

Connecteur JOnAS-MQ

Manuel d'utilisation

Date : Avril 2007

Référence : JMQ-MN7420-01 v 1.2



François Waeselynck

BuLL
1 rue de Provence
38130 Echirolles

francois.waeselynck@bull.net



SOMMAIRE

1.	Structure du document	3
2.	Mise en oeuvre de WebSphere MQ	4
2.1.	Installation.....	4
2.2.	Mise en route de Websphere MQ.....	4
2.2.1.	Mise en place et premier démarrage	4
2.2.2.	Démarrage ultérieur	6
2.2.3.	Arrêt de Websphere MQ	6
3.	Installation du connecteur.....	8
3.1.	Prérequis.....	8
3.2.	jonas-ra.xml	9
3.2.1.	Modules « Resource Adapter »	9
3.2.1.1.	Les propriétés communes des connexions.....	9
3.2.1.2.	Les propriétés pour l'authentification JNDI de MQ.....	11
3.2.1.3.	Les propriétés des fabriques de connexions JMS	11
3.2.1.4.	Les noms JNDI des fabriques de connexions non-gérées par JOnAS.....	13
3.2.1.5.	Autres propriétés.....	14
3.2.2.	Module « Outbound ».....	14
3.2.3.	Module « Inbound »	15
3.3.	jonasmq.xml.....	16
3.4.	Le déploiement du connecteur	17
4.	Déploiement d'un MDB	18
4.1.	ejb-jar.xml	18
4.2.	jonas-ejb-jar.xml.....	22
5.	Les MBeans	24
5.1.	Les MBeans JonasMQConnector.....	24
5.2.	Les MBeans MQDestination	26
5.3.	Les MBeans MDB	28



1. Structure du document

Ce manuel utilisateur détaille la mise en œuvre du connecteur JCA JOnAS-MQ. Il s'adresse :

- A l'administrateur du serveur d'application.
- Au déployeur d'application.
- Dans une moindre mesure, il peut également concerner un administrateur MQ.

La partie 2. « Installation de WebSphere MQ pour JOnAS » explique la procédure d'installation de WebSphere MQ pour pouvoir faire fonctionner les applications déployées sur JOnAS avec WebSphere MQ.

La partie 3. « Installation du connecteur » explique comment configurer et mettre en œuvre le connecteur.

La partie 4. « Déploiement des MDBs » explique comment configurer les MDBs qui utilisent le connecteur JOnAS-MQ.

La partie 5. « Les MBeans » explique les interfaces d'administration des MBeans du connecteur.

L' ANNEXE A liste quelques références utiles.



2. Mise en oeuvre de WebSphere MQ

2.1. Installation

Pour faire interopérer JOnAS et MQ, il faut disposer d'une installation de WebSphere MQ comprenant :

- Le serveur MQ.
- Le « Support des transactions client étendues » (en anglais : « Client extended transaction support »).
- Le « Support des transactions Java étendues » (en anglais : « Java extended transaction support »).

WebSphere MQ 6 peut être téléchargé depuis le site d'IBM à l'URL :

<http://www-306.ibm.com/software/integration/wmq/>.

La distribution contient un guide utilisateur intitulé « Quick Beginnings » que l'on suivra pour installer MQ. Ce guide est aussi disponible au téléchargement sur le site des librairies de WebSphere MQ, où il est décliné par plate-forme, à l'URL :

<http://www-306.ibm.com/software/integration/wmq/library/library6x.html>

2.2. Mise en route de Websphere MQ

Cette section décrit une procédure de démarrage et de configuration de WebSphere MQ permettant l'usage de l'interface JMS de WebSphere MQ. Notez que cette configuration est **minimale** : pour obtenir une configuration complète de WebSphere, consulter les guides d'utilisateur « WebSphere MQ Quick Beginnings », « WebSphere MQ Using Java », « WebSphere MQ Publish/Subscribe User's Guide » ce qui se trouvent sur le site des librairies mentionné ci-dessus.

2.2.1. Mise en place et premier démarrage

Pour une première configuration et démarrage de WebSphere MQ, appliquer la procédure suivante :

- Créer un gestionnaire de files d'attente si vous n'en avez aucun :

```
crtmqm QUEUE.MANAGER
```

```
par exemple : crtmqm QM_JONAS
```



- Démarrer le gestionnaire de files d'attente :

```
strmqm QUEUE.MANAGER
```

par exemple : strmqm QM_JONAS

- Créer les files d'attente pour le service publish/suscription :

```
runmqsc QUEUE.MANAGER < MQ_INSTALL_PATH/java/bin/MQJMS_PSQ.mqsc
```

par exemple : runmqsc QM_JONAS < /opt/mqm/java/bin/MQJMS_PSQ.mqsc

- Démarrer le service publish/suscription avec la commande :

```
strmqbrk -m QUEUE.MANAGER
```

par exemple : strmqbrk -m QM_JONAS

- Ouvrir la console de commandes du gestionnaire de files d'attente :

```
runmqsc QUEUE.MANAGER
```

par exemple : runmqsc QM_JONAS

- Dans la console de commandes de WebSphere MQ, créer un module d'écoute pour le gestionnaire de files d'attente :

```
DEFINE LISTENER(listenerName) TRPTYPE(TCP) PORT(portNumber)
```

par exemple :

```
DEFINE LISTENER(LISTENER1) TRPTYPE(TCP) PORT(1414)
```

- Dans la console de commandes de WebSphere MQ, créer un canal de type connexion serveur si le gestionnaire de files MQ est installé sur Linux :

```
DEFINE CHANNEL(channelName) CHLTYPE(SVRCONN) TRPTYPE(TCP)  
MCAUSER('mqm')
```

par exemple :

