



Notes de Version de Sun Java System Message Queue 4.2



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Référence : 820-5640
Juillet 2008

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle de la technologie utilisée par le produit décrit dans le présent document. Notamment, mais non exclusivement, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets des États-Unis ou des demandes de brevet en attente aux États-Unis et dans d'autres pays.

Droits énoncés par le gouvernement américain – Logiciel commercial. Les utilisateurs du gouvernement sont soumis au contrat de licence standard de Sun Microsystems, Inc. ainsi qu'aux dispositions applicables du FAR et de ses suppléments.

Cette distribution peut intégrer des éléments conçus par des tiers.

Il est possible que des parties du produit soient dérivées des systèmes Berkeley BSD, concédés en licence par la University of California. UNIX est une marque déposée aux États-Unis et dans d'autres pays, exclusivement concédée en licence par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. et ses filiales aux États-Unis et dans d'autres pays. Toutes les marques déposées SPARC sont utilisées sous licence et sont des marques commerciales ou déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques déposées SPARC sont constitués selon une architecture développée par Sun Microsystems, Inc.

OPEN LOOK et l'interface graphique utilisateur SunTM sont développés par Sun Microsystems, Inc. pour ses utilisateurs et ses concessionnaires. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces utilisateur visuelles ou graphiques pour l'industrie de l'informatique. Sun est sous licence non-exclusive de Xerox pour Xerox Graphical User Interface, dont la licence couvre également les détenteurs de licence Sun qui implémentent OPEN LOOK GUIs en accord avec les contrats de licence écrits de Sun.

Les produits couverts et les informations contenues dans cette publication sont contrôlés par les lois régissant les exportations aux États-Unis et peuvent être soumises aux lois régissant les exportations ou les importations dans d'autres pays. L'utilisation d'armes nucléaires, de missiles, d'armes biologiques et chimiques ou d'armes nucléaires maritimes, qu'elle soit directe ou indirecte, est strictement interdite. Son exportation ou réexportation vers des pays soumis à l'embargo américain ou à des entités exclues des listes d'exportation américaines, notamment mais pas exclusivement, les personnes et pays figurant sur des listes noires, est strictement interdite.

LA DOCUMENTATION EST FOURNIE « EN L'ÉTAT » ET TOUTES LES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

Table des matières

1	Notes de version de Sun Java System Message Queue 4.2	5
	Historique de révision des notes de mise à jour	6
	Installation ou mise à niveau de Message Queue 4.2	7
	Message Queue 4.2 Plate-formes et composants pris en charge	7
	Prise en charge de la plate-forme du système exploitation	7
	Prise en charge de la virtualisation du système	8
	Dépendances entre composants	8
	Nouvelles fonctions de Message Queue 4.2 et versions récentes	10
	Nouvelles fonctions de Message Queue 4.2	10
	Nouvelles fonctions de Message Queue 4.1	18
	Nouvelles fonctions de Message Queue 4.0	22
	Fonction sera désapprouvée dans les versions futures	26
	Bogues résolus dans Message Queue 4.2 et ses versions récentes	26
	Bogues résolus dans Message Queue 4.2	27
	Bogues résolus dans Message Queue 4.1	28
	Bogues résolus dans Message Queue 4.0	29
	Mises à jour de la documentation relative à Message Queue 4.2	30
	Problèmes de compatibilité	30
	Modifications dans le jeu de documentation de Message Queue 4.2	31
	Nouvelles métriques de destination	32
	Démarrage automatique de courtier sur le SE Solaris 10	33
	Modifications apportées à JMX API	34
	Prise en charge pour le format du nom d'utilisateur dynamique pour l'authentification de client	39
	Amélioration de l'authentification JAAS	39
	Problèmes connus et restrictions	40
	Problèmes d'installation	40
	Option de mot de passe désapprouvée	46
	Problèmes d'administration/de configuration	47

Problèmes relatifs au courtier	48
Clusters de courtiers	49
Problèmes relatifs à JMX	51
Prise en charge de SOAP	51
Fichiers redistribuables	52
Accessibilité des personnes handicapées	52
Comment signaler des problèmes et apporter des commentaires	52
Forum de Sun Java System	53
Forum sur la technologie Java	53
Vos commentaires sont les bienvenus	53
Ressources Sun supplémentaires	53

Notes de version de Sun Java System Message Queue 4.2

Version 4.2

Numéro de référence 820-3701

Ces notes de version contiennent des informations importantes, disponibles au moment de la commercialisation de Sun Java™ System Message Queue 4.2. Vous y trouverez une description des nouvelles fonctions et des améliorations, des problèmes connus et des limites, ainsi que d'autres informations sur cette version. Lisez attentivement ce document avant de commencer à utiliser Message Queue 4.2.

Ces notes de version contiennent également des informations sur les versions 4.0 et 4.1 de Message Queue: Reportez-vous à “Nouvelles fonctions de Message Queue 4.0” à la page 22 et “Nouvelles fonctions de Message Queue 4.1” à la page 18, respectivement, pour obtenir des informations sur les fonctions introduites dans ces versions.

Vous trouverez la dernière édition de ces notes de version sur le site Web de documentation de Sun Java System Message Queue <http://docs.sun.com/coll/1307.3> Consultez ce site Web avant d'installer et de configurer votre logiciel, puis consultez-le régulièrement pour vous procurer la documentation concernant le produit et les notes de mise à jour les plus récentes.

Ces notes de version se composent des sections suivantes :

- “Historique de révision des notes de mise à jour” à la page 6
- “Installation ou mise à niveau de Message Queue 4.2” à la page 7
- “Message Queue 4.2 Plate-formes et composants pris en charge” à la page 7
- “Nouvelles fonctions de Message Queue 4.2 et versions récentes” à la page 10
- “Fonction sera désapprouvée dans les versions futures” à la page 26
- “Bogues résolus dans Message Queue 4.2 et ses versions récentes” à la page 26
- “Modifications dans le jeu de documentation de Message Queue 4.2” à la page 31
- “Problèmes connus et restrictions” à la page 40
- “Fichiers redistribuables” à la page 52
- “Accessibilité des personnes handicapées” à la page 52
- “Comment signaler des problèmes et apporter des commentaires” à la page 52

- “Vos commentaires sont les bienvenus” à la page 53
- “Ressources Sun supplémentaires” à la page 53

Des URL de sites tiers, qui renvoient à des informations complémentaires connexes, sont référencés dans ce document.

Sun ne saurait être tenu responsable de la disponibilité des sites Web tiers mentionnés dans ce manuel. Sun décline toute responsabilité quant au contenu, à la publicité, aux produits ou tout autre matériel disponibles dans ou par l’intermédiaire de ces sites ou ressources. Sun ne pourra en aucun cas être tenu responsable, directement ou indirectement, de tous dommages ou pertes, réels ou invoqués, causés par ou liés à l’utilisation des contenus, biens ou services disponibles dans ou par l’intermédiaire de ces sites ou ressources.

