaxe Wildcard JMS**

| Caractère | Description |
|---|---|
| . (point simple) | Marque l'espace entre les mots au sein d'une expression wildcard. |
| # (symbole de hachage) | Fait correspondre une séquence de zéros ou de plusieurs mots. |
| * (un astérisque) | Faire correspondre à un mot unique. |

**Tableau 17.2. Exemples de JMS Wildcards**

| Exemple | Description |
|---|---|
| news.europe.# | Correspond à **`news.europe`**, **`news.europe.sport`**, **`news.europe.politic`**, mais pas à **`news.usa`** or **`europe`**. |
| news. | Correspond à **`news.europe`** mais pas à **`news.europe.sport`**. |
| news.*.sport | Correspond à **`news.europe.sport`** et **`news.usa.sport`**, mais pas à **`news.europe.politics`**. |

**Exemple 17.2. Configuration des paramètres d'adresse par défaut**

Les valeurs de cet exemples sont utilisées pour illustrer le reste de ce topic.

```
<address-settings>
    <!--default for catch all-->
    <address-setting match="#">
        <dead-letter-address>jms.queue.DLQ</dead-letter-address>
        <expiry-address>jms.queue.ExpiryQueue</expiry-address>
        <redelivery-delay>0</redelivery-delay>
        <max-size-bytes>10485760</max-size-bytes>
        <address-full-policy>BLOCK</address-full-policy>
        <message-counter-history-day-limit>10</message-counter-history-
day-limit>
    </address-setting>
</address-settings>
```

**Tableau 17.3. Description de la configuration des paramètres de l'adresse JMS**

| Élément | Description | Valeur par défaut | Type |
|---|---|---|---|
| **address-full-policy** | Détermine ce qui se passe quand une adresse dont la max-size-bytes est spécifiée, est remplie. | PAGE | STRING |
| **dead-letter-address** | Si une adresse de lettres mortes est spécifiée, les messages seront déplacés vers l'adresse de lettres mortes si les tentatives de livraison **max-livraison-tentatives** ont échoué. Dans le cas contraire, ces messages non remis sont ignorés. Les caractères génériques (wildcard) sont autorisés. | jms.queue.DLQ | STRING |
| **expiry-address** | Si l'adresse d'expiration est présente, les messages expirés seront envoyés à l'adresse ou aux adresses correspondantes, au lieu d'être jetés. Les caractères génériques (wildcards) sont autorisés. | jms.queue.ExpiryQueue | STRING |
| **last-value-queue** | Définit si une file d'attente utilise uniquement les dernières valeurs ou non. | false | BOOLÉEN |
| **max-delivery-attempts** | Le nombre max de tentatives d'envoi d'un message avant qu'il soit envoyés à **dead-letter-address** ou qu'il soit ignoré. | 10 | INT |
| **max-size-bytes** | La taille maximum d'octets. | 10485760L | LONG |
| **message-counter-history-day-limit** | Limite en Jour de l'historique du compteur de messages. | 10 | INT |

| Élément | Description | Valeur par défaut | Type |
|---|---|---|---|
| **page-max-cache-size** | Le nombre de pages de fichiers à conserver en mémoire pour optimiser IO en cours de navigation de pagination. | 5 | INT |
| **page-size-bytes** | La taille de pagination. | 5 | INT |
| **redelivery-delay** | La durée entre les tentatives de re-livraison, exprimée en millisecondes. Si défini sur la valeur **0**, les tentatives de re-livraison auront lieu indéfiniment. | 0L | LONG |
| **redistribution-delay** | Définit la durée à attendre lorsque le dernier consommateur est fermé dans une file d'attente avant de pouvoir redistribuer des messages. | -1L | LONG |
| **send-to-dla-on-no-route** | Un paramètre pour une adresse qui définit la condition d'un message non acheminé vers une file d'attente pour qu'il soit envoyé à la place vers une file d'attente des messages morts ou DLA (de l'anglais Dead Letter Queue) indiquée pour cette adresse. | false | BOOLÉEN |

- **Configurer les paramètres de l'adresse et les attributs du modèle**
  Choisir le Management CLI ou la Console de Management pour configurer vos attributs de modèle selon les besoins.

  - **Configurer les paramètres de l'adresse par le Management CLI**
    Utiliser le Management CLI pour configurer les paramètres de l'adresse.

    a. **Ajouter un nouveau Modèle**
    Utiliser l'opération **add** pour créer un nouveau paramètre d'adresse, si nécessaire. Vous pouvez exécuter cette commande à partir de la racine de la session de Management CLI, qui, dans les exemples suivants, crée un nouveau modèle ou motif intitulé *patternname*, avec un attribut **max-delivery-attempts** déclaré à *5*. Voici des exemples pour les modifications sur le serveur autonome et le domaine géré pour le profil **full**.

    ```
    [domain@localhost:9999 /]
    /profile=full/subsystem=messaging/hornetq-
    ```

```
server=default/address-setting=patternname/:add(max-delivery-
attempts=5)
```

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-
server=default/address-setting=patternname/:add(max-delivery-
attempts=5)
```

b. **Modifier les attributs de modèle**
Utiliser l'opération **`write`** pour écrire une nouvelle valeur dans un attribut. Vous pouvez utiliser l'onglet de complétion pour terminer la chaîne de commande en cours, ainsi que pour exposer les attributs disponibles. L'exemple suivant met à jour la valeur de **`max-delivery-attempts`** à *10*

```
[domain@localhost:9999 /]
/profile=full/subsystem=messaging/hornetq-
server=default/address-setting=patternname/:write-
attribute(name=max-delivery-attempts,value=10)
```

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-
server=default/address-setting=patternname/:write-
attribute(name=max-delivery-attempts,value=10)
```

c. **Confirmer les attributs de modèle**
Confirmer que les valeurs ont changé en exécutant **`read-resource`** accompagné du paramètre **`include-runtime=true`** pour exposer toutes les valeurs actives en cours du modèle de serveur.

```
[domain@localhost:9999 /]
/profile=full/subsystem=messaging/hornetq-
server=default/address-setting=patternname/:read-resource
```

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-
server=default/address-setting=patternname/:read-resource
```

- **Configurer les paramètres de l'adresse par la Console de gestion**
Utiliser la Console de gestion pour configurer les paramètres de l'adresse.

a. **Connectez-vous à la Console de gestion.**
Connectez-vous à la Console de gestion de votre domaine géré ou de votre serveur autonome.

b. **Si vous utilisez le Domaine géré, sélectionner le profil qui convient.**
Sélectionner l'onglet **`Profiles`** en haut à droite, puis sélectionner le profil qui convient à partir du menu **`Profile`** en haut et à gauche de l'écran suivant. Seuls les profils **`full`** et **`full-ha`** ont le sous-système **`messaging`** activé.

c. **Sélectionner l'item `Messaging` à partir du menu de navigation.**
Étendre l'item de menu **`Messaging`** à partir du menu de navigation, et cliquer sur **`Destinations`**.

d. **Voir le fournisseur JMS.**

Une liste de fournisseurs JMS s'affiche. Dans la configuration par défaut, on ne voit que le fournisseur par **default**. Cliquer sur le lien **View** pour afficher les paramètres de ce fournisseur en détail.

e. **Voir les paramètre de configuration de l'adresse.**
Cliquer sur l'onglet **Addressing**. Ensuite, vous pourrez soit ajouter un nouveau modèle en cliquant sur le bouton **Add**, ou en modifiant un modèle existant en cliquant sur son nom et en cliquant sur le bouton **Edit**.

f. **Configurer les options.**
Si vous ajoutez un modèle, le champ **Pattern** s'en référera au paramètre **match** de l'élément **address-setting**. Vous pourrez aussi modifier **Dead Letter Address**, **Expiry Address**, **Redelivery Delay**, et **Max Delivery Attempts**. Les autres options doivent être configurées par la Management CLI.

Report a bug

### 17.6.4. Configurer la Messagerie dans HornetQ

La méthode recommandée pour configurer la messagerie dans JBoss Enterprise Application Platform 6 est soit la Console de gestion, soit Management CLI. Vous pouvez effectuer des modifications persistantes avec l'un ou l'autre de ces outils de gestion sans avoir besoin de modifier manuellement les fichiers de configuration **standalone.xml** ou **domain.xml**. Cependant, il est utile de se familiariser avec les composants de messagerie des fichiers de configuration par défaut, où les exemples de documentation utilisant des outils de gestion donnent des extraits de fichiers de configuration comme référence.

Report a bug

### 17.6.5. Configurer la re-livraison différée

#### Introduction

La re-livraison différée est définie dans l'élément **<redelivery-delay>**, qui est un élément dépendant de l'élément de configuration **<address-setting>** de la configuration du sous-système JMS (Java Messaging Service).

```
<!-- delay redelivery of messages for 5s -->
<address-setting match="jms.queue.exampleQueue">
  <redelivery-delay>5000</redelivery-delay>
</address-setting>
```

Si un retard de livraison est spécifié, le système JMS attendra pendant la durée de ce délai avant de re-livrer les messages. Si **<redelivery-delay>** est défini à **0**, il n'y aura pas de livraison à nouveau. Les caractères génériques (wildcards) peuvent être utilisés sur l'élément **<address-setting-match>** pour configurer la livraison à nouveau des adresses qui correspondent au(x) caractère(s) générique(s).

Report a bug

### 17.6.6. Configurer les adresses de lettres mortes

#### Introduction

Une adresse de lettre morte est définie dans l'élément **<address-setting>** de configuration du sous-système de JMS (Java Messaging Service).

```
<!-- undelivered messages in exampleQueue will be sent to the dead letter
address
deadLetterQueue after 3 unsuccessful delivery attempts
-->
<address-setting match="jms.queue.exampleQueue">
  <dead-letter-address>jms.queue.deadLetterQueue</dead-letter-address>
  <max-delivery-attempts>3</max-delivery-attempts>
</address-setting>
```

Si **<dead-letter-address>** n'est pas spécifié, les messages sont supprimés au bout de **<max-delivery-attempts>** envois. Par défaut, les messages sont envoyés 10 fois. Si vous définissez **<max-delivery-attempts>** à **-1** vous autorisez un nombre d'envois indéterminé. Ainsi, une lettre morte peut être définie globalement pour un ensemble d'adresses correspondantes et vous pouvez définir **<max-delivery-attempts>** à **-1** pour qu'une adresse particulière soit configurée sur un nombre d'envois indéfini. Les astérisques peuvent aussi être utilisés pour faire correspondre à un ensemble d'adresses particulier.

Report a bug

### 17.6.7. Configurer les adresses d'expiration de messages

#### Introduction

Les adresses d'expiration de messages sont définies dans la configuration address-setting de JMS (Java Messaging Service). Ainsi :

```
<!-- expired messages in exampleQueue will be sent to the expiry address
expiryQueue -->
<address-setting match="jms.queue.exampleQueue">
  <expiry-address>jms.queue.expiryQueue</expiry-address>
</address-setting>
```

Si les messages sont expirés et qu'aucune adresse d'expiration n'est spécifiée, les messages sont tout simplement retirés de la file d'attente et abandonnés. *Address wildcards* peut également être utilisé pour configurer des plages de données d'adresses d'expiration spécifiques pour un ensemble d'adresses.

#### Wildcards (*)pour les adresses

Les wildcards peuvent être utilisées pour que plusieurs adresses similaires puissent être reconnues en un seul énoncé, de la même façon dont de nombreux systèmes utilisent les astérisques (*****) pour faire correspondre des fichiers ou strings multiples en une seule recherche. Les caractères suivants sont une signification particulière dans un énoncé wildcard.

**Tableau 17.4. Syntaxe Wildcard JMS**

| Caractère | Description |
|---|---|
| . (point simple) | Marque l'espace entre les mots au sein d'une expression wildcard. |
| # (a pound or hash symbol) | Fait correspondre une séquence de zéros ou de plusieurs mots. |
| * (un astérisque) | Faire correspondre à un mot unique. |

| Caractère | Description |
|---|---|

**Tableau 17.5. Exemples de JMS Wildcards**

| Exemple | Description |
|---|---|
| news.europe.# | Correspond à **`news.europe`**, **`news.europe.sport`**, **`news.europe.politic`**, mais pas à **`news.usa`** or **`europe`**. |
| news. | Correspond à **`news.europe`** mais pas à **`news.europe.sport`**. |
| news.*.sport | Correspond à **`news.europe.sport`** et **`news.usa.sport`**, mais pas à **`news.europe.politics`**. |

Report a bug

### 17.6.8. Référence pour les attributs de configuration d'HornetQ

L'implémentation d'HornetQ de JBoss Enterprise Application Platform 6 expose les attributs de configuration suivants. Vous pouvez utiliser le Management CLI pour exposer plus particulièrement les attributs configurables ou affichables par l'opération **`read-resource`**.

**Exemple 17.3. Exemple**

```
[standalone@localhost:9999 /] /subsystem=messaging/hornetq-
server=default:read-resource
```

**Tableau 17.6. Attributs HornetQ**

| Attribut | Exemple de valeur | Type |
|---|---|---|
| **`allow-failback`** | true | BOOLÉEN |
| **`async-connection-execution-enabled`** | true | BOOLÉEN |
| **`backup`** | false | BOOLÉEN |
| **`cluster-password`** | *somethingsecure* | STRING |

| Attribut | Exemple de valeur | Type |
|---|---|---|
| **cluster-user** | HORNETQ.CLUSTER.ADMIN.USER | STRING |
| **clustered** | false | BOOLÉEN |
| **connection-ttl-override** | -1 | LONG |
| **create-bindings-dir** | true | BOOLÉEN |
| **create-journal-dir** | true | BOOLÉEN |
| **failback-delay** | 5000 | LONG |
| **failover-on-shutdown** | false | BOOLÉEN |
| **id-cache-size** | 2000 | INT |
| **jmx-domain** | org.hornetq | STRING |
| **jmx-management-enabled** | false | BOOLÉEN |
| **journal-buffer-size** | 100 | LONG |
| **journal-buffer-timeout** | 100 | LONG |
| **journal-compact-min-files** | 10 | INT |
| **journal-compact-percentage** | 30 | INT |
| **journal-file-size** | 102400 | LONG |
| **journal-max-io** | 1 | INT |
| **journal-min-files** | 2 | INT |

| Attribut | Exemple de valeur | Type |
|---|---|---|
| **journal-sync-non-transactional** | true | BOOLÉEN |
| **journal-sync-transactional** | true | BOOLÉEN |
| **journal-type** | ASYNCIO | STRING |
| **live-connector-ref** | référence | STRING |
| **log-journal-write-rate** | false | BOOLÉEN |
| **management-address** | jms.queue.hornetq.management | STRING |
| **management-notification-address** | hornetq.notifications | STRING |
| **memory-measure-interval** | -1 | LONG |
| **memory-warning-threshold** | 25 | INT |
| **message-counter-enabled** | false | BOOLÉEN |
| **message-counter-max-day-history** | 10 | INT |
| **message-counter-sample-period** | 10000 | LONG |
| **message-expiry-scan-period** | 30000 | LONG |
| **message-expiry-thread-priority** | 3 | INT |
| **page-max-concurrent-io** | 5 | INT |
| **perf-blast-pages** | -1 | INT |

| Attribut | Exemple de valeur | Type |
|---|---|---|
| **persist-delivery-count-before-delivery** | false | BOOLÉEN |
| **persist-id-cache** | true | BOOLÉEN |
| **persistence-enabled** | true | BOOLÉEN |
| **remoting-interceptors** | Non défini | LIST |
| **run-sync-speed-test** | false | BOOLÉEN |
| **scheduled-thread-pool-max-size** | 5 | INT |
| **security-domain** | autre | STRING |
| **security-enabled** | true | BOOLÉEN |
| **security-invalidation-interval** | 10000 | LONG |
| **server-dump-interval** | -1 | LONG |
| **shared-store** | true | BOOLÉEN |
| **started** | true | BOOLÉEN |
| **thread-pool-max-size** | 30 | INT |
| **transaction-timeout** | 300000 | LONG |
| **transaction-timeout-scan-period** | 1000 | LONG |
| **version** | 2.2.16.Final (HQ_2_2_16_FINAL, 122) | STRING |

| Attribut | Exemple de valeur | Type |
| --- | --- | --- |
| **`wild-card-routing-enabled`** | true | BOOLÉEN |

**AVERTISSEMENT**

La valeur de **`journal-file-size`** doit être plus élevée que celle de la taille du message envoyé au serveur, ou bien le serveur ne pourra pas stocker le message.

Report a bug

### 17.6.9. Définir l'expiration des messages

**Introduction**

Avec Hornet Core API, l'expiration peut être définie directement sur le message. Par exemple :

```
// message will expire in 5000ms from now
message.setExpiration(System.currentTimeMillis() + 5000);
```

**JMS `MessageProducer`**

JMS **`MessageProducer`** inclut un paramètre **`TimeToLive`** qui contrôle l'expiration de message du message qu'il envoie :

```
// messages sent by this producer will be retained for 5s (5000ms) before
expiration
producer.setTimeToLive(5000);
```

Messages expirés qui sont consommés à partir d'une adresse d'expiration ont les propriétés suivantes :

- _HQ_ORIG_ADDRESS

Une propriété de string qui contient l'adresse d'origine du message expiré.

- _HQ_ACTUAL_EXPIRY

Une propriété longue qui contient l'expiration du message expiré.

Report a bug

## 17.7. PERSISTANCE

### 17.7.1. Persistance dans HornetQ

HornetQ gère sa propre persistance. Il est livré avec un journal de haute performance, qui est optimisé pour les cas d'utilisation de messagerie spécifique.

Le journal HornetQ journal est en «append» uniquement avec une taille de fichier configurable, ce qui améliore les performances en permettant des opérations d'écriture simples. Il se compose d'un ensemble de fichiers sur le disque, qui sont initialement pré-créés à une taille fixe et remplis. au fur et à mesure que les opérations de serveur (ajouter un message, supprimer le message, mise à jour de message, etc.) sont effectuées, les enregistrements des opérations sont ajoutées au journal jusqu'à ce que le fichier journal soit plein, moment à partir duquel le fichier journal suivant est utilisé.

Un algorithme de Nettoyage de la mémoire

Le journal supporte également les transaction locales et XA.

La majorité du journal est imprimée en Java, mais l'interaction avec le système de fichier est rendu abstrait pour autoriser plusieurs implémentations enfichables. Les deux implémentations livrées avec HornetQ sont :

- *Java Non-blocking IO (NIO)*

  Utilise Java NIO pour l'interface avec le système de fichiers. Cela donne une excellence performance et exécute sur n'importe quelle plate-forme avec Java 6 ou un runtime plus récent.

- *Linux Asynchronous IO (AIO)*

  Utilise un encapsuleur de code natif pour parler à la bibliothèque d'e/s asynchrone Linux (AIO). Avec AIO, HornetQ reçoit un message lorsque les données ont été rendues persistantes. Cette commande supprime le besoin de synchronisation explicite. AIO fournira généralement une meilleure performance que Java NIO, mais nécessite le noyau Linux 2.6 ou version ultérieur et le paquet libaio.

  AIO nécessite également les systèmes de fichiers ext2, ext3, ext4, jfs or xfs.

Le serveur standard HornetQ utilise les instances de journaux suivants :

- *bindings journal*

  Stocke les données relatives à des liaisons, y compris l'ensemble des files d'attente déployées sur le serveur et leurs attributs. Il stocke également des données comme les compteurs de séquence ID. Le journal de liaisons est toujours un journal NIO, car il a généralement un débit faible en comparaison au journal de messages.

  Les fichiers de ce journal ont comme préfixe hornetq-bindings. Chaue fichier a des extensions de liaison. La taille du fichier est de 1048576 octets, et le fichier se trouve dans le dossier de liaisons.

- *JMS journal*

  Stocke toutes les données liées à JMS, comme les files d'attente JMS, les sujets ou fabriques de connexions et toutes les liaisons JNDI de ces ressources. Toutes les ressources JMS créées avec l'API de gestion sont persistées dans ce journal. Toutes les ressources configurées par des fichiers de configuration ne le sont pas. Ce journal n'est créé que si JMS est utilisé.

- *message journal*

Stocke toutes les données liées à des messages, y compris les messages eux-mêmes et les duplicate-id caches. Par défaut, HornetQ utiliser AIO pour son journal. Si AIO est disponible, il retombera automatiquement sur NIO.

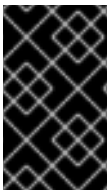
Les gros messages sont persistés en dehors du journal de messages. Dans les situations de moindre mémoire, configurer HornetQ pour qu'il envoie les messages sur le disque. Si la persistance n'est pas requise, HornetQ peut être configuré pour ne persister aucune donnée.

Report a bug

## 17.8. HAUTE DISPONIBILITÉ

### 17.8.1. HornetQ Shared Stores

Lorsque vous utilisez un magasin partagé (Shared Store), les serveurs live et de sauvegarde partagent le répertoire de données même, ensemble, à l'aide d'un système de fichiers partagé. Cela inclut le répertoire de pagination, le répertoire de journaux, des messages volumineux et le journal de liaison. Lorsque le basculement et le serveur de sauvegarde reprennent, il chargent le stockage persistant de système de fichiers partagé. Les clients peuvent alors s'y connecter.



**IMPORTANT**

HornetQ prend en charge les stores GFS2 sur SAN (Storage Area Network), ainsi que la haute disponibilité sur NFSv4. L'attribut de type de journal doit être défini à ASYNCIO pour ces options, car NIO ne peut pas être utilisé en haute disponibilité.

**IMPORTANT**

HornetQ supporte NFS, sous des directives strictes de configuration qui sont soulignées ci-dessous.

Cette forme de haute disponibilité diffère que la réplication de données, car elle requiert que le système de fichiers soit accessible à la fois par les noeuds de sauvegarde live et de sauvegarde. Cela correspondra le plus souvent à un SAN de haute performance.

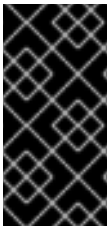
**AVERTISSEMENT**

Ne pas utiliser les montages NFS pour stocker un journal avec NIO (non-blocking I/O), à moins que vous utilisiez Red Hat Enterprise Linux. Cela est dû à l'implémentation NFS utilisée.

L'implémentation NFS de Red Hat Linux utilisée supporte à la fois le direct I/O (ouverture des fichiers avec l'indicateur O_DIRECT défini), et l'I/O asynchrone basé noyau. Avec ces deux fonctionnalités présentes, il est possible d'utiliser NFS comme option de stockage, sous conditions de configuration strictes :

- HornetQ doit être configuré pour utiliser le type de journal ASYNCIO.

- Le cache client Red Hat Enterprise Linux NFS doit être désactivé.



**IMPORTANT**

Le journal du serveur doit être vérifié après le démarrage de JBoss Enterprise Application Platform 6, pour s'assurer que la bibliothèque native est bien chargée, et que le type de journal ASYNCIO est utilisé. Si la bibliothèque native ne se charge pas, HornetQ échouera dans le journal NIO, et cela va être précisé dans le journal du serveur.

**IMPORTANT**

La bibliothèque native qui implémente des e/s asynchrones exige que **`libaio`** soit installée sur le système Red Hat Enterprise Linux sur lequel JBoss Enterprise Application Platform 6 exécute.

**NOTE**

Il est recommandé que, si vous utilisez NFS en vertu des stipulations ci-dessus, une configuration NFS hautement disponible soit utilisée.

L'avantage de share-store haute disponibilité est qu'aucune réplication ne se produit entre les nœuds live et de sauvegarde. Autrement dit, il n'y a pas de dégradation des performances en raison de la surcharge de réplication pendant le fonctionnement normal.

L'inconvénient la réplication shared-store est qu'elle nécessite un système de fichiers partagé, et que lorsque le serveur de sauvegarde est activé, il faut charger le journal à partir d'un shared-store. Cela peut prendre un certain temps, selon la quantité de données dans le store.

S'il est exigé d'avoir des performances élevées durant le fonctionnement normal sont requise, il y a accès à un réseau SAN rapide et un taux de basculement légèrement plus lent est acceptable (en fonction de la quantité de données). Shared-store haute disponibilité est recommandée.

Report a bug

### 17.8.2. High-availability (HA) Failover

High-availability Failover est disponible, soit en basculement automatique des clients, ou en basculement au niveau des applications, grâce à une structure de live-backup. Chaque serveur est un serveur de sauvegarde, qui peut également être sauvegardé par autant de serveurs que nécessaire.

Le serveur de sauvegarde ne prend le relais que si le serveur se plante ou en cas de basculement. Simultanément, l'un des serveurs de sauvegarde secondaires reprend comme serveur de sauvegarde passif, à partir du nouveau serveur. Après le basculement, et après que l'ancien serveur ait été redémarré, il devient un serveur de sauvegarde secondaire, ou un serveur de sauvegarde, s'il n'y a que deux.

**IMPORTANT**

La mise en cluster doit être activée même si vous n'utilisez pas les fonctionnalités de clustering. C'est parce que chaque nœud du cluster HA doit avoir une connexion de cluster pour tous les autres nœuds, afin de négocier des rôles avec d'autres serveurs.

**IMPORTANT**

On a besoin d'un file-system-directory (répertoire de système de fichier) pour que le serveur de sauvegarde puisse envoyer/recevoir des messages en réponse aux messages reçus par l'ancien serveur live.

La topologie de cluster haute disponibilité est atteinte par les serveurs direct et de sauvegarde car ils envoient des informations sur leurs ids de connexion en multidiffusion IP. Si la multidiffusion IP ne peut pas être utilisée, il est également possible d'utiliser une configuration statique des connexions initiales. Après la connexion initiale, le client est informé de la topologie. Si la connexion en cours est périmée, le client établit une connexion vers un autre nœud.

Report a bug

## 17.9. RÉPLICATION DE MESSAGES

### 17.9.1. La réplication de messages HornetQ

HornetQ supporte la possibilité de continuer à fonctionner après un basculement d'un ou de plusieurs serveurs. Ceci est en partie réalisé grâce au support de basculement lorsque des connexions de clients migrent d'un serveur live vers un serveur de sauvegarde en cas d'un basculement de serveur live. Pour conserver le serveur de sauvegarde actuel, les messages sont répliqués du serveur live vers le serveur de sauvegarde en continu par deux stratégies: store partagé et réplication. Cette section couvre la stratégie de réplication.

**AVERTISSEMENT**

Seuls les messages persistés peuvent être répliqués. Tout message non persistant ne peut pas survivre à un basculement.

La réplication de messages entre un serveur direct et un serveur de sauvegarde est effectué par le biais du trafic réseau car les serveurs live et de sauvegarde ne partagent pas les mêmes stores de données. Toutes les revues sont répliquées entre les deux serveurs, tant que les deux serveurs sont dans le même cluster et ont le même nom d'utilisateur et mot de passe de cluster. Tout le trafic de données (persistantes) reçu par le serveur live est répliqué sur le serveur de sauvegarde.

Quand le serveur de sauvegarde est en ligne, il cherche à trouver et à se connecter à un serveur live pour tenter la synchronisation. Une fois synchronisé, il n'est plus disponible en tant que serveur de sauvegarde. Le synchronisation peut prendre un long moment selon le volume de données à synchroniser et la vitesse du réseau.

La façon dont un serveur de sauvegarde recherche un serveur live pour répliquer les données dépend de savoir si le paramètre de ***backup-group-name*** a été défini dans le fichier **hornetq-configuration.xml**. Un serveur de sauvegarde se connectera à un serveur live qui partage le même nom de groupe uniquement. En l'absence de ce paramètre, un serveur de sauvegarde va essayer de se connecter à un serveur live.

Dans le cas d'un serveur live ayant échoué, le serveur de sauvegarde correctement configuré et synchronisé reprendra ses fonctions. Le serveur de sauvegarde permettra d'établir si le serveur live a

échoué, s'il n'a pas pu s'y connecter, mais est encore capable de se connecter à plus de la moitié des autres serveurs dans le cluster. Si plus de la moitié des autres serveurs du cluster échouent également de répondre, cela indique une panne générale de réseau et le serveur de sauvegarde attendra pour réessayer la connexion au serveur live.

Report a bug

### 17.9.2. Configurer les Serveurs HornetQ pour la Réplication

Pour configurer les serveurs live et de sauvegarde en tant que paire de réplication, configurer les deux fichiers **`hornetq-configuration.xml`** pour qu'ils aient :

```
<shared-store>false</shared-store>
.
.
.
<cluster-connections>
   <cluster-connection name="my-cluster">
      ...
   </cluster-connection>
</cluster-connections>
```

Le serveur de sauvegarde doit également être marqué explicitement en tant que serveur de sauvegarde.

```
<backup>true</backup>
<connectors>
   <connector name="nameOfConfiguredLiveServerConnector">
      <factory-class>
         org.hornetq.core.remoting.impl.netty.NettyConnectorFactory
      </factory-class>
   <param key="port" value="5445"/>
   </connector>
<!-- a real configuration could have more connectors here -->
<connectors>
```

Report a bug

# CHAPITRE 18. SOUS-SYSTÈME DE TRANSACTION

## 18.1. CONFIGURATION DE SOUS-SYSTÈME DE TRANSACTION

### 18.1.1. Configuration des transactions

**Introduction**

Les procédures suivantes vous montrent comment configurer le sous-système de transactions de JBoss Enterprise Application Platform.

- Section 18.1.3, « Configurez votre base de données pour utiliser les Transaction JTA »
- Section 18.1.4, « Configuration d'une source de données XA »
- Section 18.1.2, « Configurer le Transaction Manager »
- Section 18.1.6, « Configurer la Journalisation des Sous-systèmes de transactions »

Report a bug

### 18.1.2. Configurer le Transaction Manager

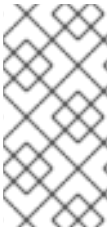
Vous pouvez configurer le Transaction Manager (TM) à l'aide de la Console de gestion sur le web ou la ligne de commande Management CLI. Pour chaque commande ou option donnée, on assume que vous exécutez JBoss Enterprise Application Platform comme un domaine géré. Si vous utilisez un serveur autonome ou que vous souhaitez modifier un profil différent de la valeur par défaut **`default`**, il se peut que vous deviez modifier les étapes et les commandes de la manière suivante.

**Notes sur les commandes d'exemple**

- Pour la console de gestion, le profil par défaut **`default`** est celui qui sera sélectionné quand vous vous connectez. Si vous souhaitez modifier la configuration du Transaction Manager dans un autre profile, sélectionnez votre profile à la place, et non pas **`default`**, pour chaque instruction.

  De même, substituez votre profil à la place du profil par défaut **`default`** pour les commandes CLI de l'exemple.

- Si vous utilisez un Serveur Autonome, un seul profil existe. Ignorer toute instruction pour choisir un profil spécifique. Dans les commandes CLI, retirer la partie **`/profile=default`** des commandes d'échantillon.



**NOTE**

Pour que les options du TM soient visibles dans la Console de gestion ou dans le Management CLI, le sous-système **`transactions`** doit être activé. Il est activé par défaut, et il faut pour cela qu'un certain nombre d'autres sous-systèmes fonctionnent correctement, donc il est improbable qu'il soit désactivé.

**Configurer le TM par la Console de gestion**

Pour configurer le TM à l'aide de la Console de gestion sur le web, sélectionnez l'onglet **`Runtime`** de la liste dans la partie supérieure gauche de l'écran de la Console de gestion. Si vous utilisez un domaine

géré, vous avez le choix de plusieurs profils. Choisir le bon **profil** de la boîte de sélection dans la partie supérieure droite de l'écran Profils. Étendez le menu **Container**, et sélectionnez **Transactions**.

Vous verrez la plupart des options dans la page de configuration du Transaction Manager. Les options **Recovery** sont cachées par défaut. Cliquer sur l'en-tête **Recovery** pour les étendre. Cliquer sur le bouton **Edit** pour éditer une des options. Les changements prendront place immédiatement.

Cliquer sur l'étiquette **Need Help?** pour afficher le texte d'aide en ligne.

#### Configurer le TM par le Management CLI

Dans le Management CLI, vous pouvez configurer le TM en utilisant une série de commandes. Les commandes commencent toutes par **/profile=default/subsystem=transactions/** pour un domaine géré avec **default** de profil, ou par **/subsystem=transactions** pour un serveur autonome.

**Tableau 18.1. Options de configuration de la TM**

| Option | Description | Commande CLI |
|---|---|---|
| Activer les Statistiques | Indique s'il faut activer les statistiques de transaction. Ces statistiques se trouvent dans la Console de Gestion dans la section **Subsystem Metrics** de l'onglet **Runtime**. | **/profile=default/subsystem=transactions/:write-attribute(name=enable-statistics,value=true)** |
| Activer le statut TSM | Indique si l'on doit activer le service de gestion du statut de transaction (TSM), qui est utilisé pour le recouvrement hors-processus. | **/profile=default/subsystem=transactions/:write-attribute(name=enable-tsm-status,value=false)** |
| Délai d'attente par défaut | Délai d'attente de transaction par défaut. La valeur par défaut est de **300** secondes. Vous pouvez la remplacer par programmation, sur la base d'une transaction. | **/profile=default/subsystem=transactions/:write-attribute(name=default-timeout,value=300)** |
| Chemin | Le chemin d'accès relatif ou absolu du système de fichiers dans lequel le cœur du gestionnaire de transactions stocke les données. Par défaut, la valeur est un chemin d'accès relatif à la valeur de l'attribut **relative-to**. | **/profile=default/subsystem=transactions/:write-attribute(name=path,value=var)** |

| Option | Description | Commande CLI |
| --- | --- | --- |
| Relatif à | Référence une configuration de chemin global dans le modèle du domaine. La valeur par défaut correspond au répertoire de données de JBoss Enterprise Application Platform 6, qui correspond à la valeur de la propriété **jboss.server.data.dir**, et qui a pour valeur par défaut ***EAP_HOME*/domain/data/** pour un Domaine Géré, ou ***EAP_HOME*/standalone/data/** pour une instance de Serveur Autonome. La valeur de l'attribut TM du chemin **path** est relative à ce chemin. Utiliser une chaîne vide pour désactiver le comportement par défaut et forcer la valeur de l'attribut du chemin **path** qui doit être traité comme un chemin absolu. | **/profile=default/subsystem=transactions/:write-attribute(name=relative-to,value=jboss.server.data.dir)** |
| Chemin de Store Objet | Un chemin de système de fichiers relatif ou absolu où le store objet TM stocke des données. Relatif, par défaut, à la valeur du paramètre **object-store-relative-to**. | **/profile=default/subsystem=transactions/:write-attribute(name=object-store-path,value=tx-object-store)** |
| Chemin de Store Objet Relatif à | Référence une configuration de chemin global dans le modèle du domaine. La valeur par défaut correspond au répertoire de données de JBoss Enterprise Application Platform 6, qui correspond à la valeur de la propriété **jboss.server.data.dir**, et qui a pour valeur par défaut ***EAP_HOME*/domain/data/** pour un Domaine Géré, ou ***EAP_HOME*/standalone/data/** pour une instance de Serveur Autonome. La valeur de l'attribut TM du chemin **path** est relative à ce chemin. Utiliser une chaîne vide pour désactiver le comportement par défaut et forcer la valeur de l'attribut du chemin **path** qui doit être traité comme un chemin absolu. | **/profile=default/subsystem=transactions/:write-attribute(name=object-store-relative-to,value=jboss.server.data.dir)** |

| Option | Description | Commande CLI |
|---|---|---|
| Liaisons de sockets | Indique le nom de la liaison du socket utilisé par le Gestionnaire de Transactions pour la récupération et la création des identificateurs de transaction, lorsque le mécanisme du socket est utilisé. Se référer à **`processus-id-socket-max-ports`** pour plus d'informations sur la génération de l'identificateur unique. Les liaisons de socket sont spécifiées par le groupe de serveurs dans l'onglet **`Serveur`** de la Console de gestion. | **`/profile=default/subsystem=transactions/:write-attribute(name=socket-binding,value=txn-recovery-environment)`** |
| Liaison de socket de statut | Indique la liaison de socket à utiliser pour le gestionnaire de Statut de transaction. | **`/profile=default/subsystem=transactions/:write-attribute(name=status-socket-binding,value=txn-status-manager)`** |
| Listener de recouvrement | Indique si oui ou non le processus de Recouvrement de transaction doit écouter au socket de réseau. La valeur par défaut est **`false`**. | **`/profile=default/subsystem=transactions/:write-attribute(name=recovery-listener,value=false)`** |

Les options suivantes sont pour une utilisation avancée et ne peuvent être modifiées qu'à l'aide du Management CLI. Soyez prudent lors de leur modification à partir de la configuration par défaut. Communiquer avec Red Hat Global Support Services pour plus d'informations.

**Tableau 18.2. Options de configuration TM avancées**

| Option | Description | Commande CLI |
|---|---|---|
| jts | Indique si l'on doit utiliser les transactions Java Transaction Service (JTS). La valeur par défaut est **`false`**, qui utilise des transactions JTA uniquement. | **`/profile=default/subsystem=transactions/:write-attribute(name=jts,value=false)`** |
| Identifiant de nœud | L'identifiant de nœud pour le service JTS. Ce dernier doit être unique pour le service JTS, parce que le Gestionnaire de Transaction l'utilise pour la récupération. | **`/profile=default/subsystem=transactions/:write-attribute(name=node-identifier,value=1)`** |

| Option | Description | Commande CLI |
| --- | --- | --- |
| process-id-socket-max-ports | Le Gestionnaire de Transaction crée un identifiant unique pour chaque journal des transactions. Deux mécanismes différents sont fournis pour générer des identificateurs uniques : un mécanisme basé sur le socket et un mécanisme fondé sur l'identificateur de processus du processus.<br><br>Dans le cas de l'identifiant basé-socket, le socket est ouvert et son numéro de port est utilisé pour l'identifiant. Si le port est déjà utilisé, on cherchera le port suivant, jusqu'à ce qu'un port libre soit trouvé. Les **`processus-id-socket-max-ports`** représentent le nombre maximal de sockets que le TM va essayer avant d'abandonner. La valeur par défaut est **`10`**. | **`/profile=default/subsystem=transactions/:write-attribute(name=process-id-socket-max-ports,value=10)`** |
| process-id-uuid | Définir à **`true`** avec un identifiant de processus pour créer un identifiant unique pour chaque transaction. Sinon, le mécanisme basé socket sera utilisé. La valeur par défaut est **`true`**. Se référer à **`process-id-socket-max-ports`** pour obtenir davantage d'informations. | **`/profile=default/subsystem=transactions/:write-attribute(name=process-id-uuid,value=true)`** |
| use-hornetq-store | Utiliser les mécanismes de stockage journalisés de HornetQ au lieu du stockage basé sur des fichiers, pour les journaux de transactions. Ceci est désactivé par défaut, mais peut améliorer les performances I/O. Il n'est pas recommandé pour les transactions JTS sur les gestionnaires de transactions séparés. . | **`/profile=default/subsystem=transactions/:write-attribute(name=use-hornetq-store,value=false)`** |

Report a bug

### 18.1.3. Configurez votre base de données pour utiliser les Transaction JTA

#### Résumé

Cette tâche vous montre comment activer JTA (Java Transactions API) sur votre source de données.

#### Prérequis

Vous devez remplir les conditions suivantes avant de continuer cette tâche :

- Votre base de données ou autre ressource devra supporter JTA. Dans le doute, veuillez consulter la documentation.
- Créer une source de données. Veuillez vous référer à Section 6.3.1, « Créer une source de données Non-XA avec les Interfaces de gestion ».
- Arrêter JBoss Enterprise Application Platform.
- Obtenez un accès pour pouvoir éditer les fichiers de configuration directement, dans un éditeur de texte.

**Procédure 18.1. Configurer la Source de données pour utiliser les Transactions JTA.**

1. **Ouvrir le fichier de configuration dans l'éditeur de texte.**
   Selon si vous exécutez JBoss Enterprise Application Platform sur un domaine géré ou un serveur autonome, votre fichier de configuration ne se trouvera pas au même endroit.
   - **Domaine géré**
     Le fichier de configuration par défaut d'un domaine géré se trouve dans `EAP_HOME/domain/configuration/domain.xml` pour Red Hat Enterprise Linux, et `EAP_HOME\domain\configuration\domain.xml` pour Microsoft Windows Server.
   - **Serveur autonome**
     Le fichier de configuration par défaut d'un serveur autonome se trouve dans `EAP_HOME/standalone/configuration/domain.xml` pour Red Hat Enterprise Linux, et `EAP_HOME\standalone\configuration\domain.xml` pour Microsoft Windows Server.
2. **Chercher la balise `<datasource>` qui correspond à votre source de données.**
   La source de données aura un attribut **`jndi-name`** correspondant à celui que vous aviez indiqué quand vous l'avez créé. Par exemple, la source de données ExampleDS ressemble à ceci :

   ```
   <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-
   name="H2DS" enabled="true" jta="true" use-java-context="true" use-
   ccm="true">
   ```

3. **Définir l'attribut `jta` à `true`.**
   Ajouter l'élément suivant au contenu de votre balise **`<datasource>`**, tel qu'il apparaît à l'étape précédente : **`jta="true"`**
4. **Sauvegarder le fichier de configuration.**
   Sauvegarder le fichier de configuration et sortir de l'éditeur de texte.
5. **Démarrer JBoss Enterprise Application Platform.**
   Relancer le serveur JBoss Enterprise Application Platform 6.

**Résultat :**

JBoss Enterprise Application Platform démarre, et votre source de données est configurée pour utiliser les transactions JTA.

Report a bug

### 18.1.4. Configuration d'une source de données XA

**Prérequis**

Pour pouvoir ajouter une source de données XA, vous devrez vous connecter à la Console de gestion. Voir Section 3.4.2, « Connectez-vous à la Console de management » pour plus d'informations.

1. **Ajouter une nouvelle source de données.**
   Ajouter une nouvelle source de données à la plateforme JBoss Enterprise Application Platform. Suivre les instructions qui se trouvent dans Section 6.3.1, « Créer une source de données Non-XA avec les Interfaces de gestion », puis, cliquer sur l'onglet **`XA Datasource`** en haut.

2. **Configurer les propriétés supplémentaires suivant les besoins.**
   Tous les paramètres de la source de données se trouvent dans Section 6.6.1, « Paramètres de source de données ».

**Résultat**

Votre source de données XA est configurée et prête à l'utilisation.

Report a bug

### 18.1.5. A propos des Messages de Journalisation de Transaction

Pour suivre le statut de la transaction tout en gardant les fichiers de journalisation lisible, utiliser le niveau de journalisation **DEBUG** pour le logger de transaction. Pour un débogage détaillé, utiliser le niveau de journalisation **TRACE**. Veuillez consulter Section 18.1.6, « Configurer la Journalisation des Sous-systèmes de transactions » pour plus d'informations sur la configuration du logger de transaction.

Le gestionnaire de transaction peut générer beaucoup d'informations de journalisation si configuré pour se connecter au niveau de journalisation **TRACE**. Vous trouverez ci-dessous quelques-uns des messages les plus courants. Cette liste n'est pas exhaustive, il se peut que vous rencontriez d'autres messages.

**Tableau 18.3. Changement d'état de transaction**

| | |
|---|---|
| Début de transaction | Quand une transaction commence, le code suivant s'exécute :<br>`com.arjuna.ats.arjuna.coordinator.BasicAction::Begin:1342`<br>`tsLogger.logger.trace("BasicAction::Begin() for action-id "+ get_uid());` |
| Validation de Transaction | Quand une transaction est validée, le code suivant s'exécute :<br>`com.arjuna.ats.arjuna.coordinator.BasicAction::End:1342`<br>`tsLogger.logger.trace("BasicAction::End() for action-id "+ get_uid());` |

| | |
|---|---|
| Restauration de Transaction | Quand une transaction est restaurée, le code suivant s'exécute :<br>`com.arjuna.ats.arjuna.coordinator.BasicAction::Abort:1575`<br>`tsLogger.logger.trace("BasicAction::Abort() for action-id "+ get_uid());` |
| Délai d'expiration de Transaction | Quand une transaction expire, le code suivant s'exécute :<br>`com.arjuna.ats.arjuna.coordinator.TransactionReaper::doCancellations:349`<br>`tsLogger.logger.trace("Reaper Worker " + Thread.currentThread() + " attempting to cancel " + e._control.get_uid());`<br>Vous verrez ensuite le même thread restaurer la transaction tel que montré ci-dessus. |

Report a bug

### 18.1.6. Configurer la Journalisation des Sous-systèmes de transactions

**Résumé**

Utiliser cette procédure pour contrôler la quantité d'informations enregistrées sur les transactions, indépendamment des autres paramètres de journalisation dans JBoss Enterprise Application Platform. La procédure montre comment procéder dans la Console de gestion sur le web. La commande de gestion CLI est donnée par la suite.

**Procédure 18.2. Configurer le Transaction Logger par la Console de gestion**

1. **Naviguer vers la zone de configuration de la Journalisation**
   Dans la Console de gestion, cliquer sur l'onglet **`Profiles`** en haut et à gauche de l'écran. Si vous utilisez un domaine géré, fermer le profil du serveur que vous souhaitez configurer, à partir de la case de sélection **`Profile`** qui se trouve en haut et à droite.

   Dérouler le menu **`Core`**, et cliquer sur l'étiquette **`Logging`**.

2. **Modifier les attributs de `com.arjuna`.**
   Cliquer sur le bouton **`Edit`** dans la section **`Details`** qui se situe en bas de la page. Vous pourrez ajouter ici les informations de journalisation spécifiques à la classe. La classe **`com.arjuna`** est déjà présente. Vous pourrez modifier le niveau de journalisation et décider si vous souhaitez utiliser les gestionnaires parents.

**Niveau de Journalisation**

Le niveau de journalisation est **`WARN`** par défaut. Comme les transactions peuvent produire une grande quantité de messages de journalisation, la signification des niveaux de journalisation standard est légèrement différente pour le Transaction Logger. En général, les messages avec des niveaux de gravité moins élevés que le niveau choisi sont ignorés.

**Niveaux de journalisation des transactions, du plus au moins détaillé.**

- TRACE
- DEBUG
- INFO
- WARN
- ERROR
- FAILURE

**Utiliser les gestionnaires parents**

Indique si l'enregistreur d'événements doit envoyer ses sorties vers l'enregistreur d'événements parent. Le comportement par défaut est **`true`**.

3. Les changements prennent effet immédiatement.

Report a bug

## 18.2. ADMINISTRATION DES TRANSACTIONS

### 18.2.1. Naviguer et gérer les transactions

Le CLI de gestion basé de ligne de commande prend en charge la capacité de naviguer et de manipuler les enregistrements des transactions. Cette fonctionnalité est fournie par l'interaction entre le gestionnaire de transactions et l'API de Gestion (Management) de JBoss Enterprise Application Platform 6.

Le gestionnaire de transactions stocke des informations sur chaque transaction en attente et les participants impliqués dans la transaction, dans un stockage persistant appelé *object store*. L'API de gestion expose le store objet sous forme de ressource appelée **`log-store`**. Une opération API nommée **`probe`** lit les journaux de transaction et crée un noeud pour chaque journal. Vous pouvez invoquer la commande **`probe`** manuellement, quand vous souhaitez réactualiser le **`log-store`**. Il est normal pour les journaux de transaction d'apparaitre ou de disparaitre rapidement.

**Exemple 18.1. Réactualiser le Log Store**

Cette commande réactualise le Log Store des groupes de serveurs qui utilisent le profil par défaut **`default`** dans un domaine géré. Dans le cas d'un serveur autonome, supprimer **`profile=default`** de la commande.