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) MCAUSER('mqm')
```

Si le gestionnaire de files MQ est installé sur Windows, le script est comme suit :

```
DEFINE CHANNEL(channelName) CHLTYPE(SVRCONN) TRPTYPE(TCP)  
MCAUSER('MUSR_MQADMIN')
```

par exemple :

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP)  
MCAUSER('MUSR_MQADMIN')
```

Note : le paramètre MCAUSER dans la commande est établi à 'mqm' pour Linux ou 'MUSR_MQADMIN' pour Windows signifie que le canal accepte les connexions qui viennent de n'importe quelle machine dans le même réseau. C'est un point de sécurité à améliorer !

- Dans la console de commandes, démarrer le module d'écoute :



```
START LISTENER(listenerName)
```

```
par exemple : START LISTENER(LISTENER1)
```

- Pour sortir de la console de commande script de MQ :

```
end
```

Le gestionnaire de file d'attente est opérationnel et prêt à servir des applications JMS.

2.2.2. Démarrage ultérieur

Pour tout démarrage ultérieur de webspere MQ, appliquer la procédure suivante :

- Démarrer le gestionnaire de files d'attente :

```
strmqm QUEUE.MANAGER
```

```
par exemple : strmqm QM_JONAS
```

- Démarrer le service publish/suscription avec la commande :

```
strmqbrk -m QUEUE.MANAGER
```

```
par exemple : strmqbrk -m QM_JONAS
```

- Ouvrir la console de commandes du gestionnaire de files d'attente :

```
runmqsc QUEUE.MANAGER
```

```
par exemple : runmqsc QM_JONAS
```

- Dans la console de commandes, démarre le module d'écoute :

```
START LISTENER(listenerName)
```

```
par exemple : START LISTENER(LISTENER1)
```

- Pour sortir de la console de commande script de MQ :

```
end
```

Le gestionnaire de file d'attente est opérationnel et prêt à servir des applications JMS.

2.2.3. Arrêt de Websphere MQ

Pour arrêter le service publish/suscription du gestionnaire de files exécuter la commande suivante :

```
endmqbrk -m QUEUE.MANAGER
```

```
par exemple : endmqbrk -m QM_JONAS
```



Connecteur JOnAS-MQ

Manuel d'utilisation

Extension de la console d'administration JonasAdmin

Pour arrêter le gestionnaire de files, exécuter la commande suivante :

```
endmqm QUEUE.MANAGER
```

```
par exemple : endmqm QM_JONAS
```



3. Installation du connecteur

Cette section décrit la configuration du connecteur. Cette configuration se fait avant le déploiement du connecteur.

La configuration repose sur deux fichiers :

- le fichier `jonas-ra.xml`, le descripteur de déploiement spécifique à JOnAS, qui est inclus dans le répertoire META-INF de l'archive du connecteur.
- Le fichier complémentaire `jonasmq.xml` (le nom de ce fichier peut être différent selon la configuration dans `jonas-ra.xml`) qui est placé dans `%JONAS_BASE%/conf`.

3.1. Prérequis

Côté JOnAS, pour pouvoir utiliser le connecteur JOnAS-MQ, vous devez ajouter les bibliothèques suivantes dans le répertoire `%JONAS_ROOT%/lib/commons` ou `%JONAS_BASE%/lib/commons` :

- `com.ibm.mqjms.jar`: inclut les classes JMS de MQ.
- `com.ibm.mqetclient.jar`: permet de supporter les transactions XA.
- `com.ibm.mq.jar`: l'interface propriétaire de programmation MQ.
- `dhbcore.jar`: utilisé par les autres fichiers jar.
- `com.ibm.mq.pcf.jar`: permet d'administrer les gestionnaires de files MQ.
- `mqcontext.jar`: permet de créer la fabrique de contexte JNDI de MQ.

Les archives `com.ibm.mqjms.jar`, `com.ibm.mqetclient.jar`, `com.ibm.mq.jar` et `dhbcore.jar` sont à récupérer dans le répertoire `MQ_INSTALL_PATH/Java/lib`, où `MQ_INSTALL_PATH` est le répertoire d'installation de MQ.

Le fichier `mqcontext.jar` peut être téléchargé depuis le lien suivant :

<ftp://ftp.software.ibm.com/software/integration/support/supportpacs/individual/me01.zip>

Le fichier `com.ibm.mq.pcf.jar` peut être téléchargé depuis le lien suivant :

<ftp://ftp.software.ibm.com/software/integration/support/supportpacs/individual/ms0b.zip>

Pour qu'une application client JMS de JOnAS puisse reconnaître les objets JMS de WebSphere MQ, les archives `com.ibm.mqjms.jar` et le fichier `dhbcore.jar` doivent être dans la variable d'environnement `CLASSPATH` de l'application client.