Historique de révision des notes de mise à jour

Le tableau suivant répertorie les dates correspondant à toutes les versions 4.x du produit Message Queue et décrit, pour chaque version, les modifications apportées à ce document.

TABLEAU 1-1 Historique des révisions

Date	Description des modifications
Mai 2006	Version initiale de ce document pour Message Queue 4.0.
Janvier 2007	Version initiale de ce document pour Message Queue 4.1 Beta. Ajout d'une description concernant la prise en charge de JAAS.
Avril 2007	Seconde version de ce document pour Message Queue 4.1 Beta. Ajout de la fonctionnalité de haute disponibilité.
Septembre 2007	Troisième version de ce document pour Message Queue 4.1. Ajoute une description de prise en charge pour Java Enterprise System Monitoring Framework, ports C fixes, corrections de bogues et autres fonctions.
Avril 2008	Première version bêta de ce document pour Message Queue 4.2. Ajoute de nouvelles fonctions pour cette version.

Installation ou mise à niveau de Message Queue 4.2

Vous pouvez procéder à une nouvelle installation de Message Queue 4.2 ou à une mise à niveau à partir de la version Message Queue 3.6 ou supérieure en utilisant le programme d'installation de Message Queue 4.2. La procédure et toutes les autres informations pertinentes pour l'installation ou la mise à niveau sur les plate-formes Solaris, Linux et Windows sont documentées dans le Guide d'installation de *Sun Java System Message Queue 4.2 Installation Guide*, qui n'a pas été mis à jour pour Message Queue 4.2.

Si vous procédez à une mise à niveau à partir d'une version de Message Queue antérieure à la version 3.6, reportez-vous au Guide de mise à niveau de *Sun Java Enterprise System 5 pour UNIX*, et au Guide de mise à niveau de *Sun Java Enterprise System 5 Update 1 Upgrade Guide for UNIX*.

Vérifiez également les “[Problèmes d'installation](#)” à la page 40 pour les problèmes et les limites d'installation et de mise à niveau connus.

Message Queue 4.2 Plate-formes et composants pris en charge

Cette section couvre les sujets suivants liés aux exigences du système Message Queue 4.2 requises :

- “[Prise en charge de la plate-forme du système exploitation](#)” à la page 7
- “[Prise en charge de la virtualisation du système](#)” à la page 8
- “[Dépendances entre composants](#)” à la page 8

Prise en charge de la plate-forme du système exploitation

Message Queue 4.2 est pris en charge par les plate-formes des systèmes d'exploitation Solaris, Linux et Windows. Le [Tableau 1–2](#) présente les versions prises en charge pour chaque plate-forme. Pour les configurations matérielles requises de chaque plate-forme, reportez-vous au *Sun Java System Message Queue 4.2 Installation Guide*.

TABLEAU 1–2 Versions de plate-formes prises en charge

Plate-forme	Versions prises en charge
Solaris	Solaris 9 (SunOS 5.9), toutes les mises à jour (SPARC, x86)
	Solaris 10 (SunOS 5.10), toutes les mises à jour (SPARC, x86, x64)

TABLEAU 1-2 Versions de plate-formes prises en charge (Suite)

Plate-forme	Versions prises en charge
Linux	Red Hat Enterprise Linux Advanced Server 3.0, 4.0, 5.0, toutes les mises à jour, version 32- et 64-bit (x86, x64) Red Hat Enterprise Linux Enterprise Server 3.0, 4.0, 5.0, toutes les mises à jour, versions 32- et 64-bit (x86, x64)
Windows	Windows Vista Windows XP Professional, SP2 (x86) ¹ Windows 2000 Advanced Server, SP4 (x86) ² Windows Server 2003 Standard Enterprise Editions, SP2, versions 32- et 64-bit (x86, x64) ³

¹ Pas de prise en charge sur Home, tablette PC ou Media Center Edition

² Pas de prise en charge de Windows 2000 Professional ou Server Edition

³ Pas de prise en charge Web ou Small Business Server Edition

Prise en charge de la virtualisation du système

La virtualisation du système est une technologie permettant l'exécution indépendante de plusieurs instances du système d'exploitation sur un matériel partagé. Le logiciel déployé sur un système d'exploitation hébergé dans un environnement virtualisé ne détecte généralement pas que la plate-forme a été virtualisée. Sun teste ses produits Sun Java System sur des combinaisons de virtualisation de système et de systèmes d'exploitation afin de s'assurer qu'ils fonctionnent aussi bien dans des environnements virtualisés correctement dimensionnés et configurés que dans des environnements non virtualisés. Pour plus d'informations sur la prise en charge des produits Sun Java System dans les environnements virtualisés, reportez-vous à la page <http://docs.sun.com/doc/820-46511>.

Dépendances entre composants

Outre des configurations requises spécifiques à la plate-forme, Message Queue 4.2 dépend également de certains composants de base à installer pour développer et exécuter les clients Message Queue. [Tableau 1-3](#) décrit ces composants. Vous pouvez également utiliser d'autres versions ou implémentations d'autres fournisseurs mais ces dernières n'ont pas fait l'objet de tests par Sun Microsystems et ne sont, par conséquent, pas prises en charge officiellement.

Remarque – Le programme d'installation de Message Queue vous permet de sélectionner une version JDK/JRE existante ou d'installer la version JDK version (1.5.0_15).

TABLEAU 1-3 Composants de prise en charge requis

Composant	Prend en charge	Versions prises en charge
Java Runtime Environment (JRE)	Outils d'administration et de courtier Message Queue	J2SE™ Runtime Environment version 1.5.0_15 ou supérieure Java™ SE Runtime Environment 1.6.0 (versions Sun Microsystems uniquement)
Java Software Development Kit (JDK), Standard Edition	Développement et déploiement du client Java	J2SE™ Development Kit version 1.5.0_15 ou supérieure Java SE Development Kit 1.6.0 (versions de production Sun Microsystems uniquement)

Tableau 1-4 présente les composants supplémentaires que vous pouvez installer pour fournir une plus grande prise en charge des clients de Message Queue. Vous n'avez peut-être pas besoin de tous les composants de la liste : par exemple, si vous n'écrivez pas pour un client C, vous n'aurez pas besoin du compiler C, de la librairie d'exécution C++ runtime library, de NSPR ou de NSS.