```
/profile=default/subsystem=transactions/log-store=log-store/:probe
```

**Exemple 18.2. Voir toutes les transactions préparées**

Pour voir toutes les transactions préparées, commencer par réactualiser le log store (voir Exemple 18.1, « Réactualiser le Log Store »), puis exécuter la commande suivante, qui fonctionne de la même manière qu'une commande **ls** de système de fichier.

```
ls /profile=default/subsystem=transactions/log-store=log-
store/transactions
```

Chaque transaction est visible, ainsi que son identifiant unique. Les opérations individuelles peuvent être exécutées contre une transaction individuelle (voir Gérer une transaction).

### Gérer une transaction

**Voir des attributs de transaction.**

Pour voir des informations sur une transaction, comme son nom JNDI, son nom de produit EIS ou sa version, ou statut, utiliser la commande CLI **:read-resource**.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:ffff7f000001\:-b66efc2\:4f9e6f8f\:9:read-resource
```

**Voir tous les participants d'une transaction.**

Chaque journal de transaction contient un élément enfant nommé **participants**. Utiliser la commande CLI **read-resource** CLI sur cet élément pour voir les participants des transactions. Les participants sont identifiés par leurs noms JNDI.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:ffff7f000001\:-
b66efc2\:4f9e6f8f\:9/participants=java\:\/JmsXA:read-resource
```

Le résultat devrait ressembler à ceci :

```
{
   "outcome" => "success",
   "result" => {
       "eis-product-name" => "HornetQ",
       "eis-product-version" => "2.0",
       "jndi-name" => "java:/JmsXA",
       "status" => "HEURISTIC",
       "type" => "/StateManager/AbstractRecord/XAResourceRecord"
   }
}
```

Le statut du résultat affiché ici est dans un état **HEURISTIC** et est susceptible d'être recouvert. Voir Recouvrement d'une transaction. pour plus d'informations.

**Supprimer une transaction.**

Chaque journal de transaction supporte une opération **:delete** pour effacer l'enregistrement qui représente la transaction.

```
/profile=default/subsystem=transactions/log-store=log-
```

```
store/transactions=0\:ffff7f000001\:-b66efc2\:4f9e6f8f\:9:delete
```

**Recouvrement d'une transaction.**

Chaque journal de transaction supporte le recouvrement par la commande CLI **`:recover`**.

**Recouvrement des transactions heuristiques et des participants**

- Si le statut de la transaction est **`HEURISTIC`**, l'opération de recouvrement change l'état en **`PREPARE`** et déclenche un recouvrement.
- Si l'un des participants de la transaction est heuristique, l'opération de recouvrement tente de répondre à l'opération **`commit`** (validation). En cas de succès, le participant est retiré du journal des transactions. Vous pouvez vérifier cela en exécutant à nouveau l'opération **`:probe`** sur le **`log-store`** et en vérifiant que le participant n'est plus inscrit. Si c'est le dernier participant, la transaction sera également supprimée.

**Réactualiser le statut de la transaction qui a besoin d'être recouvrée.**

Si une transaction a besoin d'être recouvrée, vous pourrez utiliser la commande CLI **`:refresh`** pour vous assurer qu'elle a toujours besoin d'être recouvrée, avant de tenter le recouvrement.

```
/profile=default/subsystem=transactions/log-store=log-
store/transactions=0\:ffff7f000001\:-b66efc2\:4f9e6f8f\:9:refresh
```

**NOTE**

Pour les transactions JTS, si les participants se trouvent sur des serveurs à distance, un nombre limité d'informations peut être disponible au Gestionnaire de Transactions. Dans ce cas, il est conseillé d'utiliser le store objet basé sur fichiers, plutôt que le mode stockage HornetQ. Cela constitue le comportement par défaut. Pour utiliser le mode de stockage HornetQ, vous pouvez définir la valeur de l'option **`use-hornetq-store`** sur **`true`**, dans la configuration de Gestionnaire de Transaction. Veuillez consulter Section 18.1.2, « Configurer le Transaction Manager » pour plus d'informations concernant la configuration du Gestionnaire de Transaction.

**Voir les statistiques de transaction**

Si les statistiques de TM (Transaction Manager) sont activées, vous pouvez consulter les statistiques à propos du Gestionnaire de Transaction et du sous-système de transaction. Veuillez consulter Section 18.1.2, « Configurer le Transaction Manager » pour plus d'informations sur l'activation des statistiques de TM.

Vous pouvez consulter les statistiques soit par la Console de gestion basée-web, soit par la gestion de ligne de commande CLI. Dans la Console de gestion basée-web, les statistiques de transaction seront disponibles via **Runtime** → **Subsystem Metrics** → **Transactions**. Les statistiques de transaction sont disponibles pour chaque serveur dans un domaine géré, également. Vous pourrez spécifier le serveur dans la case de sélection **`Server`** située en haut à gauche.

La table suivante affiche chaque statistique disponible, sa description et la commande CLI pour afficher le statistique.

**Tableau 18.4. Les statistiques de sous-système de transaction**

| Statistique | Description | Commande CLI |
| --- | --- | --- |
| Total | Le nombre total de transactions exécutées par le Gestionnaire de Transaction sur ce serveur. | `/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-transactions,include-defaults=true)` |
| Validé | Le nombre de transactions validées exécutées par le Gestionnaire de Transaction sur ce serveur. | `/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-committed-transactions,include-defaults=true)` |
| Abandonné | Le nombre de transactions interrompues exécutées par le Gestionnaire de Transaction sur ce serveur. | `/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-aborted-transactions,include-defaults=true)` |
| Délai expiré | Le nombre de transactions expirées exécutées par le Gestionnaire de Transaction sur ce serveur. | `/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-timed-out-transactions,include-defaults=true)` |
| Heuristiques | Pas disponible dans la Console de Gestion. Nombre de transactions dans un état heuristique. | `/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-heuristics,include-defaults=true)` |

| Statistique | Description | Commande CLI |
| --- | --- | --- |
| Transactions In-Flight | Pas disponible dans la Console de Gestion. Nombre de transactions commencées mais pas encore achevées. | `/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-inflight-transactions,include-defaults=true)` |
| Origine de l'échec - Applications | Le nombre de transactions échouées dont l'origine de l'échec était une application. | `/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-application-rollbacks,include-defaults=true)` |
| Origine de l'échec - Ressources | Le nombre de transactions échouées dont l'origine de l'échec était une ressource. | `/host=master/server=server-one/subsystem=transactions/:read-attribute(name=number-of-resource-rollbacks,include-defaults=true)` |

Report a bug

## 18.3. RÉFÉRENCES DE TRANSACTIONS

### 18.3.1. Erreurs et Exceptions pour les transactions JBoss

Pour obtenir des informations lancées par les méthodes de la classe **`UserTransaction`**, voir la spécification *UserTransaction API* dans http://download.oracle.com/javaee/1.3/api/javax/transaction/UserTransaction.html.

Report a bug

### 18.3.2. Limitations de JTA Clustering

Les transactions JTA ne peuvent pas être clusterisées à travers les instances multiples de JBoss EAP. Pour ce comportement, utiliser les transactions JTS.

Pour pouvoir utiliser les transactions JTS, vous devrez configurer l'ORB: Section 18.4.2, « Configurer l'ORB pour les transactions JTS ».

Report a bug

## 18.4. CONFIGURATION ORB

### 18.4.1. A propos de CORBA (Common Object Request Broker Architecture)

*Common Object Request Broker Architecture (CORBA)* est une norme qui autorise les applications et services à travailler ensemble même lorsqu'ils sont écrits dans de multiples langages d'habitude incompatibles ou lorsqu'ils sont hébergés sur des plateformes différentes. Les requêtes CORBA sont émises par un composant côte seveur appelé *Object Request Broker (ORB)*. JBoss Enterprise Application Platform 6 fournit une instance ORB au moyen du composant JacORB.

L'ORB est utilisé en interne pour les transactions *Java Transaction Service (JTS)*, et peut également être utilisé par votre propre application.

Report a bug

### 18.4.2. Configurer l'ORB pour les transactions JTS

Dans une installation JBoss Enterprise Application Platform par défaut, l'ORB est désactivé. Vous pouvez activer l'ORB en utilisant le Management CLI de ligne de commande.

**NOTE**

Dans un domaine géré, le sous-système JacORB est disponible dans les profils **`full`** et **`full-ha`** uniquement. Dans un serveur autonome, il est disponible uniquement quand vous utilisez les configurations **`standalone-full.xml`** ou **`standalone-full-ha.xml`**.

**Procédure 18.3. Configurer l'ORB par la Console de gestion**

1. **Voir les paramètres de configuration du profil.**
   Sélectionner **`Profiles`** (domaine géré) ou **`Profile`** (serveur autonome) dans la partie supérieure droite de la console de gestion. Si vous utilisez un domaine géré, sélectionner soit le profil **`full`** ou **`full-ha`** à partir de la boîte de dialogue de sélection en haut à gauche.

2. **Modifier les paramètres d' `Initializers`**
   Étendre le menu **`Subsystems`** (sous-systèmes) sur la gauche, si nécessaire. Étendre le sous-menu **`Container`** et cliquer sur **`JacORB`**.

   Sur le formulaire qui apparaît sur l'écran principal, sélectionner l'onglet **`Initializers`**, et cliquer sur le bouton **`Edit`** (modifier).

   Activer les intercepteurs de sécurité en configurant la valeur de **`Security`** à **`on`**.

   Pour activer ORB sur JTS, définir la valeur des **`Transaction Interceptors`** à **`on`**, au lieu de la valeur par défaut **`spec`**.

   Voir le lien **`Need Help?`** sur le formulaire pour accéder à des explications sur ces valeurs. Cliquer sur **`Save`** quand vous aurez fini de modifier les valeurs.

3. **Configuration ORB avancée**

Voir les autres sections du formulaire pour les options de configuration avancées. Chaque section inclut un lien **Need Help?** avec des informations détaillées sur les paramètres.

### Configurer l'ORB par le Management CLI

Vous pouvez configurer chaque aspect de l'ORB à l'aide du Management CLI. Les commandes suivantes configurent les initialiseurs aux mêmes valeurs que celles de la procédure ci-dessus, pour la Console de gestion. Il s'agit de la configuration minimale pour l'ORB, si utilisé avec JTS.

Ces commandes sont configurées pour un domaine de sécurité utilisant le profil **full**. Si nécessaire, modifier le profil pour qu'il convienne mieux à celui que vous aurez besoin de configurer. Si vous utilisez un serveur autonome, n'utilisez pas la portion **/profile=full** des commandes.

**Exemple 18.3. Activer les intercepteurs de sécurité**

```
/profile=full/subsystem=jacorb/:write-attribute(name=security,value=on)
```

**Exemple 18.4. Activer l'ORB pour JTS**

```
/profile=full/subsystem=jacorb/:write-
attribute(name=transactions,value=on)
```

Report a bug

# CHAPITRE 19. ENTERPRISE JAVABEANS

## 19.1. INTRODUCTION

### 19.1.1. Entreprise JavaBeans

Enterprise JavaBeans (EJB) 3.1 est une API pour développer des applications de Java EE distribuées, transactionnelles, sécurisées et portables grâce à l'utilisation des composants côté serveur appelés Enterprise Beans. Enterprise Beans implémente la logique métier d'une application, de manière découplée, qui encourage la réutilisation. Enterprise JavaBeans 3.1 est documenté dans la spécification Java EE JSR-318.

JBoss Enterprise Application Platform 6 prend en charge la génération d'applications qui utilisent la spécification Enterprise JavaBeans 3.1. Le conteneur EJB est implémenté par le projet communautaire JBoss EJB3, http://www.jboss.org/ejb3.

Report a bug

### 19.1.2. Enterprise JavaBeans pour Administrateurs

Les administrateurs JBoss ont de nombreuses options de configuration disponibles pour contrôler la performance des Beans Enterprise dans JBoss Enterprise Application Platform 6. Ces options sont accessibles par la Console de gestion ou par l'outil de configuration de ligne de commande. Éditer le fichier de configuration du serveur XML pour appliquer les modifications est également possible mais non recommandé

Les options de configuration EJB se situent dans des endroits légèrement différents de la Console de gestion, selon que le serveur exécute ou non.

Si le serveur exécute en tant que serveur autonome :

1. Cliquer sur le lien **`Profile`** qui se trouve en haut et à droite pour changer de vue de Profil.
2. Expand the **`Profile`** menu on the left by clicking the arrow next to the label.
3. Cliquer sur **`Container`** pour le développer, puis cliquer sur **`EJB 3`**.

Si le serveur exécute dans le cadre d'un domaine géré :

1. Cliquer sur le lien **`Profile`** qui se trouve en haut pour changer de vue de Profil.
2. Expand the **`Subsystems`** menu on the left by clicking the arrow next to the label.
3. Sélectionner le profile que vous modifiez à partir du menu **`Profile`**.
4. Cliquer sur **`Container`** pour le développer, puis cliquer sur **`EJB 3`**.

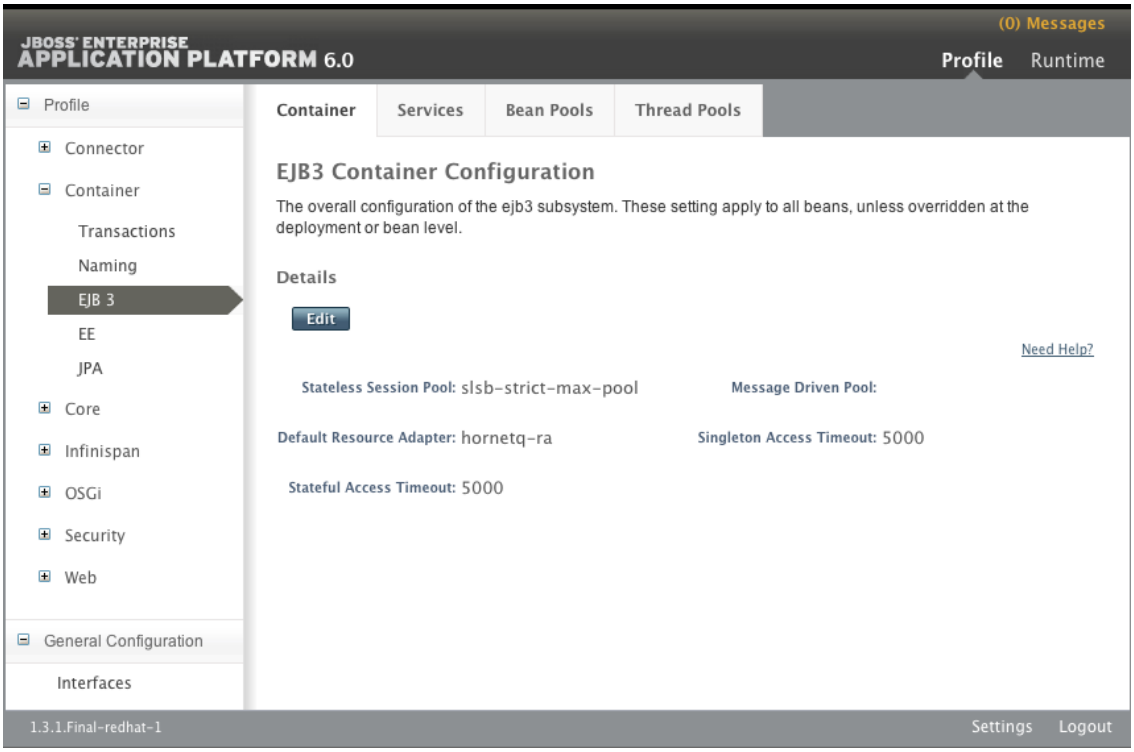



**Figure 19.1. Options de configuration EJB de la Console de gestion (Serveur autonome)**

Report a bug

### 19.1.3. Beans Enterprise

Les beans Enterprise sont des composants d'applications côté serveur, ainsi définis dans la spécification Enterprise JavaBeans (EJB) 3.1, JSR-318. Les beans Enterprise sont conçus pour l'implémentation d'une logique commerciale d'application d'une manière découplée, pour encourager sa réutilisation.

Les beans Enterprise sont écrits comme des classes Java et sont annotés avec les annotations EJB appropriées. Ils peuvent être déployés sur le serveur d'application dans leur propre archive (un fichier JAR) ou être déployés dans le cadre d'une application Java EE. Le serveur d'applications gère le cycle de vie de chaque bean Entreprise et leur fournit des services comme la sécurité, les transactions et la gestion de concurrence.

Un bean Enterprise peut également définir un nombre d'interfaces de métier. Les interfaces de métier vous proposent un plus grand contrôle sur les méthodes Bean qui sont disponibles pour les clients. et peut également vous donner accès aux clients qui exécutent dans les JVM à distance.

Il existe trois types d'Enterprise Bean : les Session beans, les Message-driven beans et les Entity beans.

**IMPORTANT**

Les Entity beans sont maintenant obsolètes dans EJB 3.1 et Red Hat recommande d'utiliser des entités JPA à la place. Red Hat ne recommande d'utiliser des Entity beans que pour les compatibilités rétroactives avec les systèmes hérités.

Report a bug

### 19.1.4. Session Beans

Les Session Beans sont des Beans Enterprise qui encapsulent un ensemble de processus métier connexes ou tâches qui sont injectés dans les classes qui en ont fait la demande. Il existe trois types de Session Beans : sans état, avec état et singleton.

Report a bug

### 19.1.5. Message-Driven Beans

Les Message-driven Beans (MDB) fournissent un modèle basé-événement pour le développement de l'application. Les méthodes des MDB ne sont pas injectées ou invoquées du code client mais sont déclenchées par la réception de messages d'un service de messagerie tel que le serveur Java Messaging Service (JMS). La spécification Java EE 6 exige que JMS soit pris en charge, mais les autres systèmes de messagerie peuvent être supportés également.

Report a bug

## 19.2. CONFIGURER LES BEAN POOLS

### 19.2.1. Bean Pools

JBoss Enterprise Application Platform 6 maintient un certain nombre d'instances de beans stateless enterprise déployés en mémoire pour procurer une performance plus rapide. Cette technique s'appelle le Bean Pooling. Quand on a besoin d'un bean, le serveur de l'application peut en prendre un du pool qui convient parmi les beans déjà existants au lieu d'en instancier un nouveau. Quand le bean n'est plus requis, il est renvoyé dans le pool en vue d'être réutilisé.

Les Bean Pools sont configurés et maintenus séparément dans le cas des Stateless session beans et des Beans basés messages.

Report a bug

### 19.2.2. Créer un Bean Pool

Les Bean Pools peuvent être créés par l'intermédiaire de la Console de gestion et du CLI.

Les Bean Pools peuvent également être créés en ajoutant la configuration du Bean Pool requis au fichier de configuration du serveur en utilisant l'éditeur de texte. Exemple 19.2, « Exemple de configuration XML » est un exemple de configuration.

**Procédure 19.1. Créer un Bean Pool par la console de gestion**

1. Connectez-vous à la Console de gestion. Voir Section 3.4.2, « Connectez-vous à la Console de management ».

2. Naviguez dans le panneau **`EJB3 Bean Pools`**.

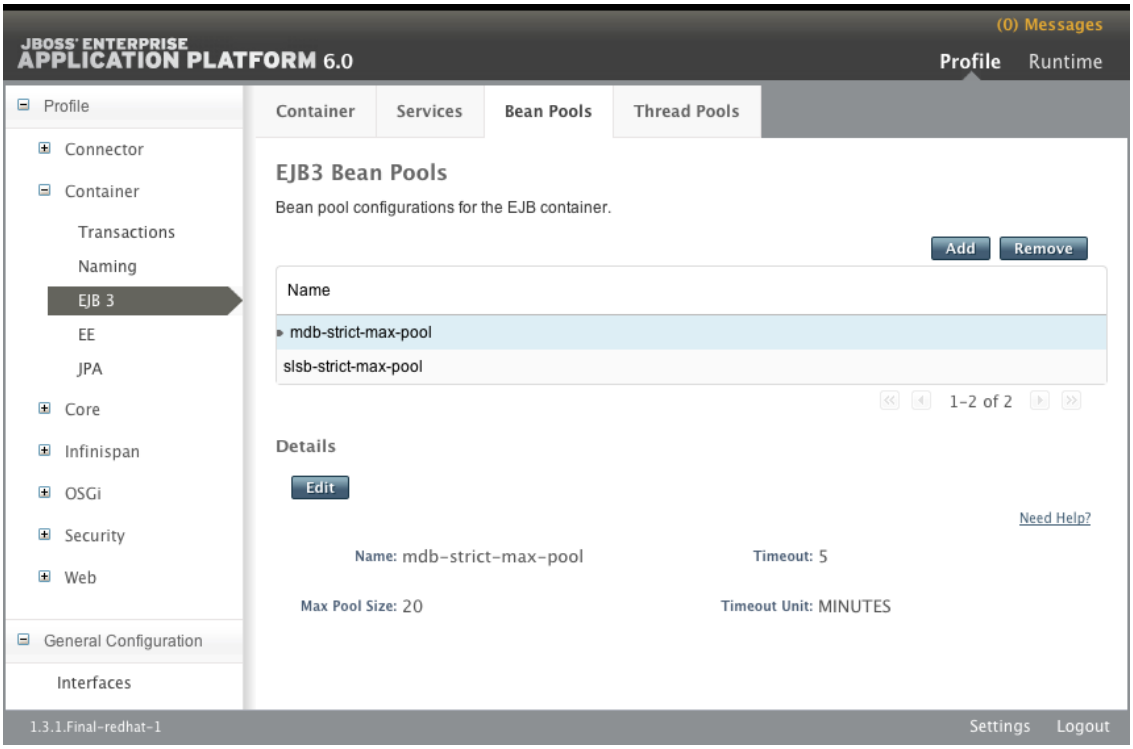



**Figure 19.2. Panneau EJB3 Bean Pools**

3. Cliquer sur le bouton **Add**. Le dialogue **Add EJB3 Bean Pools** apparaîtra.

4. Donnez les informations requises, les valeurs de **Name**, **Max Pool Size**, **Timeout** et l'unité de **Timeout**.

5. Cliquer sur le bouton **Save** pour sauvegarder le nouveau Bean Pool ou cliquer sur le lien **Cancel** pour faire cesser la procédure.

   - Si vous cliquer sur le bouton **Save**, le dialogue se fermera et le nouveau Bean Pool apparaîtra dans la liste.

   - Si vous cliquez sur **Cancel**, le dialogue se fermera et aucun autre Bean Pool ne sera créé.

**Procédure 19.2. Créer un Bean Pool par le CLI**

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI ».

2. Utiliser l'opération **add** avec la syntaxe suivante.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:add(max-
pool-size=MAXSIZE, timeout=TIMEOUT, timeout-unit="UNIT")
```

   - Remplacer *BEANPOOLNAME* par le nom requis de Bean Pool.

   - Remplacer *MAXSIZE* par le nom requis de Bean Pool.

   - Remplacer *TIMEOUT*

   - Remplacer *UNIT* par l'unité de temps requise. Les valeurs permises sont les suivantes : **NANOSECONDS**, **MICROSECONDS**, **MILLISECONDS**, **SECONDS**, **MINUTES**, **HOURS**, et **DAYS**.

3. Utiliser l'opération **`read-resource`** pour confirmer la création d'un Bean Pool.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:read-resource
```

**Exemple 19.1. Créer un Bean Pool par la CLI**

```
[standalone@localhost:9999 /] /subsystem=ejb3/strict-max-bean-instance-pool=ACCTS_BEAN_POOL:add(max-pool-size=500, timeout=5000, timeout-unit="SECONDS")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Exemple 19.2. Exemple de configuration XML**

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">

   <pools>

      <bean-instance-pools>

         <strict-max-pool  name="slsb-strict-max-pool" max-pool-size="20"
            instance-acquisition-timeout="5"
            instance-acquisition-timeout-unit="MINUTES" />

         <strict-max-pool name="mdb-strict-max-pool" max-pool-size="20"
            instance-acquisition-timeout="5"
            instance-acquisition-timeout-unit="MINUTES" />

      </bean-instance-pools>

   </pools>

</subsystem>
```

Report a bug

### 19.2.3. Supprimer un Bean Pool

Les Bean Pool non utilisés peuvent être supprimés par la Console de Management.

**Prérequis :**

- La Bean Pool que vous souhaitez supprimer ne peut pas être en cours d'utilisation. Consulter Section 19.2.5, « Assigner des Bean Pools aux Beans de session et aux Beans basés messages » pour vérifier qu'elle n'est pas en cours d'utilisation.

**Procédure 19.3. Supprimer un Bean Pool par la Console de management**

1. Se connecter à la Console de management Console. Consulter Section 3.4.2, « Connectez-vous à la Console de management ».

2. Cliquer sur **Profile** en haut à droite, puis développer l'item **Container** dans le panneau de Profil sur la gauche, et sélectionner **EJB 3**. Puis sélectionner l'onglet **Bean Pools** dans le panneau principal.

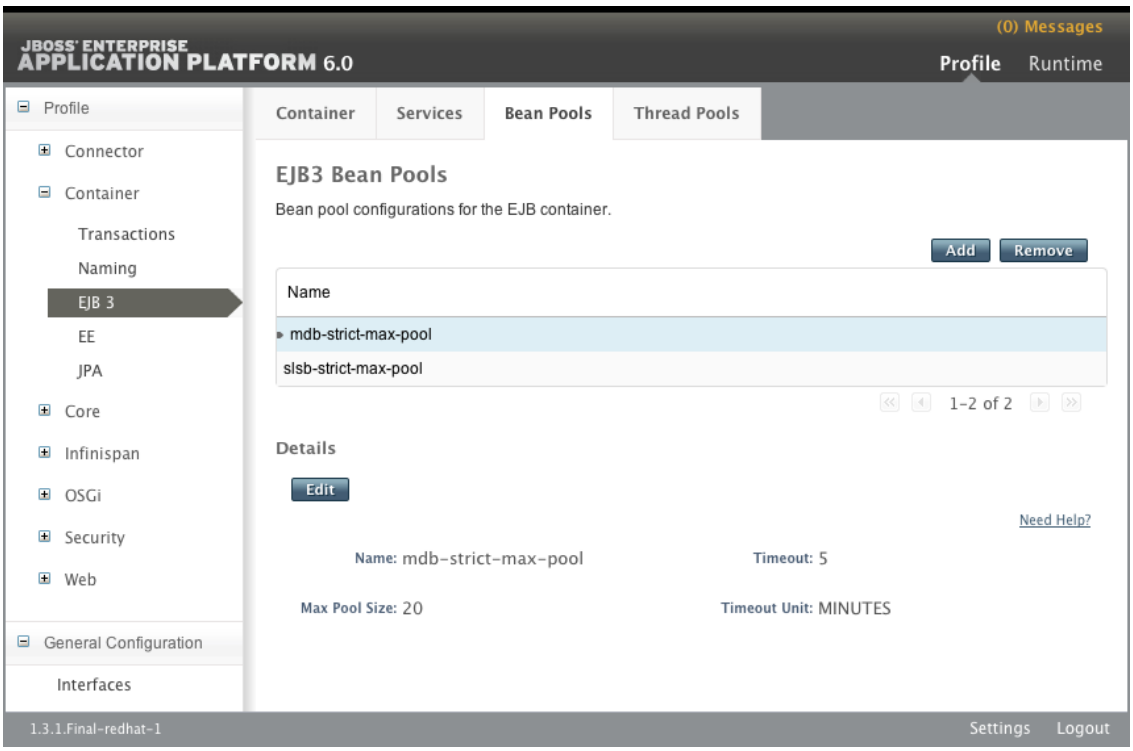



**Figure 19.3. Panneau EJB3 Bean Pools**

3. Cliquer sur le Bean Pool que vous souhaiter supprimer de la liste.

4. Cliquer sur le bouton **`Remove`**. La boîte de dialogue **`Remove Item`** dialog appears.

5. Cliquer sur le bouton **`OK`** pour confirmer la suppression ou cliquer sur le lien **`Cancel`** pour abandonner l'opération.

   Si vous cliquer sur le bouton **`Ok`**, le dialogue se fermera et le Bean Pool sera supprimé et retiré de la liste.

   Si vous cliquez sur le bouton Cancel, la boîte de dialogue se fermera et il n'y aura aucun changement.

**Procédure 19.4. Supprimer un Bean Pool par le CLI**

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI ».

2. Utiliser l'opération **`remove`** avec la syntaxe suivante.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:remove
```

   - Remplacer *BEANPOOLNAME* par le nom requis de Bean Pool.

**Exemple 19.3. Supprimer un Bean Pool par le CLI**

```
[standalone@localhost:9999 /] /subsystem=ejb3/strict-max-bean-instance-
pool=ACCTS_BEAN_POOL:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Report a bug

### 19.2.4. Modifier un Bean Pool

Les Bean Pools peuvent être modifiés par la Console de management.

**Procédure 19.5. Modifier un Bean Pool par la Console de management**

1. Connectez-vous à la Console de gestion. Section 3.4.2, « Connectez-vous à la Console de management »

2. Cliquer sur **Profile** en haut à droite, puis développer l'item **Container** dans le panneau de Profil sur la gauche, et sélectionner **EJB 3**. Puis sélectionner l'onglet **Bean Pools** dans le panneau principal.

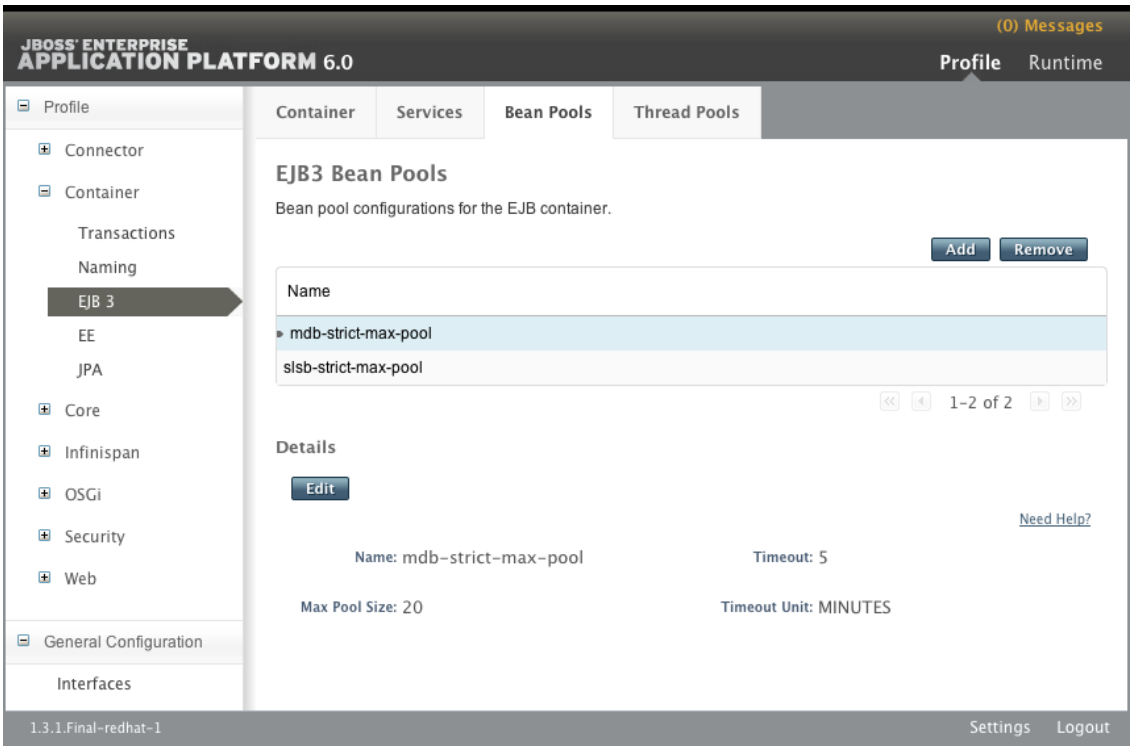



**Figure 19.4. Panneau EJB3 Bean Pools**

3. Cliquer sur le Bean Pool que vous souhaiter modifier dans la liste.

4. Cliquer sur le bouton **`Edit`**. Le champ de la zone **`Details`** est maintenant modifiable.

5. Modifier les informations que vous souhaitez. Seules les valeurs de **`Max Pool Size`**, **`Timeout`**, et l'unité de **`Timeout`** peuvent être modifiés.

6. Cliquer sur le bouton **`Save`** si les changements vous conviennent, ou bien, cliquer sur le lien **`Cancel`** si vous souhaitez ignorer les changements.

Si vous cliquez sur le bouton **Ok**, la zone **Details** sera à nouveau une zone non modifiable et le Bean Pool sera mis à jour avec les nouvelles informations.

Si vous cliquez sur le lien **Cancel**, la zone **Details** sera à nouveau une zone non modifiable et aucun changement n'aura lieu.

**Procédure 19.6. Modifier un Bean Pool par le CLI**

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI ».
2. Utiliser l'opération **write_attribute** avec la syntaxe suivante pour chaque attribut du Bean Pool à modifier.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:write-attribute(name="ATTRIBUTE", value="VALUE")
```

   - Remplacer *BEANPOOLNAME* par le nom requis de Bean Pool.
   - Remplacer *ATTRIBUTE* par le nom de l'attribut à modifier. Les attributs ne pouvant pas être modifiés de cette façon sont **max-pool-size**, **timeout**, et **timeout-unit**.
   - Remplacer *VALUE* par la valeur requise de l'attribut.

3. Utiliser l'opération **read-resource** pour confirmer les changements au Bean Pool.

```
/subsystem=ejb3/strict-max-bean-instance-pool=BEANPOOLNAME:read-resource
```

**Exemple 19.4. Définir la Valeur de timeout d'un Bean Pool par le CLI**

```
[standalone@localhost:9999 /] /subsystem=ejb3/strict-max-bean-instance-pool=HSBeanPool:write-attribute(name="timeout", value="1500")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Report a bug

### 19.2.5. Assigner des Bean Pools aux Beans de session et aux Beans basés messages

Les administrateurs de systèmes JBoss peuvent assigner des Bean Pools individuels que les Session beans et les Bean basés-messages peuvent utiliser. Les Bean Pools peuvent être alloués par la Console de gestion ou le CLI.

Par défaut, deux Bean Pools sont fournis, **slsb-strict-max-pool** et **mdb-strict-max-pool** pour les Stateless sessions beans et les Beans basés-messages respectivement.

Pour créer ou modifier des Bean Pools, consulter Section 19.2.2, « Créer un Bean Pool » et Section 19.2.4, « Modifier un Bean Pool ».

**Procédure 19.7. Allouer des Bean Pools pour les Session beans ou pour les Beans basés-message par la Console de gestion.**

1. Connectez-vous à la Console de gestion. Section 3.4.2, « Connectez-vous à la Console de management »
2. Naviguer vers le panneau de Configuration de conteneurs EJB3.

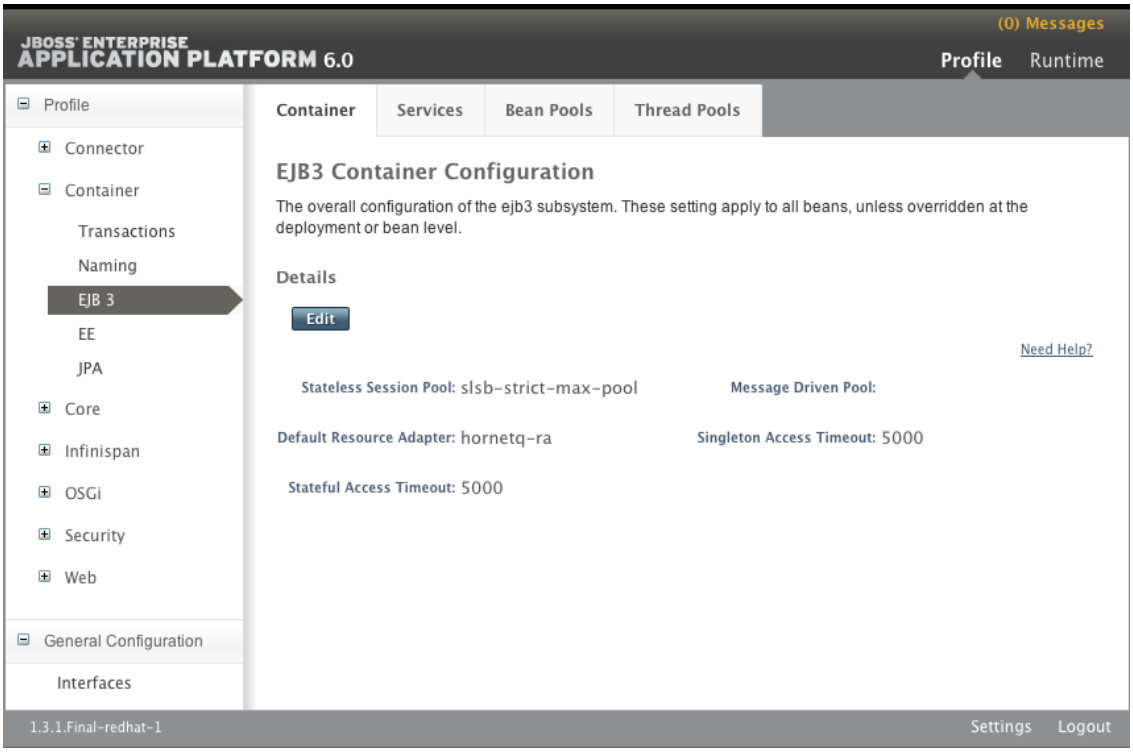



**Figure 19.5. Panneau de configuration de conteneurs EJB de la Console de gestion (Serveur autonome)**

3. Cliquer sur le bouton **`Edit`**. Les champs de la zone **`Details`** sont modifiables.
4. Sélectionner le Bean Pool à utiliser pour chaque type de bean à partir de la combo-box appropriée.
5. Cliquer sur le bouton **`Save`** si les changements vous conviennent, ou bien, cliquer sur le lien **`Cancel`** si vous souhaitez ignorer les changements.
6. La zone Détails sera maintenant modifiable et affichera la sélection Bean Pool qui convient.

**Procédure 19.8. Allouer des Bean Pools pour les Session beans ou pour les Beans basés-message par le CLI.**

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI ».
2. Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
/subsystem=ejb3:write-attribute(name="BEANTYPE", value="BEANPOOL")
```

   - Remplacer *BEANTYPE* par **`default-mdb-instance-pool`** pour les Bean basés-messages ou **`default-slsb-instance-pool`** pour les Stateless sessions beans.

- Remplacer *BEANPOOL* par le nom du Bean Pool à assigner.

3. Utiliser l'opération **`read-resource`** pour confirmer les changements.

```
/subsystem=ejb3:read-resource
```

**Exemple 19.5. Assigner un Bean Pool pour les Session beans par le CLI**

```
[standalone@localhost:9999 /] /subsystem=ejb3:write-
attribute(name="default-slsb-instance-pool", value="LV_SLSB_POOL")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Exemple 19.6. Exemple de configuration XML**

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
   <session-bean>
      <stateless>
         <bean-instance-pool-ref pool-name="slsb-strict-max-pool"/>
      </stateless>
      <stateful default-access-timeout="5000" cache-ref="simple"/>
      <singleton default-access-timeout="5000"/>
   </session-bean>
   <mdb>
      <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
      <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
   </mdb>


</subsystem>
```

Report a bug

## 19.3. CONFIGURER LES EJB THREAD POOLS

### 19.3.1. Enterprise Bean Thread Pools

JBoss Enterprise Application Platform 6 maintient un certain nombre d'instances d'objets de thread Java en mémoire à utiliser par les services de beans enterprise, y compris l'invocation à distante, le service de minuteur et l'invocation asynchrone.

Cette technique s'appelle le «thread pooling». Elle fournit une meilleure performance en éliminant l'étape de création de threads et procure à l'administrateur de services un moyen de contrôler l'utilisation des ressources.

On peut créer des pools de threads multiples par divers paramètres et chaque service peut recevoir ainsi un thread de pool différent.

Report a bug

### 19.3.2. Créer un Thread Pool

Les Thread Pool EJB peuvent être créés par la Console de gestion ou le CLI.

**Procédure 19.9. Créer un Thread Pool EJB par la Console de gestion**

1. Connectez-vous à la Console de gestion. Section 3.4.2, « Connectez-vous à la Console de management »

2. Cliquer sur **Profile** en haut à droite, puis développer l'item **Container** dans le panneau de Profil sur la gauche, et sélectionner **EJB 3**. Puis sélectionner l'onglet **Thread Pools** dans le panneau principal.