Pour qu'une application Java (qu'elle soit un client JMS, ou qu'elle s'exécute au sein de JOnAS) puissent utiliser MQ en mode « binding » (Le mode par lequel l'application et le gestionnaire de files MQ résident sur la même machine et communiquent par une mémoire



partagée), ajouter la variable d'environnement `MQ_JAVA_LIB_PATH = MQ_INSTALL_PATH/Java/lib`. Ce répertoire contient des bibliothèques natives utilisées pour la communication du mode « binding ». Pour plus de détail, veuillez consulter le document « WebSphere MQ Using Java » page 8-9-10.

3.2. jonas-ra.xml

Ce fichier contient tous les informations de configurations essentielles pour le connecteur. Ces configurations sont définies par le fichier `ra.xml`. Elles contiennent les informations suivantes :

- Les propriétés du modules « Resource Adapter ».
- Les propriétés du modules « Outbound ».
- Les propriétés du modules « Inbound ».
- Les propriétés des destinations.

Ce fichier peut-être configuré commodément via l'éditeur de RAR intégré de la console jonasAdmin. Cette opération se fait connecteur non déployé.

3.2.1. Modules « Resource Adapter »

Les propriétés de ce module sont définies par les éléments XML `<jonas-config-property>`. Chaque élément contient une paire `<jonas-config-property-name>` et `<jonas-config-property-value>` qui déclarent le nom d'une propriété et sa valeur.

Les propriétés de ce module sont classées comme suit.

3.2.1.1. Les propriétés communes des connexions

Ces propriétés sont les plus essentielles pour pouvoir connecter à un gestionnaire de files d'attente MQ.

Nom de la propriété	Valeur de la propriété	Description
<code>TransportType</code>	'client' ou 'binding'	Le mode « client » utilise le protocole TCP/IP tandis que le mode « binding » utilise le protocole de mémoire partagée pour connecter à un gestionnaire de file d'attente. Le mode « binding » ne peut donc être utilisé que si le gestionnaire de files est exécuté sur la même machine avec JOnAS.
<code>HostName</code>	Une adresse IP ou le nom du hôte	Doit être renseigné si le mode de transport est « client ». C'est l'adresse IP ou le nom du hôte qui héberge le gestionnaire de files d'attente.



Nom de la propriété	Valeur de la propriété	Description
Port	Un Integer	Doit être renseigné si le mode de transport est « client ». C'est le port sur lequel écoute le gestionnaire de files d'attente (port d'écoute du LISTENER).
Channel	Un String	Doit être renseigné si le mode de transport est « client ». C'est le nom du canal de type connexion serveur du gestionnaire de files d'attente.
QueueManager	Un String	Doit être renseigné si le mode de transport est « binding ». C'est le nom du gestionnaire de files d'attente.

Ces propriétés peuvent être changées à partir de la console de JOnAS une fois le connecteur déployé, via l'extension à jonasAdmin (cf la section « **Erreur ! Source du renvoi introuvable. Erreur ! Source du renvoi introuvable.** »).

Voici un exemple des configurations de ces propriétés :

```
...
<jonas-config-property>
  <jonas-config-property-name>
    TransportType
  </jonas-config-property-name>
  <jonas-config-property-value>
    client
  </jonas-config-property-value>
</jonas-config-property>

<jonas-config-property>
  <jonas-config-property-name>
    HostName
  </jonas-config-property-name>
  <jonas-config-property-value>
    ode2.dc.frec.bull.fr
  </jonas-config-property-value>
</jonas-config-property>

<jonas-config-property>
  <jonas-config-property-name>
    Port
  </jonas-config-property-name>
  <jonas-config-property-value>
    1414
  </jonas-config-property-value>
</jonas-config-property>

<jonas-config-property>
  <jonas-config-property-name>
```



```
QueueManager
</jonas-config-property-name>
<jonas-config-property-value>
</jonas-config-property-value>
</jonas-config-property>

<jonas-config-property>
  <jonas-config-property-name>
    Channel
  </jonas-config-property-name>
  <jonas-config-property-value>
    SYSTEM.DEF.SVRCONN
  </jonas-config-property-value>
</jonas-config-property>
...
```

3.2.1.2. Les propriétés pour l'authentification JNDI de MQ

Le connecteur utilise le gestionnaire de files d'attente de MQ en tant que registre JNDI pour pouvoir créer les files d'attente physiques. Si le canal de ce gestionnaire de files d'attente est sécurisé, il faut fournir des informations d'authentification.

Nom de la propriété	Valeur de la propriété	Description
AuthenticationMode	'none', 'simple', ou 'CRAM_MD5'	Si le mode d'authentification est différent de « none » les valeurs de « JndiUserName » et « JndiPassword » sont requises
JndiUserName	Un String	L'identité de l'utilisateur.
JndiPassword	Un String	Le mot de passe de l'utilisateur.

3.2.1.3. Les propriétés des fabriques de connexions JMS

Nom de la propriété	Valeur de la propriété	Description
ConnectionFactoryProperties	Un String	Une chaîne de caractères composée de paires 'propertyName=propertyValue' séparées par des virgules ','.
QueueConnectionFactoryProperties	Un String	Une chaîne de caractères composée de paires 'propertyName=propertyValue' séparées par des virgules ','.



Nom de la propriété	Valeur de la propriété	Description
TopicConnectionFactoryProperties	Un String	Une chaîne de caractères composée de paires 'propertyName=propertyValue' séparées par des virgules ','.