TABLEAU 1-4 Composants de prise en charge en option

Composant	Prend en charge	Versions prises en charge
Serveur d'applications	HTTP/HTTPS	Sun Java System Application Server Enterprise Edition, Version 9.1, mise à jour version 2
Serveur web	HTTP/HTTPS	Sun Java System Web Server Enterprise Edition, Version 7.0, mise à jour 2
Base de données	Magasin de données basé sur JDBC	HADB, Version 4.4.3.5 Java DB (Apache Derby), Version 10.2.2 MySQL Community Edition, Version 5.0 Oracle10g postgresql, Version 8.1 Remarque – La base de données PointBase n'est plus prise en charge.
Base de données hautement disponible	Clusters de courtiers haute disponibilité	HADB, Version 4.4.3.5 MySQL Cluster Edition, Version 5.0 Oracle10g

TABLEAU 1-4 Composants de prise en charge en option (Suite)

Composant	Prend en charge	Versions prises en charge
Serveur de répertoires Lightweight Directory Access Protocol (LDAP)	Référentiel utilisateur et objets administrés de Message Queue	Sun Java System Directory Server, Version 6.0
IJNDI (Java Naming and Directory Interface, interface de répertoire et de nommage Java) ;	Prise en charge des objets administrés et référentiel utilisateur LDAP	JNDI Version 1.2.1 LDAP Service Provider, Version 1.2.2 File System Service Provider, Version 1.2 Beta 3 ¹
Compilateur C et bibliothèque d'exécution compatible C++	Clients C de Message Queue	Solaris: Sun Studio, Version 11 ou supérieure, compilateur C++ avec mode standard et compilateur C Linux: gcc/g++, Version 3.2.3 Windows: Microsoft Windows Visual C++, Version 6.0 SP3
Netscape Portable Runtime (NSPR)	Clients C de Message Queue	Version 4.7-1 ²
Network Security Services (NSS)	Clients C de Message Queue	Version 3.11.9-1 ²

¹ Prise en charge des objets administrés uniquement; pris en charge pour le développement et les tests, mais pas pour le déploiement dans un environnement de production

² Regroupés dans un package partagé dans le set de téléchargement

Nouvelles fonctions de Message Queue 4.2 et versions récentes

Les nouvelles fonctions de Message Queue 4.2, versions 4.2 et 4.0 sont décrites dans les sections suivantes :

- [“Nouvelles fonctions de Message Queue 4.2”](#) à la page 10
- [“Nouvelles fonctions de Message Queue 4.1”](#) à la page 18
- [“Nouvelles fonctions de Message Queue 4.0”](#) à la page 22

Nouvelles fonctions de Message Queue 4.2

Sun Java System Message Queue est un service de messagerie complet qui offre des fonctionnalités de messagerie asynchrones et fiables conformes à la spécification de messagerie Java (JMS) 1.1. En outre, Message Queue propose des fonctionnalités supplémentaires par rapport à la spécification JMS pour répondre aux besoins des déploiements d'entreprise à grande échelle.

Message Queue 4.2 est une version mineure apportant un certain nombre d'améliorations de fonctions et de corrections de bogues. Cette section explique comment installer ou mettre à niveau Message Queue 4.2 et décrit les nouvelles fonctions incluses dans cette version :

- “Destinations multiples pour un éditeur ou un abonné” à la page 11
- “Validation des schémas des messages de charge utile XML” à la page 13
- “Prise en charge de C-API pour les transactions distribuées” à la page 14
- “Prise en charge du programme d'installation pour l'enregistrement à Sun Connection” à la page 17
- “Prise en charge de la base de données MySQL” à la page 18

Pour de plus amples informations sur les fonctions introduites dans Message Queue 4.0 et 4.1, reportez-vous respectivement aux sections “Nouvelles fonctions de Message Queue 4.0” à la page 22 et “Nouvelles fonctions de Message Queue 4.1” à la page 18.

Destinations multiples pour un éditeur ou un abonné

Dans Message Queue 4.2, un éditeur peut maintenant publier des messages vers de multiples destinations de sujet et un abonné peut recevoir des messages provenant de multiples destinations de sujet. Cette capacité est obtenue en utilisant un nom de destination de sujet incluant des caractères génériques, permettant ainsi de multiples destinations. L'utilisation de noms symboliques de ce type permet aux administrateurs de créer des destinations de sujet supplémentaires, selon le cas, de façon cohérente avec le schéma de nommage générique. Les éditeurs et les abonnés publient et consomment automatiquement à partir des destinations ajoutées. (Les abonnés de sujet générique sont plus courants que les éditeurs.)

Remarque – Cette fonction ne s'applique pas aux destinations de files d'attente.

Le format du nom de la destination du sujet symbolique se compose de multiples segments, dans lesquels les caractères génériques (*, **, >) peuvent représenter un ou plusieurs segments du nom. Par exemple, supposons que vous ayez un schéma de nommage de destination comme suit :

taille.couleur.forme

où les segments de nom du sujet avoir les valeurs suivantes :

- *taille*: grand, moyen, petit, ...
- *couleur*: rouge, vert, bleu, ...
- *forme*: cercle, triangle, carré, ...

Message Queue prend en charge les caractères génériques suivants :

- * correspond à un seul segment
- ** correspond à un ou plusieurs segments
- > correspond à un nombre quelconque de segments successifs

Vous pouvez par conséquent indiquer de multiples destinations de sujet comme suit :

`grand.*.cercle` représenterait :

```
grand.rouge.cercle
grand.vert.cercle
...
```

`**.``carré` représenterait tous les noms se terminant par `.carré`, par exemple :

```
petit.vert.carré
moyen.bleu.carré
...
```

`petit.>` représenterait tous les noms de destination commençant par `petit.`, par exemple :

```
petit.bleu.cercle
petit.rouge.carré
...
```

Pour utiliser cette fonction de multiples destinations, vous créez des destinations de sujet à l'aide du schéma de nommage similaire à la description ci-dessus. Les applications client peuvent ensuite créer des éditeurs ou des consommateurs en utilisant des noms de destination symboliques. Exemple :

```
...
Chaîne DEST_LOOKUP_NAME = "grand.*.cercle";
Sujet t = (Destination) ctx.lookup(DEST_LOOKUP_NAME);
TopicPublisher myPublisher = mySession.createPublisher(t);
myPublisher.send(myMessage);

...
Chaîne DEST_LOOKUP_NAME = "**.carré";
Sujet t = (Destination) ctx.lookup(DEST_LOOKUP_NAME);
TopicSubscriber mySubscriber = mySession.createSubscriber(t);
Message m = mySubscriber.receive();
```

Dans le premier exemple, le courtier placera une copie du message sous une quelconque destination correspondant au nom symbolique `grand.*.cercle`. Dans le second exemple, un abonné sera créé si au moins une destination correspond au nom symbolique `**.``carré` et il recevra des messages de toutes les destinations correspondant à ce nom symbolique. Si aucune destination ne correspond au nom symbolique, l'éditeur ne sera créé que lorsqu'il existera une destination.