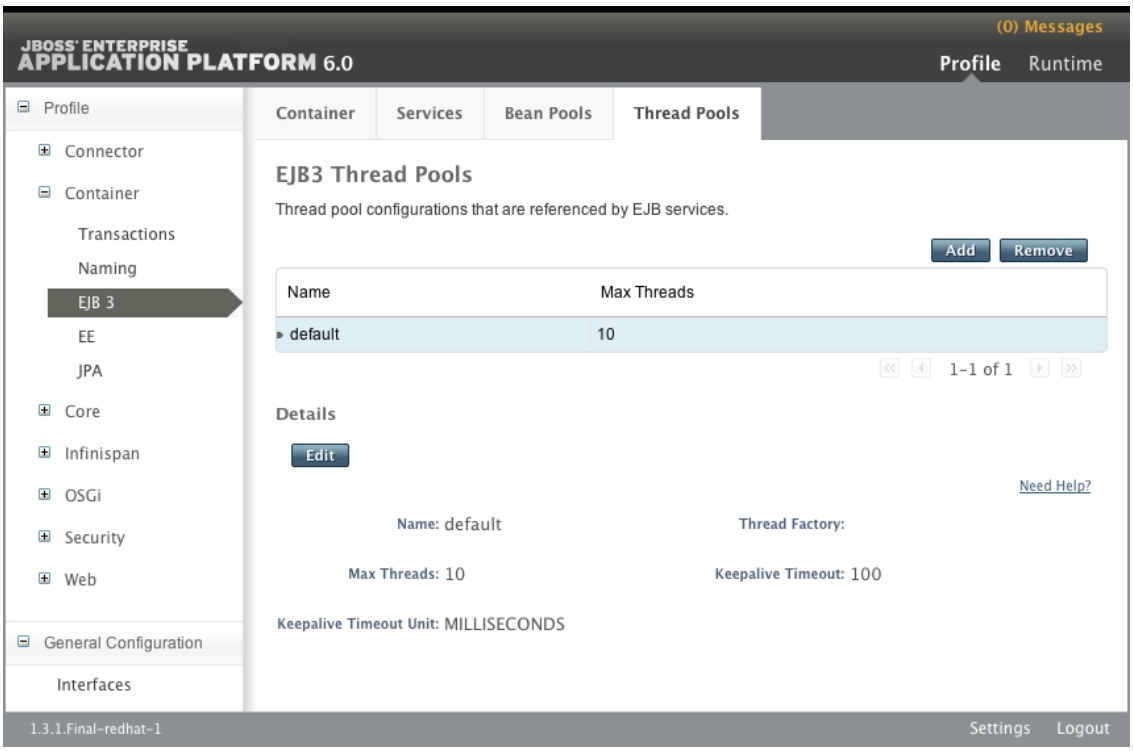



**Figure 19.6. Panneau EJB3 Thread Pools**

3. Cliquer sur le bouton **Add**. Le dialogue **Add EJB3 Thread Pools** apparaîtra.

4. Donnez les informations requises, les valeurs de **Name**, **Max Threads**, et **Keep-Alive Timeout**.

5. Cliquer sur le bouton **Save** pour sauvegarder le nouveau thread pool ou cliquer sur le lien **Cancel** pour annuler la procédure.

   - Si vous cliquez sur le bouton **Save**, le dialogue disparaîtra et le nouveau Pool Thread apparaîtra dans la liste.

   - Si vous cliquez sur **Cancel**, le dialogue se fermera et aucun autre Thread Pool ne sera créé.

**Procédure 19.10. Créer un Thread Pool par le CLI**

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI ».

2. Utiliser l'opération **add** avec la syntaxe suivante.

```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:add(max-threads=MAXSIZE,
keepalive-time={"time"=>"TIME", "unit"=>UNIT"})
```

   - Remplacer *BEANPOOLNAME* par le nom requis de Thread Pool.
   - Remplacer *MAXSIZE* par la taille maximum de thread Pool.
   - Remplacer *UNIT* par l'unité de temps requise de «keep-alive time».. Les valeurs permises sont les suivantes : **NANOSECONDS**, **MICROSECONDS**, **MILLISECONDS**, **SECONDS**, **MINUTES**, **HOURS**, et **DAYS**.
   - Remplacer *TIME* par la valeur (entier relatif) du «keep-alive time». Cette valeur doit correspondre à un nombre d'unités *UNIT*.

3. Utiliser l'opération **read-resource** pour confirmer la création d'un Bean Pool.

```
/subsystem=ejb3/strict-max-bean-instance-pool=THREADPOOLNAME:read-
resource
```

**Exemple 19.7. Créer un Thread Pool par le CLI**

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=testmepool:add(max-threads=50, keepalive-time={"time"=>"150",
"unit"=>"SECONDS"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Exemple 19.8. Exemple de configuration XML**

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">

   <thread-pools>
      <thread-pool name="default" max-threads="20" keepalive-
time="150"/>
   </thread-pools>

</subsystem>
```

Report a bug

### 19.3.3. Supprimer le Thread Pool

Les Thread Pool non utilisés peuvent être supprimés par la Console de Management.

**Prérequis**

- La Thread Pool que vous souhaitez supprimer ne peut pas être en cours d'utilisation. Voir les tâches suivantes pour vérifier que le thread pool n'est pas en cours d'utilisation :

- Section 19.6.2, « Configurer le Service de la minuterie EJB3 »
- Section 19.7.2, « Configurer le Thread Pool du Service d'invocations asynchrones EJB3 »
- Section 19.8.2, « Configurer EJB3 Remote Service »

**Procédure 19.11. Supprimer un Thread Pool EJB par la Console de management**

1. Connectez-vous à la console de management. Section 3.4.2, « Connectez-vous à la Console de management ».
2. Cliquer sur **Profile** en haut à droite, puis développer l'item **Container** dans le panneau de Profil sur la gauche, et sélectionner **EJB 3**. Puis sélectionner l'onglet **Thread Pools** dans le panneau principal.

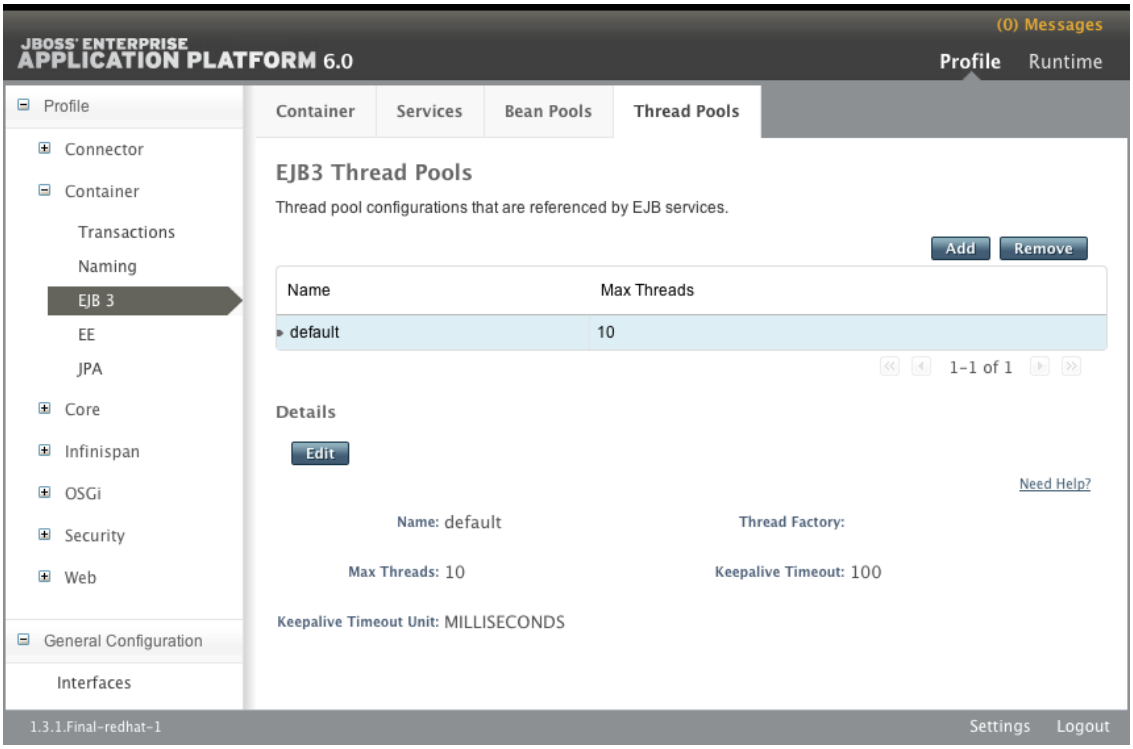



**Figure 19.7. Panneau EJB3 Thread Pools**

3. Cliquer sur le Thread Pool que vous souhaiter supprimer de la liste.
4. Cliquer sur le bouton **`Remove`**. La boîte de dialogue **`Remove Item`** dialog appears.
5. Cliquer sur le bouton **`OK`** pour confirmer la suppression ou cliquer sur le lien **`Cancel`** pour abandonner l'opération.

   Si vous cliquer sur le bouton **`Ok`**, le dialogue se fermera et le Thread Pool sera supprimé et retiré de la liste.

   Si vous cliquez sur le bouton **`Cancel`**, la boîte de dialogue se fermera et il n'y aura aucun changement.

**Procédure 19.12. Supprimer un Thread Pool par le CLI**

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI ».

2. Utiliser l'opération **`remove`** avec la syntaxe suivante.

```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:remove
```

   - Remplacer *THREADPOOLNAME* par le nom requis de Thread Pool.

**Exemple 19.9. Supprimer un Thread Pool par le CLI**

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=ACCTS_THREADS:remove
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Report a bug

### 19.3.4. Modifier un Thread Pool

Les administrateurs JBoss peuvent modifier les Thread Pools par la Console de gestion ou le CLI.

**Procédure 19.13. Modifier un Thread Pool par la Console de management**

1. **Login**
   Connectez-vous à la Console de gestion.

2. **Naviguez dans l'onglet EJB3 Thread Pools**
   Cliquer sur **Profile** en haut à droite, puis développer l'item **Container** dans le panneau de Profil sur la gauche, et sélectionner **EJB 3**. Puis sélectionner l'onglet **Thread Pools** dans le panneau principal.

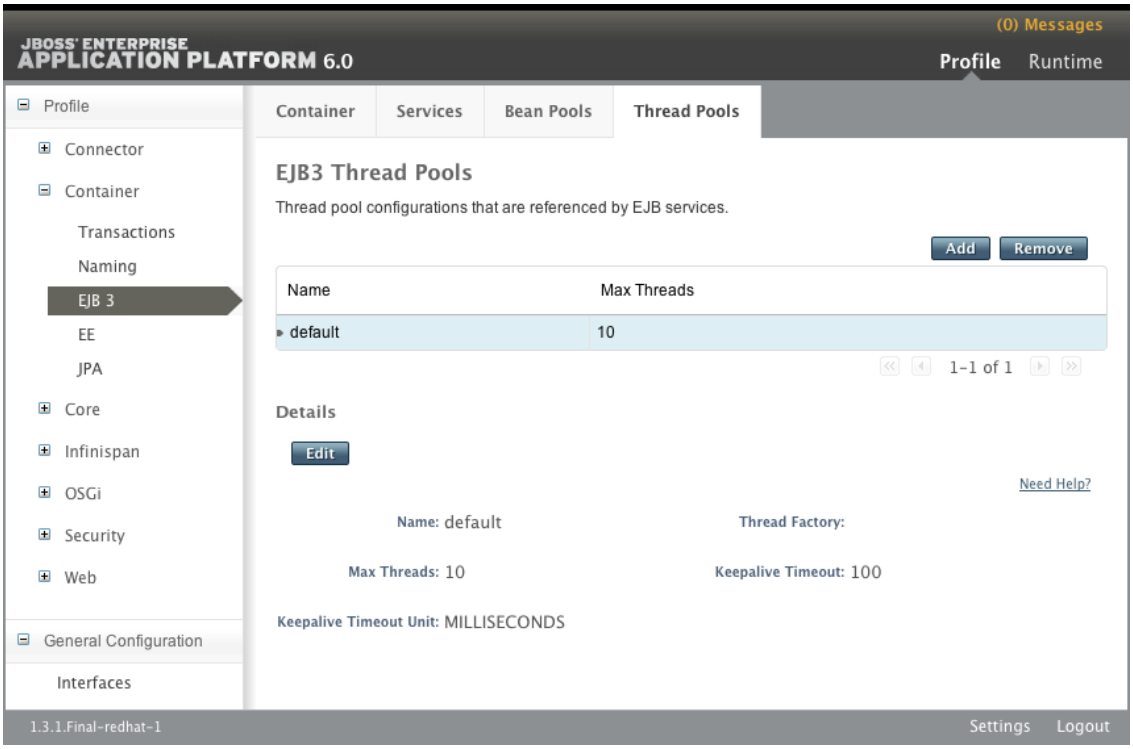



**Figure 19.8. Onglet EJB3 Thread Pools**

3. **Sélectionner le Thread Pool à modifier**
   Sélectionner le Thread Pool que vous souhaiter supprimer de la liste.

4. **Cliquer sur le bouton Edit**
   Les champs et les espaces réservés aux détails sont maintenant modifiables.

5. **Modifier Détails**
   Modifier les détails que vous souhaitez modifier. Vous ne pourrez modifier que les valeurs suivantes : **`Thread Factory`**, **`Max Threads`**, **`Keepalive Timeout`**, et **`Keepalive Timeout Unit`**.

6. **Save ou Cancel**
   Cliquer sur le bouton **`Save`** si les changements vous conviennent, ou bien, cliquer sur le lien **`Cancel`** si vous souhaitez ignorer les changements.

**Procédure 19.14. Modifier un Thread Pool par le CLI**

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI ».

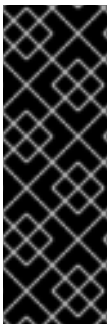
2. Utiliser l'opération **`write_attribute`** avec la syntaxe suivante pour chaque attribut du Thread Pool à modifier.

   ```
   /subsystem=ejb3/thread-pool=THREADPOOLNAME:write-
   attribute(name="ATTRIBUTE", value="VALUE")
   ```

   - Remplacer *THREADPOOLNAME* par le nom requis de Thread Pool.
   - Remplacer *ATTRIBUTE* par le nom de l'attribut à modifier. Les attributs ne pouvant pas être modifiés de cette façon sont **`keepalive-time`**, **`max-thread`**, et **`thread-factory`**.
   - Remplacer *VALUE* par la valeur requise de l'attribut.

3. Utiliser l'opération **`read-resource`** pour confirmer les changements au Thread Pool.

   ```
   /subsystem=ejb3/thread-pool=THREADPOOLNAME:read-resource
   ```



**IMPORTANT**

Quand vous changez la valeur de l'attribut **`keepalive-time`** par le CLI, la valeur requise correspond à une représentations d'objet. La syntaxe sera la suivante :

```
/subsystem=ejb3/thread-pool=THREADPOOLNAME:write-
attribute(name="keepalive-time", value={"time" =>
"VALUE","unit" => "UNIT"}
```

**Exemple 19.10. Définir la Valeur maximum Maxsize d'un Thread Pool par le CLI**

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=HSThreads:write-attribute(name="max-threads", value="50")
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Exemple 19.11. Définir la valeur de temps `keepalive-time` d'un Thread Pool par le CLI**

```
[standalone@localhost:9999 /] /subsystem=ejb3/thread-
pool=HSThreads:write-attribute(name="keepalive-time", value=
{"time"=>"150"})
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

Report a bug

## 19.4. CONFIGURER LES SESSION BEANS

### 19.4.1. Session Bean Access Timeout

Les Stateful beans et les Singleton Session Beans ont une valeur de délai d'accès précisée pour les accès simultanés. Cette valeur correspond à la période pendant laquelle une demande de méthode de bean de session peut être bloquée avant qu'il y ait timeout.

La valeur de timeout et l'unité de temps utilisées peut être spécifiée grâce à l'annotation **`@javax.ejb.AccessTimeout`** sur la méthode.

Si non spécifié, JBoss Enterprise Application Platform 6 fournit une valeur de timeout de 5000 millisecondes.

Consulter Javadocs pour AccessTimeout à l'adresse suivante :
http://docs.oracle.com/javaee/6/api/javax/ejb/AccessTimeout.html

Report a bug

### 19.4.2. Définir les valeurs de timeout d'accès aux beans de session par défaut

Les administrateurs de systèmes JBoss peuvent spécifier les valeurs de timeout par défaut des beans de session Stateful ou Singleton. Les valeurs de timeout par défaut peuvent être modifiées par la Console de gestion ou le CLI. La valeur par défaut est de 5000 millisecondes.

**Procédure 19.15. Définir les valeurs de timeout d'accès aux beans de session par défaut par la Console de gestion**

1. Connectez-vous à la Console de gestion. Voir Section 3.4.2, « Connectez-vous à la Console de management ».

2. Cliquer sur **Profile** en haut à droite, puis développer l'item **Container** dans le panneau de Profil sur la gauche, et sélectionner **EJB 3**. Puis sélectionner l'onglet **Container** dans le panneau principal.

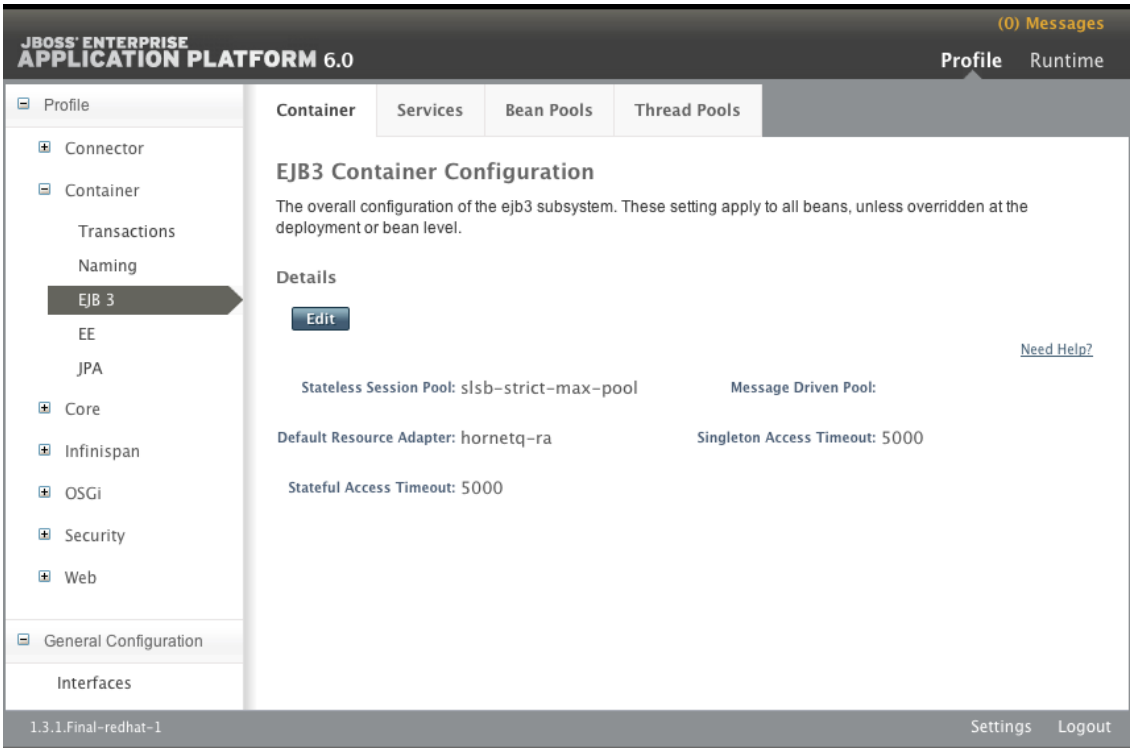



**Figure 19.9. Panneau de configuration de conteneurs EJB de la Console de gestion (Serveur autonome)**

3. Cliquer sur le bouton **`Edit`**. Le champ de la zone **`Details`** est maintenant modifiable.
4. Saisir les valeurs qui conviennent dans **`Stateful Access Timeout`** et/ou dans les cases de texte **`Singleton Access Timeout`**.
5. Cliquer sur le bouton **`Save`** si les changements vous conviennent, ou bien, cliquer sur le lien **`Cancel`** si vous souhaitez ignorer les changements.
6. La zone de **`Details`** sera alors non modifiable et affichera les valeurs de timeout qui conviennent.

**Procédure 19.16. Définir les valeurs de timeout d'accès aux beans de session par par le CLI.**

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI ».
2. Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
/subsystem=ejb3:write-attribute(name="BEANTYPE", value=TIME)
```

   - Remplacer *BEANTYPE* par **`default-stateful-bean-access-timeout`** pour les sessions beans Stateful, ou **`default-singleton-bean-access-timeout`** pour les sessions bean Singleton.
   - Remplacer *TIME* par la valeur de timeout qui convient.

3. Utiliser l'opération **`read-resource`** pour confirmer les changements.

```
/subsystem=ejb3:read-resource
```

**Exemple 19.12. Définir la valeur de timeout d'accès aux beans Stateful par par le CLI à 9000.**

```
[standalone@localhost:9999 /] /subsystem=ejb3:write-
attribute(name="default-stateful-bean-access-timeout", value=9000)
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

**Exemple 19.13. Exemple de configuration XML**

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">
   <session-bean>
      <stateless>
         <bean-instance-pool-ref pool-name="slsb-strict-max-pool"/>
      </stateless>
      <stateful default-access-timeout="5000" cache-ref="simple"/>
      <singleton default-access-timeout="5000"/>
   </session-bean>

</subsystem>
```

Report a bug

## 19.5. CONFIGURER LES MESSAGE-DRIVEN BEANS

### 19.5.1. Définir l'Adaptateur de ressources par défaut des Beans basés-messages

Les administrateurs de systèmes JBoss peuvent spécifier l'adaptateur de ressources par défaut utilisé par les beans basés-message. L'adaptateur de ressources par défaut peut être spécifié par la Console de gestion ou le CLI. L'adaptateur de ressources fourni par défaut dans JBoss Enterprise Applications Platform 6 est **`hornetq-ra`**.

**Procédure 19.17. Définir l'adaptateur de ressources par défaut pour les beans basés-messages par la Console de gestion.**

1. Connectez-vous à la Console de gestion. Section 3.4.2, « Connectez-vous à la Console de management »

2. Cliquer sur **Profile** en haut à droite, puis développer l'item **Container** dans le panneau de Profil sur la gauche, et sélectionner **EJB 3**. Puis sélectionner l'onglet **Container** dans le panneau principal.

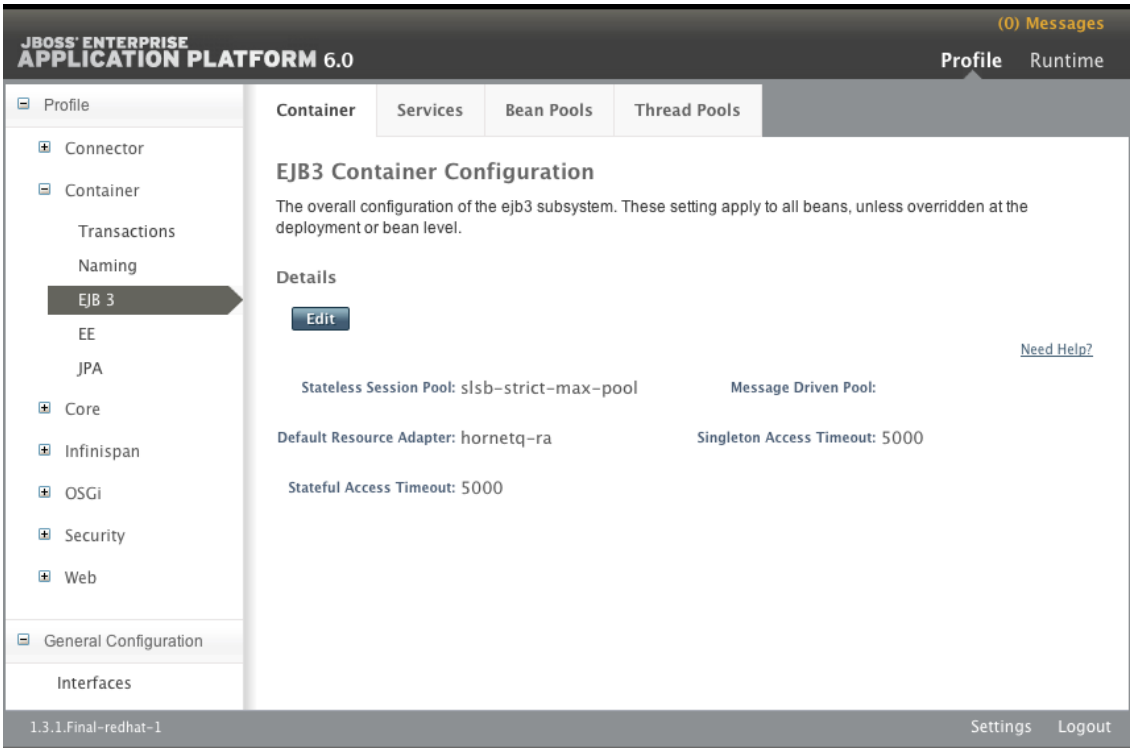



**Figure 19.10. Panneau de configuration de conteneurs EJB de la Console de gestion (Serveur autonome)**

3. Cliquer sur le bouton **`Edit`**. Le champ de la zone **`Details`** est maintenant modifiable.
4. Saisir le nom de l'adaptateur de la ressource à utiliser dans la case de texte **`Default Resource Adapter`**.
5. Cliquer sur le bouton **`Save`** pour conserver les changements, ou bien cliquer sur le lien **`Cancel`** si vous souhaitez les ignorer.
6. La zone **`Details`** sera alors non modifiable et affichera le nom d'adaptateur de ressources qui convient.

**Procédure 19.18. Définir l'adaptateur de ressources par défaut pour les beans basés-messages par le CLI**

1. Lancer l'outil CLI et connectez-vous à votre serveur. Voir Section 3.5.4, « Se connecter à une instance de serveur géré par le Management CLI ».
2. Utiliser l'opération **`write-attribute`** avec la syntaxe suivante.

```
/subsystem=ejb3:write-attribute(name="default-resource-adapter-name", value="RESOURCE-ADAPTER")
```

   Remplacer *RESOURCE-ADAPTER* par le nom de l'adaptateur de ressources à utiliser.
3. Utiliser l'opération **`read-resource`** pour confirmer les changements.

```
/subsystem=ejb3:read-resource
```

**Exemple 19.14. Définir l'adaptateur de ressources par défaut pour les beans basés-messages par le CLI**

```
[standalone@localhost:9999 subsystem=ejb3] /subsystem=ejb3:write-
attribute(name="default-resource-adapter-name", value="EDIS-RA")
{"outcome" => "success"}
[standalone@localhost:9999 subsystem=ejb3]
```

**Exemple 19.15. Exemple de configuration XML**

```
<subsystem xmlns="urn:jboss:domain:ejb3:1.2">

   <mdb>
      <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
      <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
   </mdb>


</subsystem>
```

Report a bug

## 19.6. CONFIGURER LE SERVICE DE MINUTERIE EJB3

### 19.6.1. Service de minuterie EJB3

Le service de minuterie EJB3 est un service standard Java EE 6 pour programmer l'invocation de méthodes à partir de beans enterprise. Les beans de sessions Stateless, les beans de sessions Singleton et les beans basés-messages peuvent tous programmer un rappel de n'importe quelle méthode qui leur appartient à un moment précis, après un intervalle de temps, ou à un intervalle récurrent, ou encore sur la base d'un calendrier.

Report a bug

### 19.6.2. Configurer le Service de la minuterie EJB3

Les administrateurs JBoss peuvent configurer le Service de la minuterie EJB3 dans la Console de gestion de JBoss Enterprise Application Platform 6. Les fonctions pouvant être configurées sont le thread pool utilisé pour l'invocation de beans programmés et le répertoire où les données du Service de la minuterie sont stockées.

**Procédure 19.19. Configurer la Service du timer EJB3**

1. **Login**
   Connectez-vous à la Console de gestion.

2. **Ouvrir l'onglet de Service de minuterie**
   Cliquer sur **Profile** en haut à droite, puis développer l'item **Container** dans le panneau de Profil sur la gauche, et sélectionner **EJB 3**. Puis sélectionner l'onglet **Service** dans le panneau principal, puis l'onglet **Timer Service**.

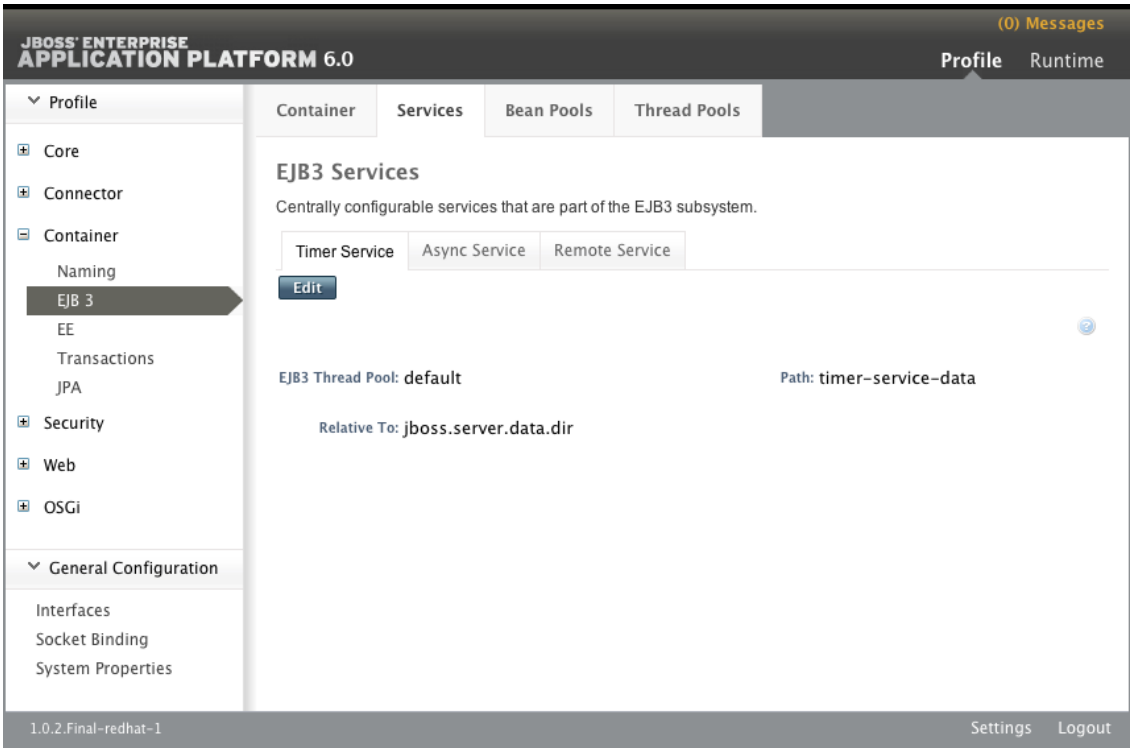



**Figure 19.11. L'onglet de Service de minuterie du panneau EJB3 Services**

3. **Saisir le Mode Édition**
   Cliquer sur le bouton 'Edit'. Les champs devraient maintenant être modifiables.

4. **Effectuer les changements requis.**
   Vous pouvez sélectionner un Thread Pool EJB3 différent pour le Service de minuterie si les Thread Pools supplémentaires sont été configurés, et vous pouvez changer le répertoire utilisé pour sauvegarder les données du Service de minuterie. La configuration de répertoire de données du Service de minuterie comprend deux valeurs: **`Path`**, le répertoire qui contient les données, et **`Relative To`**, le répertoire qui contient **`Path`**. Par défaut **`Relative To`** est défini à une variable de chemin de système de fichiers.

5. **Sauvegarder ou Annuler**
   Cliquer sur le bouton **`Save`** si les changements vous conviennent, ou bien, cliquer sur le lien **`Cancel`** si vous souhaitez ignorer les changements.

Report a bug

# 19.7. CONFIGURER LE SERVICE D'INVOCATION ASYNCHRONE EJB

## 19.7.1. EJB3 Service d'invocations asynchrones

Le service d'invocations asynchrones est un service de conteneurs JavaBeans Enterprise qui gère l'invocation asynchrone des méthodes de beans de sessions. Ce service maintient un certain nombre d'invocations asynchrones configurables (Thread Pool) qui sont allouées pour l'exécution de méthodes asynchrones.

Enterprise JavaBeans 3.1 permet à toute méthode de bean de session (stateful, stateless, ou singleton) d'être annotée pour permettre l'exécution asynchrone.

Report a bug

### 19.7.2. Configurer le Thread Pool du Service d'invocations asynchrones EJB3

Les administrateurs JBoss peuvent configurer le Service d'invocations asynchrones EJB3 dans la console JBoss Enterprise Application Platform 6 Management Console pour permettre l'utilisation d'un Thread pool spécifique.

**Procédure 19.20. Configurerhttp://francegourmet.com.au/product-category/snails-and-mushrooms/**

1. **Login**
   Connectez-vous à la Console de gestion.

2. **Ouvrir l'onglet Async Service**
   Cliquer sur **Profile** en haut à droite, étendre l'item **Container** dans le panneau Profile qui se trouve sur la gauche et sélectionner **EJB 3**. Sélectionner l'onglet **Services** à partir du panneau principal, puis l'onglet **Async Service**.

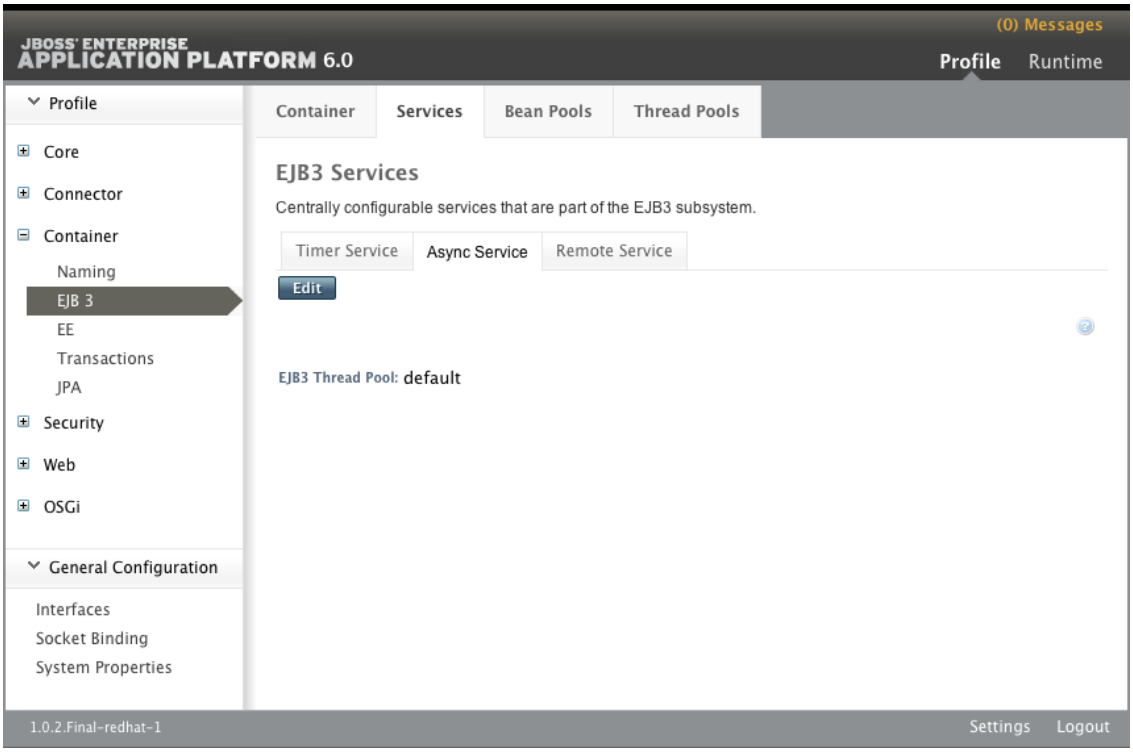



**Figure 19.12. L'onglet Async Service du panneau Services EJB3**

3. **Saisir le Mode Édition**
   Cliquer sur le bouton **`Edit`**. Les champs sont alors modifiables.

4. **Sélectionner le Thread Pool**
   Sélectionner le Thread Pool à utiliser à partir de la liste. Le Thread Pool devra déjà avoir été créé.

5. **Sauvegarder ou Annuler**
   Cliquer sur le bouton **`Save`** si les changements vous conviennent, ou bien, cliquer sur le lien **`Cancel`** si vous souhaitez ignorer les changements.

Report a bug

## 19.8. CONFIGURER EJB3 REMOTE INVOCATION SERVICE

### 19.8.1. EJB3 Remote Service

Le service EJB3 Remote gère l'exécution à distance des Beans Enterprise dans les interfaces commerciales à distance.

Report a bug

### 19.8.2. Configurer EJB3 Remote Service

Les administrateurs JBoss peuvent configurer EJB3 Remote Service dans la Console de gestion de JBoss Enterprise Application Platform 6. Les fonctions pouvant être configurées sont le thread pool utilisé pour l'invocation de beans programmés et le connecteur sur lequel le réseau EJB3 Remoting est enregistré.

**Procédure 19.21. Configurer EJB3 Remote Service**

1. **Login**
   Connectez-vous à la Console de gestion.

2. **Ouverture de l'onglet Remote Service**
   Cliquer sur **Profile** en haut à droite, puis développer l'item **Container** dans le panneau de Profil sur la gauche, et sélectionner **EJB 3**. Puis sélectionner l'onglet **Service** dans le panneau principal, puis l'onglet **Remote Service**.

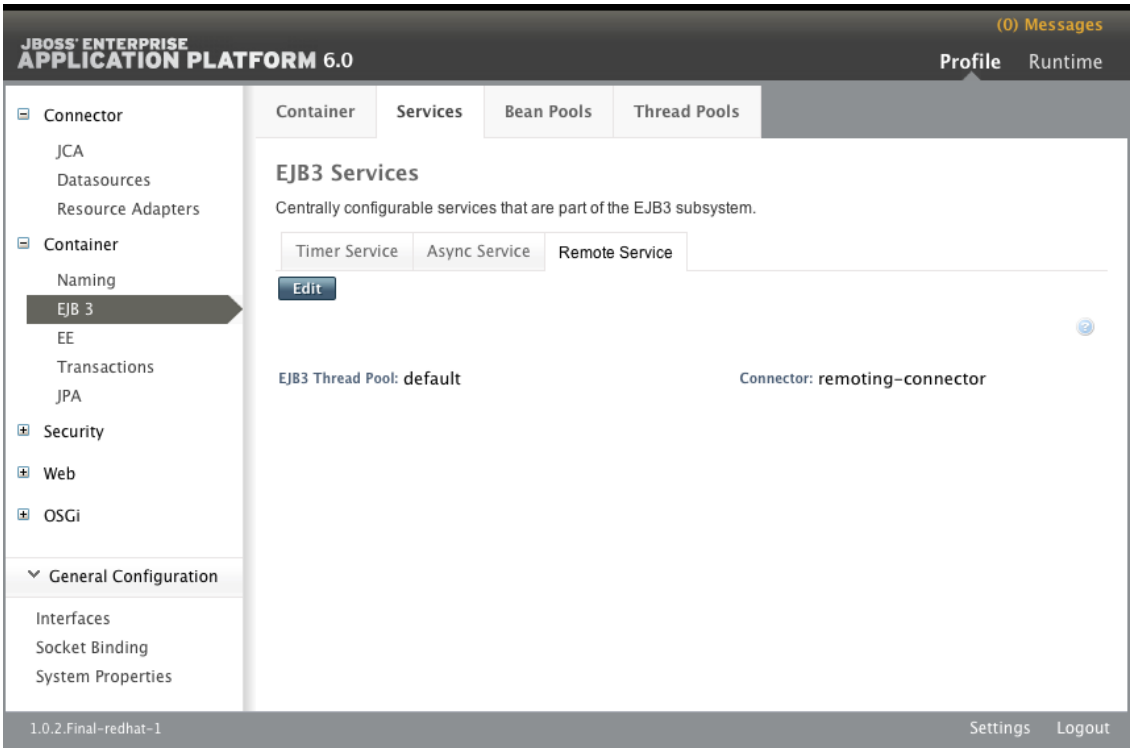



**Figure 19.13. L'onglet Remote Service du panneau Services EJB3**

3. **Saisir le Mode Édition**
   Cliquer sur le bouton Édit. Les champs seront alors modifiables.

4. **Effectuer les changements requis.**
   Vous pouvez sélectionner un EJB3 Thread Pool différent pour Remote Service si les Thread Pools supplémentaires ont été configurés. Vous pouvez changer le connecteur utilisé pour enregistrer le Canal EJB Remoting.

5. **Sauvegarder ou Annuler**
   Cliquer sur le bouton **Save** si les changements vous conviennent, ou bien, cliquer sur le lien **Cancel** si vous souhaitez ignorer les changements.

Report a bug

## 19.9. CONFIGURER LES EJB 2.X ENTITY BEANS

### 19.9.1. EJB Entity Beans

Les EJB Entity Beans sont un type de bean enterprise de la version 2.x de la spécification EJB qui représentait des données persistantes qui étaient maintenues dans une base de données. Les Entity Beans ont été remplacés par les entité JPA et ont été officiellement listées pour être retirées (nettoyées) des versions futures de la spécification. Red Hat ne recommande pas l'utilisation des Entity Beans, mis à part pour raison de compatibilité rétro-active.

Le support des Entity Beans est désactivé par défaut dans JBoss Enterprise Application Platform 6.

Report a bug

### 19.9.2. Container-Managed Persistence

Container-Managed Persistence (CMP) est un service fourni par un serveur d'applications qui procure la persistance des données pour les beans Entity.

Report a bug

### 19.9.3. Activer EJB 2.x Container-Managed Persistence

Container-Managed Persistence (CMP) est géré par l'extension **org.jboss.as.cmp**. CMP est activé par défaut dans le domaine géré et dans la configuration complète du serveur autonome, par ex. **standalone-full.xml**.

Pour activer CMP dans une configuration différente, ajouter le module **org.jboss.as.cmp** à la liste d'extensions actives dans le fichier de configuration du serveur.

```
<extensions>
        <extension module="org.jboss.as.cmp"/>
</extensions>
```

Pour désactiver CMP dans une configuration de serveur, supprimer l'entrée d'extension du module **org.jboss.as.cmp**.

Report a bug

### 19.9.4. Configurer EJB 2.x Container-Managed Persistence

Le sous-système EJB 2.x Container Managed Persistence (CMP) peut être configuré pour spécifier un certain nombre de générateurs de clés. Les générateurs de clés sont utilisés pour produire des clés uniques pour identifier chaque entité persistée par le service CMP.

Il existe deux types de générateurs de clés: les générateurs de clés basés-UUID et les générateurs de clés HiLO

### Les générateurs de clés basés-UUID

Un générateur de clés basé-UUID crée des clés qui utilisent un Identifiant Unique Universel (UUI). Les générateurs de clés UUID ont besoin uniquement d'avoir un nom unique; ils n'ont pas d'autre configuration.

Les générateurs de clés basé-UUID peuvent être ajoutés par le CLI avec la syntaxe suivante.