Pour le détail des propriétés des fabriques de connexions MQ, consulter le guide d'utilisateur « WebSphere MQ Using Java » - Chapter 16. Package com.ibm.mq.jms – MQConnectionFactory – Page 476.

Voici un exemple des configurations de ces propriétés :

```
...
<jonas-config-property>
  <jonas-config-property-name>
    ConnectionFactoryProperties
  </jonas-config-property-name>
  <jonas-config-property-value>
    ClientID=jonasmq
  </jonas-config-property-value>
</jonas-config-property>

<jonas-config-property>
  <jonas-config-property-name>
    QueueConnectionFactoryProperties
  </jonas-config-property-name>
  <jonas-config-property-value>
    ClientID=jonasmq
  </jonas-config-property-value>
</jonas-config-property>

<jonas-config-property>
  <jonas-config-property-name>
    TopicConnectionFactoryProperties
  </jonas-config-property-name>
  <jonas-config-property-value>
    ClientID=jonasmq
  </jonas-config-property-value>
</jonas-config-property>
...
```

Ces propriétés peuvent être changées à partir de la console de JOnAS une fois le connecteur déployé, via l'extension à jonasAdmin (cf la section « **Erreur ! Source du renvoi introuvable. Erreur ! Source du renvoi introuvable.** »).



3.2.1.4. Les noms JNDI des fabriques de connexions non-gérées par JOnAS

Ce sont les noms JNDI pour les fabriques de connexions non gérées, utilisées par les programmes Java « stand-alone ». Ces noms doivent être différents de ceux des autres connecteurs JMS afin que l'on puisse déployer plusieurs connecteurs JMS en même temps.

Nom de la propriété	Valeur de la propriété	Description
JConnectionFactoryJndiName	Un String	Le nom JNDI pour l'objet ConnectionFactory.
JQueueConnectionFactoryJndiName	Un String	Le nom JNDI pour l'objet QueueConnectionFactory.
JtopicConnectionFactoryJndiName	Un String	Le nom JNDI pour l'objet TopicConnectionFactory.

Voici un exemple de configuration de ces propriétés :

```
...
<jonas-config-property>
  <jonas-config-property-name>
    JConnectionFactoryJndiName
  </jonas-config-property-name>
  <jonas-config-property-value>
    JCF
  </jonas-config-property-value>
</jonas-config-property>

<jonas-config-property>
  <jonas-config-property-name>
    JQueueConnectionFactoryJndiName
  </jonas-config-property-name>
  <jonas-config-property-value>
    JQCF
  </jonas-config-property-value>
</jonas-config-property>

<jonas-config-property>
  <jonas-config-property-name>
    JTopicConnectionFactoryJndiName
  </jonas-config-property-name>
  <jonas-config-property-value>
    JTCF
  </jonas-config-property-value>
</jonas-config-property>
...
```

Attention, ces propriétés ne peuvent pas être modifiées une fois le connecteur déployé.



3.2.1.5. Autres propriétés

Nom de la propriété	Valeur de la propriété	Description
MdbName	Un String	Le nom MBean de cette instance de connecteur. Si plusieurs connecteurs sont déployés, leurs noms doivent être différents.
ConfigFileName	Un String	Le nom du fichier qui contient les configurations des destinations JMS de MQ.

Voici un exemple des configurations de ces propriétés :

```
...
<!--
    MBEAN NAME FOR THIS INSTANCE OF THE CONNECTOR
-->
<jonas-config-property>
    <jonas-config-property-name>
        MdbName
    </jonas-config-property-name>
    <jonas-config-property-value>
        jonasmq
    </jonas-config-property-value>
</jonas-config-property>

<!--
    DESTINATION CONFIGURATION FILE NAME
-->
    <jonas-config-property>
    <jonas-config-property-name>
        ConfigFileName
    </jonas-config-property-name>
    <jonas-config-property-value>
        jonasmqconfig.xml
    </jonas-config-property-value>
</jonas-config-property>
...
```

3.2.2. Module « Outbound »

Les configurations dans cette partie contiennent les propriétés des fabriques de connexion gérées par JOnAS (les objets ManagedConnectionFactory). Chaque fabrique de connexions est définie dans un élément XML <jonas-connection-definition>. La propriété essentielle de chaque fabrique de connexions est son nom JNDI. Ce nom doit être différent de ceux des autres connecteurs afin que plusieurs connecteurs et plusieurs instances de ce connecteur puissent être déployés concurremment.

Voici un exemple des configurations de ce module :



```
...
<jonas-connection-definition>
  <description>
    com.bull.jonasmqconnector.outbound.ManagedQueueConnectionFactoryImpl
  </description>
  <jndi-name>
    QCF
  </jndi-name>
</jonas-connection-definition>

<jonas-connection-definition>
  <description>
    com.bull.jonasmqconnector.outbound.ManagedTopicConnectionFactoryImpl
  </description>
  <jndi-name>
    TCF
  </jndi-name>
</jonas-connection-definition>

<jonas-connection-definition>
  <description>
    com.bull.jonasmqconnector.outbound.ManagedJMSConnectionFactoryImpl
  </description>
  <jndi-name>
    CF
  </jndi-name>
</jonas-connection-definition>
...
```

3.2.3. Module « Inbound »

Les configurations dans cette partie contiennent les propriétés du module « Inbound ». La propriété essentielle de ce module est le nom JNDI. C'est le nom avec lequel, les MDBs peuvent identifier le connecteur qu'ils veulent utiliser. Il faut donc que ce nom soit différent de ceux des autres connecteurs JCA pour qu'on puisse utiliser en même temps plusieurs connecteurs JCA.