Si un administrateur crée des destinations supplémentaires correspondant à un nom symbolique, alors les éditeurs génériques créés à l'aide du nom symbolique publieront par la suite vers cette destination et les éditeurs génériques créés à l'aide de ce nom symbolique recevront par la suite des messages à partir de cette destination.

De plus, les outils d'administration de Message Queue, outre le fait de signaler le nombre total d'éditeurs (producteurs) et d'abonnés (consommateurs) pour une destination de sujet, signaleront également s'ils existent, le nombre des éditeurs génériques et des abonnés génériques (y compris, pour chacun, leurs noms de destination symboliques).

Validation des schémas des messages de charge utile XML

Cette nouvelle fonction de Message Queue 4.2 permet de valider le contenu d'un message XML texte (pas objet) XML par rapport à un schéma XML au moment où le message est envoyé au courtier. L'emplacement du schéma XML (XSD, XML Schema Directory) est spécifié comme une propriété d'une destination de Message Queue. Si aucun emplacement XSD n'est spécifié, la déclaration DTD du document XML est utilisée pour exécuter une validation DTD. (La validation XSD, qui comprend la validation du type de données et de la fourchette de valeurs, est plus rigoureuse que la validation DTD.)

Les applications client utilisant cette nouvelle fonction doivent procéder à une mise à niveau de Java version SE à JRE version 1.5 ou supérieure.

Afin d'activer la validation du schéma XML, vous devez définir les propriétés de destination physiques suivantes :

TABLEAU 1-5 Propriétés de destination physiques pour la validation du schéma XML

Propriétés	Type	Valeur par défaut	Description
validateXMLSchemaEnabled	Booléen	false	Validation du schéma XML activée ? Si l'option est définie non définie ou définie sur false, alors la validation du schéma XML n'est pas activée pour la destination.
XMLSchemaURLList	String (Chaîne)	null	Liste des chaînes URI du document (XSD) du schéma XML, séparées par des espaces Les URI pointent vers l'emplacement d'un ou plusieurs XSD à utiliser pour la validation du schéma XML, si cette option est activée. Mettez cette valeur entre guillemets si plusieurs URI sont spécifiés. Exemple : "http://foo/flap.xsd http://test.com/test.xsd" Si cette propriété n'est pas définie ou null et si la validation XML est activée, la validation XML est effectuée à l'aide d'un DTD spécifié dans le document XML.

TABLEAU 1-5 Propriétés de destination physiques pour la validation du schéma XML (Suite)

Propriétés	Type	Valeur par défaut	Description
<code>reloadXMLSchemaOnFailure</code>	Booléen	<code>false</code>	Rechargement du schéma XML en cas d'échec activé ? Si l'option est définie non définie ou définie sur <code>false</code> , alors le schéma n'est pas rechargé en cas d'échec de la validation.

Lorsque la validation XML est activée, l'exécution du client Message Queue tentera de valider un message XML par rapport aux XSD spécifiés (ou par rapport au DTD, si aucun XSD n'est spécifié) avant de l'envoyer au courtier. S'il est impossible de localiser le schéma spécifié ou si le message ne peut pas être validé, le message n'est pas envoyé et une exception est émise.

Les propriétés de validation XML peuvent être définies lors de la création de la destination ou au moment de la mise à jour en utilisant la commande `imqcmd create dst` ou `imqcmd update dst`, respectivement. Les propriétés de validation XML doivent être définies lorsqu'une destination est inactive : c'est-à-dire lorsqu'il n'y a ni consommateurs ni producteurs et lorsqu'il n'y a pas de messages dans la destination.

Remarque – Si un XSD n'est pas accessible lors de l'exécution, il peut s'avérer nécessaire de modifier la `XMLSchemaURList` pendant qu'une destination est active.

Si l'une quelconque des propriétés de validation XML alors qu'une destination est active (par exemple, si un producteur est connecté à la destination), ce changement prendra effet lors de la prochaine connection du producteur au courtier. De même, si un XSD est modifié, à la suite d'un changement des configurations requises de l'application, toutes les applications client produisant des messages XML sur la base du XSD modifié doivent se reconnecter au courtier.

Si la propriété `reloadXMLSchemaOnFailure` est définie comme `true` que la validation XML échoue, alors l'exécution du client Message Queue tentera de recharger le XSD avant d'essayer à nouveau de valider un message. L'exécution du client émettra une exception si la validation échoue en utilisant le XSD rechargé.

Prise en charge de C-API pour les transactions distribuées

Selon le modèle de transaction distribuée X/Open, la prise en charge pour les transactions distribuées dépend d'un gestionnaire de transactions distribuées qui suit et gère les opérations exécutées par un ou plusieurs gestionnaires de ressources. Dans Message Queue 4.2, le C-API de Message Queue prend désormais en charge l'interface XA (entre un gestionnaire de transactions distribuées et Message Queue sous forme de gestionnaire de ressources conforme à XA), permettant aux clients C-API de Message Queue s'exécutant dans un environnement de traitement des transactions distribuées (comme BEA Tuxedo) de participer aux transactions distribuées.

Cette prise en charge des transactions distribuées se compose des nouvelles fonctions C-API suivantes (et de nouveaux paramètres et codes d'erreur) utilisées pour implémenter la spécification de l'interface XA :

```
MQGetXAConnection()
```

```
MQCreateXASession()
```

Si une application client C doit être utilisée dans le contexte d'une transaction distribuée, alors elle doit obtenir une connexion à l'aide de `MQGetXAConnection()` et créer une session pour produire et consommer des messages à l'aide de `MQCreateXASession()`. Le lancement, la validation et la répétition de toute transaction distribuée sont gérés via des API fournies par le gestionnaire de transactions distribuées.