```
/subsystem=cmp/uuid-keygenerator=UNIQUE_NAME:add
```

**Exemple 19.16. Ajouter le générateur de clés UUID**

Pour ajouter un générateur de clés basé-UUID ayant pour nom **`uuid_identities`**, utiliser cette commande CLI :

```
/subsystem=cmp/uuid-keygenerator=uuid_identities:add
```

La configuration XML créée par cette commande est :

```
<subsystem xmlns="urn:jboss:domain:cmp:1.0">
   <key-generators>
      <uuid name="uuid_identities" />
   </key-generators>
</subsystem>
```

### Générateurs de clés HiLo

Les générateurs de clés HiLo utilisent une base de données pour créer et stocker les clés d'identité des enttités. Le générateur de clés HiLo doivent posséder des noms uniques et sont configurés avec des propriétés qui spécifient la source de données utilisée pour stocker les données, ainsi que les noms du tableau et colonnes qui stockent les clés.

Les générateurs de clés HiLo peuvent être ajoutés par le CLI grâce à la syntaxe de commande suivante:

```
/subsystem=cmp/hilo-keygenerator=UNIQUE_NAME/:add(property=value,
property=value, ...)
```

**Exemple 19.17. Ajouter un générateur de clés HiLo**

```
/subsystem=cmp/hilo-keygenerator=DB_KEYS/:add(create-
table=false,data-source=java:jboss/datasources/ExampleDS,drop-
table=false,id-column=cmp_key_ids,select-hi-ddl=select
max(cmp_key_ids) from cmp_key_seq,sequence-column=cmp_key_seq,table-
name=cmp-keys))
```

La configuration XML créée par cette commande est :

```
<subsystem xmlns="urn:jboss:domain:cmp:1.0">
   <key-generators>
      <hilo name="DB_KEYS">
         <create-table>false</create-table>
         <data-source>java:jboss/datasources/ExampleDS</data-source>
```

```
            <drop-table>false</drop-table>
            <id-column>cmp_key_ids</id-column>
            <select-hi-ddl>select max(cmp_key_ids) from
cmp_key_seq</select-hi-ddl>
            <sequence-column>cmp_key_seq</sequence-column>
            <table-name>cmp-keys</table-name>
         </hilo>
      </key-generators>
   </subsystem>
```

Report a bug

### 19.9.5. Les propriétés de sous-système CMP pour les Générateurs de clés HiLo

**Tableau 19.1. Les propriétés de sous-système CMP pour les Générateurs de clés HiLo**

| Propriété | Type des données | Description |
|---|---|---|
| **block-size** | long | - |
| **create-table** | booléen | Si défini sur **TRUE**, le tableau **table-name** sera créé avec le contenu **create-table-ddl** si le tableau n'est pas trouvé. |
| **create-table-ddl** | chaîne | Les commandes DDL utilisées pour créer le tableau spécifié dans **table-name** si on ne **create-table** is set to **TRUE**. |
| **data-source** | token | La source de données utilisée pour se connecter à la base de données. |
| **drop-table** | booléen | - |
| **id-column** | token | - |
| **select-hi-ddl** | chaîne | La commande SQL qui retournera la plus grande clé actuellement stockée. |
| **sequence-column** | token | - |
| **sequence-name** | token | - |
| **table-name** | token | Nom de la table utilisée pour stocker les informations sur les clés |

Report a bug

# CHAPITRE 20. JAVA CONNECTOR ARCHITECTURE (JCA)

## 20.1. INTRODUCTION

### 20.1.1. Java EE Connector API (JCA)

JBoss Enterprise Application Platform 6 fournit un support complet à la spécification Java EE Connector API (JCA). Voir JSR 322: Java EE Connector Architecture 1.6 pour obtenir plus d'informations sur la spécification JCA.

Un adaptateur de ressources est un composant qui implémente l'architecture de Java EE Connector API. Il ressemble à un objet de source de données, mais fournit une connectivité à partir d'EIS (Enterprise Information System) vers un grand nombre de systèmes hétérogènes, comme des bases de données, systèmes de messagerie, traitement de transactions, et systèmes ERP (Enterprise Resource Planning).

Report a bug

### 20.1.2. Java Connector Architecture (JCA)

La Java EE Connector Architecture (JCA) définit une architecture standard pour les systèmes de Java EE pour les systèmes externes hétérogènes Enterprise Information Systems (EIS). Exemples de systèmes EISs : Enterprise Resource Planning (ERP), transaction central de traitement (TP), bases de données et systèmes de messagerie.

JCA 1.6 fournit des fonctionnalités de gestion :

- connections
- transactions
- sécurité
- cycle de vie
- Instances de travail
- Flux interne de transactions
- Flux interne de messages

JCA 1.6 a été développé sous Java Community Process en tant que JSR-322, http://jcp.org/en/jsr/detail?id=313.

Report a bug

### 20.1.3. Adaptateurs de ressources

Un adaptateur de ressources est un composant Java EE déployable qui permet la communication entre une application Java EE et une entreprise d'informations système (EIE) à l'aide de la spécification Java Connector Architecture (JCA). Un adaptateur de ressources est souvent fourni par les fournisseurs de l'EIE pour permettre une intégration facile de leurs produits avec des applications Java EE.

Un système d'Information Enterprise peut être n'importe quel autre système de logiciel au sein d'une organisation. Les exemples incluent les systèmes ERP (Enterprise Resource Planning), les systèmes de base de données, les serveurs d'e-mails et les systèmes de messagerie propriétaires.

Un adaptateur de ressources est empaqueté dans un fichier de Ressources Adaptateur Archive (RAR) qui peut être déployé dans JBoss Enterprise Application Platform 6. Un fichier RAR peut également être inclus dans un déploiement Enterprise Archive (EAR).

Report a bug

## 20.2. CONFIGURATION DU SOUS-SYSTÈME JAVA CONNECTOR ARCHITECTURE (JCA)

Le sous-système JCA du fichier de configuration de JBoss Enterprise Application Platform contrôle les paramètres de configuration généraux du conteneur JCA et déploiements d'adaptateurs de ressources.

**Éléments clés du sous-système JCA**

**Validation d'archive**

- Ce paramétrage indique si la validation d'archivage doit avoir lieu sur les unités de déploiement.
- Le tableau suivant décrit les attributs que vous pouvez définir pour la validation d'archivage.

**Tableau 20.1. Attributs de validation d'archivage**

| Attribut | Valeur par défaut | Description |
|---|---|---|
| `enabled` | true | Indique si la validation d'archivage est activée. |
| `fail-on-error` | true | Indique si un rapport d'erreur de validation d'archivage a fait échouer le développement. |
| `fail-on-warn` | false | Indique si un rapport d'avertissement de validation d'archivage a fait échouer le développement. |

- Si une archive n'implémente pas la spécification Java EE Connector Architecture correctement, et que la validation d'archivage est activée, un message d'erreur s'affichera pendant le déploiement pour décrire le problème, comme par exemple :

```
Severity: ERROR
Section: 19.4.2
Description: A ResourceAdapter must implement a "public int
hashCode()" method.
Code: com.mycompany.myproject.ResourceAdapterImpl

Severity: ERROR
Section: 19.4.2
Description: A ResourceAdapter must implement a "public boolean
equals(Object)" method.
Code: com.mycompany.myproject.ResourceAdapterImpl
```

- Si la validation d'archivage n'est pas spécifiée, on la considérera comme présente et l'attribut **`enabled`** aura comme valeur true par défaut.

**Validation de bean**

- Ce paramètre indique si la validation de bean (JSR-303) aura lieu sur les unités de déploiement.
- Le tableau ci-dessous décrit les attributs que vous pouvez déterminer pour la validation de bean.

**Tableau 20.2. Attributs de validation de bean**

| Attribut | Valeur par défaut | Description |
|---|---|---|
| **`enabled`** | true | Indique si la validation de bean est activée. |

- Si la validation de bean n'est pas spécifiée, on la considérera comme présente et l'attribut **`enabled`** aura comme valeur true par défaut.

**Work Managers**

- Il y a deux types de Work Managers :

  **Work Manager par défaut**

  Le Work Manager par défaut et ses Thread Pools.

  **Work Manager personnalisé**

  Une définition de Work Manager et ses Thread Pools.

- Le tableau suivant décrit les attributs que vous pouvez définir pour les Work Managers.

**Tableau 20.3. Attributs de Work Managers**

| Attribut | Description |
|---|---|
| **`name`** | Indique le nom du Work Manager. Requis pour les Work Managers personnalisés. |
| **`short-running-threads`** | Thread Pool pour les instances Work standards. Chaque Work Manager un un Thread Pool à exécution courte. |
| **`long-running-threads`** | Les instances Work de Thread pool de JCA 1.6 qui définissent **`LONG_RUNNING`**. Chaque Work Manager peut avoir un Thread pool de longue durée en option. |

- Le tableau ci-dessous décrit les attributs que vous pouvez définir pour les Thread pools de Work Managers.

**Tableau 20.4. Attributs de Thread pool**

| Attribut | Description |
|---|---|
| `allow-core-timeout` | Paramètre booléen qui détermine quels threads principaux risquent d'expirer. La valeur par défaut est false. |
| `core-threads` | La taille du pool de threads. Doit être inférieure à la taille de pool de threads maximum. |
| `queue-length` | La longueur maximum de la file d'attente. |
| `max-thread` | Taille de pool de threads maximum. |
| `keepalive-time` | Indique le durée pendant laquelle les threads de pool doivent être conservés après avoir complété leur tâche. |
| `thread-factory` | Référence à la fabrique de threads. |

### Bootstrap Contexts

- Utilisé pour définir les contextes de bootstapping (démarrage) personnalisés.
- Le tableau suivant décrit les attributs à définir pour les contextes de bootstraping.

**Tableau 20.5. Attributs de contexte de bootstrapping**

| Attribut | Description |
|---|---|
| `name` | Indique le nom du contexte de bootstrapping |
| `workmanager` | Indique le nom du Work Manager à utiliser dans ce contexte. |

### Gestionnaire de connexion mis en cache

- Utilisé pour déboguer les connexions et pour supporter l'inscription tardive d'une connexion dans une transaction, pour vérifier leur bonne utilisation et fonctionnement.
- Le tableau suivant décrit les attributs que vous pouvez définir pour le manager de connexions mis en cache.

**Tableau 20.6. Attributs de manager de connexion mis en cache**

| Attribut | Valeur par défaut | Description |
|---|---|---|
| `debug` | false | Sorties avertissement en cas d'échec de fermeture explicite des connexions |
| `error` | false | Envoie une exception en cas d'échec de fermeture explicite des connexions. |

**Procédure 20.1. Configurer le sous-système JCA par la Console de management**

1. Le sous-système JCA de JBoss Enterprise Application Platform 6 peut être configuré dans la Console de management. Les options de configuration de JCA sont situées dans des endroits légèrement différents dans la Console de management, selon la façon dont le serveur est exécuté.

   - Si le serveur exécute ne autonome, suivre les étapes suivantes :

     a. Cliquer sur le lien **`Profile`** qui se trouve en haut et à droite de la vue **`Profile`**.

     b. Veillez à ce que la section **`Profile`** du panneau de navigation à gauche soit développée.

     c. Cliquer sur **`Connector`** pour l'étendre, puis cliquer sur **`JCA`**.

   - Si le serveur exécute dans le cadre d'une Managed Domain, suivre les étapes suivantes :

     a. Cliquer sur le lien **`Profile`** qui se trouve en haut et à droite de la vue **`Profile`**

     b. Sélectionner le profil que vous souhaitez modifier à partir du menu **`Profile`** en haut du panneau de navigation sur la gauche.

     c. Cliquer sur **`Connector`** pour l'étendre, puis cliquer sur **`JCA`**.

2. Configurer les paramètres du sous-système JCA à l'aide des trois onglets.

   a. **Config courante**
   L'onglet **`Common Config`** contient des paramètres pour le gestionnaire de connexions en cache, la validation de l'archive et la validation de bean (JSR-303). Chacun d'entre eux est contenu dans son onglet propre. Ces réglages peuvent être changés en ouvrant l'onglet correspondant, en cliquant sur le bouton Edit, en effectuant les changements nécessaires et puis en cliquant sur le bouton Save de sauvegarde.

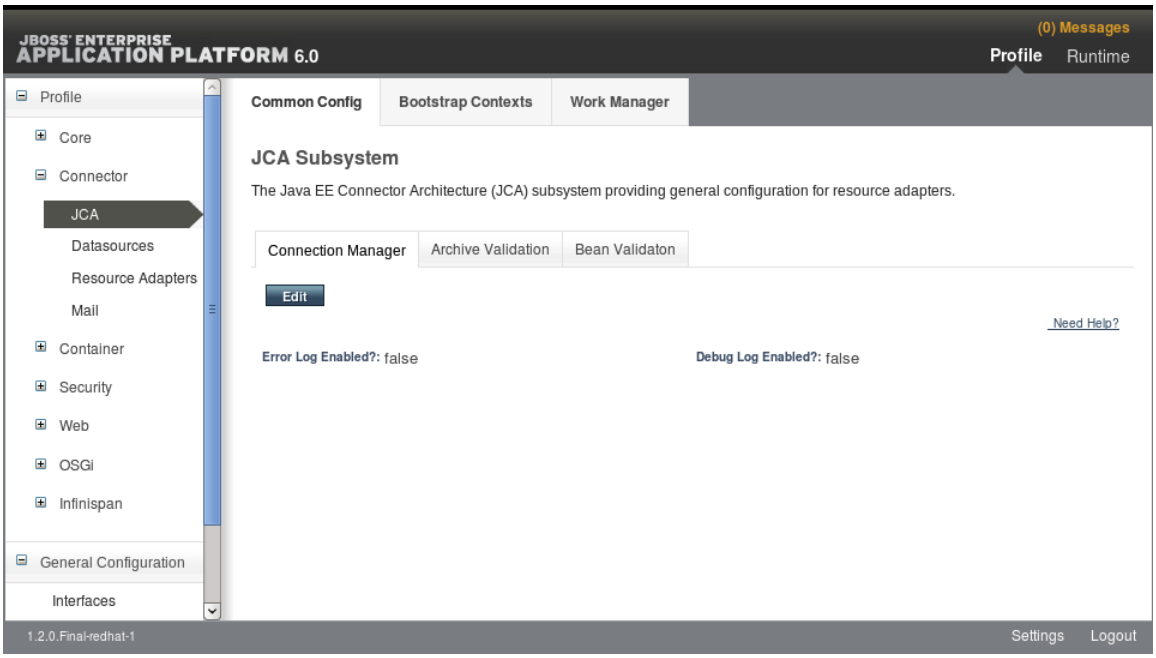



**Figure 20.1. Configuration commune JCA**

   b. **Work Managers**

L'onglet **`Work Manager`** contient la liste des Work Managers (gestionnaires de travail) configurés. Les nouveaux Work Managers peuvent être ajoutés, supprimés, et leurs pools de threads configurés ici. Chaque Work Manager peut avoir un pool de threads d'exécution courte et un pool de threads de longue durée en option.

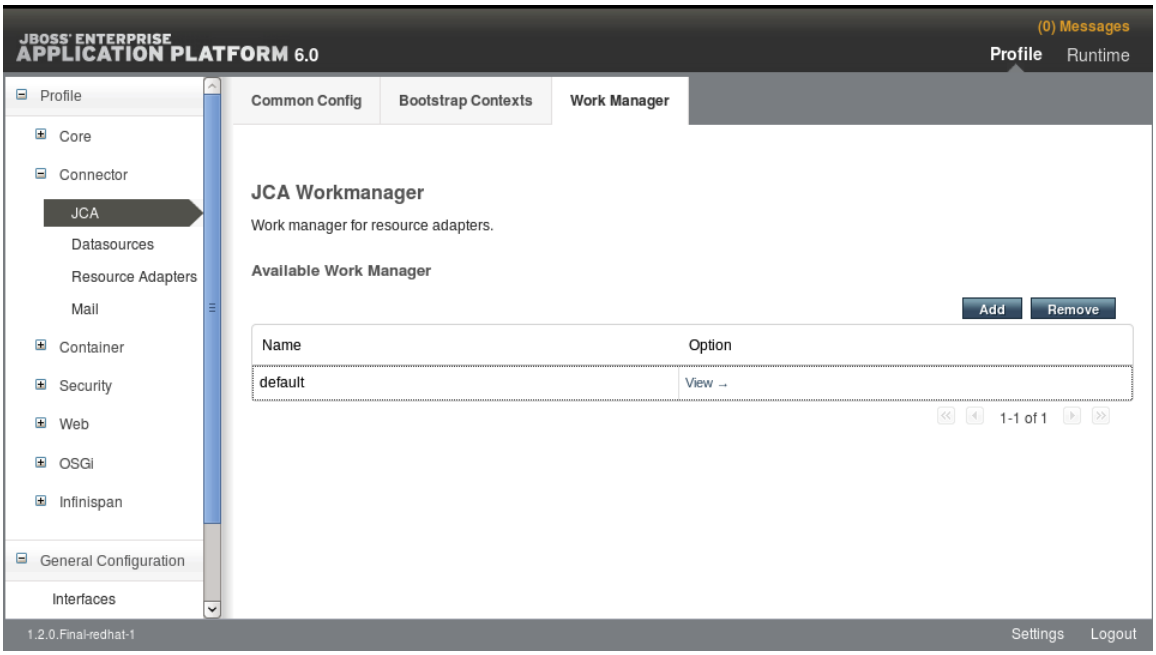



**Figure 20.2. Work Managers**

Les attributs de pools de threads peuvent être configurés ici :

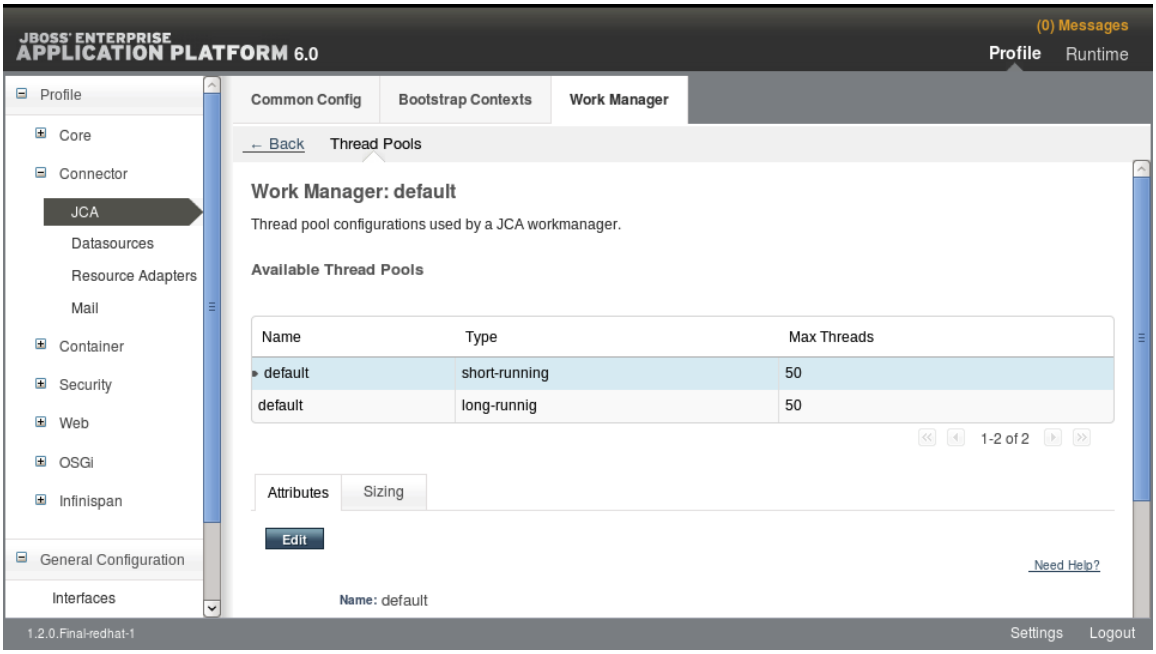



**Figure 20.3. Work Manager Thread Pools**

c. **Bootstrap Contexts**
L'onglet **`Bootstrap Contexts`** contient la liste des contextes d'amorçage configurés. De nouveaux objets de contexte d'amorçage peuvent être ajoutés, supprimés ou configurés. Un Work Manager doit être assigné à chaque contexte de Bootstrap.

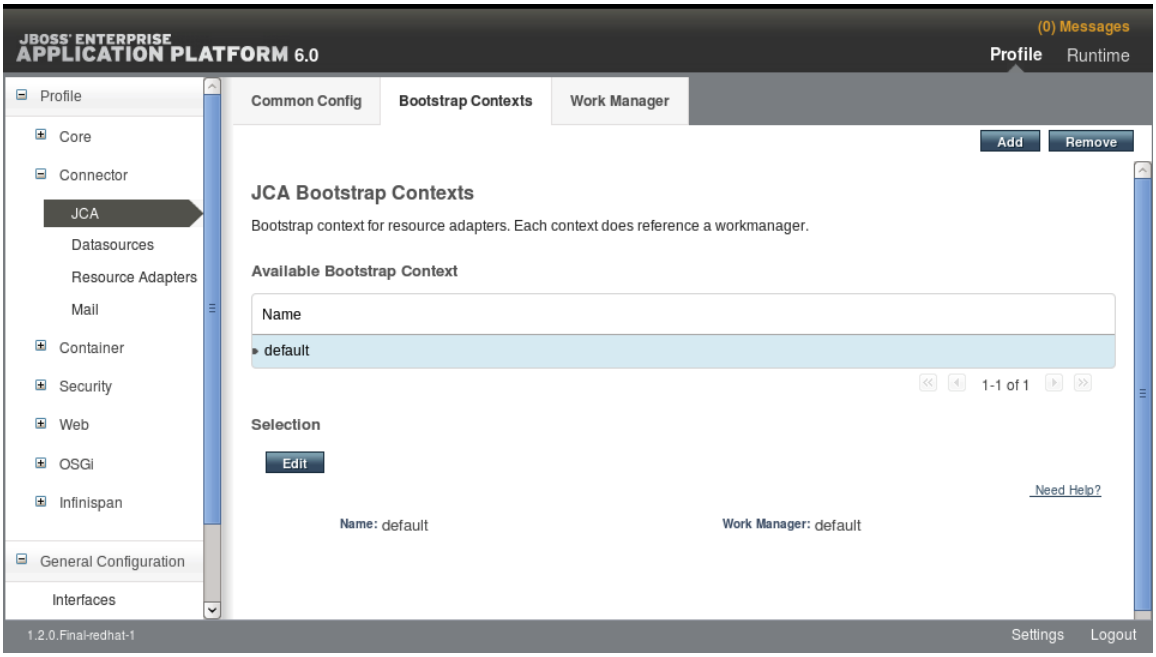



**Figure 20.4. Bootstrap Contexts**

Report a bug

## 20.3. DÉPLOYER UN ADAPTATEUR DE RESSOURCES

Les adaptateurs de ressources peuvent être déployés dans JBoss Enterprise Application Platform 6 à l'aide du Management CLI, de la Console de management basée-web, ou en copiant manuellement les fichiers. Le processus est le même que celui d'autres artefacts déployables.

**Procédure 20.2. Déployer un adaptateur de ressources par la Management CLI**

1. Ouvrir une invite de commande pour votre système d'exploitation.
2. Se connecter au Management CLI.
   - Dans Linux, saisir ce qui suit au niveau de la ligne de commande :

     ```
     $ EAP_HOME/bin/jboss-cli.sh --connect
     $ Connected to standalone controller at localhost:9999
     ```

   - Dans Windows, saisir ce qui suit au niveau de la ligne de commande :

     ```
     C:\>EAP_HOME\bin\jboss-cli.bat --connect
     C:\> Connected to standalone controller at localhost:9999
     ```

3. Déployer l'adaptateur de ressources.
   - Pour déployer l'adaptateur de ressources dans un serveur autonome, saisir ce qui suit dans une ligne de commande :

     ```
     $ deploy path/to/resource-adapter-name.rar
     ```

   - Pour déployer l'adaptateur de ressources dans tous les serveurs d'un domaine géré, saisir ce qui suit dans une ligne de commande :

```
$ deploy path/to/resource-adapter-name.rar --all-server-groups
```

**Procédure 20.3. Déployer un adaptateur de ressources par la Console de gestion basée-web**

1. Démarrer votre serveur JBoss Enterprise Application Platform 6.
2. Si vous n'avez pas encore ajouté d'utilisateur, ajoutez-en un maintenant. Pour plus d'informations, voir le chapitre Getting Started du Guide d'installation de JBoss Enterprise Application Platform 6.
3. Ouvrir un navigateur web et naviguer dans la Console de management. L'emplacement par défaut est http://localhost:9990/console/. Pour plus d'informations sur la Console de gestion, voir Section 3.4.2, « Connectez-vous à la Console de management ».
4. Cliquer sur le lien **`Runtime`** qui se trouve en haut et à droite pour passer à la vue de Runtime, puis choisir **`Manage Deployments`** dans le panneau de navigation de gauche, et cliquer sur **`Add Content`** (Ajouter Contenu) en haut et à droite.

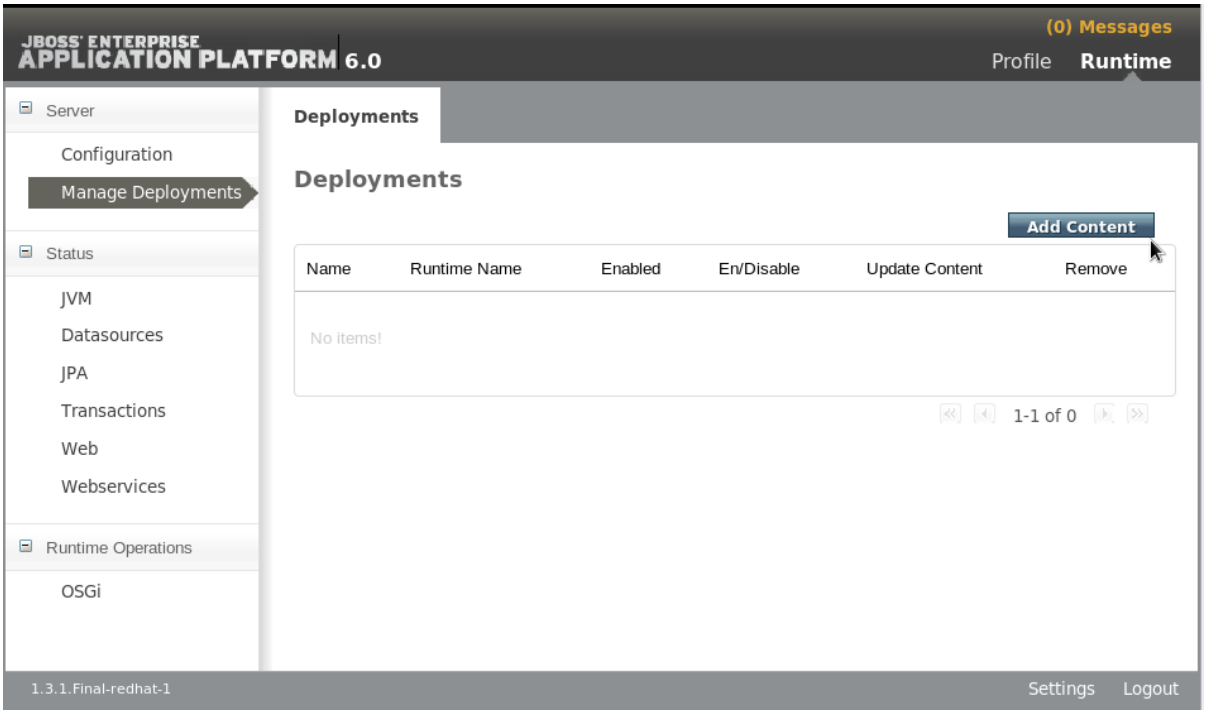



**Figure 20.5. Gérer les déploiements - Ajouter Contenu**

5. Naviguer dans l'archive d'adaptateur de ressources et le sélectionner. Puis, cliquer sur le bouton **`Next`**.

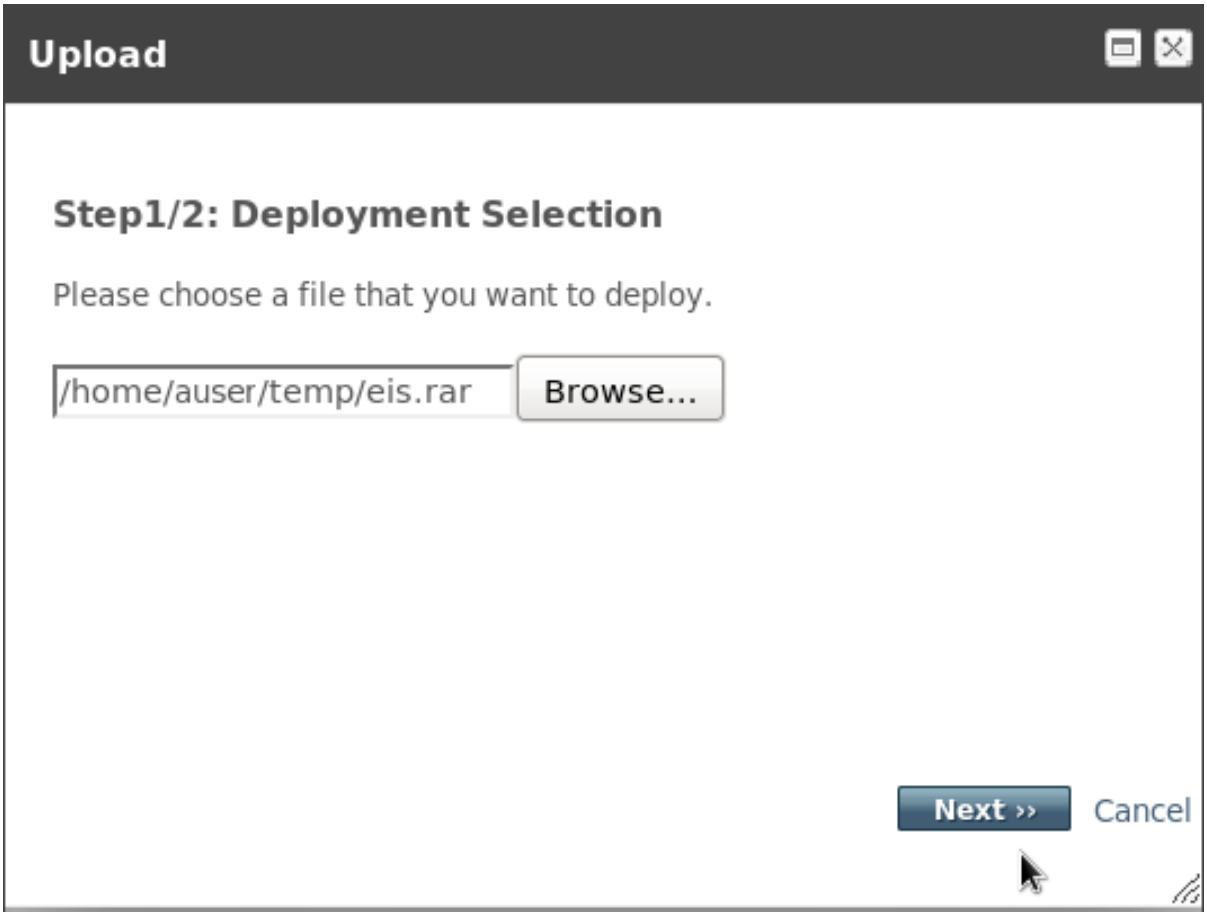



**Figure 20.6. Sélection de déploiement**

6. Vérifier les noms de déploiement, puis cliquer sur le bouton **Save**.

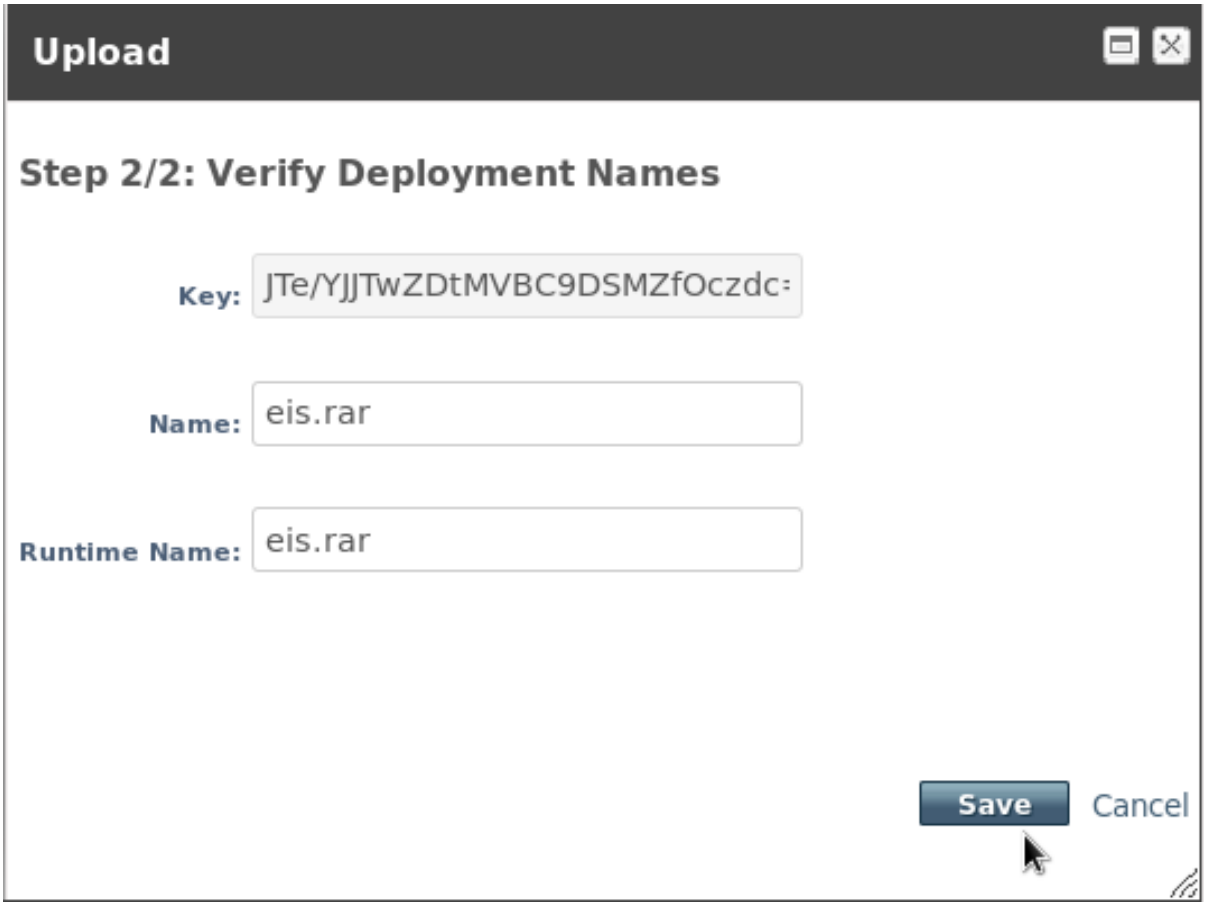



**Figure 20.7. Vérifier les noms de déploiement**

7. L'archive d'adaptateur de ressources devrait maintenant s'afficher dans la liste dans un état désactivé. Cliquer sur le lien **`Enable`** pour l'activer.

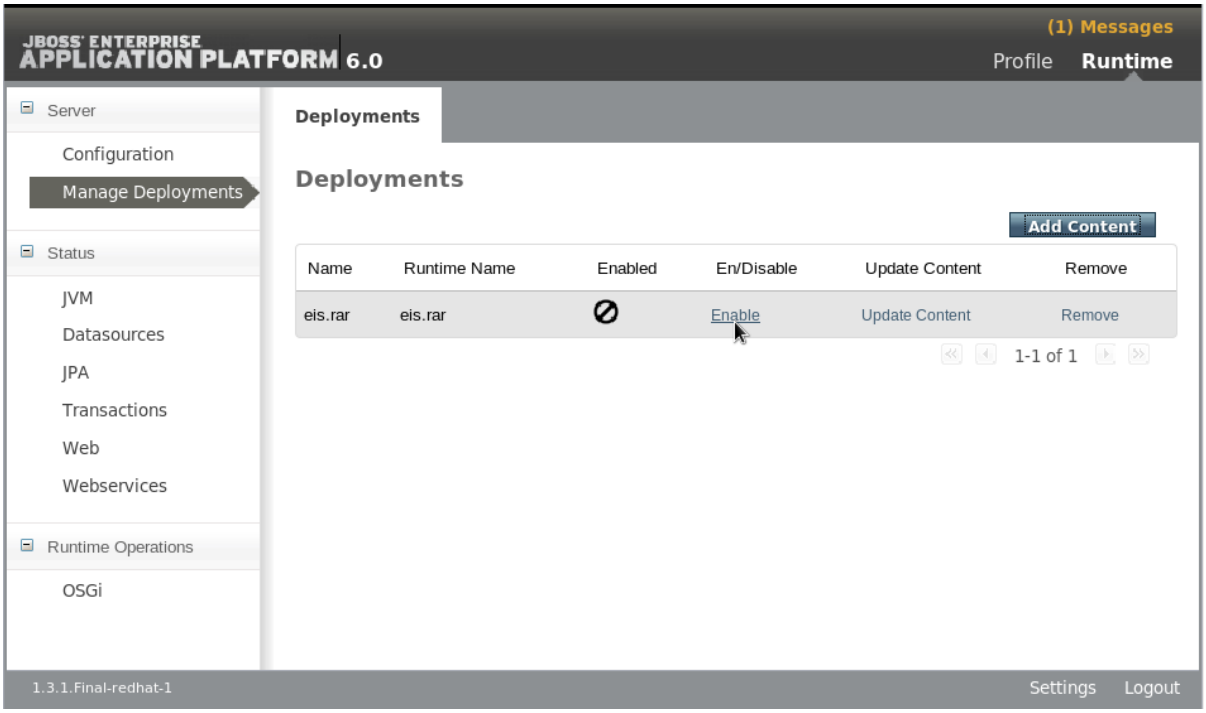



**Figure 20.8. Activer le déploiement**

8. Un dialogue vous demande "êtes-vous certain?" que vous souhaitez activer le RAR listé. Cliquer sur **`Confirm`**. L'archive d'adaptateur de déploiement devrait maintenant afficher **`Enabled`**.

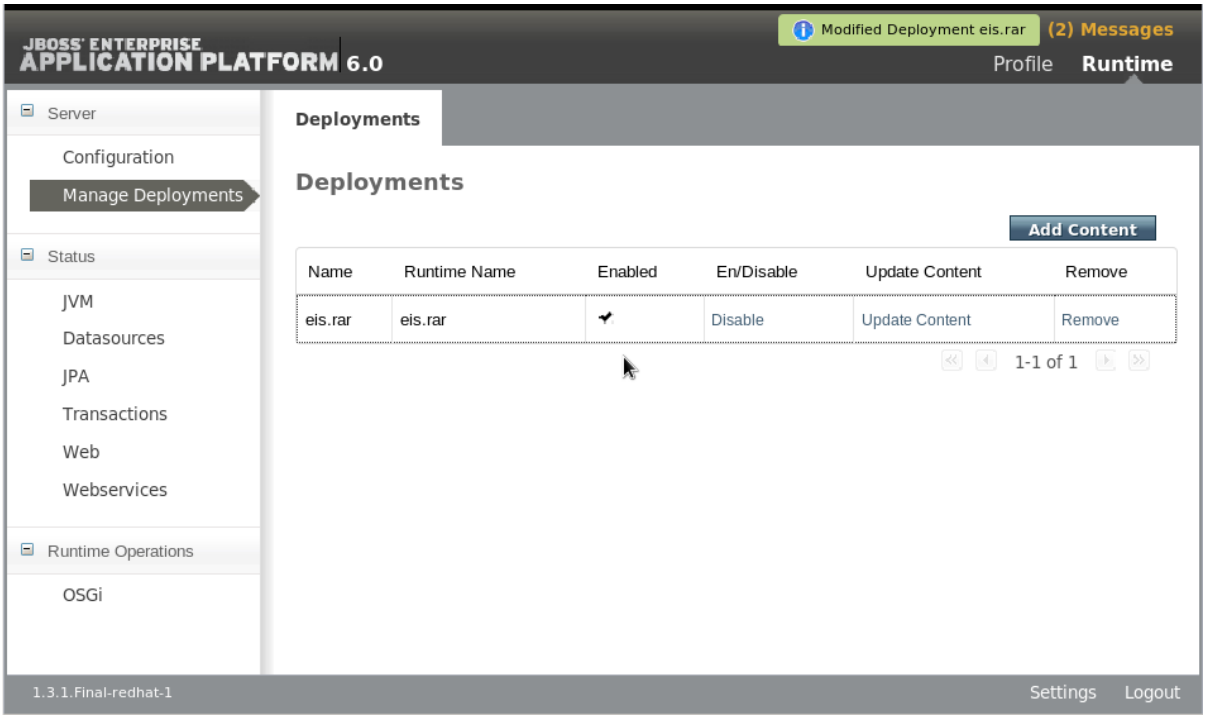



**Figure 20.9. Déploiements**

**Procédure 20.4. Déployer un adaptateur de ressources manuellement**

- Copier l'archive d'adaptateur de ressources dans le répertoire des déploiements de serveur,
  - Pour un serveur autonome, copier l'archive d'adptateur de ressources dans le répertoire ***EAP_HOME*/standalone/deployments/**.
  - Pour un domaine géré, copier l'archive d'adptateur de ressources dans le répertoire ***EAP_HOME*/domain/deployments/** du contrôleur de domaine.

Report a bug

## 20.4. CONFIGURATION D'UN ADAPTATEUR DE RESSOURCES DÉPLOYÉES

Les administrateurs JBoss peuvent configurer les adaptateurs de ressources pour JBoss Enterprise Application Platform 6 à l'aide du Management CLI, de la Console de management basée-web, ou en modifiant manuellement la configuration des fichiers.

Voir le document du fournisseur pour votre adaptateur de ressources pour obtenir des informations sur les propriétés prises en charge et autres informations.

**NOTE**

Dans la procédure suivante, la ligne de commande que vous devez saisir suit l'invite suivante **[standalone@localhost:9999 /]**. Ne PAS saisir le texte qui se trouve à l'intérieur des accolades. Voici la sortie que vous devriez apercevoir comme résultat, ainsi, **{"outcome" => "success"}**.

**Procédure 20.5. Configurer un adaptateur de ressources par le Management CLI**

1. Ouvrir une invite de commande de votre système d'exploitation.
2. Connectez-vous au Management CLI.
   - Dans Linux, saisir ce qui suit au niveau de la ligne de commande :

     ```
     $ EAP_HOME/bin/jboss-cli.sh --connect
     ```

     Vous devriez voir le résultat de sortie suivant :

     ```
     $ Connected to standalone controller at localhost:9999
     ```

   - Dans Windows, saisir ce qui suit au niveau de la ligne de commande :

     ```
     C:\>EAP_HOME\bin\jboss-cli.bat --connect
     ```

     Vous devriez voir le résultat de sortie suivant :

     ```
     C:\> Connected to standalone controller at localhost:9999
     ```

3. Ajouter la configuration d'adaptateur de ressource.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar:add(archive=eis.rar, transaction-
support=XATransaction)
{"outcome" => "success"}
```

4. Configurer la <config-property> **`server`** niveau adaptateur de ressources.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar/config-properties=server/:add(value=localhost)
{"outcome" => "success"}
```

5. Configurer la <config-property>**`port`** niveau adaptateur de ressources

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar/config-properties=port/:add(value=9000)
{"outcome" => "success"}
```

6. Ajouter une définition de connexion à la fabrique de connexions gérées.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar/connection-definitions=cfName:add(class-
name=com.acme.eis.ra.EISManagedConnectionFactory, jndi-
name=java:/eis/AcmeConnectionFactory)
{"outcome" => "success"}
```

7. Configurer <config-property> **`port`** niveau usine de connexions gérées.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar/connection-definitions=cfName/config-
properties=name/:add(value=Acme Inc)
```

```
{"outcome" => "success"}
```

8. Ajouter un objet admin.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar/admin-objects=aoName:add(class-
name=com.acme.eis.ra.EISAdminObjectImpl, jndi-
name=java:/eis/AcmeAdminObject)
{"outcome" => "success"}
```

9. Configurer la propriété **`threshold`** de l'objet admin.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar/admin-objects=aoName/config-
properties=threshold/:add(value=10)
{"outcome" => "success"}
```

10. Activer l'adaptateur de ressource.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar:activate
{"outcome" => "success"}
```

11. Voir l'adaptateur de ressources nouvellement configuré et activé.