Voici un exemple des configurations de ce module :

```
...
<jonas-activation-spec>
  <description>javax.jms.MessageListener</description>
  <jndi-name>jonasmq_as</jndi-name>
  <defaultAS>>true</defaultAS>
</jonas-activation-spec>
...
```

La propriété `<defaultAS>` positionnée à « true » signifie que ce connecteur est celui utilisé par défaut par tout MDB ne précisant pas le nom JNDI du connecteur dans son fichier de déploiement `jonas-ejb-jar.xml` (Au lieu de préciser le nom JNDI du connecteur par l'élément XML `<jndi-name>` dans `jonas-ejb-jar.xml`, le déployeur du MDB peut préciser à la place le nom JNDI de la destination).



3.3. jonasmq.xml

Ce fichier est situé dans le répertoire `%JONAS_BASE%/conf`. Il contient des configurations de destinations JMS MQ qui sont créées et initialisées ou retrouvées par le connecteur lors de son démarrage une fois déployé. Ce fichier peut être modifié à travers un MBean et via l'extension de la console JOnAS.

Dans ce fichier, chaque élément XML `<destination>` correspond à la définition d'une destination JMS MQ. Chaque élément `<destination>` contient un attribut `name` pour définir le nom JNDI de la destination et un attribut `type` pour indiquer si cette destination est une file d'attente ou un topic. Chaque sous-élément de l'élément `<destination>` contient une paire d'attributs : nom de la propriété et la valeur de cette propriété pour la destination.

Pour le détail des propriétés des destinations JMS MQ, consulter le guide d'utilisateur « WebSphere MQ Using Java » - Chapter 16. Package `com.ibm.mq.jms - MQDestination - Page 505`.

Voici un exemple de ce fichier :

```
<?xml version="1.0" encoding="UTF-8"?>
<jonasmq>

  <destinations>

    <destination name="sampleTopic" type="Topic">
      <property name="BaseTopicName" value="sampleTopic"/>
    </destination>

    <destination name="sampleQueue" type="Queue">
      <property name="BaseQueueName" value="sampleQueue"/>
    </destination>

    <destination name="mdbTopic" type="Topic">
      <property name="BaseTopicName" value="AN_MDB_TOPIC"/>
    </destination>

    <destination name="mdbQueue" type="Queue">
      <property name="BaseQueueName" value="AN_MDB_QUEUE"/>
    </destination>

    <destination name="Q_MDB_REPLY_QUEUE_NAME" type="Queue">
      <property
        name="BaseQueueName"
        value="Q_MDB_REPLY_QUEUE_NAME"/>
    </destination>

    <destination name="Q_MDB_TEST_QUEUE_NAME" type="Queue">
      <property
        name="BaseQueueName"
        value="Q_MDB_TEST_QUEUE_NAME"/>
    </destination>

  </destinations>
```



</jonasmq>

3.4. Le déploiement du connecteur

Une fois configuré, le connecteur est déployé selon les mécanismes standard de JOnAS :

- Soit par configuration d `ejonas.properties`
- Soit en le déposant dans le répertoire `autoload` et en relançant JOnAS.
- Par une commande : `jonas admin -a connector_name.rar`
- Depuis la console `jonasAdmin`



4. Déploiement d'un MDB

Le déploiement d'un MDB est configuré à l'aide des descripteurs de déploiement standard et spécifique de l'archive le contenant : `ejb-jar.xml` et `jonas-ejb-jar.xml`.

4.1. `ejb-jar.xml`

On suppose que ce fichier suit le schéma `ejb-jar_2_1.xsd` :

http://java.sun.com/xml/ns/j2ee/ejb-jar_2_1.xsd

La configuration du MDB est définie sous l'élément `<activation-config>`.

L'élément `<activation-config>` se compose de sous éléments `<activation-config-property>`. Chaque élément `<activation-config-property>` contient deux sous-éléments :

- `<activation-config-property-name>` : cette élément contient le nom de la propriété.
- `<activation-config-property-value>` : cette élément contient la valeur de la propriété.

Voici les propriétés de livraison des messages d'un MDB.

Nom de la propriété	Valeur de la propriété	Description
<code>destination</code>	Un String	Le nom JNDI de la destination. A ne pas confondre pas avec le nom de la destination (le nom de file d'attente et le nom de topic). Cette propriété ne doit pas être renseignée si l'élément XML <code><jndi-name></code> dans <code>jonas-<i>ejb-jar.xml</i></code> définit le nom JNDI de la destination (et non pas le nom JNDI du connecteur)(voir 4.2).
<code>destinationType</code>	Un String	Le type de la destination : « <code>javax.jms.Topic</code> » OU « <code>javax.jms.Queue</code> ».