Informations publiques

La spécification de l'interface X/Open XA requiert les informations publiques suivantes concernant le gestionnaire de ressources conforme à XA de Message Queue :

- Nom de la structure `xa_switch_t` : `sun_my_xa_switch`
- Nom du gestionnaire de ressources : `SUN_RM`
- La bibliothèque MQ C-API doit être liée : `mqcrt`
- La chaîne `xa_close` et le format : `none`
- La chaîne `xa_open` et le format : “;” separated name=value pairs

Les paires de nom/valeur suivantes sont prises en charge :

TABLEAU 1-6 Paires de nom/valeur du gestionnaire de ressources de Message Queue

Nom	Valeur	Description	Par défaut
adresse	<i>hôte:port</i>	L'hôte:port du service Portmapper du courtier.	<code>localhost:7676</code>
Identifiant (username)	chaîne	L'identifiant (username) de connexion au courtier	<code>hôte (guest)</code>
mot de passe (password)	chaîne (string)	Le mot de passe de l'identifiant	<code>hôte</code>
conntype	TCP ou SSL	Le type de protocole de la connexion au courtier	TCP
Hôte sécurisé	<code>true/false</code>	détermine si l'hôte du courtier est sécurisé ou non (uniquement applicable pour <code>conntype=SSL</code>)	<code>true</code>

TABLEAU 1-6 Paires de nom/valeur du gestionnaire de ressources de Message Queue (Suite)

Nom	Valeur	Description	Par défaut
certdbpath	chaîne	Le chemin complet vers le répertoire qui contient un certificat NSS et des fichiers de base de données clé	non défini
clientid	chaîne	Requis uniquement pour les abonnements JMS durables	non défini
se reconnecte	entier	Le nombre de tentatives de reconnexion au courtier (0 signifie aucune reconnexion)	0

Exemples de programmation

Pour programmer une application utilisant des transactions distribuées, vous créez un service côté serveur qui s'exécute dans l'environnement du gestionnaire de transactions un code côté client appelant les API du gestionnaire de transactions. Message Queue 4.2 fournit des exemples de programmation sur la base du gestionnaire de transactions Tuxedo. Ces exemples sont situés dans le répertoire de programmes d'exemples sur chaque plate-forme, dans le répertoire `./C/tuxedo`.

Ce répertoire comprend un fichier README qui explique comment paramétrer Tuxedo pour utiliser le gestionnaire de ressources de Message Queue et comment développer les programmes d'exemples suivants dans l'environnement Tuxedo :

Programme d'exemples	Description
<code>jmsserver.c</code>	Implémente les services Tuxedo qui envoient et reçoivent des messages à l'aide de Message Queue.
<code>jmsclient_sender.c</code>	Client Tuxedo qui utilise le service de génération de messages dans le programme <code>jmsserver.c</code> .
<code>jmsclient_receiver.c</code>	Client Tuxedo qui utilise le service de réception de messages dans le programme <code>jmsserver.c</code> .
<code>async_jmsserver.c</code>	Implémente un service Tuxedo qui consomme de façon asynchrone des messages en utilisant Message Queue.
<code>jmsclient_async_receiver.c</code>	Client Tuxedo qui utilise le service de consommation de messages asynchrones dans le programme <code>jmsserver.c</code> .

Prise en charge du programme d'installation pour l'enregistrement à Sun Connection

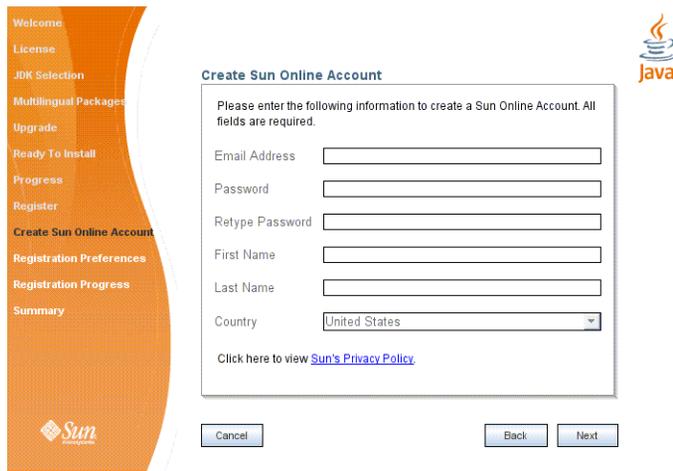
Le programme d'installation de Message Queue a été amélioré pour permettre l'enregistrement de Message Queue avec Sun Connection, un service hébergé par Sun qui vous aide à suivre, organiser et maintenir votre logiciel et matériel Sun.

Lors de l'installation de Message Queue, vous pouvez choisir d'enregistrer Message Queue avec Sun Connection. Les informations concernant le programme Message Queue installé, comme la version, le nom d'hôte, le système d'exploitation, la date d'installation et d'autres informations de base du même type, sont transmis de façon sécurisée à la base de données de Sun Connection. Le service d'inventaire de Sun Connection peut vous aider à organiser votre logiciel et matériel Sun, tandis que le service de mise à jour peut vous informer des dernières corrections de sécurité disponibles, des mises à jour recommandées et des améliorations des fonctions.

L'écran suivant du programme d'installation a été ajouté à Message Queue 4.2 pour l'enregistrement à Sun Connection :

The screenshot shows the 'Register' dialog box in the Message Queue 4.2 installation wizard. The dialog is titled 'Register' and asks 'Why register? Gain convenient access to'. It lists benefits: patch information and bug updates, screencasts and tutorials, and support and training offerings. There are three radio button options: 'I don't have a Sun Online Account. Sign me up.' (selected), 'I already have a Sun Online Account.', and 'Skip Registration.'. Below the options are input fields for 'User Name:' and 'User Password:'. At the bottom are 'Cancel', 'Back', and 'Next' buttons. The background shows the installation wizard's progress bar with 'Register' highlighted, and the Sun logo.

Pour vous enregistrer, vous devez avoir un compte Sun Online ou en créer un. Si vous n'avez pas encore de compte, le programme d'installation affiche l'écran suivant pour créer un compte Sun Online :



Vous pouvez enregistrer Message Queue au cours de l'installation en utilisant les écrans ci-dessus, ou attendre que l'installation soit terminée pour exécuter le programme d'installation sous le mode enregistrement uniquement, comme suit :

```
# installer -r
```

Le mode enregistrement uniquement nécessite que Message Queue 4.2 soit déjà installé et affichera uniquement les écrans du programme d'installation liés à l'enregistrement.

Prise en charge de la base de données MySQL

Message Queue 4.2 prend en charge la base de données MySQL comme un magasin de données basé sur JDBC. MySQL Cluster Edition peut être utilisé comme base de données JDBC pour un courtier autonome et comme le magasin de données partagées hautement nécessaire à un cluster de courtiers à haute disponibilité. Pour de plus amples informations sur la configuration de Message Queue pour utiliser MySQL, reportez-vous à “Configuring a JDBC-Based Data Store” du *Sun Java System Message Queue 4.2 Administration Guide* et également aux “High-Availability Cluster Properties” du *Sun Java System Message Queue 4.2 Administration Guide*.