```
[standalone@localhost:9999 /] /subsystem=resource-adapters/resource-
adapter=eis.rar:read-resource(recursive=true)
{
    "outcome" => "success",
    "result" => {
        "archive" => "eis.rar",
        "beanvalidationgroups" => undefined,
        "bootstrap-context" => undefined,
        "transaction-support" => "XATransaction",
        "admin-objects" => {"aoName" => {
            "class-name" => "com.acme.eis.ra.EISAdminObjectImpl",
            "enabled" => true,
            "jndi-name" => "java:/eis/AcmeAdminObject",
            "use-java-context" => true,
            "config-properties" => {"threshold" => {"value" => 10}}
        }},
        "config-properties" => {
            "server" => {"value" => "localhost"},
            "port" => {"value" => 9000}
        },
        "connection-definitions" => {"cfName" => {
            "allocation-retry" => undefined,
            "allocation-retry-wait-millis" => undefined,
            "background-validation" => false,
            "background-validation-millis" => undefined,
            "blocking-timeout-wait-millis" => undefined,
            "class-name" =>
"com.acme.eis.ra.EISManagedConnectionFactory",
            "enabled" => true,
```

```
            "flush-strategy" => "FailingConnectionOnly",
            "idle-timeout-minutes" => undefined,
            "interleaving" => false,
            "jndi-name" => "java:/eis/AcmeConnectionFactory",
            "max-pool-size" => 20,
            "min-pool-size" => 0,
            "no-recovery" => undefined,
            "no-tx-separate-pool" => false,
            "pad-xid" => false,
            "pool-prefill" => false,
            "pool-use-strict-min" => false,
            "recovery-password" => undefined,
            "recovery-plugin-class-name" => undefined,
            "recovery-plugin-properties" => undefined,
            "recovery-security-domain" => undefined,
            "recovery-username" => undefined,
            "same-rm-override" => undefined,
            "security-application" => undefined,
            "security-domain" => undefined,
            "security-domain-and-application" => undefined,
            "use-ccm" => true,
            "use-fast-fail" => false,
            "use-java-context" => true,
            "use-try-lock" => undefined,
            "wrap-xa-resource" => true,
            "xa-resource-timeout" => undefined,
            "config-properties" => {"name" => {"value" => "Acme
Inc"}}
        }}
    }
}
```

**Procédure 20.6. Configurer un adaptateur de ressources par la Console de management basée-web**

1. Démarrer votre serveur de JBoss Enterprise Application Platform 6.

2. Si vous n'avez pas encore ajouté d'utilisateur, ajoutez-en un maintenant. Pour plus d'informations, voir le chapitre Getting Started du Guide d'installation de JBoss Enterprise Application Platform 6.

3. Ouvrir un navigateur web et naviguez dans la Console de management. L'emplacement par défaut est http://localhost:9990/console/. Pour plus d'informations sur la Console de management, voir Section 3.4.2, « Connectez-vous à la Console de management ».

4. Cliquer sur le lien **Runtime** qui se trouve en haut et à droite pour passer à la vue Profile, puis choisir **Resource Adapters** dans le panneau de navigation de gauche, et cliquer sur **Add** (Ajouter) en haut et à droite.

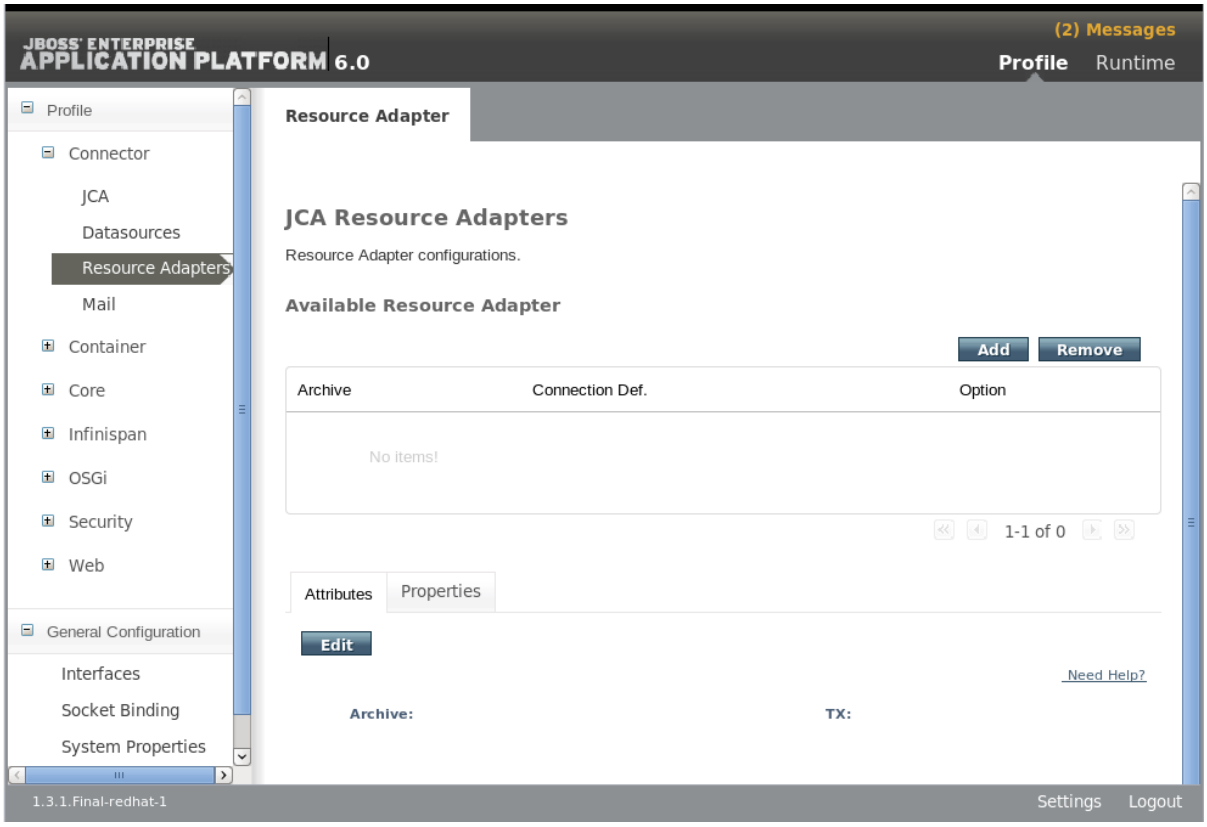



**Figure 20.10. Adaptateurs de ressources JCA**

5. Saisir le nom de l'archive et choisir le type de transaction **`XATransaction`** à partir du menu déroulant **`TX:`**. Ensuite, cliquer sur **Save**.

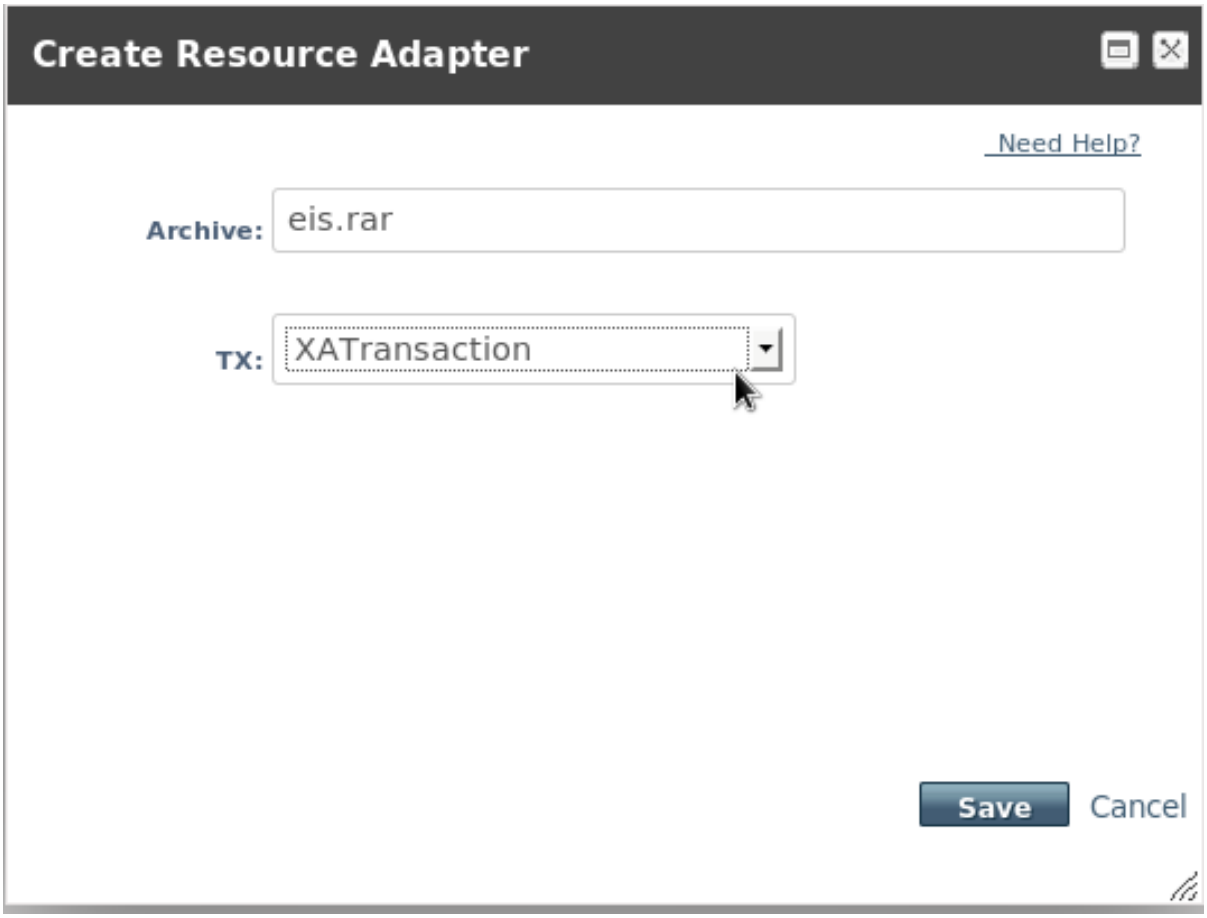



**Figure 20.11. Créer un adaptateur de ressources**

6. Sélectionner l'onglet **Properties**, puis cliquer sur **Add** pour ajouter des propriétés d'adaptateur de ressources.

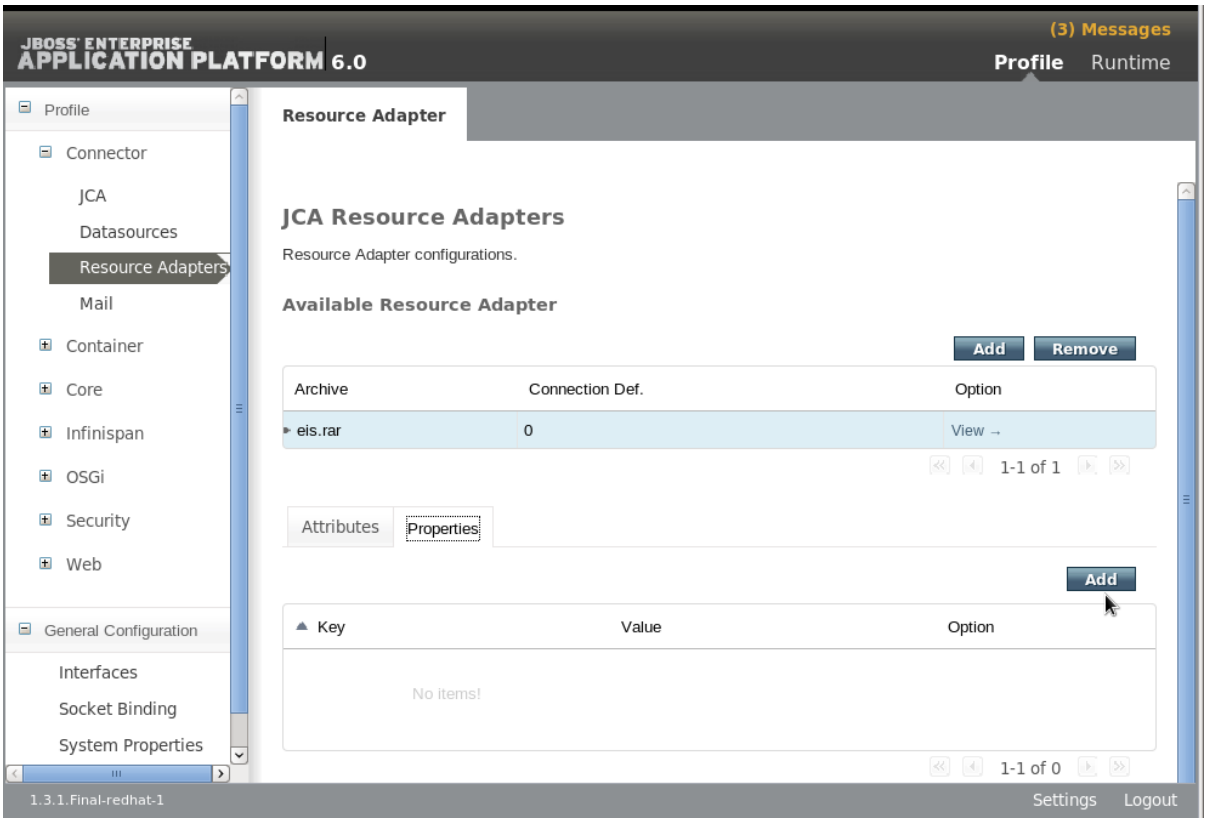



**Figure 20.12. Ajouter des propriétés d'adaptateur de ressources**

7. Saisir le **serveur** pour le **Name** (nom) et le nom d'hôte, par exemple **localhost**, pour la valeur **Value**. Puis cliquer sur **Save** pour sauvegarder la propriété.

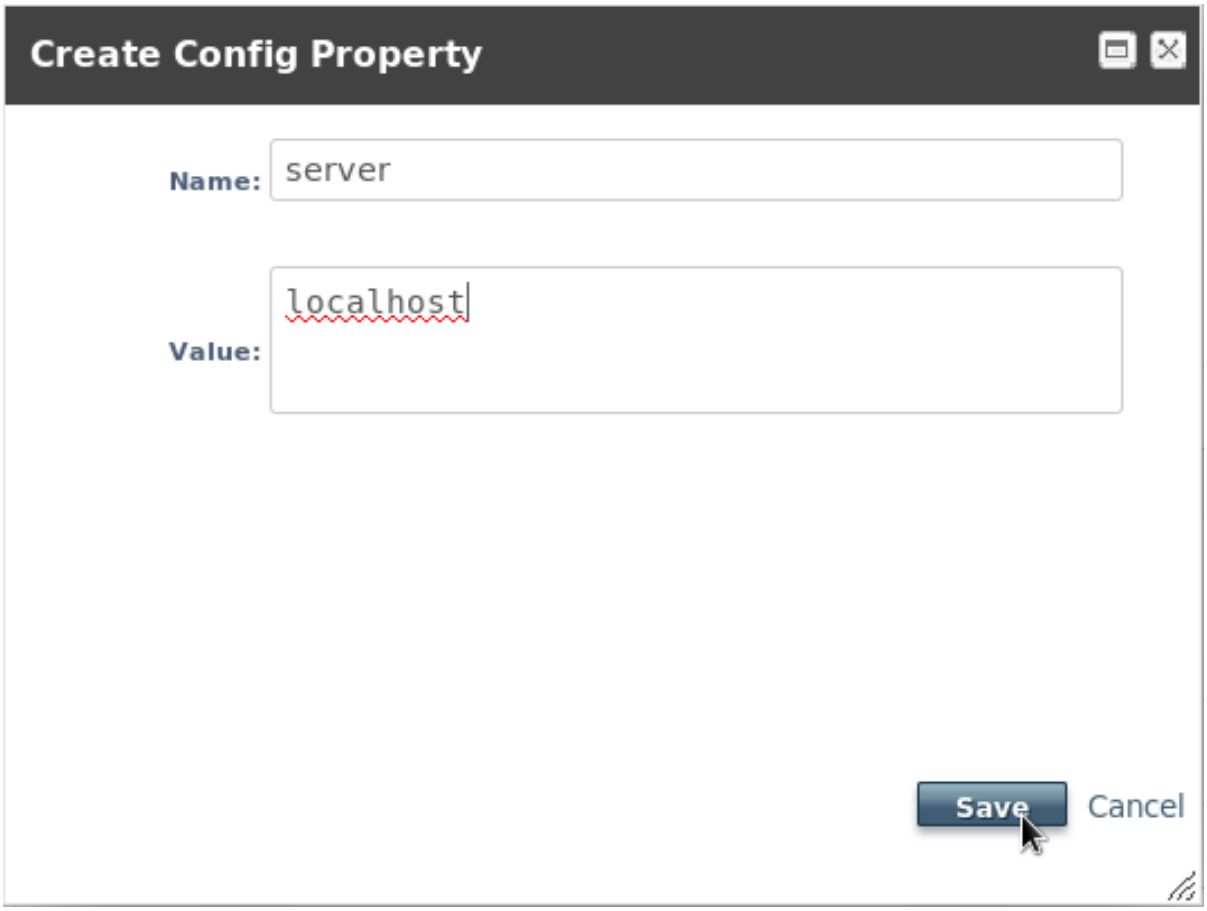



**Figure 20.13. Ajouter une propriété d'adaptateur de ressources**

8. Saisir le **`port`** pour le **`Name`** (nom) et le nom d'hôte, par exemple **`9000`**, pour la valeur **`Value`**. Puis cliquer sur **`Save`** pour sauvegarder la propriété.

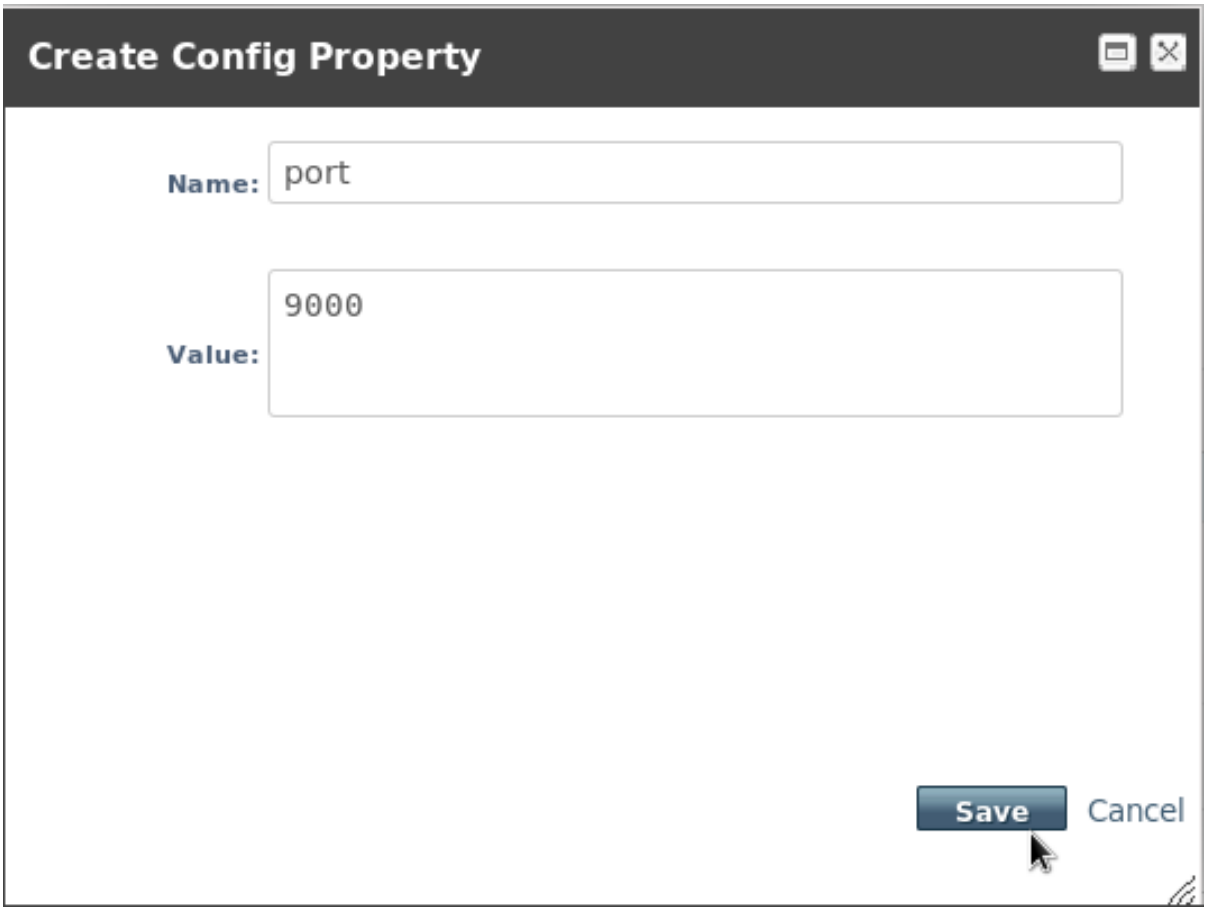



**Figure 20.14. Ajouter la propriété de port d'adaptateur de ressource.**

9. Les propriétés **`server`** et **`port`** apparaissent maintenant dans le panneau **`Properties`**. Cliquer sur le lien **`View`** (Vue) sous la colonne **`Option`** pour l'adaptateur de ressources listées pour visualiser les définitions de connexion or **`Connection Definitions`**.

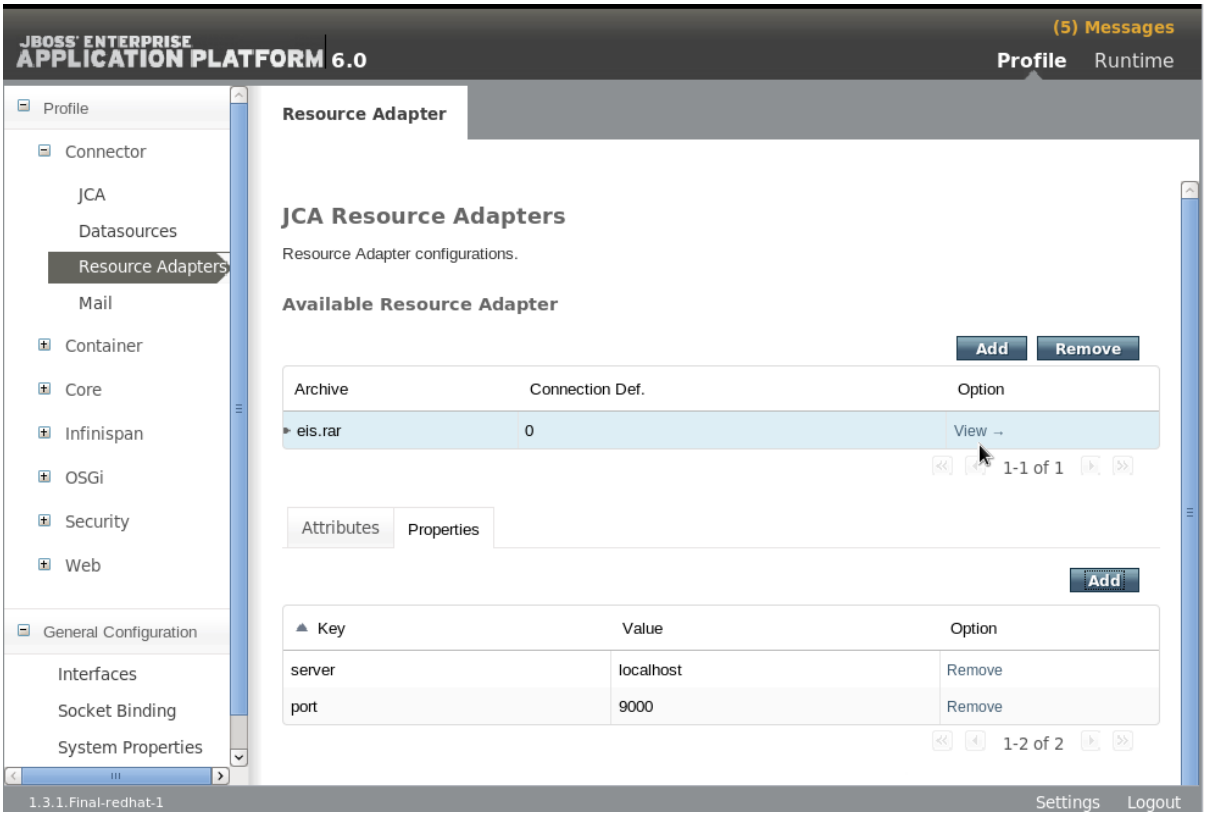



**Figure 20.15. Propriétés de serveur d'adaptateur de ressources terminé**

10. Cliquer sur **Add** en haut et à droite de la page pour ajouter une définiton de connexion.

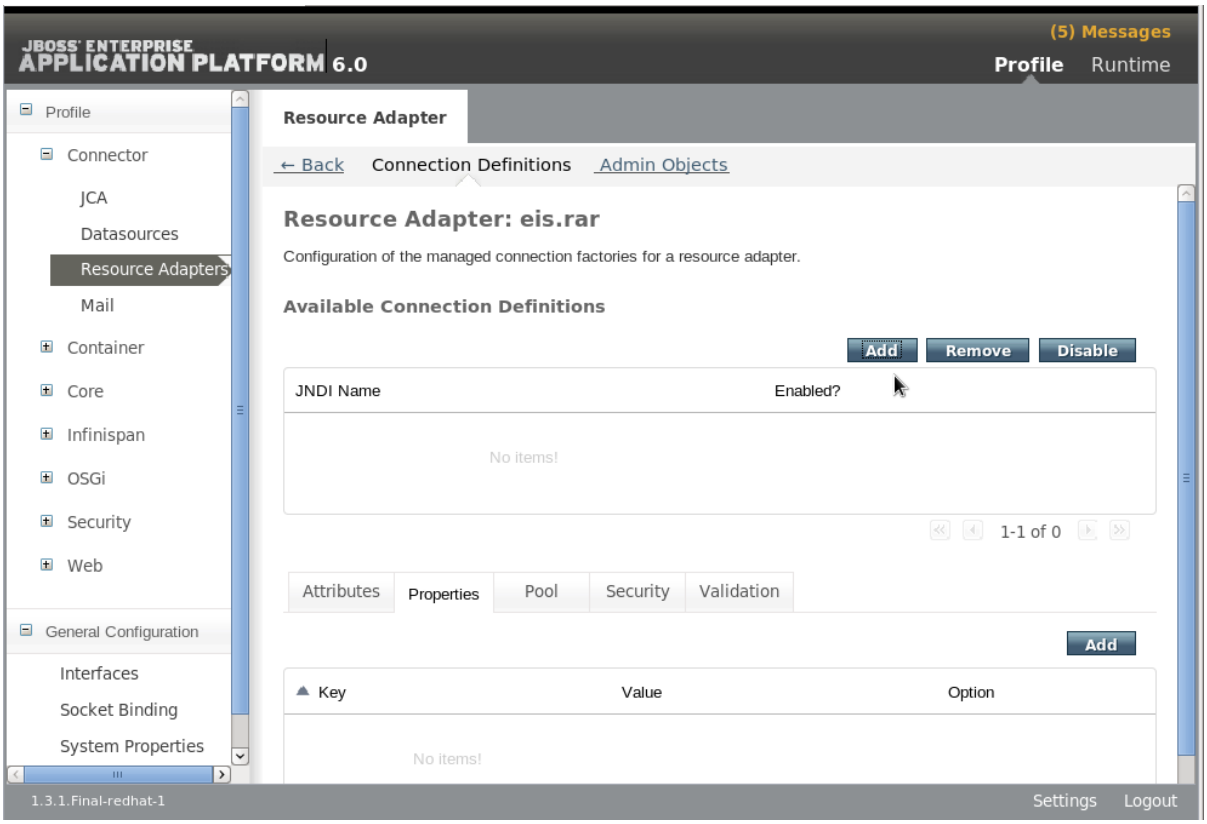



**Figure 20.16. Ajouter une définition de connexion**

11. Saisir le **`JNDI Name`** et le nom de classe complet de la **`Connection Class`**. Puis cliquer sur **`Next`**.

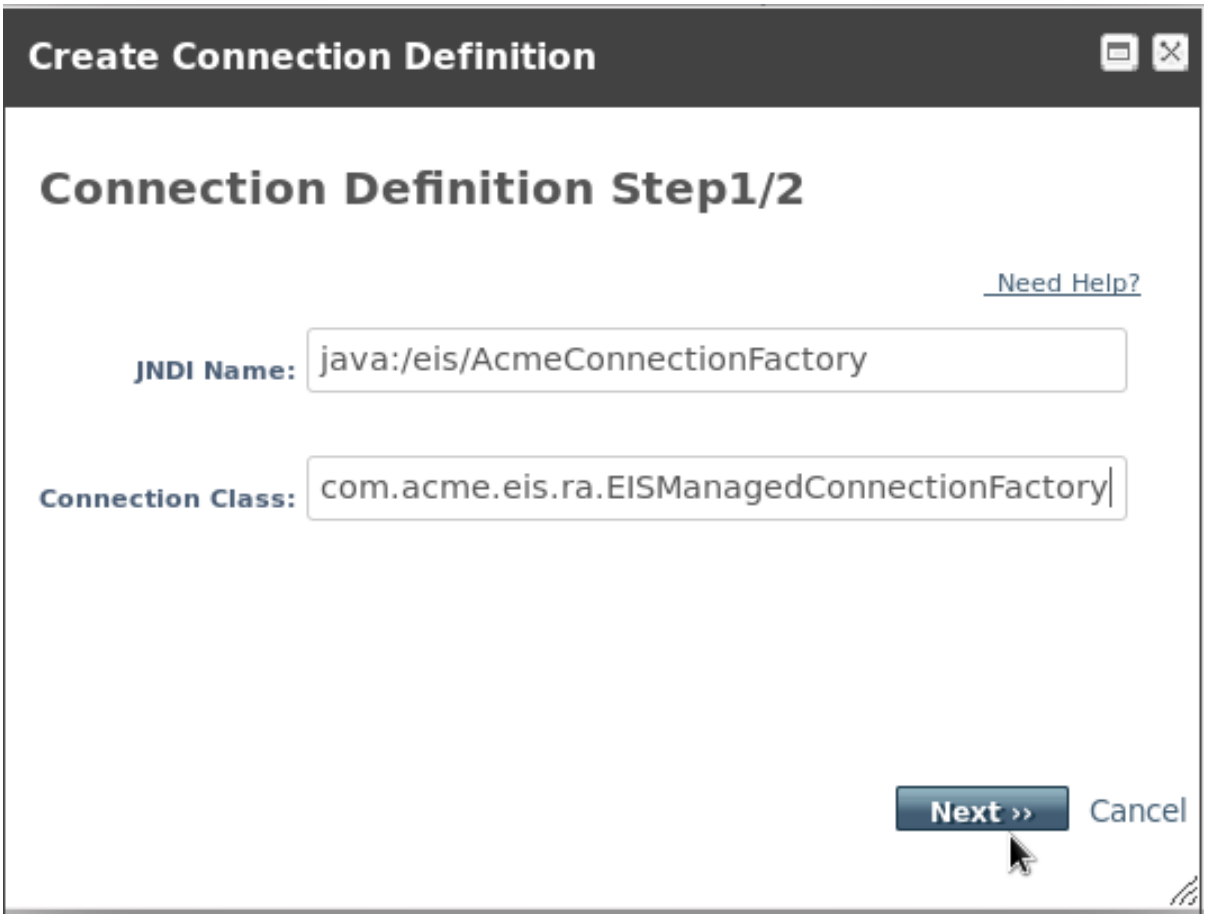



**Figure 20.17. Créer la propriété de définition de connexion - Étape 1**

12. Cliquer sur **Add** pour saisir la **`Key`** (clé) et la **`Value`** pour cette définition de connexion.

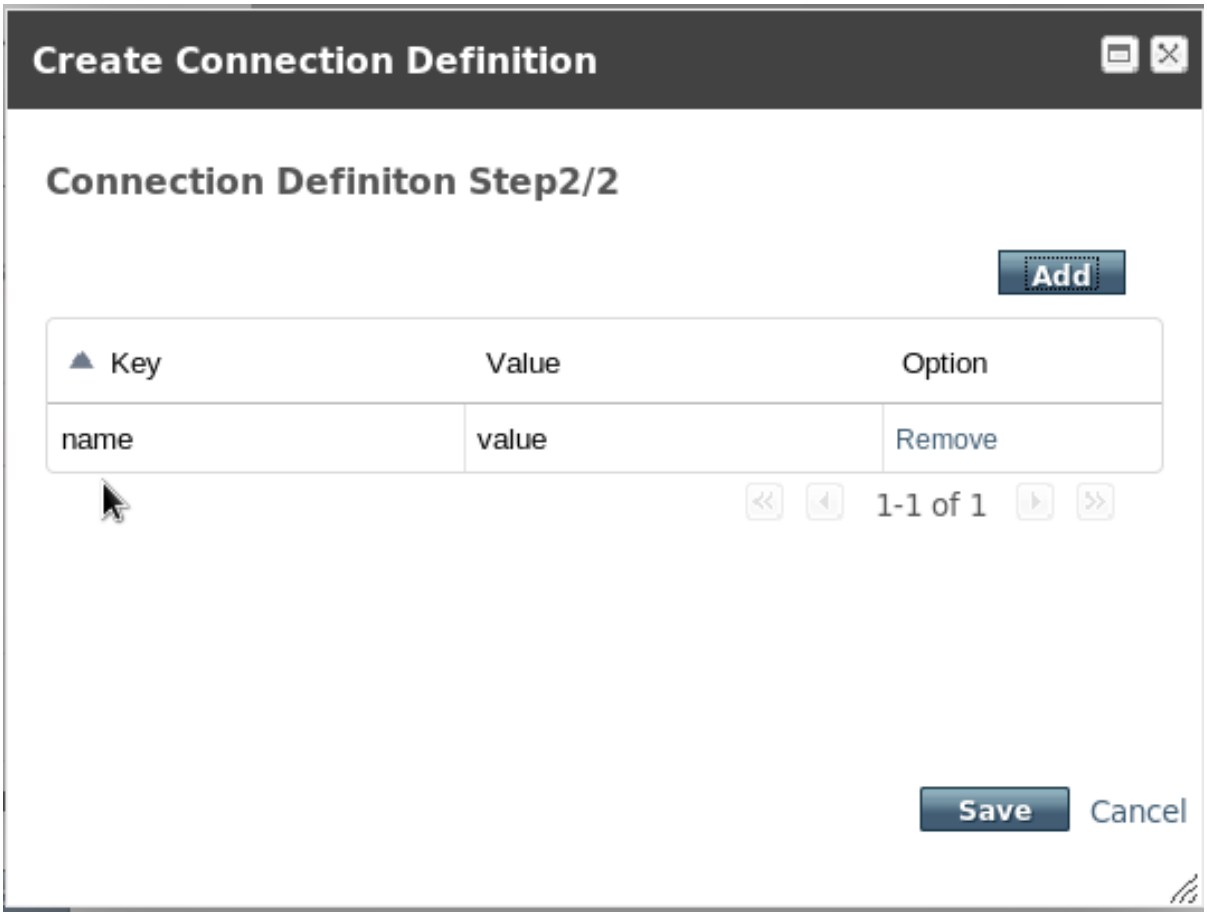



**Figure 20.18. Créer la propriété de définition de connexion - Étape 2**

13. Cliquer sur le champ **name** dans la colonne **Key** pour autoriser la saisie des données sur ce champ. Saisir le nom de propriété et appuyer sur Entrée pour ce champ. Cliquer sur le champ **value** dans la colonne **Value** pour activer la saisie sur ce champ. Saisir la valeur de la propriété et appuyer sur Entrée une fois que c'est fait. Puis, cliquer sur **Save** pour sauvegarder la propriété.

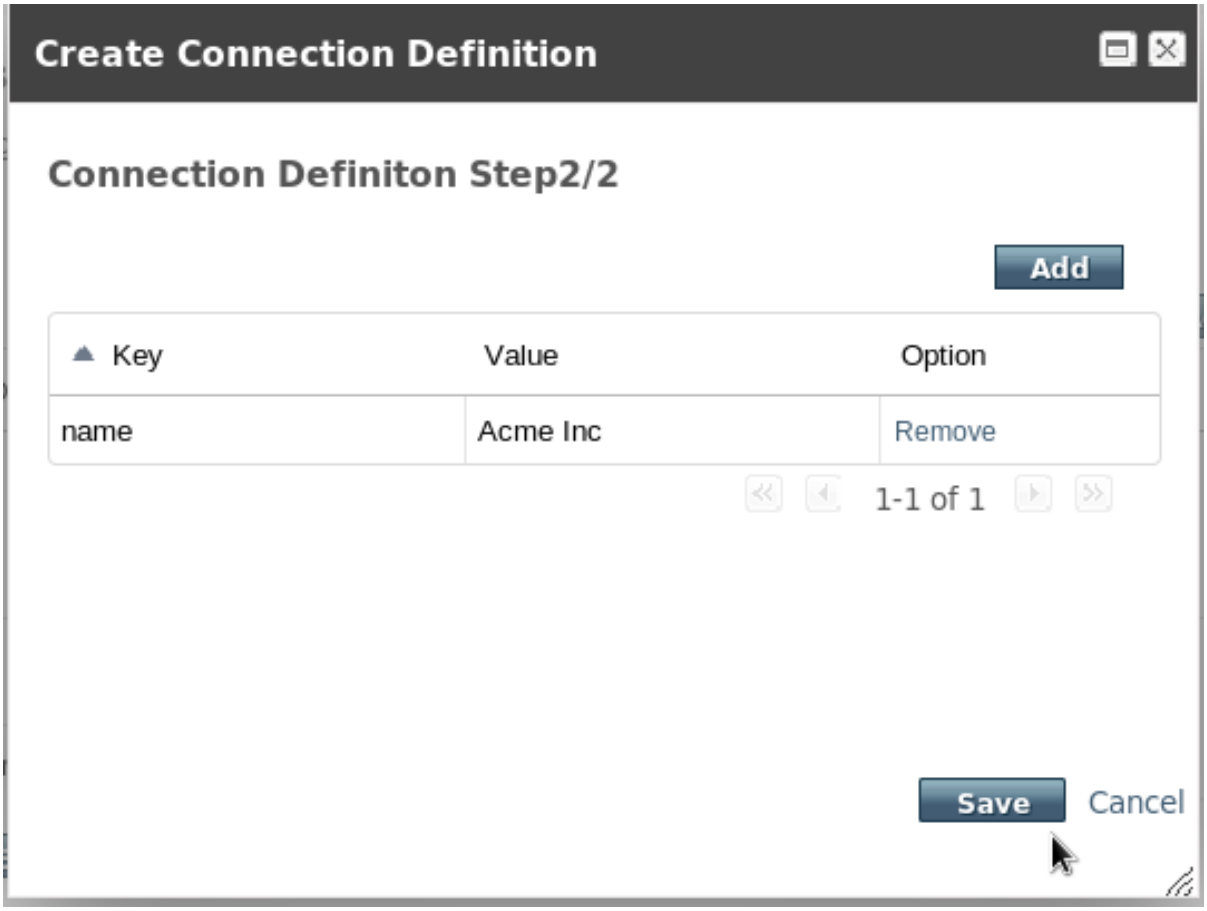



**Figure 20.19. Créer la propriété de définition de connexion - Étape 2**

14. La définition de connexion est terminée, mais non activée. Cliquer sur le bouton **`Enable`** pour activer la définition de connexion.

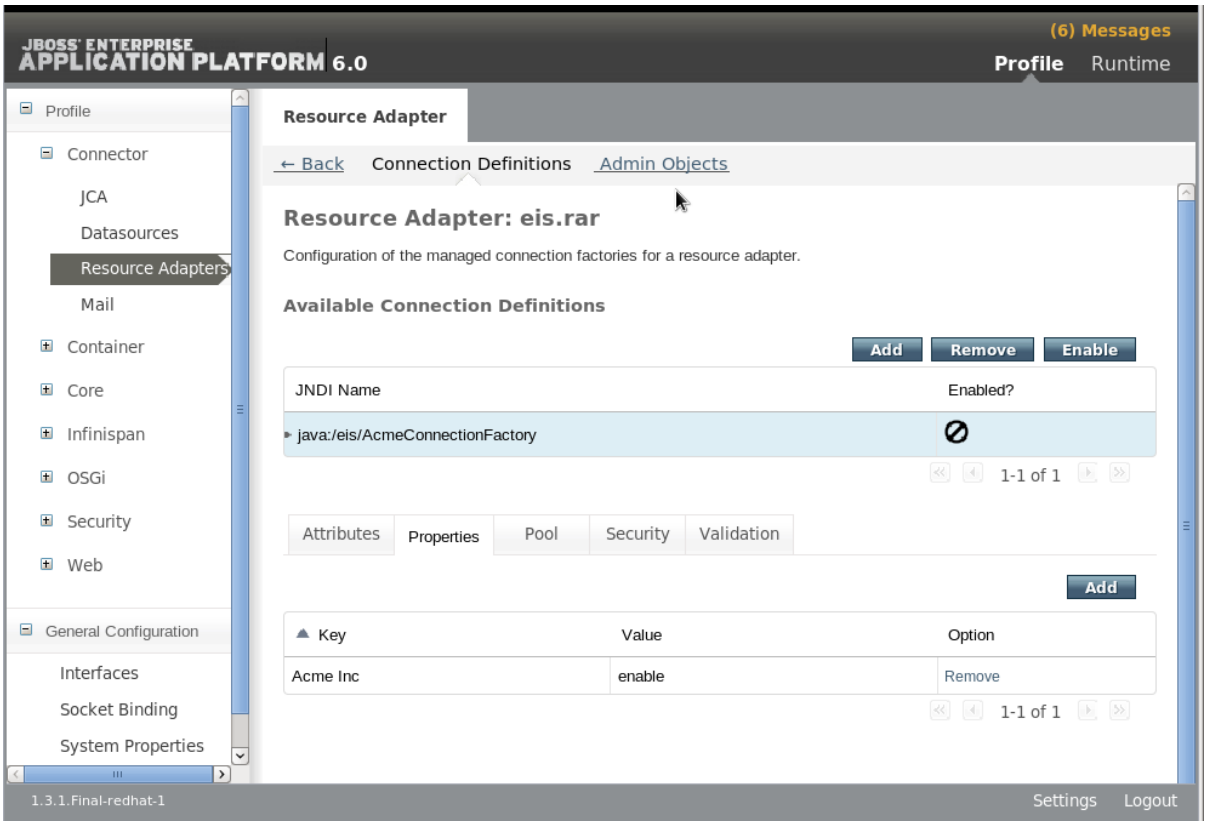



**Figure 20.20. Créer une définition de connexion - Non activée**

15. Un dialogue vous demande "Souhaitez-vous réellement modifier la définition de connexion?" du nom JNDI. Cliquer sur **Confirm**. La définition de connexion devrait maintenant afficher **Enabled** (activée).

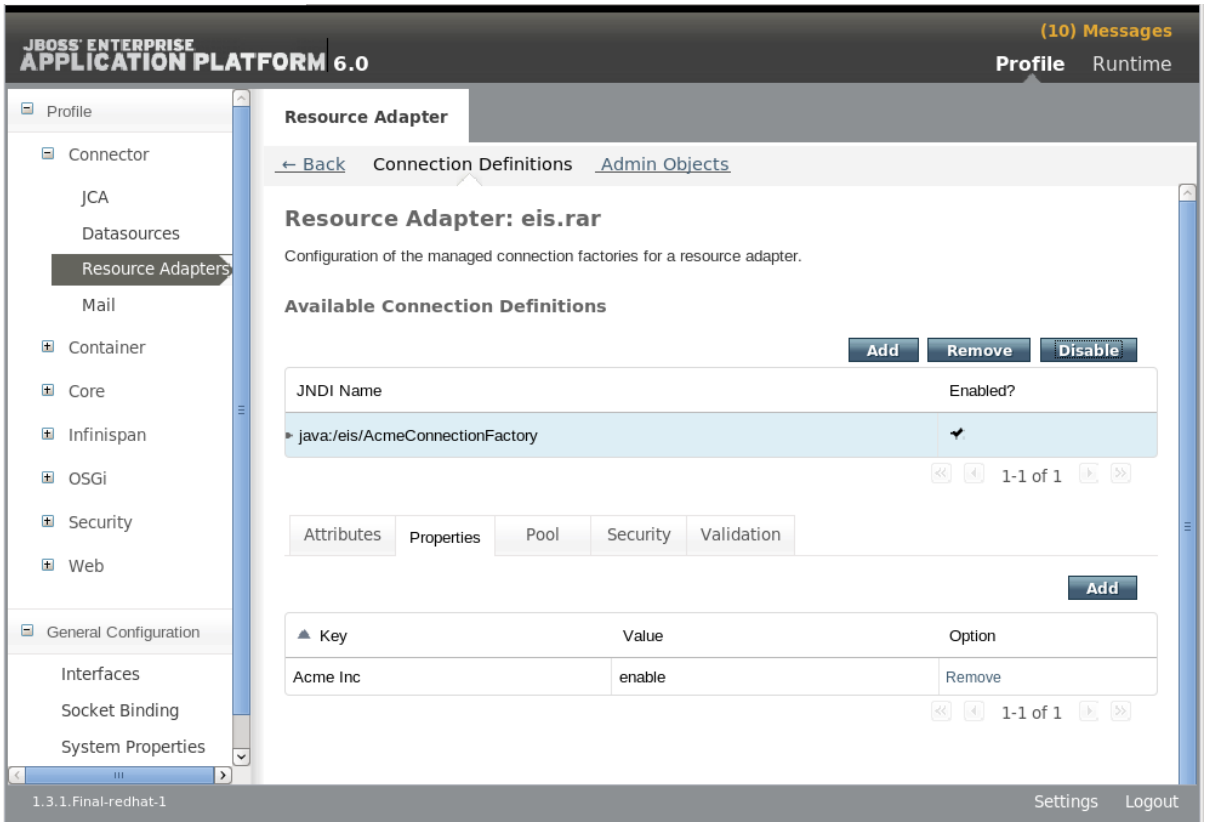



**Figure 20.21. Définition de connexion maintenant activée**

16. Cliquer sur l'onglet **Admin Objects** qui se trouve dans la partie supérieure de la page pour créer et configurer des objets admin. Puis, cliquer sur **Add**.

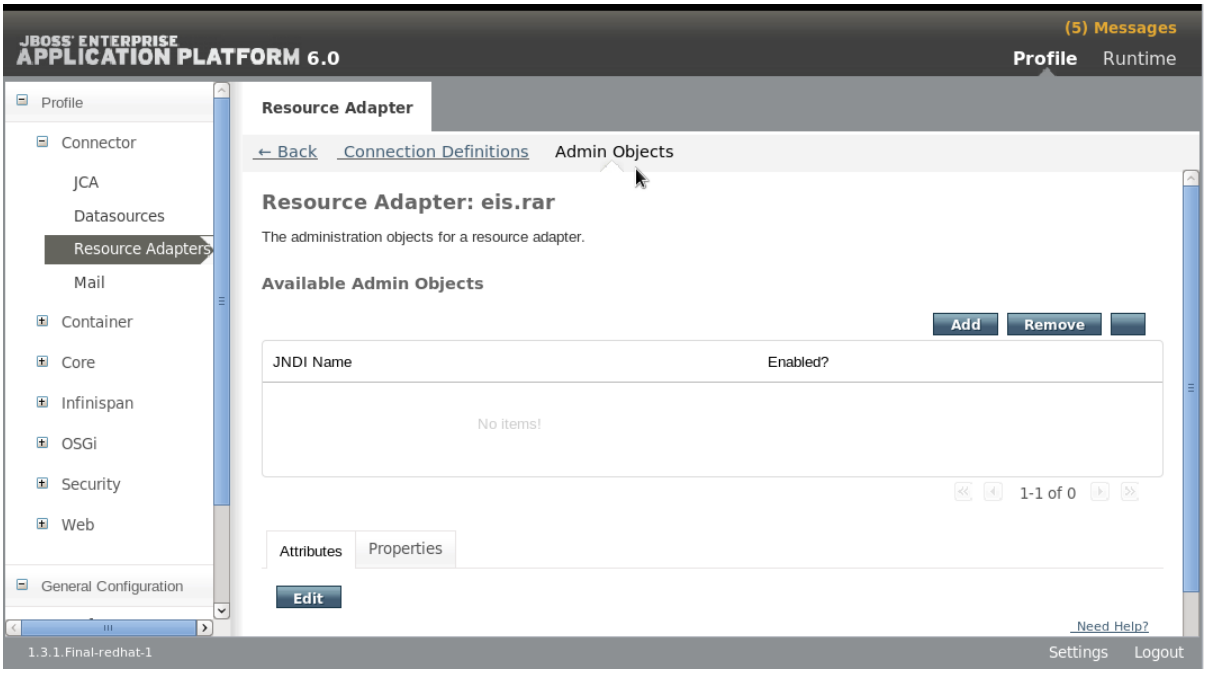



**Figure 20.22. Objets admin disponibles**

17. Saisir le **JNDI Name** et le nom de classe **Class Name** complet de l'objet admin. Puis cliquer sur **Save**.

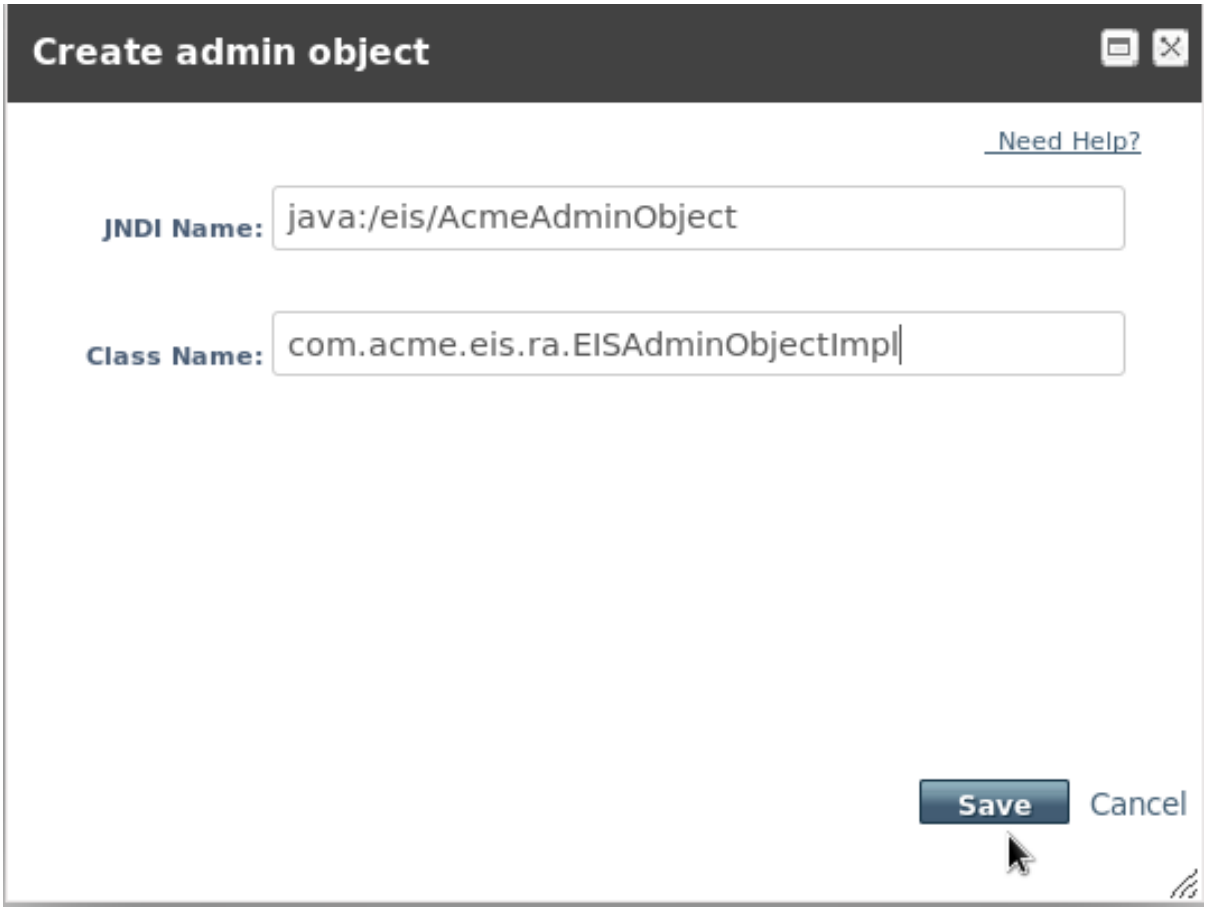



**Figure 20.23. Créer un objet admin**

18. Sélectionner l'onglet **`Properties`**, puis cliquer sur **Add** pour ajouter des propriétés d'objet admin.

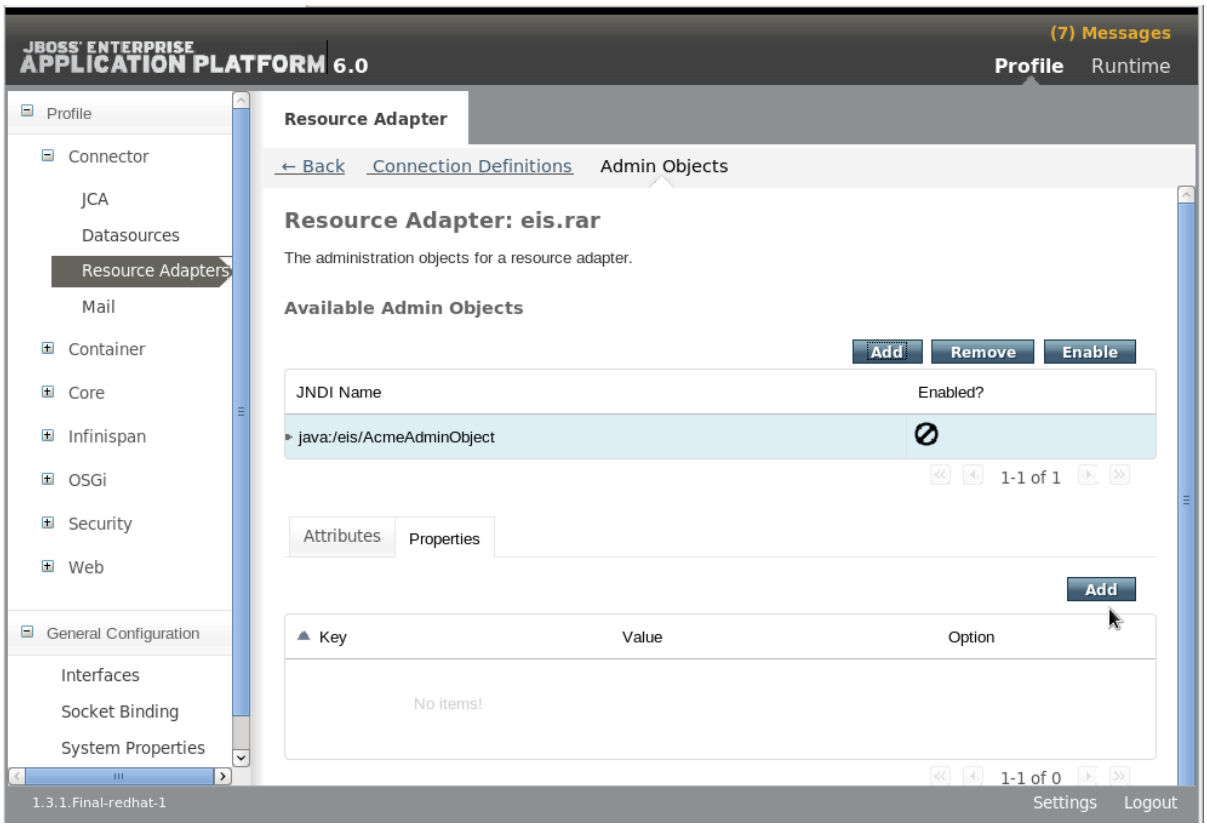



**Figure 20.24. Ajouter les propriétés d'objet admin**

19. Saisir une propriété de configuration d'objet admin, comme par exemple la limite **`threshold`**, dans le champ **`Name`** (nom). Saisir la valeur de la propriété de configuration, comme par exemple **`10`**, pour la valeur **`Value`**. Puis cliquer sur **`Save`** pour sauvegarder la propriété.

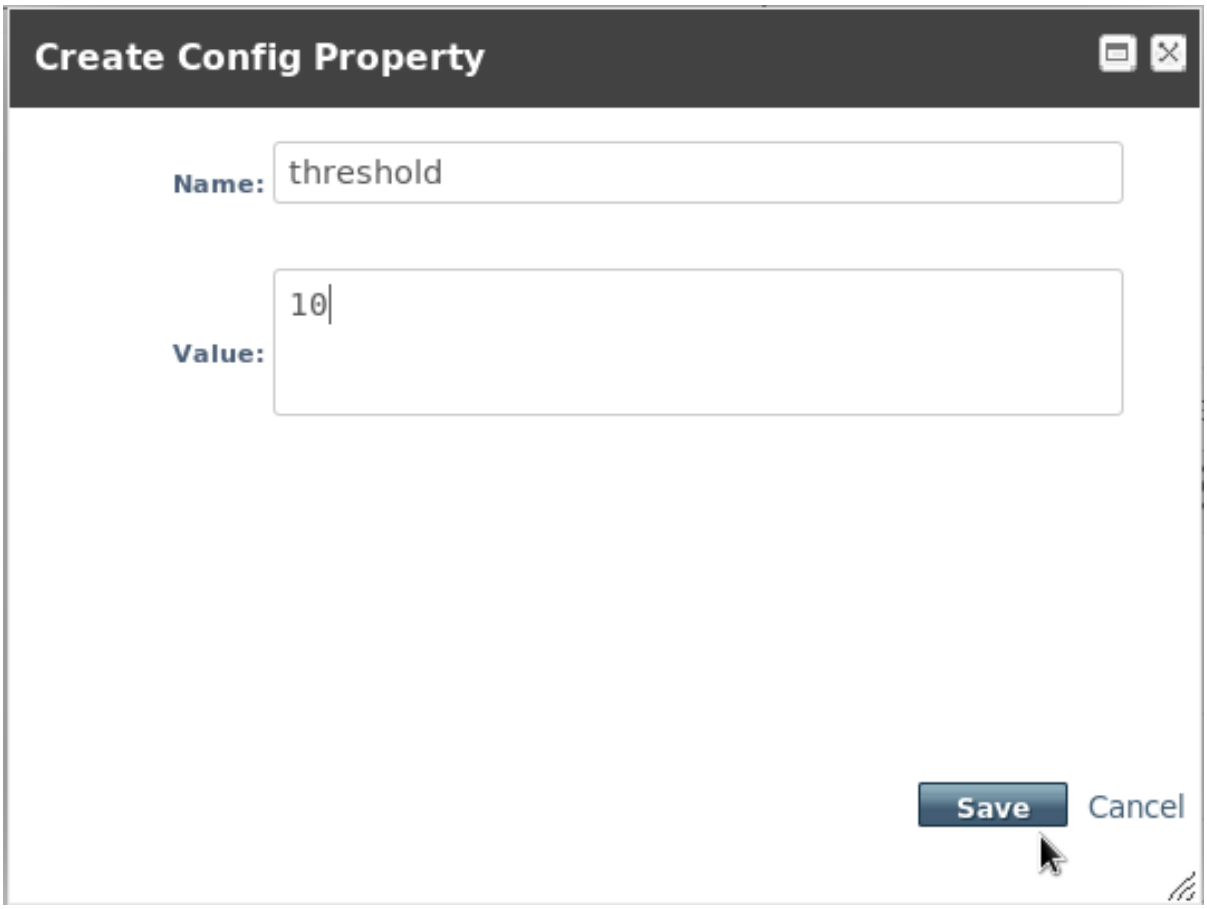



**Figure 20.25. Créer une propriété de configuration d'objet admin**

20. L'objet admin est maintenant complété, mais non actif. CLiquer sur **`Enable`** pour activer l'objet admin.

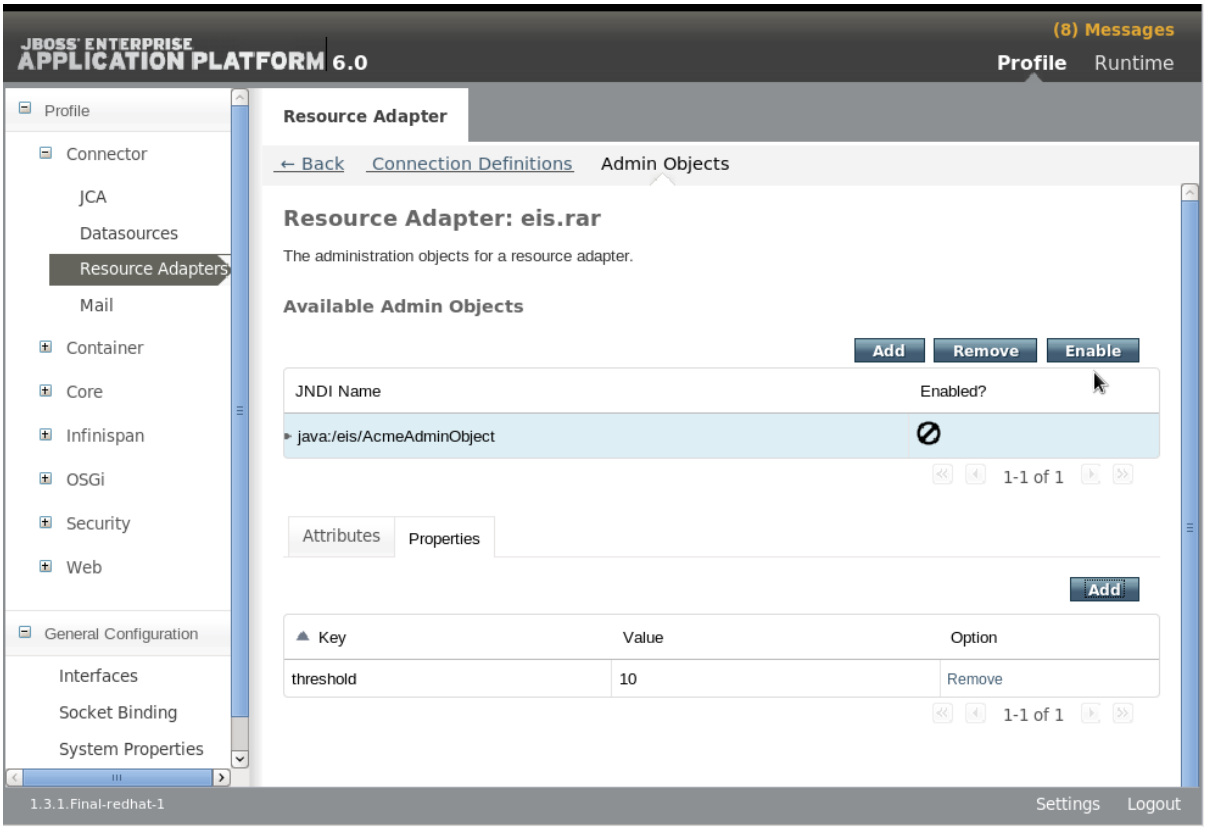



**Figure 20.26. Objet admin - Non actif**

21. Un dialogue vous demande "Souhaitez-vous réellement modifier l'Objet admin?" du nom JNDI. Cliquer sur **`Confirm`**. L'objet admin devrait maintenant afficher **`Enabled`** (activé).

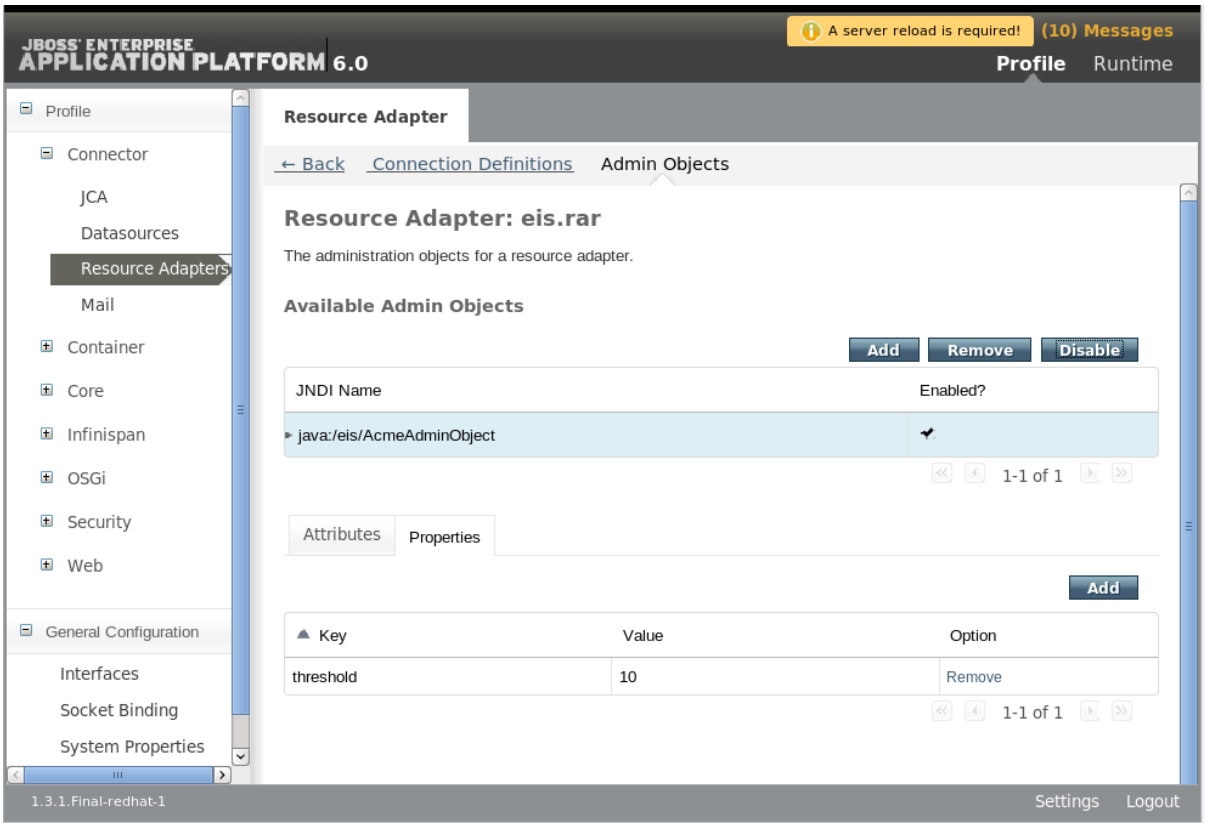



**Figure 20.27. Définition de connexion maintenant activée**

22. Vous devez charger à nouveau la configuration du serveur pour terminer ce processus. Cliquer

sur le lien **Runtime** qui se trouve en haut et à droite pour passer à la vue de Runtime, puis choisir **Configuration** dans le panneau de navigation de gauche, et cliquer sur **Reload** (Charger à nouveau) sur la droite.

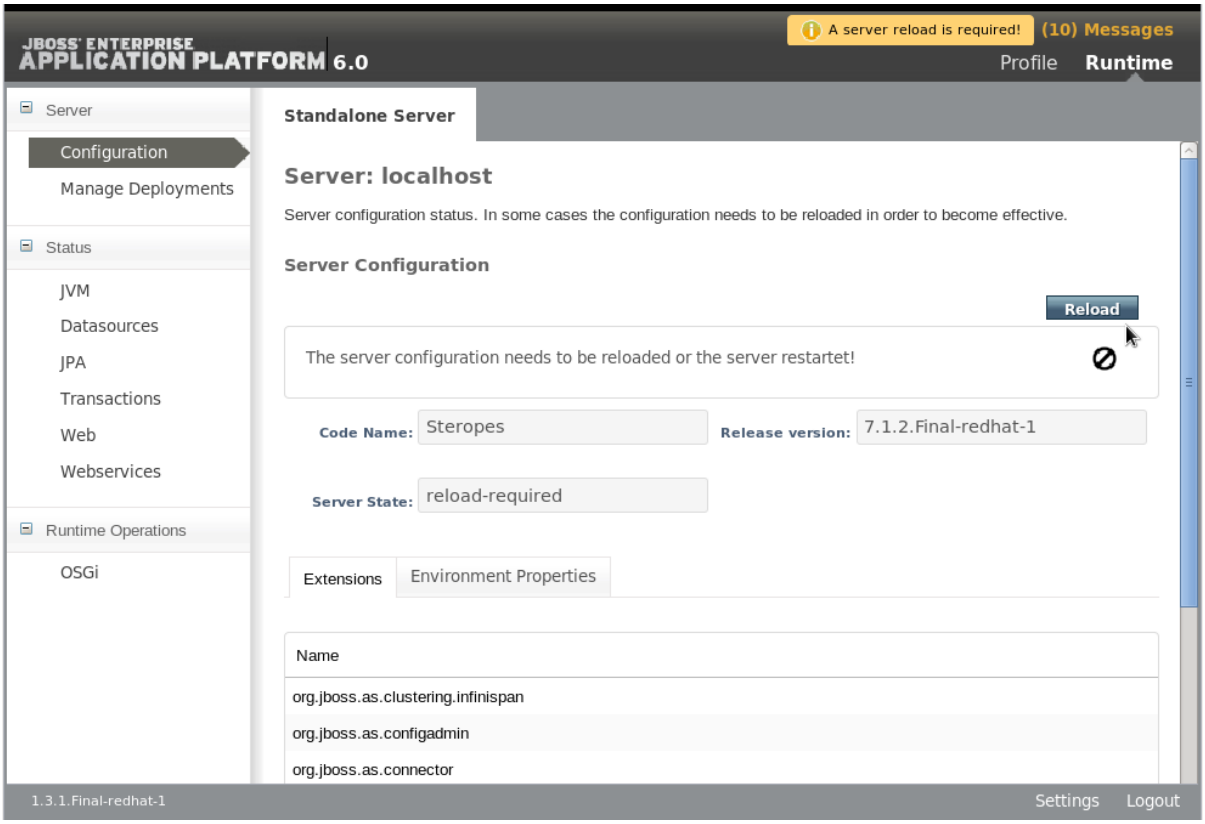



**Figure 20.28. Chargement à nouveau de la configuration de serveur**

23. Un dialogue vous demande « Souhaitez-vous charger à nouveau la configuration du serveur ? » pour le serveur indiqué ? Cliquer sur **Confirm**. La configuration du serveur sera à jour.

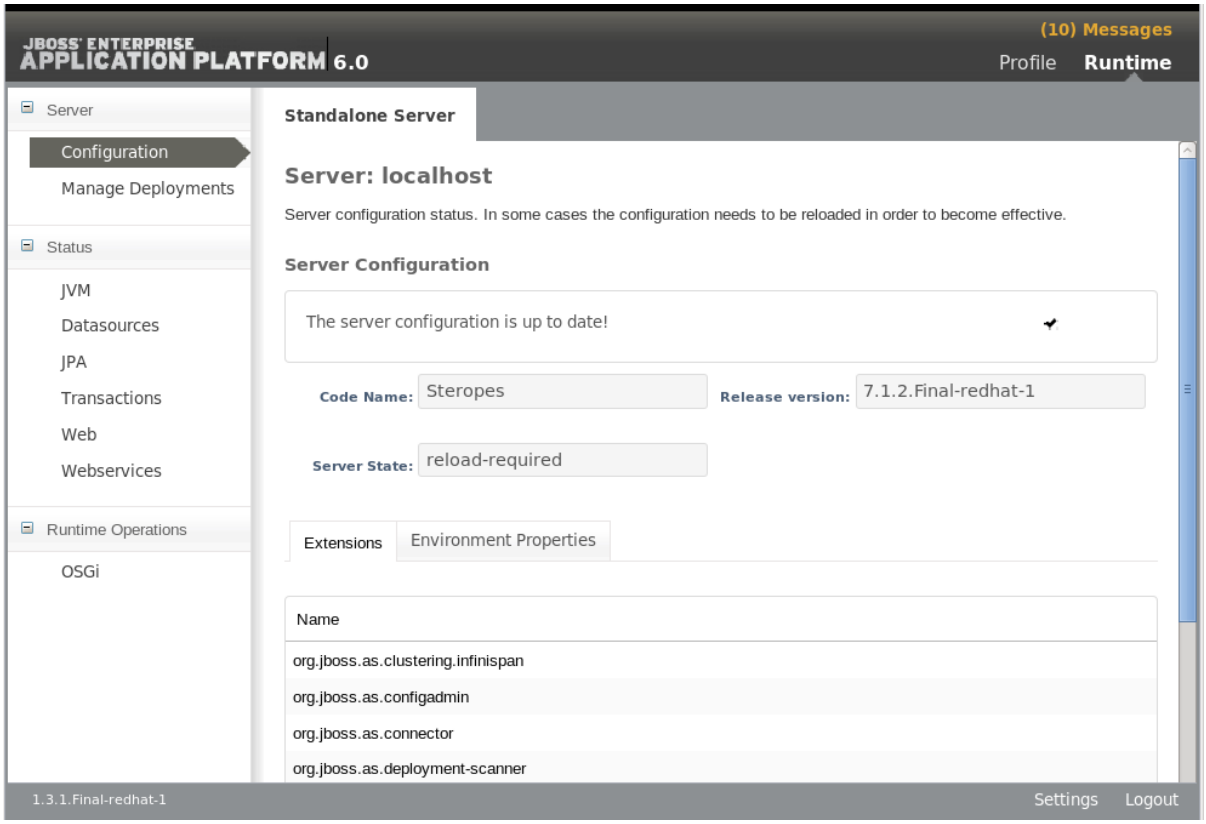



**Figure 20.29. Définition de connexion maintenant activée**

### Procédure 20.7. Configurer un adaptateur de ressources manuellement

1. Stopper le serveur de JBoss Enterprise Application Platform.

**IMPORTANT**

Vous devez interrompre le serveur avant de modifier le fichier de configuration du serveur pour que votre changement puisse être persisté au redémarrage du serveur.

2. Ouvrir le fichier de configuration du serveur pour l'éditing.
   - Pour les serveurs autonomes, il s'agit du fichier ***EAP_HOME*/standalone/configuration/standalone.xml**.
   - Si vous exécutez dans un domaine géré, il s'agira du fichier ***EAP_HOME*/domain/configuration/domain.xml**.
3. Chercher le sous-système **urn:jboss:domain:resource-adapters** dans le fichier de configuration.
4. Il n'y a pas d'adaptateurs de ressources définis pour ce système. Veuillez commencer par remplacer :