Nom de la propriété	Valeur de la propriété	Description
destinationProperties	Un String	Une chaîne de caractères composée de paires 'propertyName=propertyValue' séparées par des virgules ',' pour définir les propriétés JMS pour la destination.
acknowledgeMode	Un String	A définir si le mode de transaction de ce MDB est « NotSupported » : « Auto-acknowledge » ou « Dups-ok-acknowledge » Cf la spécification JMS 1.1.
messageSelector	Un String	Une chaîne de caractères de requête pour sélectionner des messages. Cf la spécification JMS 1.1.
subscriptionDurability	Un String	Cf la spécification JMS 1.1 : « Durable » ou « NonDurable »
subscriptionName	Un String	A renseigner si la propriété « subscriptionDurability » est « Durable » Cf la spécification JMS 1.1.
maxMessages	Un entier	Le nombre maximal de messages reçus par une session en même temps. Valeur par défaut : 1. Cf la spécification JMS 1.1.
userName	Un String	Nom de l'utilisateur pour la connexion.
password	Un String	Le mot de passe de l'utilisateur pour la connexion.
poolSize	Un entier	Par défaut est « 1 ». La taille du pool de ServerSession.
initPoolSize	Un entier	La taille initiale du pool de ServerSession.



Connecteur JOnAS-MQ

Manuel d'utilisation

Extension de la console d'administration JonasAdmin

Nom de la propriété	Valeur de la propriété	Description
maxNumberOfWorks	Un entier	Le nombre maximal de processus que le pool peut ouvrir simultanément pour exécuter des ServerSession.
mdbName	Un String	Un nom pour identifier ce MDB sur le serveur MBean. Ceci devrait être identique avec le nom du MDB.
reconnect	Boolean	Par défaut est « true ». Cette propriété spécifie si le MDB doit tenter de rétablir la connexion en cas de rupture de la connexion physique au serveur MQ.
maxReconnect	Un entier	Par défaut est « 6 ». Cette propriété spécifie le nombre de tentatives pour rétablir la connexion en cas de rupture de la connexion physique au serveur MQ.
reconnectInterval	Un entier long	Par défaut est « 10 000 ». Cette propriété spécifie le délai (en millisecondes) entre deux tentatives de reconnexion.

Voici un extrait de fichier `ejb-jar.xml` :

```
...
<message-driven>
  <description>
    A Queue Transactional MDB for Test
  </description>
  <display-name>QT Mdb</display-name>
  <ejb-name>QTMdbTest</ejb-name>
  <ejb-class>com.bull.jonasmqconnector.test.qtmdb.QTMdbTestBean</ejb-
class>
  <messaging-type>javax.jms.MessageListener</messaging-type>
  <transaction-type>Container</transaction-type>

  <activation-config>

    <activation-config-property>
      <activation-config-property-name>
        destination
      </activation-config-property-name>
      <activation-config-property-value>
        QT_MDB_TEST_QUEUE
      </activation-config-property-value>
    </activation-config-property>

    <activation-config-property>
      <activation-config-property-name>
```



```
        destinationType
    </activation-config-property-name>
    <activation-config-property-value>
        javax.jms.Queue
    </activation-config-property-value>
</activation-config-property>

<activation-config-property>
    <activation-config-property-name>
        acknowledgeMode
    </activation-config-property-name>
    <activation-config-property-value>
        Auto-acknowledge
    </activation-config-property-value>
</activation-config-property>

<activation-config-property>
    <activation-config-property-name>
        subscriptionDurability
    </activation-config-property-name>
    <activation-config-property-value>
        NonDurable
    </activation-config-property-value>
</activation-config-property>

<activation-config-property>
    <activation-config-property-name>
        initPoolSize
    </activation-config-property-name>
    <activation-config-property-value>
        7
    </activation-config-property-value>
</activation-config-property>

<activation-config-property>
    <activation-config-property-name>
        poolSize
    </activation-config-property-name>
    <activation-config-property-value>
        15
    </activation-config-property-value>
</activation-config-property>

<activation-config-property>
    <activation-config-property-name>
        maxNumberOfWorks
    </activation-config-property-name>
    <activation-config-property-value>
        20
    </activation-config-property-value>
</activation-config-property>

<activation-config-property>
    <activation-config-property-name>
        mdbName
    </activation-config-property-name>
```



```
        <activation-config-property-value>
            QTMdbTest
        </activation-config-property-value>
    </activation-config-property>

    <activation-config-property>
        <activation-config-property-name>
            reconnect
        </activation-config-property-name>
        <activation-config-property-value>
            true
        </activation-config-property-value>
    </activation-config-property>

    <activation-config-property>
        <activation-config-property-name>
            maxReconnect
        </activation-config-property-name>
        <activation-config-property-value>
            60
        </activation-config-property-value>
    </activation-config-property>

    <activation-config-property>
        <activation-config-property-name>
            reconnectInterval
        </activation-config-property-name>
        <activation-config-property-value>
            3000
        </activation-config-property-value>
    </activation-config-property>

    </activation-config>
</message-driven>
...
```

Pour pouvoir recevoir des messages en mode transactionnel, la méthode « onMessage » du MDB doit avoir l'attribut transactionnel :

```
...
<assembly-descriptor>
    <container-transaction>
        <method>
            <ejb-name>QTMdbTest</ejb-name>
            <method-name>*</method-name>
        </method>
        <trans-attribute>Required</trans-attribute>
    </container-transaction>
</assembly-descriptor>
...
```