Nouvelles fonctions de Message Queue 4.1

Message Queue 4.1 était une version mineure apportant un certain nombre de nouvelles fonctions, des améliorations de fonctions et des corrections de bogues. Cette section décrit les nouvelles fonctions de la version 4.1 et fournit de plus amples références destinées à votre usage :

- “Clusters de courtiers haute disponibilité” à la page 19

- “Prise en charge de JAAS” à la page 20
- “Modification du format du magasin de données persistant” à la page 20
- “Configuration de l’environnement du courtier” à la page 21
- “Prise en charge de Java ES Monitoring Framework” à la page 21
- “Gestion améliorée des transactions” à la page 21
- “Ports fixes pour les connexions de clients C” à la page 21

Pour obtenir des informations sur les nouvelles fonctionnalités de Message Queue 4.0, reportez-vous à la section “Nouvelles fonctions de Message Queue 4.0” à la page 22.

Clusters de courtiers haute disponibilité

Message Queue 4.1 a introduit des clusters de courtiers haute disponibilité. Comparés aux clusters de courtiers traditionnels assurant la disponibilité du *service de messagerie* (si un courtier échoue, un autre courtier est disponible pour fournir un service de messagerie), les clusters de courtiers haute disponibilité assurent eux la disponibilité des *données* (si un courtier échoue, ses messages persistants et données d’état sont disponibles pour qu’un autre courtier les utilise pour le relayer et délivrer les messages).

La nouvelle implémentation haute disponibilité de Message Queue 4.1 utilise un magasin de données basé sur JDBC partagé : Le courtier d’un cluster n’a plus de magasin de données persistantes propre mais partage la même base de données (conforme à JDBC) avec tous les autres courtiers du cluster. Si un courtier particulier échoue, un autre courtier du cluster prend sa relève pour acheminer et livrer ses message. Le courtier de basculement utilise alors les données et les informations d’état du magasin de données partagées. Les clients du courtier défaillant ayant émis les messages se reconnectent au courtier de basculement, et le service de messagerie est ininterrompu.

Le magasin partagé basé sur JDBC utilisé dans l’implémentation haute disponibilité de Message Queue 4.1 doit être lui-même hautement disponible. Si vous ne possédez pas de base de données hautement disponible ou si la livraison ininterrompue de messages n’est pas importante pour vous, vous pouvez continuer à utiliser les clusters traditionnels qui fournissent une disponibilité de service sans disponibilité de données.

Pour configurer un cluster de courtiers haute disponibilité de Message Queue 4.1, vous devez spécifier les propriétés de courtier suivantes pour chaque courtier du cluster :

- *Propriétés d’appartenance au cluster* : spécifient l’appartenance du courtier à un cluster haute disponibilité, l’ID de ce cluster et l’ID du courtier dans le cluster.
- *Propriétés de base de données hautement disponible* : spécifient le modèle de données persistantes (JDBC), le nom du fournisseur et les propriétés de configuration spécifiques au fournisseur.
- *Propriétés de détection d’échecs et de basculement* : spécifient comment un échec de courtier est détectée et géré à l’aide d’un courtier de substitution.

L’implémentation de clusters de courtiers haute disponibilité se fait selon les étapes suivantes :

1. Installez une base de données hautement disponible.
2. Installez le fichier .jar du pilote JDBC.
3. Créez le schéma de base de données pour le magasin de données persistant hautement disponible.
4. Paramétrez des propriétés haute disponibilité pour chaque courtier du cluster.
5. Démarrez chaque courtier du cluster.

Si vous souhaitez consulter une discussion conceptuelle sur les clusters de courtier haute disponibilité et obtenir une comparaison de ceux-ci par rapport aux clusters conventionnels, reportez-vous au [Chapitre 4, “Broker Clusters” du *Sun Java System Message Queue 4.2 Technical Overview*](#). Pour obtenir des informations procédurales et référentielles sur les clusters de courtiers haute disponibilité, reportez-vous au [Chapitre 8, “Managing Broker Clusters” du *Sun Java System Message Queue 4.2 Administration Guide*](#) et à la section “Cluster Configuration Properties” du *Sun Java System Message Queue 4.2 Administration Guide*.

Si vous avez utilisé une base de données hautement disponible avec Message Queue 4.0 et si vous souhaitez passer à un cluster de courtiers haute disponibilité, vous pouvez utiliser l'utilitaire Gestionnaire de base de données (`imqdbmgr` pour une conversion vers un magasin de données persistant partagé. Reportez-vous également à “Clusters de courtiers” à la [page 49](#) pour plus de problèmes et restrictions connus.

Prise en charge de JAAS

En plus des mécanismes d'authentification intégrés basés sur les fichiers et le LDAP, Message Queue introduit également la prise en charge de JAAS (Java Authentication and Authorization Service), qui vous permet de connecter un mécanisme d'authentification au courtier pour authentifier les clients Message Queue.

Pour obtenir une description des informations qu'un courtier met à disposition à un service d'authentification conforme à JAAS et une explication sur la façon de configurer le courtier pour utiliser un tel service, reportez-vous à “Using JAAS-Based Authentication” du *Sun Java System Message Queue 4.2 Administration Guide*.

Modification du format du magasin de données persistant

Message Queue 4.1 a modifié le magasin de données basé sur JDBC pour prendre en charge des clusters de courtiers haute disponibilité. Pour cette raison, le format du magasin de données basé sur JDBC— est augmenté pour passer à la version 410. Les versions de format 350, 370 et 400 sont migrées automatiquement vers la version 410.

Veillez noter que le format du magasin de données persistant basé sur les fichiers reste à la version 370 car aucun changements n'ont été apportés à cette version.

Configuration de l'environnement du courtier

La propriété `IMQ_DEFAULT_EXT_JARS` a été ajoutée au fichier de configuration de l'environnement Message Queue 4.1 `imqenv.conf`. Vous pouvez définir cette propriété pour spécifier les noms de chemin des fichiers `.jar` externes à inclure dans `CLASSPATH` au démarrage du courtier. Si vous utilisez cette propriété pour spécifier l'emplacement des fichiers `JAR` externes, il ne vous sera plus nécessaire de copier ces fichiers dans le répertoire `lib/ext`. Ces fichiers `.jar` externes peuvent faire référence aux pilotes `JDBC` ou aux modules de connexion `JAAS`. L'exemple de propriété suivant spécifie l'emplacement des pilotes `JDBC`.

```
IMQ_DEFAULT_EXT_JARS=/opt/SUNWhadb4/lib/hadbjdbc4.jar:/opt/SUNWjavadb/derby.jar
```

Prise en charge de Java ES Monitoring Framework

Message Queue 4.1 a introduit la prise en charge de Sun Java Enterprise System (Java ES) Monitoring Framework, qui permet de surveiller les composants Java ES à l'aide d'une interface graphique usuelle. Cette interface est implémentée par une console basée sur le Web, Sun Java System Monitoring Console. Les administrateurs peuvent utiliser la console pour visualiser les statistiques de performance, créer des règles de surveillance automatique et accuser réception d'alarmes. Si vous exécutez Message Queue en même temps que d'autres composants Java ES, vous trouverez peut-être plus pratique d'utiliser une seule interface pour tous les gérer.