```
<subsystem xmlns="urn:jboss:domain:resource-adapters:1.0"/>
```

par ceci :

```
<subsystem xmlns="urn:jboss:domain:resource-adapters:1.0">
    <resource-adapters>
        <!-- <resource-adapter> configuration listed below -->
    </resource-adapters>
</subsystem>
```

5. Remplacer la configuration **<!-- <resource-adapter> listée ci-dessous -->** par la définition XML de l'adaptateur de ressources. Ce qui suit représente la représentation XML de la configuration de l'adaptateur de ressources créé par le Management CLI et la Console de management basée-web décrite ci-dessus.

```
<resource-adapter>
    <archive>
        eis.rar
    </archive>
    <transaction-support>XATransaction</transaction-support>
    <config-property name="server">
        localhost
    </config-property>
    <config-property name="port">
        9000
    </config-property>
    <connection-definitions>
        <connection-definition class-
name="com.acme.eis.ra.EISManagedConnectionFactory"
                jndi-name="java:/eis/AcmeConnectionFactory"
                pool-name="java:/eis/AcmeConnectionFactory">
            <config-property name="name">
                Acme Inc
            </config-property>
        </connection-definition>
    </connection-definitions>
    <admin-objects>
        <admin-object class-
name="com.acme.eis.ra.EISAdminObjectImpl"
                jndi-name="java:/eis/AcmeAdminObject"
                pool-name="java:/eis/AcmeAdminObject">
            <config-property name="threshold">
                10
            </config-property>
        </admin-object>
    </admin-objects>
</resource-adapter>
```

6. **Démarrer ler serveur**

Lancer à nouveau le serveur JBoss Enterprise Application Platform pour qu'il commence d'exécuter avec la nouvelle configuration.

Report a bug

## 20.5. RÉFÉRENCE DE DESCRIPTION D'ADAPTATEUR DE RESSOURCES

Les tableaux suivants décrivent les éléments de description d'adaptateurs de ressources.

**Tableau 20.7. Éléments principaux**

| Élément | Description |
|---|---|
| `bean-validation-groups` | Indique le groupe de validation du bean qui doit être utilisé |
| `bootstrap-context` | Indique le nom unique du contexte de bootstrapping qui doit être utilisé |
| `config-property` | Config-property spécifie les propriétés de configuration de l'adaptateur de ressources. |
| `transaction-support` | Indique le type de transactions pris en charge par l'adaptateur de ressources. La valeurs valides sont : NoTransaction, LocalTransaction, XATransaction |
| `connection-definitions` | Indique les définitions de connexion |
| `admin-objects` | Indique les objets d'administration |

**Tableau 20.8. Éléments de groupes de validation de beans**

| Élément | Description |
|---|---|
| `bean-validation-group` | Indique le nom de classe complet d'un groupe de validation de beans devant être utilisé pour la validation. |

**Tableau 20.9. Définition de connexion / attributs d'objets admin**

| Attribut | Description |
|---|---|
| `class-name` | Indique le nom de classe complet d'une usine de connexions gérée ou d'un objet admin |
| `jndi-name` | Indique le nom JNDI |
| `enabled` | L'objet doit-il être activé ? |

| Attribut | Description |
|---|---|
| `use-java-context` | Indique si on doit utiliser un contexte java:/ JNDI |
| `pool-name` | Indique le nom de pool de l'objet |
| `use-ccm` | Active le gestionnaire de connexion mis en cache |

**Tableau 20.10. Éléments de définition de connexion**

| Élément | Description |
|---|---|
| `config-property` | Config-property spécifie les propriétés de configuration de l'usine de connexions. |
| `pool` | Indique les paramètres de pooling |
| `xa-pool` | Indique les paramètres de pooling XA |
| `security` | Indique les paramètres de sécurité |
| `timeout` | Indique les paramètres de timeout |
| `validation` | Indique les paramètres de validation |
| `recovery` | Indique les paramètres de recouvrement XA |

**Tableau 20.11. Éléments de pooling**

| Élément | Description |
|---|---|
| `min-pool-size` | L'élément min-pool-size indique le nombre minimal de connexions qu'un pool peut contenir. Celles-ci ne sont pas créées tant que l'on ne connaît pas le Sujet de la demande de connexion. Cette valeur par défaut à 0 |
| `max-pool-size` | L'élément max-pool-size indique le nombre maximal de connexions d'un pool. On ne pourra pas créer plus de connexions que ce nombre indiqué pour chaque sub-pool. Cette valeur par défaut à 20. |
| `prefill` | Indique si l'on doit essayer de pré-remplir le pool de connexion. La valeur par défaut est false. |
| `use-strict-min` | Indique si la min-pool-size doit être considérée sérieusement. La valeur par défaut est false. |

| Élément | Description |
| --- | --- |
| **flush-strategy** | Indique comment le pool doit être vidé en cas d'erreur. Les valeurs valides sont FailingConnectionOnly (default), IdleConnections, EntirePool |

**Tableau 20.12. Éléments de pool XA**

| Élément | Description |
| --- | --- |
| **min-pool-size** | L'élément min-pool-size indique le nombre minimal de connexions qu'un pool peut contenir. Celles-ci ne sont pas créées tant que l'on ne connaît pas le Sujet de la demande de connexion. Cette valeur par défaut à 0 |
| **max-pool-size** | L'élément max-pool-size indique le nombre maximal de connexions d'un pool. On ne pourra pas créer plus de connexions que ce nombre indiqué pour chaque sub-pool. Cette valeur par défaut à 20. |
| **prefill** | Indique si l'on doit essayer de pré-remplir le pool de connexion. La valeur par défaut est false. |
| **use-strict-min** | Indique si la min-pool-size doit être considérée sérieusement. La valeur par défaut est false. |
| **flush-strategy** | Indique comment le pool doit être vidé en cas d'erreur. Les valeurs valides sont FailingConnectionOnly (default), IdleConnections, EntirePool |
| **is-same-rm-override** | L'élément is-same-rm-override element permet de définir inconditionnellement si javax.transaction.xa.XAResource.isSameRM(XAResource) doit renvoyer true ou false |
| **interleaving** | Élément qui permet ou non l'entrelacement des usines de connexions XA |
| **no-tx-separate-pools** | Oracle n'aime pas que les connexions XA soient utilisées à la fois à l'intérieur et à l'extérieur d'une connexion JTA. Pour résoudre ce problème, vous pourrez créer des sub-pools pour ces contextes différents. |
| **pad-xid** | Est-ce que le Xid doit être capitonné ? |
| **wrap-xa-resource** | Est-ce que les instances XAResource doivent être encapsulées dans une instance org.jboss.tm.XAResourceWrapper |

**Tableau 20.13. Éléments de sécurité**

| Élément | Description |
| --- | --- |
| **application** | Indique si les paramètres de sécurité fournis (comme getConnection(user, pw) sont utilisés pour distinguer les connexions d'un pool. |
| **security-domain** | Indique si des Sujets (de domaine de sécurité) sont utilisés pour distinguer les connexions d'un pool. Le contenu du domaine de sécurité correspond au nom du gestionnaire de sécurité JAAS qui gère l'authentification. Ce nom est en corrélation à l'attribut « JAAS login-config.xml descriptor application-policy/name ». |
| **security-domain-and-application** | Indique que les paramètres de l'application fournis (par exemple, à partir de getConnection (utilisateur, pw)) ou que le Sujet (du domaine de la sécurité) soient utilisés pour distinguer les connexions du pool. Le contenu du domaine de sécurité est le nom du gestionnaire de sécurité JAAS qui gère l'authentification. Ce nom correspond à l'attribut « JAAS login-config.xml descriptor application-policy/name ». |

**Tableau 20.14. Éléments de timeout**

| Élément | Description |
| --- | --- |
| **blocking-timeout-millis** | L'élément « blocking-timeout-millis » indique la durée maximale en millisecondes de blocage pendant que vous attendez une connexion, avant de lever une exception. Notez que cela bloque uniquement pendant que vous attendez un permis de connexion, et ne soulèvera pas d'exception si la création d'une nouvelle connexion prend un temps excessivement long. La valeur par défaut est 30000 (30 secondes). |
| **idle-timeout-minutes** | Les éléments idle-timeout-minutes indiquent la durée maximum, en minutes, avant qu'une connexion inutile puisse être fermée. La durée maximum dépend du temps de balayage de l'idleRemover, qui correspond à la moitié du temps « idle-timeout-minutes » le plus petit de n'importe quel pool. |
| **allocation-retry** | Cet élément de tentative d'allocation indique le nombre de fois que l'on doit allouer un connexion avant de lancer une exception. La valeur par défaut est 0. |
| **allocation-retry-wait-millis** | Le temps, en millisecondes, qu'il faut attendre avant de retenter d'allouer une connexion. La valeur par défaut est 5 000, soit 5 secondes. |
| **xa-resource-timeout** | Passé à XAResource.setTransactionTimeout(). La valeur par défaut est 0 sans invoquer le setter. Indiqué en secondes. |

**Tableau 20.15. Éléments de validation**

| Élément | Description |
| --- | --- |
| **background-validation** | Élément pour spécifier que les connexions doivent être validées en arrière-plan plutôt qu'avant utilisation |
| **background-validation-minutes** | L'élément « background-validation-minutes » indique la durée, en minutes, d'exécution de la validation d'arrière-plan. |
| **use-fast-fail** | Indique s'il y a échec d'allocation de connexion à la première connexion si invalide (true) ou s'il y a de nouvelles tentatives jusqu'à ce que le pool soit épuisé de toutes les essais de connexion possibles (false). La valeur par défaut est false. |

**Tableau 20.16. Éléments d'objets admin**

| Élément | Description |
| --- | --- |
| **config-property** | Spécifie une propriété de configuration d'objet d'administration. |

**Tableau 20.17. Éléments de recouvrement**

| Élément | Description |
| --- | --- |
| **recover-credential** | Indique la paire nom / mot de passe ou le domaine de sécurité qui doit être utilisé pour le recouvrement. |
| **recover-plugin** | Spécifie l'implémentation de org.jboss.jca.core.spi.recovery.RecoveryPlugin class. |

Les schéma de déploiement sont définis dans **jboss-as-resource-adapters_1_0.xsd** and http://docs.jboss.org/ironjacamar/schema/ironjacamar_1_0.xsd pour l'activation automatique.

Report a bug

## 20.6. AFFICHAGES DES STATISTIQUES DE CONNEXION

Vous pouvez lire les statistiques d'une connexion définie à partir de la sous-arborescence **deployment=name.rar**.

Les statistiques sont définis à ce niveau et non pas au niveau **/subsystem** afin d'être accessible à partir de tout **rar** non défini dans les configurations de fichiers **standalone.xml** ou **domain.xml**.

Par exemple :

**Exemple 20.1.**

```
/deployment=example.rar/subsystem=resource-
adapters/statistics=statistics/connection-
definitions=java\:\/testMe:read-resource(include-runtime=true)
```

**NOTE**

Veillez à ce que l'argument ***`include-runtime=true`*** et à ce que tous les statistiques soient en runtime uniquement et la valeur par défaut est **`false`**.

Report a bug

## 20.7. STATISTIQUES D'ADAPTATEUR DE RESSOURCES

**Statistiques principaux**

Le tableau suivant contient une liste de statistiques principaux d'adaptateurs de ressources pris en charge :

**Tableau 20.18. Statistiques principaux**

| Nom | Description |
|---|---|
| **`ActiveCount`** | Le nombre de connexions actives. Chacune de ces connexions est soit utilisée par une application, ou disponible via pool |
| **`AvailableCount`** | Le nombre de connexions disponibles dans le pool |
| **`AverageBlockingTime`** | Le durée moyenne passée à bloquer l'obtention d'un verrou exclusif sur le pool. La valeur est en millisecondes. |
| **`AverageCreationTime`** | Le durée moyenne passée à créer une connexion. La valeur est en millisecondes. |
| **`CreatedCount`** | Le nombre de connexions créées. |
| **`DestroyedCount`** | Le nombre de connexions détruites. |
| **`InUseCount`** | Le nombre de connexions actuellement utilisées. |
| **`MaxCreationTime`** | La durée maximum pour créer une connexion. La valeur est en millisecondes. |
| **`MaxUsedCount`** | Le nombre maximum de connexions utilisées |
| **`MaxWaitCount`** | Le nombre maximum de requêtes attendant une connexion en même temps. |
| **`MaxWaitTime`** | Le durée maximum à attendre un verrou exclusif sur le pool. |
| **`TimedOut`** | Le nombre de connexions expirées |
| **`TotalBlockingTime`** | Le durée à attendre un verrou exclusif sur le pool. La valeur est en millisecondes. |

| Nom | Description |
| --- | --- |
| `TotalCreationTime` | La durée passée à créer des connexions. La valeur est en millisecondes. |
| `WaitCount` | Le nombre de requêtes en attente de connexion. |

Report a bug

## 20.8. DÉPLOYER L'ADAPTATEUR DE RESSOURCES WEBSPHERE MQ

### Websphere MQ

WebSphere MQ est un logiciel de messagerie Oriented Middleware (MOM) d'IBM qui permet à des applications sur des systèmes distribués à communiquer entre eux. Ceci est accompli grâce à l'utilisation des messages et des files d'attente de messages. WebSphere MQ est chargé de remettre des messages à des files d'attente de messages et pour transférer des données à d'autres gestionnaires de file d'attente à l'aide de canaux de message. Pour plus d'informations sur WebSphere MQ, voir WebSphere MQ.

### Résumé

Cette section couvre les étapes à suivre pour déployer et configurer l'adaptateur de ressource Websphere MQ dans JBoss Enterprise Application Platform 6. Cela peut se faire manuellement en modifiant les fichiers de configuration, par le Management CLI, ou par la Console de gestion basée-web.

### Prérequis

Avant de démarrer, vous devrez vérifier votre version d'adaptateur de ressource WebSphere MQ et comprendre certaines propriétés de configuration de WebSphere MQ.

- L'adaptateur de ressources WebSphere MQ est fourni en tant que fichier RAR (Resource Archive) nommé **`wmq.jmsra-VERSION.rar`**. Vous devrez utiliser la version **`7.0.1.7`** ou version supérieure.
- Vous devez connaître les valeurs des propriétés de configuration Websphere MQ suivantes. Voir la documentation de produit WebSphere MQ pour obtenir des détails sur ces propriétés.
  - MQ.QUEUE.MANAGER: le nom du gestionnaires de files d'attentes de WebSphere MQ
  - MQ.HOST.NAME: le nom d'hôte utilisé pour se connecter au gestionnaire de files d'attente de WebSphere MQ
  - MQ.CHANNEL.NAME: le canal de serveur utilisé pour se connecter au gestionnaire de files d'attente de WebSphere MQ
  - MQ.QUEUE.NAME: le nom de la file d'attente de destination
  - MQ.PORT: le port utilisé pour se connecter au gestionnaire de files d'attente de WebSphere MQ
  - MQ.CLIENT: le type de transport
- Pour les connexions sortantes, vous devrez vous familiariser avec la propriété de configuration suivante :

- MQ.CONNECTIONFACTORY.NAME: le nom de l'instance d'usine de connexion qui fournit la connexion vers le système à distance.

**NOTE**

Voici les configurations par défaut fournies par IBM. Elles sont assujetties au changement. Veuillez vous référer à la documentation Websphere MQ pour plus d'informations.

**Procédure 20.8. Déployer l'adaptateur de ressources manuellement**

1. Si vous avez besoin d'un support de transactions avec l'adaptateur de ressources de WebSphereMQ, vous devrez re-paquager l'archive **`wmq.jmsra-VERSION.rar`** pour qu'elle inclue **`mqetclient.jar`**. Vous devrez utiliser la commande suivante :

   ```
   [user@host ~]$ jar -uf wmq.jmsra-VERSION.rar mqetclient.jar
   ```

   Soyez certain de remplacer *VERSION* par le numéro de version correct.

2. Copier le fichier **`wmq.jmsra-VERSION.rar`** dans le répertoire **`EAP_HOME/standalone/deployments/`**.

3. Ajouter l'adaptateur de ressources au fichier de configuration du serveur.

   a. Ouvrir le fichier **`EAP_HOME/standalone/configuration/standalone-full.xml`** dans un éditeur.

   b. Chercher le sous-système **`urn:jboss:domain:resource-adapters`** dans le fichier de configuration.

   c. Il n'y a pas d'adaptateurs de ressources définis pour ce système. Veuillez commencer par remplacer :

   ```
   <subsystem xmlns="urn:jboss:domain:resource-adapters:1.0"/>
   ```

   par ceci :

   ```
   <subsystem xmlns="urn:jboss:domain:resource-adapters:1.0">
       <resource-adapters>
           <!-- <resource-adapter> configuration listed below -->
       </resource-adapters>
   </subsystem>
   ```

   d. La configuration de l'adaptateur de ressources dépend de si vous avez besoin de la restauration et du support de transactions. Si vous n'avez pas besoin de support de transaction, choisissez la première étape de configuration ci-dessous. Si vous avez besoin de support de transaction, choisissez la deuxième étape de la configuration.

- Pour les déploiements non transactionnels, veuillez remplacer la configuration **<!-- <resource-adapter> listée ci-dessous -->** par ce qui suit :