4.2. jonas-ejb-jar.xml



Pour un MDB, la propriété essentielle définie dans ce fichier est le nom JNDI du module « Inbound » du connecteur JMS que le MDB utilise. Cette propriété est définie comme suit :

```
<jonas-message-driven>
  <ejb-name>DTMdbTest</ejb-name>
  <jonas-message-driven-destination>
    <jndi-name>jonasmq_as</jndi-name>
  </jonas-message-driven-destination>
  <max-cache-size>20</max-cache-size>
  <min-pool-size>10</min-pool-size>
</jonas-message-driven>
```

L'élément XML `<jndi-name>` dans ce fichier peut être utilisé autrement. Il peut définir le nom JNDI de la destination sur laquelle ce MDB écoute. Dans ce cas, ce MDB ne spécifie pas le connecteur et il utilise le connecteur par défaut (le connecteur dont le fichier `jonas-ra.xml` contient l'élément `<defaultAS>` définie comme `true`) et dans le fichier `ejb-jar.xml` de ce MDB, on ne doit pas définir la propriété `<destination>`.

`<max-cache-size>` : cette élément définit le nombre maximal des instances du MDB qui peuvent exécuter en même temps. Pour avoir la meilleure performance, cette valeur doit être identique à la propriété `maxNumberOfWorks` définie dans le fichier `ejb-jar.xml`.

`<min-pool-size>` : cette élément définit le nombre d'instances du MDB dans le pool. Pour avoir la meilleure performance, cette valeur doit être identique à la propriété `poolSize` définie dans le fichier `ejb-jar.xml`.



5. Les MBeans

Le connecteur implémente trois types de MBeans :

- Le MBean `JonasMQConnector` qui représente une instance de connecteur.
- Le MBean `MQDestination` qui représente une destination (Queue ou Topic) de MQ gérée par le connecteur.
- Le MBean `MDB` qui représente un déploiement de MDB qui utilise le connecteur.

5.1. Les MBeans `JonasMQConnector`

Ce MBean représente une instance de connecteur. Il est créé chaque fois qu'une instance du connecteur est déployée et il est détruit chaque fois que le connecteur est retiré. Ce MBean est nommé comme suit :

```
<jonas_domain>:type=JonasMQConnector,name=<mbean_name>
```

Où `<jonas_domain>` est le domaine de l'instance de JOnAS sur laquelle le connecteur est déployé ; `<mbean_name>` est le nom MBean du connecteur spécifié par la propriété de configuration `MBeanName` dans le fichier `jonas-ra.xml` du connecteur.

Voici un exemple d'un nom de ce MBean :

```
jonas:type=JonasMQConnector,name=jonasmq
```

Ce MBean contient les attributs suivants :

Nom de l'attribut	Valeur de l'attribut	Description
<code>TransportType</code>	<code>'client'</code> ou <code>'binding'</code>	Par défaut est <code>'client'</code> . Le mode « client » utilise le protocole TCP/IP tandis que le mode « binding » utilise le protocole mémoire partagée pour la connexion au gestionnaire de file d'attente. Le mode « binding » ne peut donc être utilisé que si le gestionnaire de files est exécuté sur le même serveur que JOnAS.
<code>HostName</code>	Une adresse IP ou le nom du hôte	Par défaut est <code>'localhost'</code> . Cette information doit être fournie si le mode de transport est « client ». C'est l'adresse IP ou le nom du hôte qui héberge le gestionnaire de files d'attente.



Connecteur JOnAS-MQ

Manuel d'utilisation

Extension de la console d'administration JonasAdmin

Nom de l'attribut	Valeur de l'attribut	Description
Port	Un Entier	Par défaut est 1414. Cette information doit être fournie si le mode de transport est « client ». C'est le port sur lequel le gestionnaire de files d'attente écoute (Le port défini pour le LISTENER).
Channel	Un String	Par défaut est `SYSTEM.DEF.SVRCONN`. Cette information doit être fournie si le mode de transport est « client ». C'est le nom du canal de type connexion serveur du gestionnaire de files d'attente.
QueueManager	Un String	Cette information doit être fournie si le mode de transport est « binding ». C'est le nom du gestionnaire de files d'attente.
PerformanceEventEnabled	true / false	Cet attribut indique si le gestionnaire de files permet à ses files d'attente de générer les événements concernant la performance.

Les attributs `TransportType`, `HostName`, `Port`, `Channel`, `QueueManager` correspondent aux propriétés de même nom spécifiées dans le fichier `jonas-ra.xml` du connecteur. La modification de ces attributs ne peut pas être enregistrée. Si un de ces attributs est modifié, toutes les destinations JMS créées par cette instance du connecteur seront supprimées (Les files d'attente physiques ne sont pas supprimées).

Voici toutes les opérations fournies par ce type de MBean :

- Récupérer une destination existante ou créer une nouvelle.

```
createDestination(boolean isTopic, String jndiName, String properties)
```

- Supprimer une destination.

```
deleteDestination(String jndiName, boolean deletePhysicDest)
```

- Sauvegarder la liste des destinations JMS créées par le connecteur.

```
saveDestinationConfig()
```

- Charger la liste des destinations JMS et créer toutes les destination dans cette liste.

```
loadDestinationConfig()
```

- Rafraîchir l'interface d'administration des files d'attente MQ. Après que la connexion à MQ est rétablie, il faut effectuer cette opération afin de pouvoir observer les files d'attente MQ.



```
restartAdminQm();
```

- Récupérer une propriété de l'objet ConnectionFactory.

```
String getCfProperty(String propertyName)
```