Pour obtenir des informations sur l'utilisation de Java ES monitoring framework pour surveiller Message Queue, reportez-vous à [XREF](#).

Gestion améliorée des transactions

Auparavant, seules les transactions à l'état `PREPARED` pouvaient être annulées administrativement. Par exemple, lorsqu'une session, faisant partie d'une transaction distribuée, ne s'arrêtait pas normalement, la transaction était conservée à un état qui ne pouvait pas être nettoyé par un administrateur. Dans Message Queue 4.1, vous pouvez désormais utiliser l'utilitaire Commande (`imqcmd`) pour nettoyer (annuler) des transactions qui sont à l'état suivant : `STARTED`, `FAILED`, `INCOMPLETE`, `COMPLETE` et `PREPARED`.

Pour vous aider à déterminer si une transaction particulière peut être annulée ou pas (en particulier lorsque celle-ci n'est pas à l'état `PREPARED`), l'utilitaire Commande fournit des données supplémentaires dans la sortie `imqcmd query txn` : il fournit l'ID de connexion pour la connexion ayant démarré la transaction et spécifie l'heure de création de cette dernière. À l'aide de ces informations, un administrateur peut alors décider d'annuler ou pas la transaction. En règle générale, il est préférable que l'administrateur évite d'annuler une transaction prématurément.

Ports fixes pour les connexions de clients C

Dans Message Queue 4.1, les clients `C`, comme les clients `Java`, peuvent maintenant se connecter à un port de courtier fixe plutôt qu'à un port affecté de façon dynamique par le service du

courtier Port Mapper. Les connexions de port fixe sont utiles si vous essayez de franchir un pare-feu ou si avez besoin de contourner le service Port Mapper pour d'autres raisons.

Pour configurer une connexion de port fixe, vous devez configurer à la fois l'exécution du courtier et du client C (les deux extrémités de la connexion). Par exemple, si vous souhaitez connecter votre client via `ssljms` au port 1756, il vous faut procéder comme suit :

- Côté client, paramétrez les propriétés suivantes :
`MQ_SERVICE_PORT_PROPERTY=1756`
`MQ_CONNECTION_TYPE_PROPERTY=SSL`
- Côté courtier, paramétrez la propriété `imq.serviceName.protocolType.port` comme suit :
`imq.ssljms.tls.port=1756`

Remarque – La propriété de connexion `MQ_SERVICE_PORT_PROPERTY` a été reportée dans Message Queue 3.7 Mise à jour 2.

Nouvelles fonctions de Message Queue 4.0

Message Queue 4.0 est une version mineure visant essentiellement à prendre en charge Application Server 9 PE. Elle comprend de nouvelles fonctions, des améliorations de fonctions et des corrections de bogues. Cette section inclut une description des nouvelles fonctions de cette version :

- “Prise en charge de l'API d'administration JMX” à la page 23
- “Journalisation à l'exécution client” à la page 23
- “API de notification des événements de connexion” à la page 23
- “Améliorations de l'administration du courtier” à la page 23
- “Affichage des informations concernant le magasin de données basé sur JDBC” à la page 24
- “Prise en charge du fournisseur JDBC” à la page 25
- “Modifications du format du magasin de données persistant” à la page 25
- “Propriétés supplémentaires des messages” à la page 25
- “Prise en charge de SSL” à la page 25



Attention – Parmi les modifications mineures mais radicales de la version 4.0, on peut souligner la désapprobation de l'option de ligne de commande pour spécifier un mot de passe. Désormais, vous devez stocker tous les mots de passe dans un fichier, comme décrit dans “[Option de mot de passe désapprouvée](#)” à la page 46 ou les saisir lorsque vous y êtes invité.

Prise en charge de l'API d'administration JMX

Une nouvelle API a été ajoutée dans Message Queue 4.0 pour la configuration et le contrôle des courtiers Message Queue conformément à la spécification Java Management Extensions (JMX). A l'aide de cette API, vous pouvez configurer et contrôler les fonctions de courtier, à l'aide de programmes, depuis une application Java. Dans les anciennes versions de Message Queue, ces fonctions étaient uniquement accessibles à partir des utilitaires d'administration de la ligne de commande ou de la console d'administration.

Pour de plus amples informations, reportez-vous au [Sun Java System Message Queue 4.2 Developer's Guide for JMX Clients](#).

Journalisation à l'exécution client

Message Queue 4.0 a introduit la prise en charge de la journalisation de l'exécution du client des événements liés aux connexions et aux sessions.

Pour obtenir des informations sur la journalisation de l'exécution client et la façon de la configurer, reportez-vous au Java Dev Guide page 137.

API de notification des événements de connexion

Message Queue 4.0 a introduit un API de notification des événements qui permet l'exécution du client pour informer une application des changements dans l'état de connexion. Les notifications d'événements de connexion permettent à un client Message Queue d'écouter les événements de fermeture et de reconnexion et d'entreprendre l'action appropriée selon le type de notification et l'état de connexion. Par exemple, lorsqu'un basculement se produit et que le client est reconnecté à un autre courtier, une application peut vouloir nettoyer l'état de transaction correspondant et utiliser une nouvelle transaction.

Pour obtenir des informations sur les événements de connexion et la façon de créer un listener d'événements, reportez-vous au Guide Java Dev, page 96.

Améliorations de l'administration du courtier

Dans Message Queue 4.0, une nouvelle sous-commande et plusieurs options de commande ont été ajoutées à l'utilitaire de Commande (`imqcmd`) pour permettre aux administrateurs de mettre en attente un courtier, de fermer un courtier après un intervalle spécifié, de détruire une connexion ou de paramétrer les propriétés du système java (par exemple, les propriétés liées à la connexion).

- La mise en attente du courtier place ce dernier dans un état silencieux, permettant ainsi une purge des messages avant la fermeture ou le redémarrage de celui-ci. Il est impossible de créer une nouvelle connexion vers un courtier mis en attente. Pour mettre le courtier en attente, saisissez une commande similaire à l'exemple suivant.

```
imqcmd quiesce bkr -b Wolfgang:1756
```

- Pour fermer le courtier après un intervalle spécifié, saisissez une commande similaire à l'exemple ci-dessous. (L'intervalle de temps spécifie le nombre de secondes à attendre avant la fermeture du courtier.)

```
imqcmd shutdown bkr -b Hastings:1066 -time 90
```

Si vous spécifiez un intervalle de temps, le courtier journalisera un message indiquant l'heure de fermeture. Par exemple :

Fermeture du courtier dans 29 secondes (29996 millisecondes)

Alors que le courtier est en attente de fermeture, son comportement est affecté de diverses manières :

- Les connexions JMS administratives sont encore acceptées.
 - Aucune nouvelle connexion JMS n'est acceptée.
 - Les connexions JMS existantes continuent de fonctionner.
 - Le courtier n'est pas en mesure de remplacer un autre courtier dans un cluster de courtiers haute disponibilité.
 - L'utilitaire `imqcmd` ne s'interrompt pas, il envoie la requête de fermeture au courtier et reprend directement une activité normale.
- Pour détruire une connexion, saisissez une commande similaire à l'exemple suivant.

```
imqcmd destroy cxn -n 2691475382197166336
```

Utilisez la commande `imqcmd list cxn` ou `imqcmd query cxn` pour obtenir l'ID de connexion.