```
<resource-adapter>
    <archive>
        wmq.jmsra-VERSION.rar
    </archive>
    <transaction-support>NoTransaction</transaction-support>
    <connection-definitions>
        <connection-definition
                class-
name="com.ibm.mq.connector.outbound.ManagedConnectionFactoryIm
pl"
                jndi-
name="java:jboss/MQ.CONNECTIONFACTORY.NAME"
                pool-name="MQ.CONNECTIONFACTORY.NAME">
            <config-property name="channel">
                MQ.CHANNEL.NAME
            </config-property>
            <config-property name="transportType">
                MQ.CLIENT
            </config-property>
            <config-property name="queueManager">
                MQ.QUEUE.MANAGER
            </config-property>
            <security>
                <security-domain>MySecurityDomain</security-
domain>
            </security>
       </connection-definition>
    </connection-definitions>
    <admin-objects>
        <admin-object
                class-
name="com.ibm.mq.connector.outbound.MQQueueProxy"
                jndi-name="java:jboss/MQ.QUEUE.NAME"
                pool-name="MQ.QUEUE.NAME">
            <config-property name="baseQueueName">
                MQ.QUEUE.NAME
            </config-property>
        </admin-object>
    </admin-objects>
</resource-adapter>
```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR.

- Pour les déploiements transactionnels, veuillez remplacer la configuration **<!-- <resource-adapter> listée ci-dessous -->** par ce qui suit :

```
<resource-adapter>
    <archive>
```

```
        wmq.jmsra-VERSION.rar
    </archive>
    <transaction-support>XATransaction</transaction-support>
    <connection-definitions>
        <connection-definition
                class-
name="com.ibm.mq.connector.outbound.ManagedConnectionFactoryIm
pl"
                jndi-
name="java:jboss/MQ.CONNECTIONFACTORY.NAME"
                pool-name="MQ.CONNECTIONFACTORY.NAME">
            <config-property name="channel">
                MQ.CHANNEL.NAME
            </config-property>
            <config-property name="transportType">
                MQ.CLIENT
            </config-property>
            <config-property name="queueManager">
                MQ.QUEUE.MANAGER
            </config-property>
           <security>
                <security-domain>MySecurityDomain</security-
domain>
            </security>
            <recovery>
                <recover-credential>
                    <user-name>USER_NAME</user-name>
                    <password>PASSWORD</password>
                </recover-credential>
            </recovery>
        </connection-definition>
    </connection-definitions>
    <admin-objects>
        <admin-object
                class-
name="com.ibm.mq.connector.outbound.MQQueueProxy"
                jndi-name="java:jboss/MQ.QUEUE.NAME"
                pool-name="MQ.QUEUE.NAME">
            <config-property name="baseQueueName">
                MQ.QUEUE.NAME
            </config-property>
        </admin-object>
    </admin-objects>
</resource-adapter>
```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR. Vous devrez également remplacer *USER_NAME* et *PASSWORD* avec le nom et le mot de passe valides.

**NOTE**

Pour supporter les transactions, l'élément <transaction-support> a été défini à **`XATransaction`**. Pour supporter XA recovery, l'élément <recovery> a été ajouté à une définition de connexion.

e. Si vous souhaitez changer le fournisseur par défaut en système de messagerie EJB3 dans JBoss Enterprise Application Platform 6 de HornetQ vers WebSphere MQ, modifier le sous-système the **`urn:jboss:domain:ejb3:1.2`** comme suit :

Remplacer :

```
<mdb>
    <resource-adapter-ref resource-adapter-name="hornetq-ra"/>
    <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

par :

```
<mdb>
    <resource-adapter-ref resource-adapter-name="wmq.jmsra-
VERSION.rar"/>
    <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR.

**Procédure 20.9. Modifier le code MDB pour utiliser l'adaptateur de ressources**

- Configurer ActivationConfigProperty et ResourceAdapter du code MDB comme suit :

```
@MessageDriven( name="WebSphereMQMDB",
    activationConfig =
    {
        @ActivationConfigProperty(propertyName =
"destinationType",propertyValue = "javax.jms.Queue"),
        @ActivationConfigProperty(propertyName = "useJNDI",
propertyValue = "false"),
        @ActivationConfigProperty(propertyName = "hostName",
propertyValue = "MQ.HOST.NAME"),
        @ActivationConfigProperty(propertyName = "port",
propertyValue = "MQ.PORT"),
        @ActivationConfigProperty(propertyName = "channel",
propertyValue = "MQ.CHANNEL.NAME"),
        @ActivationConfigProperty(propertyName = "queueManager",
propertyValue = "MQ.QUEUE.MANAGER"),
        @ActivationConfigProperty(propertyName = "destination",
propertyValue = "MQ.QUEUE.NAME"),
```

```
        @ActivationConfigProperty(propertyName = "transportType",
propertyValue = "MQ.CLIENT")
    })
    @ResourceAdapter(value = "wmq.jmsra-VERSION.rar")
@TransactionAttribute(TransactionAttributeType.NOT_SUPPORTED)
public class WebSphereMQMDB implements MessageListener {
}
```

Soyez certain de remplacer *VERSION* par le numéro de version correct qui se trouve dans le nom du RAR.

Report a bug

# CHAPITRE 21. DÉPLOYER JBOSS ENTERPRISE APPLICATION PLATFORM 6 SUR AMAZON EC2

## 21.1. INTRODUCTION

### 21.1.1. Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) est un service exploité par amazon.com qui offre aux clients un environnement informatique virtuel personnalisable. Une Image de Machine Amazon (AMI) peut être démarrée en utilisant le service pour créer une instance ou une machine virtuelle. Les utilisateurs peuvent installer n'importe quel logiciel dont ils ont besoin sur une instance et sont facturés en fonction de l'usage. Amazon EC2 est conçu pour être flexible et permettre aux utilisateurs de déployer rapidement leurs applications à l'échelle qui leur convient.

Vous pourrez en savoir davantage sur le site web Amazon EC2, http://aws.amazon.com/ec2/.

Report a bug

### 21.1.2. Amazon Machine Instances (AMIs)

Une Amazon Machine Image (AMI) est un modèle d'instance de machine virtuelle EC2. Les utilisateurs créent des instances EC2 en sélectionnant une AMI appropriée pour créer l'instance. La composante primaire d'une AMI est un système de fichiers lecture seule qui contient un système d'exploitation installé, mais aussi des autres logiciels. Chaque AMI a différents logiciels installés pour les cas d'utilisation différents. Amazon EC2 comprend beaucoup d'AMIs au choix offerts par amazon.com et des tierces parties. Les utilisateurs peuvent également créer leurs propres AMIs personnalisées.

Report a bug

### 21.1.3. JBoss Cloud Access

JBoss Cloud Access est une fonctionnalité de Red Hat qui fournit un support à JBoss Enterprise Application Platform 6 aux fournisseurs cloud certifiés Red Hat comme Amazon EC2. JBoss Cloud Access vous permet de déplacer vos abonnements entre les serveurs traditionnels et les ressources publiques basées-cloud d'une façon simple et peu coûteuse.

Vous trouverez des informations supplémentaires à l'adresse suivante http://www.redhat.com/solutions/cloud/access/jboss/.

Report a bug

### 21.1.4. Fonctionnalités de JBoss Cloud Access

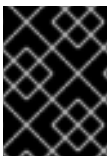
L'abonnement au programme JBoss Cloud Access donne accès aux AMI (Amazon Machine Images) privées créées par Red Hat.

Les AMI de Red Hat ont le logiciel suivant pré-installé et complètement pris en charge par Red Hat :

- Red Hat Enterprise Linux 6
- JBoss Enterprise Application Platform 6
- L'agent JBoss Operations Network (JON) 3

- Mises à jour de produit par les RPM par l'intermédiaire de l'infrastructure de mise à jour de Red Hat.

Chaque AMI de Red Hat n'est qu'un point de départ, qui requiert une configuration supplémentaire pour se confirmer aux besoins de votre application.



**IMPORTANT**

JBoss Cloud Access n'apporte pas actuellement de support au profil full-ha, ni pour les instances standalone, ni pour les domaines gérés.

Report a bug

### 21.1.5. Types d'instances Amazon EC2 prises en charge

JBoss Cloud Access prend en charge les types d'instance Amazon EC2 suivantes. Voir *Amazon EC2 User Guide* pour obtenir davantage de détails sur chaque type d'instance, http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/instance-types.html.

**Tableau 21.1. Types d'instances Amazon EC2 prises en charge**

| Type d'instance | Description |
|---|---|
| Instance standard | Les instances standard sont des environnements d'ordre général ayant un ration de mémoire-à-CPU équilibré. |
| Instance de mémoire élevée | Les Instances de mémoire élevée possède davantage de mémoire allouée que les Instances standard. Les Instances de mémoire élevée conviennent aux applications à haut débit telles que les bases de données ou les applications de mise en cache de mémoire. |
| Instance Haut CPU | Les Instance Haut CPU ont davantage de ressources CPU allouées que de mémoire et conviennent à des débits moindres mais à des applications intensives en CPU. |

**IMPORTANT**

Le type d'instance **`Micro (t1.micro)`** ne convient pas au déploiement de la plateforme JBoss Enterprise Application.

Report a bug

### 21.1.6. AMI Red Hat pris en charge

Les AMI Red Hat pris en charge peuvent être identifiés par leur nom AMI.

LEs AMI de JBoss Enterprise Application Platform 6 AMIs sont composés ainsi:

```
RHEL-osversion-JBEAP-6.0.0-arch-creationdate
```

*osversion* est le nom de version de Red Hat Enterprise Linux installé dans l'AMI. Exemple **`6.2`**.

*arch* est l'architecture de l'AMI. Correspondra à **`x86_64`** ou **`i386`**.

*creationdate* est la date de création de l'AMI sous le format *YYYYMMDD*. Exemple **`20120501`**.

Exemple de nom d'AMI : **`RHEL-6.2-JBEAP-6.0.0-x86_64-20120501`**.

Report a bug

## 21.2. DÉPLOYER JBOSS ENTERPRISE APPLICATION PLATFORM 6 SUR AMAZON EC2

### 21.2.1. Aperçu du déploiement de JBoss Enterprise Application Platform 6 sur Amazon EC2

JBoss Enterprise Application Platform 6 peut être déployé avec l'AMI Amazon EC2. L'AMI contient tout ce qui est requis pour le déploiement des instances clusterisées ou non clusterisées.

Le déploiement d'instances non clusterisées est le scénario le plus facile. Cela exige uniquement quelques changements de configuration pour spécifier votre déploiement d'application quand vous créez l'instance.

Le déploiement d'instances en cluster est plus compliqué. En plus de vos instances en cluster, vous devez déployer une instance de JBoss Enterprise Application Platform 6 pour agir comme proxy mod_cluster et S3 Bucket pour le protocole de découverte S3_PING JGroups. Red Hat recommande également la création d'un Cloud privé virtuel pour contenir votre cluster.

Chacune de ces étapes est expliquée ci-dessous mais on assume que vous êtes expérimenté avec JBoss Enterprise Application Platform 6, Red Hat Enterprise Linux 6 et Amazon EC2.

La documentation supplémentaire suivante est recommandée :

- JBoss Enterprise Application Platform 6, https://access.redhat.com/site/documentation/JBoss_Enterprise_Application_Platform/.
- Red Hat Enterprise Linux 6, https://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/.
- Amazon Web Services, http://aws.amazon.com/documentation/.

Report a bug

### 21.2.2. JBoss Enterprise Application Platform 6 non clusterisés

#### 21.2.2.1. Instances non-clusterisées

Une instance non clusterisée est une instance simple Amazon EC2 exécutant sur JBoss Enterprise Application Platform 6. Elle ne fait pas partie d'un cluster.

Report a bug

#### 21.2.2.2. Instances non clusterisées

##### 21.2.2.2.1. Lancer une instance de JBoss Enterprise Application Platform 6 non clusterisée

**Résumé**

Ce sujet couvre les étapes requises pour lancer une instance de JBoss Enterprise Application Platform 6 non clusterisée sur Red Hat AMI (Amazon Machine Image).

**Prérequis**

- Pour un AMI Red Hat, consulter Section 21.1.6, « AMI Red Hat pris en charge ».
- Groupe de sécurité pré-configuré qui autorise les requêtes entrantes sur les ports 22, 8080, et 9990 au moins.

**Procédure 21.1. Lancer une instance non clusterisée de JBoss Enterprise Application Platform 6 sur Red Hat AMI (Amazon Machine Image).**

1. Configurer le champ **`User Data`**. Les paramètres configurables sont disponibles ici : Section 21.4.1, « Paramètres de configuration permanente », Section 21.4.2, « Paramètres de scripts personnalisés ».

   **Exemple 21.1. Exemple de champ de données d'utilisateur**

   L'exemple montre le champ de données utilisateur d'une instance JBoss Enterprise Application Platform 6 non clusterisée. Le mot de passe de l'utilisateur **`admin`** a été défini à **`adminpwd`**.

   ```
   JBOSSAS_ADMIN_PASSWORD=adminpwd
   JBOSS_IP=0.0.0.0 #listen on all IPs and interfaces

   # In production, access to these ports needs to be restricted for
   security reasons
   PORTS_ALLOWED="9990 9443"

   cat> $USER_SCRIPT << "EOF"

   # Get the application to be deployed from an Internet URL
   # mkdir -p /usr/share/java/jboss-ec2-eap-applications
   # wget https://<your secure storage hostname>/<path>/<app
   name>.war -O /usr/share/java/jboss-ec2-eap-applications/<app
   name>.war

   # Create a file of CLI commands to be executed after starting the
   server
   cat> $USER_CLI_COMMANDS << "EOC"
   # deploy /usr/share/java/jboss-ec2-eap-applications/<app name>.war
   EOC

   EOF
   ```

2. **Pour les instances de production**
   Pour une instance de production, ajouter la ligne suivante sous la ligne **`USER_SCRIPT`** du champ **`User Data`** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

   ```
   yum -y update
   ```

-

**NOTE**

**`yum -y update`** doit être exécuté régulièrement, pour appliquer les correctifs de sécurité et les améliorations.

3. Lancement de l'instance AMI Red Hat

**Résultat**

Une instance non clusterisée de JBoss Enterprise Application Platform 6 a été configurée, et lancée sur un AMI Red Hat.

Report a bug

#### 21.2.2.2.2. Déployer une instance de JBoss Enterprise Application Platform 6 non clusterisée

**Résumé**

Ce sujet couvre le déploiement d'une application sur une instance de JBoss Enterprise Application Platform sur un AMI Red Hat.

1. - **Déploiement d'un exemple d'application**
   Ajouter les lignes suivantes au champ **`User Data`** :

```
# Deploy the sample application from the local filesystem
deploy --force /usr/share/java/jboss-ec2-eap-samples/hello.war
```

**Exemple 21.2. Champs de données d'utilisateur avec l'exemple d'application**

Cet exemple utilise l'exemple d'application fourni sur l'AMI Red Hat. Il inclut également une configuration de base d'une instance non clusterisée de JBoss Enterprise Application Platform 6. Le mot de passe **`admin`** de l'utilisateur a été défini à **`adminpwd`**.

```
JBOSSAS_ADMIN_PASSWORD=adminpwd
JBOSS_IP=0.0.0.0 #listen on all IPs and interfaces

# In production, access to these ports needs to be restricted
for security reasons
PORTS_ALLOWED="9990 9443"

cat> $USER_SCRIPT << "EOF"

# Create a file of CLI commands to be executed after starting
the server
cat> $USER_CLI_COMMANDS << "EOC"

# Deploy the sample application from the local filesystem
deploy --force /usr/share/java/jboss-ec2-eap-samples/hello.war
EOC

EOF
```

- **Déployer une application personnalisée**
  Ajouter les lignes suivantes au champ **`User Data`** (données utilisateur), pour configurer le nom de l'URL de l'application :

```
# Get the application to be deployed from an Internet URL
mkdir -p /usr/share/java/jboss-ec2-eap-applications
wget https://<your secure storage hostname>/<path>/<app name>.war
-O /usr/share/java/jboss-ec2-eap-applications/<app name>.war
```

**Exemple 21.3. Exemple de champ de données utilisateur avec application personnalisée**

Cet exemple utilise une application nommée **`MyApp`**, et inclut une configuration de base pour une instance JBoss Enterprise Application Platform 6 non clusterisée. Le mot de passe **`admin`** de l'utilisateur a été défini à **`adminpwd`**.

```
JBOSSAS_ADMIN_PASSWORD=adminpwd
JBOSS_IP=0.0.0.0 #listen on all IPs and interfaces

# In production, access to these ports needs to be restricted
for security reasons
PORTS_ALLOWED="9990 9443"

cat> $USER_SCRIPT << "EOF"

# Get the application to be deployed from an Internet URL
mkdir -p /usr/share/java/jboss-ec2-eap-applications
wget https://PATH_TO_MYAPP/MyApp.war -O /usr/share/java/jboss-
ec2-eap-applications/MyApp.war

# Create a file of CLI commands to be executed after starting
the server
cat> $USER_CLI_COMMANDS << "EOC"
deploy /usr/share/java/jboss-ec2-eap-applications/MyApp.war
EOC

EOF
```

2. Lancement de l'instance AMI Red Hat

**Résultat**

L'application a été déployée avec succès dans JBoss Enterprise Application Platform 6.

Report a bug

#### 21.2.2.2.3. Lancer l'instance de JBoss Enterprise Application Platform 6 non clusterisée

**Résumé**

Cette rubrique couvre les étapes nécessaires pour tester la plateforme non clusterisée de JBoss Enterprise Application Platform 6.

**Procédure 21.2. Tester que l'instance de JBoss Enterprise Application Platform 6 non clusterisée exécute correctement.**

1. Déterminer le **`Public DNS`** de l'instance, qui se situe dans le panneau d'informations de l'instance.
2. Naviguer dans **`http://<public-DNS>:8080`**.
3. Confirmer que la page d'accueil de JBoss Enterprise Application Platform apparaît, y compris le lien vers la Console admin. Si la page d'accueil n'est pas disponible, veuillez consulter : Section 21.5.1, « Résolution de problèmes dans Amazon EC2 ».
4. Cliquer sur l'hyperlink **`Admin Console`**.
5. Connectez-vous :
   - Nom d'utilisateur : **`admin`**
   - Mot de passe : Sspécifié dans le champ **`User Data`** ici : Section 21.2.2.2.1, « Lancer une instance de JBoss Enterprise Application Platform 6 non clusterisée ».
6. **Tester l'exemple d'application**
   Naviguer dans **`http://<public-DNS>:8080/hello`** pour tester que l'exemple d'application exécute avec succès. Le texte **`Hello World!`** doit apparaître dans le navigateur. Si le texte n'apparait pas, voir: Section 21.5.1, « Résolution de problèmes dans Amazon EC2 ».
7. Déconnectez-vous de la console admin de JBoss Enterprise Application Platform.

**Résultat**

L'instance de JBoss Enterprise Application Platform 6 exécute correctement.

Report a bug

### 21.2.2.3. Domaines gérés non clusterisés

#### 21.2.2.3.1. Lancer une instance pour qu'elle serve en tant que Contrôleur de domaine

**Résumé**

Cette rubrique couvre les étapes requises pour lancer un domaine géré non clusterisé de JBoss Enterprise Application Platform sur une AMI Red Hat (Amazon Machine Image)

**Prérequis**

- Pour une AMI Red Hat qui convient, consulter Section 21.1.6, « AMI Red Hat pris en charge ».
- Section 21.2.3.4, « Créer un VPC (Virtual Private Cloud) »
- Section 21.2.3.5, « Lancer une instance Apache HTTPD pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC »
- Section 21.2.3.6, « Configurer le routage par défaut du sous-système privé VPC »

**Procédure 21.3. Lancer un domaine géré de JBoss Enterprise Application Platform 6 non clusterisé sur AMI Red Hat**

1. Dans l'onglet Groupe de sécurité, veillez bien à ce que tout le trafic soit autorisé. Les capacités de pare-feu intégrées de Red Hat Enterprise Linux peuvent être utilisées pour restreindre l'accès si nécessaire.

2. Définir le sous-réseau public du VPC à *running*.

3. Sélectionner un IP statique.

4. Configurer le champ **`User Data`**. Les paramètres configurables sont disponibles ici : Section 21.4.1, « Paramètres de configuration permanente », Section 21.4.2, « Paramètres de scripts personnalisés ».

**Exemple 21.4. Exemple de champ de données d'utilisateur**

L'exemple montre le champ de données utilisateur d'un domaine géré de JBoss Enterprise Application Platform 6 non clusterisé. Le mot de passe de l'utilisateur **`admin`** a été défini à **`admin`**.

```
## password that will be used by slave host controllers to connect
to the domain controller
JBOSSAS_ADMIN_PASSWORD=admin

## subnet prefix this machine is connected to
SUBNET=10.0.0.

#### to run the example no modifications below should be needed
####
JBOSS_DOMAIN_CONTROLLER=true
PORTS_ALLOWED="9999 9990 9443"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app
name>.war -O /usr/share/java/jboss-ec2-eap-applications/<app
name>.war

## Create a file of CLI commands to be executed after starting the
server
cat> $USER_CLI_COMMANDS << "EOC"

# Add the modcluster subsystem to the default profile to set up a
proxy
/profile=default/subsystem=web/connector=ajp:add(name=ajp,protocol
=AJP/1.3,scheme=http,socket-binding=ajp)
/:composite(steps=[ {"operation" => "add", "address" => [
("profile" => "default"), ("subsystem" => "modcluster") ] },{
"operation" => "add", "address" => [ ("profile" => "default"),
("subsystem" => "modcluster"), ("mod-cluster-config" =>
"configuration") ], "advertise" => "false", "proxy-list" =>
"${jboss.modcluster.proxyList}", "connector" => "ajp"}, {
"operation" => "add", "address" => [ ("profile" => "default"),
("subsystem" => "modcluster"), ("mod-cluster-config" =>
"configuration"), ("dynamic-load-provider" => "configuration") ]},
```

```
{ "operation" => "add", "address" => [ ("profile" => "default"),
("subsystem" => "modcluster"), ("mod-cluster-config" =>
"configuration"), ("dynamic-load-provider" => "configuration"),
("load-metric" => "busyness")], "type" => "busyness"} ])

# Deploy the sample application from the local filesystem
deploy /usr/share/java/jboss-ec2-eap-samples/hello.war --server-
groups=main-server-group
EOC

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
   echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

5. **Pour les instances de production**
   Pour une instance de production, ajouter la ligne suivante sous la ligne **`USER_SCRIPT`** du champ **`User Data`** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

   ```
   yum -y update
   ```

   **NOTE**

   **`yum -y update`** soit être exécuté régulièrement, pour appliquer les correctifs de sécurité et les améliorations.

6. Lancement de l'instance AMI Red Hat

**Résultat**

Un domaine géré non clusterisé de JBoss Enterprise Application Platform 6 a été configuré, et lancé sur une AMI Red Hat.

Report a bug

#### 21.2.2.3.2. Lancer une ou plusieurs instances pour qu'elles servent en tant que contrôleurs d'hôtes

**Résumé**

Cette rubrique couvre les étapes requises pour lancer une ou plusieurs instances de JBoss Enterprise Application Platform 6 en tant que contrôleurs d'hôtes non clusterisée sur Red Hat AMI (Amazon Machine Image).

**Prérequis**

- Configurer et lancer le contrôleur de domaine non clusterisé. Consulter Section 21.2.2.3.1, « Lancer une instance pour qu'elle serve en tant que Contrôleur de domaine » .

**Procédure 21.4. Lancer les contrôleurs d'hôte**

Pour chaque instance que vous souhaitez créer, répétez les étapes suivantes :

1. Sélectionnez une AMI.
2. Définir le nombre d'instances que vous souhaitez (le nombre de contrôleurs hôtes esclaves)
3. Sélectionner le VPC et le type d'instance.
4. Cliquer sur le Groupe de sécurité.
5. Veillez à ce que tout le trafic en provenance du sous-système de JBoss Enterprise Application Platform soit autorisé.
6. Définir les autres restriction suivant les besoins.
7. Ajouter ce qui suit dans le champ User Data :

```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## host controller setup
JBOSS_DOMAIN_MASTER_ADDRESS=10.0.0.5
JBOSS_HOST_PASSWORD=<password for slave host controllers>

## subnet prefix this machine is connected to
SUBNET=10.0.1.

#### to run the example no modifications below should be needed ####
JBOSS_HOST_USERNAME=admin
PORTS_ALLOWED="1024:65535"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Server instance configuration
sed -i "s/other-server-group/main-server-group/"
$JBOSS_CONFIG_DIR/$JBOSS_HOST_CONFIG

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
    echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

8. **Pour les instances de production**
   Pour une instance de production, ajouter la ligne suivante sous la ligne **`USER_SCRIPT`** du champ **`User Data`** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

```
yum -y update
```

**NOTE**

**`yum -y update`** soit être exécuté régulièrement, pour appliquer les correctifs de sécurité et les améliorations.

9. Lancement de l'instance AMI Red Hat

**Résultat**

Les contrôleurs d'hôtes non clusterisés de JBoss Enterprise Application Platform 6 ont été configurés, et lancés sur une AMI Red Hat.

Report a bug

#### 21.2.2.3.3. Tester le domaine géré de JBoss Enterprise Application Platform 6 non clusterisée

**Résumé**

Cette rubrique couvre les étapes requises pour lancer un domaine géré non clusterisé de JBoss Enterprise Application Platform sur un AMI Red Hat (Amazon Machine Image)

Pour tester le domaine géré, vous devrez connaître les adresses IP élastiques de Apache HTTPD et du contrôleur de domaines de JBoss Enterprise Application Platform à la fois.

**Prérequis**

- Configurer et lancer le contrôleur de domaine. Consulter Section 21.2.2.3.1, « Lancer une instance pour qu'elle serve en tant que Contrôleur de domaine » .
- Configurer et lancer les contrôleurs d'hôte. Consulter Section 21.2.2.3.2, « Lancer une ou plusieurs instances pour qu'elles servent en tant que contrôleurs d'hôtes » .

**Procédure 21.5. Tester le Serveur Web**

- Naviguer dans **`http://ELASTIC_IP_OF_APACHE_HTTPD`** avec un navigateur pour confirmer que le serveur web exécute avec succès.

**Procédure 21.6. Tester le contrôleur de données**

1. Naviguer dans **`http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console`**
2. Connectez-vous en utilisant le nom d'utilisateur **`admin`** et le mot de passe spécifiés dans le champ Données d'utilisateur pour le Contrôleur de domaine et la Page d'accueil de la console admin du domaine géré s'affichera (**`http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console/App.html#server-instances`**).
3. Cliquer sur l'étiquette du serveur en haut et à droite de l'écran, et sélectionner un contrôleur d'hôtes dans le menu déroulant d'Hôtes en haut et à gauche de l'écran.
4. Vérifier que chaque contrôleur d'hôte ait deux configurations de serveurs nommées respectivement **`server-one`** et **`server-two`** qui appartiennent toutes deux au **`main-server-group`**.

5. Déconnectez-vous de la console admin de JBoss Enterprise Application Platform.

**Procédure 21.7. Tester les contrôleurs d'hôte**

1. Naviguer dans **`http://ELASTIC_IP_OF_APACHE_HTTPD/hello`** pour tester que l'exemple d'application exécute. Le texte **`Hello World!`** devrait apparaître dans le navigateur.

   Si le texte n'est pas visible, voir : Section 18.5.1, "About Troubleshooting Amazon EC2".

2. Connectez-vous à l'instance d'Apache HTTPD :

   ```
   $ ssh -L7654:localhost:7654 ELASTIC_IP_OF_APACHE_HTTPD
   ```

3. Naviguer dans **`http://localhost:7654/mod_cluster-manager`** pour confirmer que toutes les instances exécutent correctement.

**Résultat**

Le serveur web de JBoss Enterprise Application Platform 6, le contrôleur de domaine, et les contrôleurs hôtes exécutent correctement sur un AMI Red Hat.

Report a bug

## 21.2.3. JBoss Enterprise Application Platform 6 clusterisée

### 21.2.3.1. Instances clusterisées

Une instance clusterisée est une instance Amazon EC2 exécutant sur JBoss Enterprise Application Platform 6 avec le clustering activé. Une autre instance exécutant sur Apache HTTPD agira en tant que proxy pour les instances dans le cluster.

Les AMI de JBoss Enterprise Application Platform 6 comprennent deux fichiers de configuration à utiliser dans les instances en cluster, **`standalone-ec2-ha.xml`** et **`standalone-mod_cluster-ec2-ha.xml`**. Chacun de ces fichiers de configuration fournit du clustering sans multidiffusion car Amazon EC2 ne prend pas en charge la multidiffusion. Cela se fait par monodiffusion TCP pour les communications de clusters et S3_PING comme protocole de découverte. La configuration **`autonome-mod_cluster-ec2-ha.xml`** fournit également un enregistrement simple par les proxys de mod_cluster.

De même, le fichier de configuration **`domain-ec2.xml`** fournit deux profils à utiliser dans les domaines gérés clusterisés : ec2-ha, et mod_cluster-ec2-ha.

Report a bug

### 21.2.3.2. Créer une instance de base de données de service de bases de données relationnelles.

**Résumé**

Cette rubrique couvre les étapes nécessaires pour créer une instance de base de données de service de bases de données relationnelles, en utilisant MySQL comme exemple.

**AVERTISSEMENT**

Il est hautement conseillé que les fonctionnalités de sauvegarde et de maintenance demeurent actives dans les environnements de production.

**IMPORTANT**

Il est de bonne pratique de créer des paires séparées utilisateur/mot de passe pour chaque application qui accède à la base de données. Régler les options de configuration selon les besoins de votre application.

**Procédure 21.8. Créer une instance de base de données de service de bases de données relationnelles.**

1. Cliquer sur le **RDS** de la console AWS.
2. Abonnez-vous au service si nécessaire.
3. Cliquer sur **Launch DB instance**.
4. Cliquer sur **MySQL**.
    a. Sélectionner une version, comme **5.5.12**.
    b. Sélectionner **small instance**.
    c. Veillez à ce que **Multi-AZ Deployment** et **Auto upgrade** soient désactivés: **off**.
    d. Définir **Storage** à **5GB**.
    e. Définir le nom d'utilisateur et le mot de passe de l'administrateur de système et cliquer sur le bouton **Next**.
    f. Sélectionner un nom de base de données à créer avec l'instance, et cliquer sur **Next**.
    g. Désactiver les back-ups et la maintenance, si nécessaire.
    h. Confirmer les paramètres.

**Résultat**

La base de données est alors créée. Elle s'initialisera et sera prête à l'utilisation dans quelques minutes.

Report a bug

### 21.2.3.3. Clouds privés virtuels

Amazon VPC (Amazon Virtual Private Cloud) est une fonctionnalité d'AWS (Amazon Web Service) qui vous permet d'isoler un ensemble de ressources AWS dans un réseau privé. La topologie et la configuration de ce réseau privé peut être personnalisé.

Voir le site Amazon Virtual Private Cloud pour obtenir plus d'informations http://aws.amazon.com/vpc/.

Report a bug

### 21.2.3.4. Créer un VPC (Virtual Private Cloud)

**Résumé**

Cette rubrique décrit les étapes requises pour créer un cloud privé virtuel, en prenant comme exemple une base de données externe au VPC comme exemple. Vos stratégies de sécurité peuvent exiger la connexion à la base de données à crypter. Veuillez consulter *RDS FAQ* d'Amazon pour plus d'informations sur le cryptage des connexions de base de données.

**IMPORTANT**

Un VPC est recommandé pour une installation de cluster dans JBoss Enterprise Application Platform car cela simplifie grandement une communication sécurisée entre les nœuds du cluster, un Server JON et le proxy mod_cluster. Sans un VPC, ces canaux de communication doivent être chiffrés et authentifiés.

Pour obtenir des informations détaillées sur la façon de configurer SSL, voir: Section 10.11.1, « Implémentation du cryptage SSL pour le serveur de JBoss Enterprise Application Platform. ».

1. Aller dans l'onglet VPC de la console AWS.
2. Abonnez-vous au service si nécessaire.
3. Cliquer sur "**`Create new VPC`**".
4. Séelctionner un VPC avec un sous-système public et un privé.
    a. Définir le sous-système public à **`10.0.0.0/24`**.
    b. Définir le sous-système privé à **`10.0.1.0/24`**.
5. Aller dans **`Elastic IPs`**.
6. Créer un IP élastique pour que l'instance mod_cluster proxy/NAT puisse l'utiliser.
7. Aller dans **`Security groups`** et créer un groupe de sécurité pour autoriser le trafic entrant et sortant.
8. Aller sur les ACL de réseau
    a. Créer un ACL pour autoriser le trafic entrant et sortant.
    b. Créer un ACL pour autoriser le trafic vers et depuis les ports TCP **`22`**, **`8009`**, **`8080`**, **`8443`**, **`9443`**, **`9990`** et **`16163`** uniquement.

**Résultat**

Le Cloud privé virtuel (VPC) a été créé.

Report a bug

### 21.2.3.5. Lancer une instance Apache HTTPD pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC

#### Résumé

Cette section couvre toutes les étapes requises pour lancer une instance Apache HTTPD qui puisse servir de proxy mod_cluster et d'instance NAT au Virtuel Private Cloud.

#### Prérequis

- Section 21.2.3.2, « Créer une instance de base de données de service de bases de données relationnelles. ».
- Section 21.2.3.4, « Créer un VPC (Virtual Private Cloud) »

#### Procédure 21.9. Lancer une instance Apache HTTPD pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC

1. Créer un IP élastique pour cette instance.
2. Sélectionnez un AMI.
3. Allez dans le **`Security Group`** et autoriser tout le trafic (utiliser les capacités de pare-feu intégrées de Red Hat Enterprise Linux pour restreindre l'accès si nécessaire).
4. Choisir "**`running`**" dans le sous-système public du VPC.
5. Sélectionner un IP statique (comme par ex **`10.0.0.4`**).
6. Mettez ce qui suit dans le champ **`User Data:`** :

```
JBOSSCONF=disabled

cat > $USER_SCRIPT << "EOS"

echo 1 > /proc/sys/net/ipv4/ip_forward
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
echo 0 > /proc/sys/net/ipv4/conf/eth0/rp_filter

iptables -I INPUT 4 -s 10.0.1.0/24 -p tcp --dport 7654 -j ACCEPT
iptables -I INPUT 4 -p tcp --dport 80 -j ACCEPT

iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -I FORWARD -s 10.0.1.0/24 -j ACCEPT
iptables -t nat -A POSTROUTING -o eth0 ! -s 10.0.0.4 -j MASQUERADE

# balancer module incompatible with mod_cluster
sed -i -e 's/LoadModule proxy_balancer_module/#\0/'
/etc/httpd/conf/httpd.conf

cat > /etc/httpd/conf.d/mod_cluster.conf << "EOF"
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
```

```
LoadModule advertise_module modules/mod_advertise.so

Listen 7654

# workaround JBPAPP-4557
MemManagerFile /var/cache/mod_proxy/manager

<VirtualHost *:7654>
   <Location /mod_cluster-manager>
      SetHandler mod_cluster-manager
      Order deny,allow
      Deny from all
      Allow from 127.0.0.1
   </Location>

   <Location />
      Order deny,allow
      Deny from all
      Allow from 10.
      Allow from 127.0.0.1
   </Location>

   KeepAliveTimeout 60
   MaxKeepAliveRequests 0
   ManagerBalancerName mycluster
   ServerAdvertise Off
   EnableMCPMReceive On
</VirtualHost>
EOF

echo "`hostname | sed -e 's/ip-//' -e 'y/-/./'`        `hostname`"
>> /etc/hosts

semanage port -a -t http_port_t -p tcp 7654 #add port in the apache
port list for the below to work
setsebool -P httpd_can_network_relay 1 #for mod_proxy_cluster to
work
chcon -t httpd_config_t -u system_u
/etc/httpd/conf.d/mod_cluster.conf

#### Uncomment the following line when launching a managed domain
####
# setsebool -P httpd_can_network_connect 1

service httpd start

EOS
```

7. Décochez la case Amazon EC2 cloud source/destination pour cette instance pour qu'elle puisse agir en tant que router.

   a. Cliquer à droite sur l'instance Apache HTTPD et sélectionner "**Change Source/Dest check**".

   b. Cliquer sur **Yes, Disable**.

8. Créer un IP élastique pour cette instance.

**Résultat**

L'instance Apache HTTPD aura été lancée avec succès.

Report a bug

### 21.2.3.6. Configurer le routage par défaut du sous-système privé VPC

**Résumé**

Cette rubrique couvre les étapes requises pour configurer le routage par défaut du sous-système privé VPC. Les noeuds de cluster de JBoss Enterprise Application Platform exécuteront dans le sous-système privé du VPC, mais les noeuds de cluster ont besoin d'un accès internet pour la connectivité S3. Un routage par défaut doit être défini pour aller dans l'instance NAT.

**Procédure 21.10. Configurer le routage par défaut du sous-système privé VPC**

1. Naviguer dans l'instance Apache HTTPD de la console Amazon AWS.
2. Naviguer dans **VPC** → **tables de routage**.
3. Cliquer sur le tableau de routage utilisé par le sous-système privé.
4. Dans le champ de nouveau routage, saisir **`0.0.0.0/0`**.
5. Cliquer sur "**`Select a target`**".
6. Sélectionner "**`Enter Instance ID`**".
7. Choisir l'ID de l'instance Apache HTTPD en cours d'exécution.

**Résultat**

Le routage par défaut a été configurée correctement pour le sous-système VPC.

Report a bug

### 21.2.3.7. IAM (Identity and Access Management)

IAM (Identity and Access Management) fournit une sécurité configurable pour vos ressources AWS. IAM peut être configuré pour utiliser les comptes créés dans IAM ou pour fournir une fédération d'identité entre IAM et vos propres services d'identité.

Consulter le site web AWS Identity and Access Management pour plus d'informations http://aws.amazon.com/iam/.

Report a bug

### 21.2.3.8. Configurer l'installation IAM

**Résumé**

Cette rubrique couvre les étapes de configuration requises pour installer IAM pour les instances de JBoss Enterprise Application Platform. Le protocole ***`S3_PING`*** utilise Bucket S3 pour découvrir d'autres membres du cluster. La version 3.0.x de **JGroups** a besoin d'un compte d'accès Amazon AWS et de clés secrètes pour s'authentifier dans le service S3.

Il y a un risque de sécurité à entrer vos informations d'identification du compte principal dans le domaine de l'utilisateur des données, de les stocker en ligne ou dans un AMI. Pour contourner cela, un compte distinct peut être créé en utilisant la fonction Amazon IAM qui donnerait seulement accès à un seul compartiment de S3.

**Procédure 21.11. Configurer l'installation IAM**

1. Aller dans l'onglet IAM de la Console AWS.
2. Cliquer sur **`users`** (utilisateurs).
3. Sélectionner **`Create New Users`** (Créer Nouveaux Utilisateurs).
4. Choisir un nom, et veillez à ce que l'option **`Generate an access key for each User`** (Générer une clé d'accès pour chaque utilisateur) soit cochée.
5. Sélectionner **`Download credentials`**, et les sauvegarder dans un emplacement sécurisé.
6. Fermer la fenêtre.
7. Cliquer sur un utilisateur nouvellement créé.
8. Prenez note de la valeur de **`User ARM`**. Cette valeur est requise pour configurer Bucket S3, et est documenté ici: Section 21.2.3.10, « Configurer l'installation S3 Bucket ».

**Résultat**

Le compte IAM a été créé avec succès.

Report a bug

### 21.2.3.9. S3 Bucket

Les S3 Buckets représentent une unité de stockage d'organisation de base d'Amazon S3 (Amazon Simple Storage System). Une Bucket peut stocker un certain nombre d'objets arbitraires et doit posséder un nom unique pour l'identifier avec Amazon S3.

Consulter le site web Amazon S3 pour obtenir plus d'informations, http://aws.amazon.com/s3/.

Report a bug

### 21.2.3.10. Configurer l'installation S3 Bucket

**Résumé**

Cette rubrique couvre les étapes nécessaires pour configurer une nouvelle S3 Bucket.

**Prérequis**

- Section 21.2.3.8, « Configurer l'installation IAM ».

**Procédure 21.12. Configurer l'installation S3 Bucket**

1. Ouvrir l'onglet **S3** dans la console AWS.
2. Cliquer sur **`Créer Bucket`**.

3. Choisir un nom pour la Bucket et cliquer sur **`Create`** (Créer).

**NOTE**

Les noms de Bucket sont uniques dans tout S3. Les noms ne peuvent pas être réutilisés.

4. Cliquer à droite sur la nouvelle Bucket et sélectionner **`Properties`** (propriétés).

5. Cliquer sur **`Add bucket policy`** (ajouter la règle de bucket) dans l'onglet de permissions.

6. Cliquer sur **`New policy`** (Nouvelle police) pour ouvrir l'assistant de création de police.

   a. Copier le texte suivant dans la nouvelle police, en remplaçant **`arn:aws:iam::05555555555:user/jbosscluster*`** par la valeur définie ici : Section 21.2.3.8, « Configurer l'installation IAM ». Changer les deux instances de **`clusterbucket123`** au nom de Bucket défini dans l'étape 3 de cette procédure.