- Modifier une propriété pour l'objet ConnectionFactory.

```
setCfProperty(String propertyName, String propertyValue)
```

- Récupérer une propriété de l'objet QueueConnectionFactory.

```
String getQcfProperty(String propertyName)
```

- Modifier une propriété pour l'objet QueueConnectionFactory.

```
setQcfProperty(String propertyName, String propertyValue)
```

- Récupérer une propriété de l'objet TopicConnectionFactory.

```
String getTcfProperty(String propertyName) throws Exception;
```

- Modifier une propriété pour l'objet TopicConnectionFactory.

```
setTcfProperty(String propertyName, String propertyValue)
```

5.2. Les MBeans MQDestination

Ce MBean représente une destination MQ (Une file d'attente ou un topic) créée par le connecteur JOnAS-MQ. Ce MBean est nommé comme suit :

```
<jonas_domain>;type=MQDestination,name=<destination_jndi_name>,JonasMQConnector=<connector_mbean_name>
```

Où `<jonas_domain>` est le domaine de l'instance de JOnAS sur laquelle la destination est créée ; `<destination_jndi_name>` est le nom JNDI de la destination dans le contexte JNDI de JOnAS ; `<connector_mbean_name>` est le nom MBean du connecteur JOnAS-MQ qui a créé cette destination.

Voici un exemple d'un nom d'un MBean `MQDestination` :

```
jonas:type=MQDestination,name=QT_MDB_TEST_QUEUE,JonasMQConnector=jonasmq
```

Ce MBean contient les attributs suivants :

Nom de l'attribut	Valeur de l'attribut	Description
-------------------	----------------------	-------------



Nom de l'attribut	Valeur de l'attribut	Description
IsTopic	Boolean	Cet attribut indique si la destination est un topic ou une file d'attente. Si c'est un topic, les attributs suivants ne sont pas disponibles.
CurrentDepth	Un entier	Le nombre actuel de messages dans la file d'attente.
InputCount	Un entier	Le nombre de connexions vers la file d'attente pour recevoir les messages.
OutputCount	Un entier	Le nombre de connexions vers la file d'attente pour envoyer les messages.
MaxQueueDepth	Un entier	La capacité de la file d'attente ; Le nombre maximum de messages que la file d'attente peut stoker. Les messages qui sont envoyés vers une file d'attente pleine, seront transférés vers la file d'attente des messages morts du gestionnaire de files MQ.
DefPersistence	Boolean	Cet attribut indique si les messages sur cette file d'attente sont persistants ou non. Note : un message qui est marqué comme persistant peut être persistant sur une file d'attente non-persistante. Pour rendre un message non-persistant, il faut régler l'attribut <code>Persistence</code> de la destination JMS de cette file d'attente.

Les autres attributs d'une file d'attente exposés en interface MBean sont des attributs de performance de la file. Pour avoir plus de détails de ces attributs, veuillez consulter le document *Monitoring WebSphere MQ (Chapter 3. Understanding performance events)* qui se trouve sur le site des libs de WebSphere MQ.

Voici les opérations fournies par ce type de MBean :

- Effacer tous les messages de la file d'attente.

```
public void clearQueue();
```

- Récupérer une propriété JMS de la destination.

```
String getProperty(String propertyName) throws Exception;
```



- Modifier une propriété JMS pour la destination.

```
void setProperty(String propertyName, String propertyValue)
```

5.3. Les MBeans MDB

Ce MBean représente un MDB qui utilise le connecteur JOnAS-MQ. Ce MBean est nommé comme suit :

```
<jonas_domain>;type=MDB,name=<mdb_name>#<key>,MQDestination=<destination_jn  
di_name>,JonasMQConnector=<connector_mbean_name>
```

Où `<jonas_domain>` est le domaine de l'instance de JOnAS sur laquelle le MDB est déployé ; `<destination_jndi_name>` est le nom JNDI de la destination sur laquelle le MDB écoute ; `<connector_mbean_name>` est le nom MBean du connecteur JOnAS-MQ qui a créé cette destination. `<mdb_name>` est le nom MDB défini par l'élément XML `<mdbName>` dans le descripteur `ejb-jar.xml` du MDB. `<key>` est une clé construite à partir des `HashCode` des objets de ce MDB.

Voici un exemple d'un nom d'un MBean MDB :

```
jonas:type=MDB,name=QTMdbTest#72381965,MQDestination=QT_MDB_TEST_QUEUE,  
JonasMQConnector=jonasmq
```

Ce MBean contient les attributs suivants :

Nom de l'attribut	Valeur de l'attribut	Description
NumberOfReceivedMessages	Un entier	Le nombre de messages que ce MDB a reçus depuis qu'il est déployé.
NumberOfWorks	Un entier	Le nombre de sessions en train de recevoir des messages pour ce MDB.

Opérations fournies par un MBean MDB :

- Récupérer une propriété activation-config dans le descripteur `ejb-jar` de ce MDB.

```
Object getProperty(String propertyName)
```



ANNEXE A : REFERENCES

1. WebSphere MQ Quick Beginnings.
2. WebSphere MQ Using Java.
3. WebSphere MQ Publish/Subscribe User's Guide.
4. JMS 1.1 Specification.
5. JCA 1.5 Specification.



Connecteur JOnAS-MQ

Manuel d'utilisation

ANNEXE A

Fin de document