```
{
    "Version": "2008-10-17",
    "Id": "Policy1312228794320",
    "Statement": [
        {
            "Sid": "Stmt1312228781799",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::055555555555:user/jbosscluster"
                ]
            },
            "Action": [
                "s3:ListBucketVersions",
                "s3:GetObjectVersion",
                "s3:ListBucket",
                "s3:PutBucketVersioning",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion",
                "s3:GetObject",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:PutObject",
                "s3:GetBucketVersioning"
            ],
            "Resource": [
                "arn:aws:s3:::clusterbucket123/*",
                "arn:aws:s3:::clusterbucket123"
            ]
        }
    ]
}
```

**Résultat**

Une nouvelle Bucket a maintenant été créée, et configurée avec succès.

Report a bug

### 21.2.3.11. Instances clusterisées

#### 21.2.3.11.1. Lancer les AMI JBoss Enterprise Application Platform 6 clusterisés

**Résumé**

Cette rubrique couvre les étapes requises pour lancer les AMIS JBoss Enterprise Application Platform 6.

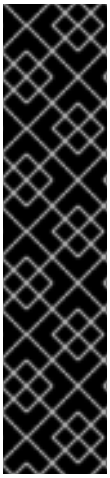
**Prérequis**

- Section 21.2.3.2, « Créer une instance de base de données de service de bases de données relationnelles. ».
- Section 21.2.3.4, « Créer un VPC (Virtual Private Cloud) ».
- Section 21.2.3.5, « Lancer une instance Apache HTTPD pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC ».
- Section 21.2.3.6, « Configurer le routage par défaut du sous-système privé VPC ».
- Section 21.2.3.8, « Configurer l'installation IAM ».
- Section 21.2.3.10, « Configurer l'installation S3 Bucket ».

**AVERTISSEMENT**

Exécuter le cluster JBoss Enterprise Application Platform dans un sous-système avec un masque de réseau inférieur à 24 bits ou bien fractionner plusieurs sous-systèmes compliquent l'obtention d'un ID homologue unique pour chaque membre du cluster.

Voir la variable ***JBOSS_CLUSTER_ID*** pour obtenir des informations sur la façon d'effectuer ce travail de configuration de façon fiable : Section 21.4.1, « Paramètres de configuration permanente ».



**IMPORTANT**

La fonctionnalité de Amazon EC2 AutoScaling peut être utilisée avec des nœuds de cluster JBoss Enterprise Application Platform. Cependant, s'assurer que c'est testé **avant** le déploiement. Vous devez vous assurer que vos charges de travail particulières sont à l'échelle du nombre de nœuds désirés, et que la performance répond à vos besoins avant d'envisager d'utiliser le type d'instance (des types d'instances différentes reçoivent une part de ressources cloud EC2 différentes).

De plus, l'emplacement de l'instance et l'utilisation machine/RDS réseau/stockage/machine hôte/RDS peuvent affecter la performance d'un cluster. Tester avec vos charges réelles attendues pour essayer de pallier à l'avance aux conditions inattendues.

**AVERTISSEMENT**

L'action *scale-down* Amazon EC2 termine les nœuds sans élégance, et, comme certaines transactions pourraient être interrompues, les autres nœuds de cluster (et équilibreurs de charge) auront besoin de temps pour basculer. Cela est susceptible d'influencer l'expérience des utilisateurs de votre application.

Il est recommandé que vous réduisiez le cluster d'applications manuellement en désactivant le serveur de l'interface de gestion du mod_cluster jusqu'à ce que toutes les sessions traitées soient complétées, ou bien que vous fermiez l'instance JBoss Enterprise Application Platform avec grâce (l'accès SSH vers l'instance ou JON peuvent être utilisés).

Vérifier que votre procédure choisie de réduction n'ait pas d'effets négatifs sur l'expérience utilisateur. Il est possible que vous ayez besoin de prendre des mesures supplémentaires pour certaines charges de travail, équilibrages de charge ou installations.

**Procédure 21.13. Lancer les AMI JBoss Enterprise Application Platform 6 clusterisés**

1. Sélectionnez un AMI.
2. Définir le nombre d'instances voulues (la taille du cluster).
3. Sélectionner le VPC et le type d'instance.
4. Cliquer sur le groupe **`Security Group`**.
5. Veillez à ce que tout le trafic en provenance du sous-système de cluster JBoss Enterprise Application Platform soit autorisé.
6. Définir les autres restriction suivant les besoins.
7. Ajouter ce qui suit dans le champ **`User Data`** :

**Exemple 21.5. Exemple de champ de données utilisateur**

```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## clustering setup
JBOSS_JGROUPS_S3_PING_SECRET_ACCESS_KEY=<your secret key>
JBOSS_JGROUPS_S3_PING_ACCESS_KEY=<your access key>
JBOSS_JGROUPS_S3_PING_BUCKET=<your bucket name>

## password to access admin console
JBOSSAS_ADMIN_PASSWORD=<your password for opening admin console>

## database credentials configuration
JAVA_OPTS="$JAVA_OPTS -
Ddb.host=instancename.something.rds.amazonaws.com -
```

```
Ddb.database=mydatabase -Ddb.user=<user> -Ddb.passwd=<pass>"

## subnet prefix this machine is connected to
SUBNET=10.0.1.

#### to run the example no modifications below should be needed
####
PORTS_ALLOWED="1024:65535"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app
name>.war -O /usr/share/java/jboss-ec2-eap-applications/<app
name>.war

## install the JDBC driver as a core module
yum -y install mysql-connector-java
mkdir -p /usr/share/jbossas/modules/com/mysql/main
cp -v /usr/share/java/mysql-connector-java-*.jar
/usr/share/jbossas/modules/com/mysql/main/mysql-connector-java.jar

cat > /usr/share/jbossas/modules/com/mysql/main/module.xml <<"EOM"

<module xmlns="urn:jboss:module:1.0" name="com.mysql">
   <resources>
      <resource-root path="mysql-connector-java.jar"/>
   </resources>
   <dependencies>
      <module name="javax.api"/>
   </dependencies>
</module>
EOM

cat > $USER_CLI_COMMANDS << "EOC"
## Deploy sample application from local filesystem
deploy --force /usr/share/java/jboss-ec2-eap-samples/cluster-
demo.war

## ExampleDS configuration for MySQL database
data-source remove --name=ExampleDS
/subsystem=datasources/jdbc-driver=mysql:add(driver-
name="mysql",driver-module-name="com.mysql")
data-source add --name=ExampleDS --connection-
url="jdbc:mysql://${db.host}:3306/${db.database}" --jndi-
name=java:jboss/datasources/ExampleDS --driver-name=mysql --user-
name="${db.user}" --password="${db.passwd}"
/subsystem=datasources/data-source=ExampleDS:enable
/subsystem=datasources/data-source=ExampleDS:test-connection-in-
pool
EOC

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
```

```
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
   echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

**Résultat**

Les AMI JBoss Enterprise Application Platform 6 ont été configurées et lancées avec succès.

Report a bug

#### 21.2.3.11.2. Tester l'instance de JBoss Enterprise Application Platform 6 clusterisée

**Résumé**

Cette rubrique couvre les étapes pour s'assurer que les instances de la plateforme clusterisée de JBoss Enterprise Application Platform 6 exécutent correctement.

**Procédure 21.14. Tester l'instance clusterisée**

1. Naviguez dans http://ELASTIC_IP_OF_APACHE_HTTPD pour confirmer que le serveur web exécute correctement.
2. **Tester les noeuds clusterisés**
   a. Naviguez dans http://ELASTIC_IP_OF_APACHE_HTTPD/cluster-demo/put.jsp.
   b. Vérifier que l'un des noeuds de cluster journalise le message suivant :

   ```
   Putting date now
   ```

   c. Stop the cluster node that logged the message in the previous step.
   d. Naviguez dans http://ELASTIC_IP_OF_APACHE_HTTPD/cluster-demo/get.jsp.
   e. Vérifier que l'heure indiquée est la même que l'heure PUT (mise) par **`put.jsp`** à l'étape 2-a.
   f. Vérifier que l'un des noeuds de cluster en cours d'exécution journalise le message suivant :

   ```
   Obtenir la date maintenant
   ```

   g. Démarrer à nouveau le noeud clusterisé qui est arrêté.
   h. Connectez-vous à l'instance Apache HTTPD :

   ```
   ssh -L7654:localhost:7654 <ELASTIC_IP_OF_APACHE_HTTPD>
   ```

   i. Naviguez dans http://localhost:7654/mod_cluster-manager pour confirmer que toutes les instances exécutent correctement.

**Résultat**

L'instance clustrisée de JBoss Enterprise Application Platform 6 a été testée, et il a été confirmé qu'elle exécutait correctement.

Report a bug

### 21.2.3.12. Domaines gérés clusterisés

#### 21.2.3.12.1. Lancer une instance pour qu'elle serve en tant que Contrôleur de domaine de cluster

**Résumé**

Cette rubrique couvre les étapes requises pour lancer un domaine géré clusterisé de JBoss Enterprise Application Platform sur une AMI Red Hat (Amazon Machine Image)

**Prérequis**

- Pour une AMI Red Hat qui convient, consulter Section 21.1.6, « AMI Red Hat pris en charge » .
- Section 21.2.3.4, « Créer un VPC (Virtual Private Cloud) »
- Section 21.2.3.5, « Lancer une instance Apache HTTPD pour qu'elle serve en tant que proxy de mod_cluster et d'instance NAT pour le VPC »
- Section 21.2.3.6, « Configurer le routage par défaut du sous-système privé VPC »

**Procédure 21.15. Lancer un contrôleur de domaine clusterisé**

1. Créer un IP élastique pour cette instance.
2. Sélectionner une AMI.
3. Allez dans le Groupe de sécurité et autoriser tout le trafic (utiliser les capacités de pare-feu intégrées de Red Hat Enterprise Linux pour restreindre l'accès si nécessaire).
4. Choisir "running" dans le sous-système public du VPC.
5. Sélectionner un IP statique (comme par ex 10.0.0.5).
6. Mettez ce qui suit dans le champ User Data :

```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654

## password that will be used by slave host controllers to connect
to the domain controller
JBOSSAS_ADMIN_PASSWORD=<password for slave host controllers>

## subnet prefix this machine is connected to
SUBNET=10.0.0.

#### to run the example no modifications below should be needed ####
JBOSS_DOMAIN_CONTROLLER=true
PORTS_ALLOWED="9999 9990 9443"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
```

```
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Get the application to be deployed from an Internet URL
# mkdir -p /usr/share/java/jboss-ec2-eap-applications
# wget https://<your secure storage hostname>/<path>/<app name>.war
-O /usr/share/java/jboss-ec2-eap-applications/<app name>.war

## Install the JDBC driver as a core module
yum -y install mysql-connector-java
mkdir -p /usr/share/jbossas/modules/com/mysql/main
cp -v /usr/share/java/mysql-connector-java-*.jar
/usr/share/jbossas/modules/com/mysql/main/mysql-connector-java.jar

cat > /usr/share/jbossas/modules/com/mysql/main/module.xml <<"EOM"

<module xmlns="urn:jboss:module:1.0" name="com.mysql">
   <resources>
      <resource-root path="mysql-connector-java.jar"/>
   </resources>
   <dependencies>
      <module name="javax.api"/>
   </dependencies>
</module>
EOM

cat > $USER_CLI_COMMANDS << "EOC"
## Deploy the sample application from the local filesystem
deploy /usr/share/java/jboss-ec2-eap-samples/cluster-demo.war --
server-groups=other-server-group

## ExampleDS configuration for MySQL database
data-source --profile=mod_cluster-ec2-ha remove --name=ExampleDS
/profile=mod_cluster-ec2-ha/subsystem=datasources/jdbc-
driver=mysql:add(driver-name="mysql",driver-module-name="com.mysql")
data-source --profile=mod_cluster-ec2-ha add --name=ExampleDS --
connection-url="jdbc:mysql://${db.host}:3306/${db.database}" --jndi-
name=java:jboss/datasources/ExampleDS --driver-name=mysql --user-
name="${db.user}" --password="${db.passwd}"
/profile=mod_cluster-ec2-ha/subsystem=datasources/data-
source=ExampleDS:enable
EOC

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
   echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

7. **Pour les instances de production**

Pour une instance de production, ajouter la ligne suivante sous la ligne **USER_SCRIPT** du champ **User Data** pour veiller à ce que les mises à jour de sécurité soient appliquées au démarrage.

```
yum -y update
```

**NOTE**

**yum -y update** doit être exécutée régulièrement, pour appliquer les correctifs et les améliorations.

8. Lancer l'instance AMI de Red Hat

**Résultat**

Un domaine géré de JBoss Enterprise Application Platform 6 est configuré et lancé sur une AMI Red Hat.

Report a bug

#### 21.2.3.12.2. Lancer une ou plusieurs instances pour qu'elles servent en tant que contrôleurs d'hôtes de cluster

**Résumé**

Cette rubrique couvre les étapes requises pour lancer une ou plusieurs instances de JBoss Enterprise Application Platform 6 en tant que contrôleurs d'hôtes clusterisés sur Red Hat AMI (Amazon Machine Image).

**Prérequis**

- Configurer et lancer le contrôleur de domaine clusterisé. Consulter Section 21.2.3.12.1, « Lancer une instance pour qu'elle serve en tant que Contrôleur de domaine de cluster » .

**Procédure 21.16. Lancer les contrôleurs d'hôte**

Pour chaque instance que vous souhaitez créer, répétez les étapes suivantes :

1. Sélectionner un AMI.
2. Définir le nombre d'instances que vous souhaitez (le nombre de contrôleurs hôtes esclaves)
3. Sélectionner le VPC et le type d'instance.
4. Cliquer sur le Groupe de sécurité.
5. Veillez à ce que tout le trafic du sous-système du cluster de JBoss Enterprise Application Platform soit autorisé.
6. Définir les autres restriction suivant les besoins.
7. Ajouter ce qui suit dans le champ User Data :

```
## mod cluster proxy addresses
MOD_CLUSTER_PROXY_LIST=10.0.0.4:7654
```

```
## clustering setup
JBOSS_JGROUPS_S3_PING_SECRET_ACCESS_KEY=<your secret key>
JBOSS_JGROUPS_S3_PING_ACCESS_KEY=<your access key>
JBOSS_JGROUPS_S3_PING_BUCKET=<your bucket name>

## host controller setup
JBOSS_DOMAIN_MASTER_ADDRESS=10.0.0.5
JBOSS_HOST_PASSWORD=<password for slave host controllers>

## database credentials configuration
JAVA_OPTS="$JAVA_OPTS -
Ddb.host=instancename.something.rds.amazonaws.com -
Ddb.database=mydatabase -Ddb.user=<user> -Ddb.passwd=<pass>"

## subnet prefix this machine is connected to
SUBNET=10.0.1.

#### to run the example no modifications below should be needed ####
JBOSS_HOST_USERNAME=admin
PORTS_ALLOWED="1024:65535"
JBOSS_IP=`hostname | sed -e 's/ip-//' -e 'y/-/./'` #listen on
public/private EC2 IP address

cat > $USER_SCRIPT << "EOF"
## Server instance configuration
sed -i "s/main-server-group/other-server-group/"
$JBOSS_CONFIG_DIR/$JBOSS_HOST_CONFIG

## install the JDBC driver as a core module
yum -y install mysql-connector-java
mkdir -p /usr/share/jbossas/modules/com/mysql/main
cp -v /usr/share/java/mysql-connector-java-*.jar
/usr/share/jbossas/modules/com/mysql/main/mysql-connector-java.jar

cat > /usr/share/jbossas/modules/com/mysql/main/module.xml <<"EOM"

<module xmlns="urn:jboss:module:1.0" name="com.mysql">
   <resources>
      <resource-root path="mysql-connector-java.jar"/>
   </resources>
   <dependencies>
      <module name="javax.api"/>
   </dependencies>
</module>
EOM

## this will workaround the problem that in a VPC, instance
hostnames are not resolvable
echo -e "127.0.0.1\tlocalhost.localdomain localhost" > /etc/hosts
echo -e "::1\tlocalhost6.localdomain6 localhost6" >> /etc/hosts
for (( i=1 ; i<255 ; i++ )); do
   echo -e "$SUBNET$i\tip-${SUBNET//./-}$i" ;
done >> /etc/hosts

EOF
```

8. **Pour les instances de production**
   Pour une instance de production, ajouter la ligne suivante sous la ligne **USER_SCRIPT** du champ **User Data** pour que les mises à jour de sécurité s'appliquent à l'amorçage.

```
yum -y update
```

**NOTE**

**yum -y update** doit être exécuté régulièrement, pour appliquer les correctifs de sécurité et les améliorations.

9. Lancement de l'instance AMI Red Hat

**Résultat**

Les contrôleurs d'hôtes de cluster de JBoss Enterprise Application Platform 6 ont été configurés, et lancés sur un AMI Red Hat.

Report a bug

#### 21.2.3.12.3. Tester le domaine géré de JBoss Enterprise Application Platform 6 clusterisée

**Résumé**

Cette rubrique couvre les étapes requises pour lancer un domaine géré clusterisé de JBoss Enterprise Application Platform sur un AMI Red Hat (Amazon Machine Image)

Pour tester le domaine géré, vous devrez connaître les adresses IP élastiques de Apache HTTPD et du contrôleur de domaines de JBoss Enterprise Application Platform à la fois.

**Prérequis**

- Configurer et lancer le contrôleur de domaine clusterisé. Consulter Section 21.2.3.12.1, « Lancer une instance pour qu'elle serve en tant que Contrôleur de domaine de cluster » .
- Configurer et lancer les contrôleurs d'hôte de cluster. Consulter Section 21.2.3.12.2, « Lancer une ou plusieurs instances pour qu'elles servent en tant que contrôleurs d'hôtes de cluster » .

**Procédure 21.17. Tester l'instance Apache HTTPD**

- Naviguer dans **http://ELASTIC_IP_OF_APACHE_HTTPD** avec un navigateur pour confirmer que le serveur web exécute avec succès.

**Procédure 21.18. Tester le contrôleur de domaine**

1. Naviguer dans **http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console**
2. Connectez-vous en utilisant le nom d'utilisateur **admin** et le mot de passe spécifiés dans le champ Données d'utilisateur pour le Contrôleur de domaine. Une fois connecté, la Page d'accueil de la console admin d'un domaine géré s'affichera (**http://ELASTIC_IP_OF_DOMAIN_CONTROLLER:9990/console/App.html#server-instances**).

3. Cliquer sur l'étiquette du serveur en haut et à droite de l'écran, et sélectionner un contrôleur d'hôtes dans le menu déroulant d'Hôtes en haut et à gauche de l'écran.

4. Vérifier que ce contrôleur d'hôte ait deux configurations de serveurs nommées respectivement **`server-one`** et **`server-two`** et vérifier qu'elles appartiennent toutes deux à **`other-server-group`**.

**Procédure 21.19. Tester les contrôleurs d'hôte**

1. Naviguez dans **`http://ELASTIC_IP_OF_APACHE_HTTPD/cluster-demo/put.jsp`**.

2. Vérifier que l'un des contrôleurs d'hôte journalise le message suivant : **`Putting date now.`**

3. Stopper l'instance du serveur qui a journalisé le message dans l'étape précédente (voir la Section 2.8.3, Stopper un serveur par la Console de gestion).

4. Naviguez dans **`http://ELASTIC_IP_OF_APACHE_HTTPD/cluster-demo/get.jsp`**.

5. Vérifier que l'heure indiquée est la même que l'heure **`PUT`** (mise) par **`put.jsp`** à l'étape 2-a.

6. Vérifier que l'une des instances de serveur en cours d'exécution journalise le message suivant : **`Getting date now.`**

7. Re-démarrer l'instance du serveur arrêtée (voir la Section 2.8.3, Stopper un serveur par la Console de gestion).

8. Connectez-vous à l'instance d'Apache HTTPD.

```
$ ssh -L7654:localhost:7654 ELASTIC_IP_OF_APACHE_HTTPD
```

9. Naviguez dans **`http://localhost:7654/mod_cluster-manager`** pour confirmer que toutes les instances exécutent correctement.

**Résultat**

Le serveur web de JBoss Enterprise Application Platform 6, le contrôleur de domaine, et les contrôleurs hôtes exécutent correctement sur un AMI Red Hat.

Report a bug

## 21.3. ÉTABLIR UN MONITORING DANS JBOSS OPERATIONS NETWORK (JON)

### 21.3.1. AMI Monitoring

Une fois que vous avez votre application commerciale déployée dans une instance AMI configurée correctement, l'étape suivante est d'instaurer le monitoring de la plateforme avec JON (JBOss Operations Network).

Le serveur JON est généralement situé à l'intérieur d'un réseau d'entreprise, donc il est nécessaire d'établir une connexion sécurisée entre le serveur et chacun de ses agents. La création d'un réseau privé virtuel entre les deux points est la solution la plus courante, mais cela complique la configuration de réseau requise. Ce chapitre fournit des directives de configuration de réseau permettant d'établir la

communication entre l'agent de JON et le serveur JON. Pour plus d'informations sur la configuration, la gestion et l'utilisation, veuillez vous référer à la documentation officielle de Red Hat pour JBoss Operations Network (JON).

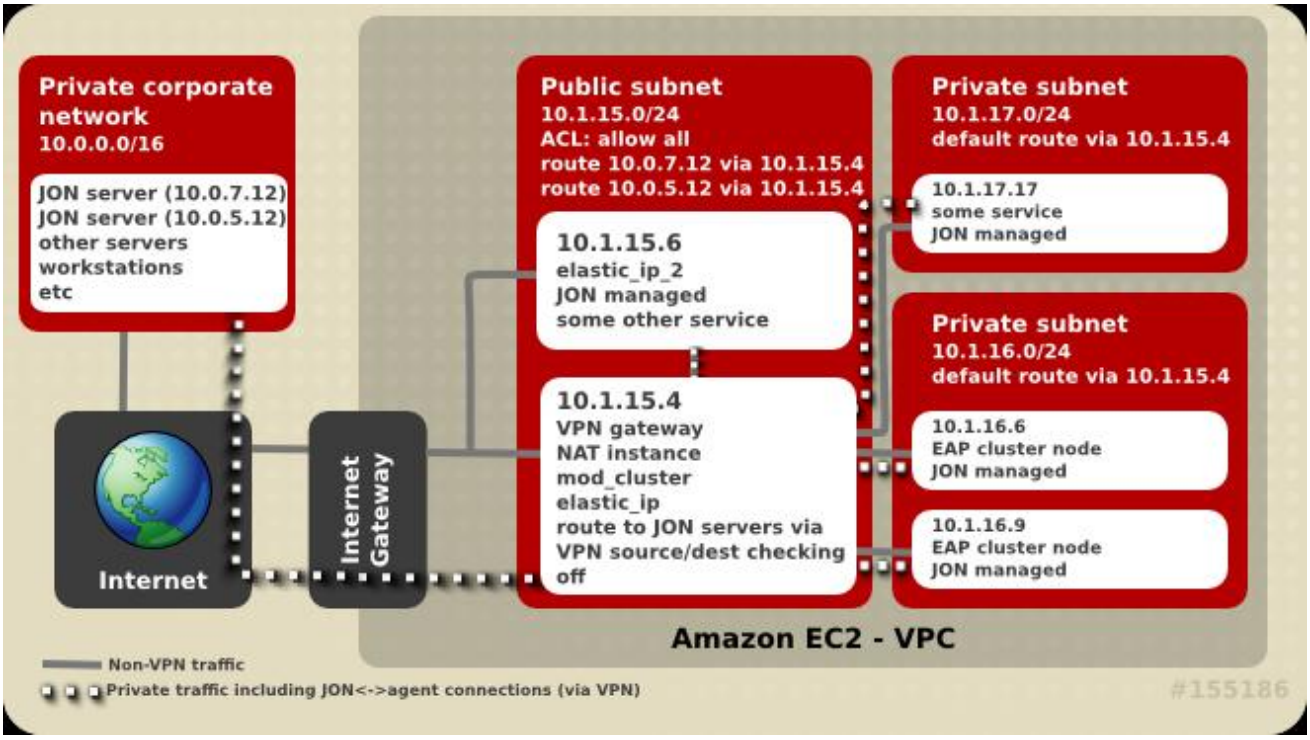



**Figure 21.1. La connectivité du serveur JON**

Report a bug

### 21.3.2. Prérequis de connectivité

L'inscription d'un agent JON avec ses serveurs requiert une communication bidirectionnelle entre l'agent et les serveurs. L'agent JON a besoin d'accéder au port 7080 sur tous les serveurs de JON, sauf dans le cas de SSL quand le port 7443 est utilisé. Chaque serveur de JON doit pouvoir accéder à chacun des agents connectés sur un hôte unique et son port homologue. Le port de l'agent est habituellement 16163.

S'il y a plusieurs serveurs JON clusterisés, veillez à ce que chaque agent puisse communiquer avec tous les serveurs dans le cluster JON via les paires IP et nom d'hôte comme configurés par la console d'administration du serveur JON. Le serveur JON utilisé par l'agent à enregistrer n'est sans doute pas le serveur qu'il essaie d'utiliser après l'initialisation.

Report a bug

### 21.3.3. Network Address Translation (NAT)

Une passerelle VPN entreprise agissant en mode routé simplifie grandement la configuration du réseau. Si toutefois votre gateway VPN entreprise opère en mode NAT, le serveur JON n'a pas de visibilité directe des agents. Dans ce cas, le réacheminement de port devra être configuré pour chaque agent.

Les configurations NAT VPN ont besoin d'un port sur la passerelle à transmettre à l'adresse de port de l'agent JON sur l'ordinateur dont il s'agit. L'agent JON doit également être configuré pour indiquer au serveur le numéro de port transféré et l'adresse IP. Vous trouverez de plus amples informations dans la description de **`rhq.communications.connector.*`** du fichier de configuration de **`agent-configuration.xml`**

Report a bug

### 21.3.4. Amazon EC2 et DNS

Les serveurs JON et les agents JON doivent être en mesure de résoudre les noms d'hôtes des uns et des autres. La résolution DNS est plus compliquée dans le cas d'une configuration VPN. Les serveurs connectés ont plusieurs options possibles. Une des options consiste à utiliser les serveurs DNS du réseau de l'entreprise ou Amazon EC2. Une autre option est d'utiliser une configuration divisée de DNS avec les serveurs DNS de l'entreprise utilisés pour résoudre les noms dans des domaines particuliers, et les serveurs d'Amazon EC2 DNS utilisés pour la résolution de tous les autres noms.

Report a bug

### 21.3.5. Routing dans EC2

Tous les serveurs de Amazon EC2 ont une fonctionnalité de routage **`source/destination checking`** activée par défaut. Cette fonction supprime tous les paquets envoyés au serveur qui ont une destination séparée de l'adresse IP de la machine. Si la solution VPN sélectionnée pour la connexion des agents au serveur JON inclut un routeur, cette fonctionnalité devra être désactivée pour le ou les serveur(s) agissant en tant que routeurs ou passerelles VPN. Ce paramètre de configuration est accessible via la console d'Amazon AWS. La désactivation de **`source/destination checking`** est également requise dans un Cloud privé virtuel (VPC).

Certaines configurations VPN renvoient le trafic internet général par le VPN entreprise par défaut. Il est conseillé de l'éviter car cela risque de ralentir et de rendre moins performante la configuration pour vos besoins particuliers.

Malgré que l'utilisation d'un schéma d'adressage approprié n'est pas un problème spécifique à JON, les mauvais schémas peuvent l'affecter. Amazon EC2 attribue des adresses IP à partir du réseau 10.0.0.0/8. Les instances ont généralement une adresse IP publique également, mais seul le trafic réseau sur l'adresse IP interne dans la même zone de disponibilité est gratuit. Pour éviter d'utiliser le réseau 10.0.0.0/8 en adressage privé, il y a plusieurs choses à considérer.

- Quand vous créez un VPC, évitez d'allouer des adresses déjà utilisées dans le réseau privé afin d'éviter les problèmes de connectivité.
- Si une instance a besoin d'accéder à des ressources locales de zone disponible, veillez à ce que les adresses privées Amazon EC2 soient utilisées et que les trafic ne soit pas dirigé par le VPN.
- Si une instance d'Amazon EC2 a accès à un petit sous-ensemble d'adresses de réseau privé d'entreprise (par exemple les serveurs JON uniquement), seules ces adresses devront être acheminées par le VPN. Cela augmente la sécurité et réduit les chances de collision d'espaces d'adresse de réseau privé et Amazon EC2.

Report a bug

### 21.3.6. Quitter ou Re-démarrer JON

Un des avantages des environnements de Cloud Computing est la facilité avec laquelle vous pouvez résilier ou lancer une instance de la machine. Vous pouvez également lancer une instance identique à l'instance initiale. Cela peut entraîner des problèmes si la nouvelle instance tente de s'enregistrer par les serveurs JON utilisant le même nom d'agent que celui de l'agent déjà en cours d'exécution. Dans un tel cas, le serveur JON ne permettra pas un agent de reconnecter avec un token d'identification manquant ou non correspondant.

Afin d'éviter cela, veillez à ce que les agents qui ont fait leur travail soient retirés de l'inventaire JON avant d'essayer de connecter un agent du même nom ou de spécifier le token d'identification qui convient quand vous démarrerez un nouvel agent.

Un autre problème que vous pourriez rencontrer est lorsqu'une machine d'agent reçoit une nouvelle adresse IP VPN qui ne correspond plus à l'adresse enregistrée dans la configuration de JON. Un exemple pourrait inclure une machine qui redémarre ou lorsque une connexion VPN a été interrompue. Dans ce cas, il est recommandé que vous liez le cycle de vie de l'agent JON au cycle de vie de la connexion VPN. Si la connexion tombe, vous pouvez arrêter l'agent. Lorsque la connexion est rétablie à nouveau, mettre à jour **`JON_AGENT_ADDR`** dans **`/etc/sysconfig/jon-agent-ec2`** pour refléter la nouvelle adresse IP, puis redémarrez l'agent.

Les informations sur la façon de changer l'adresse IP de l'agent se trouve dans Guide «Configuring JON Servers and Agents Guide» à l'adresse suivante https://access.redhat.com/site/documentation/JBoss_Operations_Network/.

S'il existe un grand nombre d'instances lancées ou résiliées, cela risque d'être difficile d'ajouter ou de supprimer les instances manuellement dans l'inventaire de JON. Les capacités de scripting de JON peuvent être utilisées pour automatiser ces étapes. Voir la documentation JON pour plus d'informations.

Report a bug

### 21.3.7. Configurer une instance pour vous enregistrer dans le Réseau d'opérations de JBoss.

Utiliser la procédure suivante pour enregistrer une instance de JBoss Enterprise Application Platform dans JBoss Operations Network.

- Dans JBoss Enterprise Application Platform, ajouter ceci dans le champ User Data (Données utilisateur).

```
JON_SERVER_ADDR=jon2.it.example.com
## if instance not already configured to resolve its hostname
JON_AGENT_ADDR=`ip addr show dev eth0 primary to 0/0 | sed -n
's#.*inet \([0-9.]\+\)/.*#\1#p'`
PORTS_ALLOWED=16163
# insert other JON options when necessary, see Appendix I
```

Report a bug

## 21.4. CONFIGURATION DU SCRIPT UTILISATEUR

### 21.4.1. Paramètres de configuration permanente

**Résumé**

Les paramètres suivants peuvent être utilisés pour influencer la configuration et les opérations de JBoss Enterprise Application Platform. Leur contenu se trouve dans **`/etc/sysconfig/jbossas`** et **`/etc/sysconfig/jon-agent-ec2`**.

**Tableau 21.2. Paramètres configurables**

| Nom | Description | Par défaut |
| --- | --- | --- |
| JBOSS_JGROUPS_S3_PING_ACCESS_KEY | Clé d'accès de compte utilisateur Amazon AWS pour S3_PING Discovery quand on utilise le clustering. | N/A |
| JBOSS_JGROUPS_S3_PING_SECRET_ACCESS_KEY | Clé d'accès secrète au compte utilisateur Amazon AWS | N/A |
| JBOSS_JGROUPS_S3_PING_BUCKET | Amazon S3 Bucket à utiliser dans S3_PING Discovery. | N/A |
| JBOSS_CLUSTER_ID | ID des noeuds de membres d'un groupement. Utilisé uniquement pour le clustering. Les valeurs acceptées sont (dans l'ordre) :<br>• Un nombre d'ID de groupement valide entre 0 - 1023.<br>• Un nom d'interface de réseau, avec le dernier octet de l'IP utilisé comme valeur.<br>• "S3" comme valeur coordonnerait l'utilisation de l'ID par la S3 Bucket utilisée par S3_PING des jgroups.<br>Il est conseillé d'utiliser le dernier octet de l'IP (par défaut) quand tous les noeuds de cluster sont situés dans le sous-système de 24 octets ou davantage (par exemple, dans un sous-ensemble VPC). | Dernier octet de l'adresse IP d'eth0 |
| MOD_CLUSTER_PROXY_LIST | Liste délimitée par des virgules d'IP/Noms d'hôte de proxies mod_cluster si mod_cluster doit être utilisé. | N/A |
| PORTS_ALLOWED | Liste des ports entrants qui seront utilisés par le pare-feu en plus des ports par défaut. | N/A |
| JBOSSAS_ADMIN_PASSWORD | Mot de passe pour l'utilisateur **`admin`**. | N/A |

| Nom | Description | Par défaut |
|---|---|---|
| JON_SERVER_ADDR | IP ou Nom d'hôte du serveur JON dans lequel s'enregistrer. Uniquement utilisé pour l'enregistrement, ensuite, l'agent peut communiquer avec les autres serveurs dans le groupement JON. | N/A |
| JON_SERVER_PORT | Port utilisé par l'agent pour communiquer avec le serveur. | 7080 |
| JON_AGENT_NAME | Nom de l'agent JON, doit être unique. | ID de l'instance |
| JON_AGENT_PORT | Port que l'agent écoute. | 16163 |
| JON_AGENT_ADDR | Adresse IP à laquelle l'agent JON est relié. Utilisé quand le serveur a plus d'une adresse publique, (par ex VPN). | L'agent JON choisit l'IP ou le nom d'hôte local par défaut. |
| JON_AGENT_OPTS | Propriétés de système d'agent JON supplémentaire pouvant être utilisé pour configurer SSL, NAT et d'autres paramètres avancés. | N/A |

| Nom | Description | Par défaut |
| --- | --- | --- |
| JBOSS_SERVER_CONFIG | Nom du fichier de configuraiton de serveur JBoss EAP à utiliser. Si JBOSS_DOMAIN_CONTROLLER =true, alors **`domain-ec2.xml`** sera utilisé. Sinon :<br>• Si la config S3 est présente, alors **`standalone-ec2-ha.xml`** sera utilisé.<br>• Si MOD_CLUSTER_PROXY_LIST est spécifié, alors **`standalone-mod_cluster-ec2-ha.xml`** sera sélectionné.<br>• Si aucune des deux premières options sont utilisées, alors le fichier **`standalone.xml`** sera utilisé.<br>• Peut également être défini à **`standalone-full.xml`**. | **`standalone.xml`**, **`standalone-full.xml`**, **`standalone-ec2-ha.xml`**, **`standalone-mod_cluser-ec2-ha.xml`**, **`domain-ec2.xml`** suivant les autres paramètres. |
| JAVA_OPTS | Valeurs personnalisées à ajouter à la variable avant que JBoss Enterprise Application Platform démarre. | JAVA_OPTS est créé à partir de valeurs appartenant à d'autres paramètres. |
| JBOSS_IP | Adresse IP à laquelle le serveur est lié. | 127.0.0.1 |
| JBOSSCONF | Le nom du profil de JBoss Enterprise Application Platform 6 pour démarrer. Pour empêcher JBoss Enterprise Application Platform 6 de démarrer, JBOSSCONF peut être défini à **`disabled`** | **`standalone`** |
| JBOSS_DOMAIN_CONTROLLER | Indique si cette instance exécute ou non comme contrôleur de domaine. | **`false`** |
| JBOSS_DOMAIN_MASTER_ADDRESS | Adresse IP de contrôleur de domaine distant. | N/A |

| Nom | Description | Par défaut |
|---|---|---|
| JBOSS_HOST_NAME | Le nom d'hôte logique (du domaine). Doit être un nom séparé. | La valeur de la variable d'environnement HOSTNAME. |
| JBOSS_HOST_USERNAME | Le nom d'utilisateur du contrôleur d'hôte doit être utilisé quand on enregistre dans le contrôleur de domaine. Si non fourni, le JBOSS_HOST_NAME sera utilisé à la place. | JBOSS_HOST_NAME |
| JBOSS_HOST_PASSWORD | Le mot de passe que le contrôleur d'hôte doit utiliser quand il s'enregistre avec le contrôleur de domaine. | N/A |
| JBOSS_HOST_CONFIG | Si JBOSS_DOMAIN_CONTROLLER =true, alors **`host-master.xml`** sera utilisé. Si JBOSS_DOMAIN_MASTER_ADDRESS est présent, alors **`host-slave.xml`** sera utilisé. | **`host-master.xml`** ou **`host-slave.xml`**, suivant les autres paramètres. |

Report a bug

### 21.4.2. Paramètres de scripts personnalisés

**Résumé**

Les paramètres suivants peuvent être utilisés dans la section de personnalisation utlisateur du champ **`User Data:`**.

**Tableau 21.3. Paramètres configurables**

| Nom | Description |
|---|---|
| JBOSS_DEPLOY_DIR | Déployer le répertoire du profil actif (par exemple, **`/var/lib/jbossas/standalone/deployments/`**). Les archives déployables de ce répertoire seront déployées. Red Hat recommande d'utiliser la Console de gestion ou le CLI pour gérer les déploiements au lieu d'utiliser le répertoire de déploiement. |
| JBOSS_CONFIG_DIR | Répertoire de config du profile actif (par exemple, **`/var/lib/jbossas/standalone/configuration`**). |
| JBOSS_HOST_CONFIG | Nom du fichier de configuration de l'hôte actif (par exemple, **`host-master.xml`**). Red Hat conseille d'utiliser la Console de gestion ou le CLI pour configurer le serveur au lieu d'éditer le fichier de configuration. |

| Nom | Description |
| --- | --- |
| JBOSS_SERVER_CONFIG | Nom du fichier de configuration du serveur actif (par exemple, **`standalone-ec2-ha.xml`**). Red Hat conseille d'utiliser la Console de gestion ou le CLI pour configurer le serveur au lieu d'éditer le fichier de configuration. |
| USER_SCRIPT | Chemin d'accès au script de configuration personnalisé disponible avant de trouver la configuraiton user-data. |
| USER_CLI_COMMANDS | Chemin d'accès à un fichier personnalisé des commandes CLI, disponible avant de trouver la configuration user-data. |

Report a bug

## 21.5. RÉSOLUTION DE PROBLÈMES

### 21.5.1. Résolution de problèmes dans Amazon EC2

EC2 ne fournit aucune méthode out of the box pour indiquer qu'une instance a démarré correctement ou que les services exécutent correctement. Il est recommandé d'utiliser un système externe de surveillance et de gestion. JBoss Operations Network (JON) peut automatiquement découvrir, surveiller et gérer de nombreux services sur une instance EC2 avec l'agent JON installé, y compris JBoss Enterprise Application Platform et ses services, Tomcat, Httpd, PostgreSQL, etc. Comme il n'y a pas de différence entre une instance hébergée par EC ou hébergée localement dans JBoss Enterprise Application Platform, le monitoring JON des deux types de déploiement est identique.

Report a bug

### 21.5.2. Information de diagnostique

En cas de problème détecté par JBoss Operations Network, Amazon CloudWatch ou une inspection manuelle, voici les sources habituelles d'informations de diagnostique :

- **`/var/log/jboss_user-data.out`** est la sortie du script init de jboss-ec2-eap et du script de configuration personnalisée utilisateur.
- **`/var/cache/jboss-ec2-eap/`** contient les données utilisateur réelles, le script personnalisé, et les commandes CLI utilisées au démarrage de l'instance.
- **`/var/log`** contient également tous les journaux collectés au démarrage de la machine, JBoss Enterprise Application Platform, httpd et la plupart des autres services.

L'accès à ces fichiers est disponible uniquement via une session SSH. Voir Amazon EC Getting Started Guide pour obtenir des informations sur la façon de configurer ou d'établir une session SSH avec une instance Amazon EC2.

Report a bug

# CHAPITRE 22. RÉFÉRENCES SUPPLÉMENTAIRES

## 22.1. TÉLÉCHARGER LES FICHIERS DU PORTAIL DES CLIENTS DE RED HAT

**Prérequis**

- Avant de commencer cette tâche, vous aurez besoin d'un compte de Portail Clients. Rendez-vous dans https://access.redhat.com et cliquer sur le lien **`Register`** qui se trouve en haut et à droite pour créer un compte.

**Procédure 22.1. Connectez-vous et téléchargez les fichiers du Portail Clients de Red Hat**

1. Aller dans http://access.redhat.com et cliquer sur le lien **`Log in`** en haut et à gauche. Saisir vos identifiants, et cliquer sur **`Log In`**.

   **Résultat**

   Vous êtes connecté dans RHN et on vous renvoie à la page web principale à https://access.redhat.com.

2. **Rendez-vous à la page `Downloads` page.**
   Utiliser une des options de pour aller dans la page **`Downloads`**.

   - Cliquer sur le lien **`Downloads`** qui se trouve en haut de la barre de navigation.
   - Rendez-vous directement dans https://access.redhat.com/downloads/.

3. **Sélectionner le produit et la version à télécharge.**
   Utiliser un de ces façons pour choisir le produit et la version qui conviennent à télécharger.

   - Procéder une étape à la fois.
   - Chercher votre produit à partir de la zône de recherche qui se trouve en haut et à droite de l'écran.

4. **Télécharger le fichier qui convient pour votre système d'exploitation et la méthode d'installation de votre choix.**
   Suivant le produit, vous aurez le choix entre un installateur natif, un RPM ou une archive Zip suivant le système d'exploitation et l'architecture. Cliquer soit sur le nom de fichier ou sur le lien **`Download`** à droite du fichier que vous souhaitez télécharger.

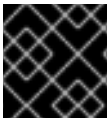
**Résultat**

Le fichier sera téléchargé dans votre ordinateur.

Report a bug

## 22.2. CONFIGURER LE JDK PAR DÉFAUT DANS RED HAT ENTERPRISE LINUX

Il est possible d'avoir plusieurs Java Development Kits (JDKs) installés sur votre système Red Hat Enterprise Linux. Cette tâche vous montre comment spécifier quel kit est utilisé par votre environnement actuel. Nécessite la commande **`alternatives`**.



**IMPORTANT**

Cette tâche ne s'applique que pour Red Hat Enterprise Linux.

**NOTE**

Il est possible de ne pas avoir à passer par cette étape. Red Hat Enterprise Linux utilise OpenJDK 1.6 comme option par défaut. Si c'est ce qui vous convient et que vous utilisez votre système correctement, vous n'aurez pas besoin d'indiquer quel JDK utiliser.

**Prérequis**

- Pour compléter cette tâche, vous devrez avoir l'accès super utilisateur, soit en connexion directe, ou par la commande **sudo**.

**Procédure 22.2. Configure the Default JDK**

1. **Déterminez les chemins qui vont conviennent le mieux pour vos binaires `java` et `javac`.**
   Vous pouvez utiliser la commande **`rpm -ql packagename |grep bin`** pour trouver les emplacements des binaires installés à partir des RPM. Les locations par défaut des binaires **`java`** et **`javac`** des systèmes Red Hat Enterprise Linux 32-bit sont les suivantes :

   **Tableau 22.1. Emplacements par défaut des binaires `java` et `javac`**

   | JDK | Chemin |
   |---|---|
   | OpenJDK 1.6 | **`/usr/lib/jvm/jre-1.6.0-openjdk/bin/java`**<br><br>**`/usr/lib/jvm/java-1.6.0-openjdk/bin/javac`** |
   | Oracle JDK 1.6 | **`/usr/lib/jvm/jre-1.6.0-sun/bin/java`**<br><br>**`/usr/lib/jvm/java-1.6.0-sun/bin/javac`** |

2. **Définir chaque solution alternative**
   Exécutez les commandes suivantes pour que votre système utilise **`java`** et **`javac`**: **`/usr/sbin/alternatives --config java`** ou **`/usr/sbin/alternatives --config javac`**. Suivez les instructions sur l'écran.

3. **Option: définir un choix alternatif `java_sdk_1.6.0`.**
   Si vous souhaitez utiliser Oracle JDK, vous devrez configurer la solution alternative **`java_sdk_1.6.0.`** également. Utiliser la commande suivante : **`/usr/sbin/alternatives --config java_sdk_1.6.0`**. Le chemin d'accès qui convient est normalement **`/usr/lib/jvm/java-1.6.0-sun`**. Vous pourrez faire une liste de fichiers pour vérifier.

**Résultat :**

Le JDK alternatif est sélectionné et actif.

Report a bug

# ANNEXE A. REVISION HISTORY

**Version 1.1-28** **Thu Jul 11 2013** **Russell Dickenson**

JBoss Enterprise Application Platform 6.1.0 GA Release.