- Pour définir une propriété système à l'aide de `imqcmd`, utilisez l'option `-D` Ceci s'avère pratique pour définir ou ignorer des propriétés de fabrique de connexion JMS ou des propriétés système Java liées à la connexion. Exemple :

```
imqcmd list svc -secure -DimqSSLIsHostTrusted=true
imqcmd list svc -secure -Djavax.net.ssl.trustStore=/tmp/mytruststore
-Djavax.net.ssl.trustStorePassword=mytrustword
```

Pour de plus amples informations sur la syntaxe de la commande `imqcmd`, reportez-vous au [Chapitre 13, "Command Line Reference" du *Sun Java System Message Queue 4.2 Administration Guide*](#).

Affichage des informations concernant le magasin de données basé sur JDBC

Dans Message Queue 4.0, une nouvelle sous-commande de requête a été ajoutée à l'utilitaire du Gestionnaire de base de données, `imqdbmgr`. Cette sous-commande est utilisée pour afficher les informations relatives au magasin de données basé sur JDBC persistant, notamment sa version base de données, l'utilisateur de la base de données et la création ou non de tables de base de données.

Voici un exemple des informations affichées par la commande.

```
requête imqdbmgr
```

```
[04/Oct/2005:15:30:20 PDT] Utilisation du magasin persistant intégré :  
    version=400  
    id du courtier=Mozart1756  
    url de connexion à la base de données=jdbc:oracle:thin:@Xhome:1521:mqdb  
    utilisateur de la base de données=scott  
Exécution en mode autonome.  
Les tables de base de données ont déjà été créées.
```

Prise en charge du fournisseur JDBC

Dans Message Queue 4.0, Apache Derby Version 10.1.1 est désormais supporté comme fournisseur de magasin de données basé sur JDBC.

Modifications du format du magasin de données persistant

Message Queue 4.0 a introduit des modifications au magasin de données basé sur JDBC à des fins d'optimisation et pour prendre en charge des améliorations futures. Pour cette raison, le format du magasin de données basé sur JDBC a été augmenté vers la version 400. Notez que dans Message Queue 4.0, la version du magasin de données basé sur les fichiers reste la version 370 car elle n'a subi aucune modification.

Propriétés supplémentaires des messages

Message Queue 4.0 a ajouté deux nouvelles propriétés paramétrées sur tous les messages placés dans la file d'attente bloquée.

- `JMS_SUN_DMQ_PRODUCING_BROKER` indique le courtier dans lequel le message a été créé.
- `JMS_SUN_DMQ_DEAD_BROKER` indique le courtier ayant marqué le message comme bloqué.

Prise en charge de SSL

À partir de Message Queue 4.0, la valeur par défaut pour la propriété de fabrication de connexion des clients `imqSSLIsHostTrusted` est `false`. Si votre application est basée sur la valeur par défaut précédente `true`, vous devez reconfigurer et définir explicitement la propriété sur `true`.

Vous pouvez choisir de faire confiance à l'hôte lorsque le courtier est configuré pour utiliser des certificats autosignés. Dans ce cas, en plus d'indiquer que la connexion doit utiliser un service de connexions SSL (à l'aide de la propriété `imqConnectionType`), vous devez définir la propriété `imqSSLIsHostTrusted` sur `true`.

Par exemple, pour exécuter de manière sécurisée les applications clientes lorsque le courtier utilise des certificats autosignés, utilisez une commande similaire à l'exemple suivant.

```
java -DimqConnectionType=TLS  
      -DimqSSLIsHostTrusted=true ClientAppName
```

Pour utiliser l'utilitaire de Commande(imqcmd) de façon sécurisée lorsque le courtier utilise des certificats auto-signés, utilisez une commande semblable à l'une des commandes suivantes (pour répertorier les services de connecteur).

```
imqcmd list svc -secure -DimqSSLIsHostTrusted=true
```

Fontion sera désapprouvée dans les versions futures

La surveillance basée sur les messages vous permettant de surveiller un courtier et ses destinations à l'aide d'informations métriques écrites vers les destinations de sujet métriques sera désapprouvée dans les versions futures.

La surveillance basée sur les messages utilise le producteur de messages métriques configurable du courtier pour écrire des données métriques dans des messages JMS envoyés aux destinations de sujets métriques, selon le type d'informations métriques contenues dans les messages. Il est ensuite possible d'accéder à ces informations métriques en écrivant une application client qui souscrit à la destination de sujet métrique appropriée, consomme ses messages et traite les données à volonté.

La fonction de surveillance basée sur les messages a été supplantée par l'API d'administration JMX implémentée dans MQ 4.0 (reportez-vous à [“Prise en charge de l'API d'administration JMX” à la page 23](#)). L'API de JMX API est plus complet (il comprend plus de données métriques que dans les destinations de sujet) et est basé sur le standard industriel JMX.

Il n'existe plus de raison valable d'utiliser une surveillance basée sur les messages maintenant que Message Queue prend en charge l'API JMX. les informations concernant la surveillance basée sur les messages restera dans la documentation de Message Queue jusqu'à ce que la fonction soit formellement désapprouvée.

Bogues résolus dans Message Queue 4.2 et ses versions récentes

Message Queue 4.2 inclut de nouvelles corrections de bogues et incorpore également des bogues résolus dans les versions Message Queue 4.1 et Message Queue 4.0.

Les sections suivantes dressent par version la liste des bogues résolus :

- [“Bogues résolus dans Message Queue 4.2” à la page 27](#)
- [“Bogues résolus dans Message Queue 4.1” à la page 28](#)
- [“Bogues résolus dans Message Queue 4.0” à la page 29](#)

Bogues résolus dans Message Queue 4.2

Le tableau suivant décrit les bogues résolus dans Message Queue 4.2.

Remarque – Le tableau suivant doit inclure les bogues méritant d’être mentionnés résolus dans les versions 4.1 patch 1 et 4.1 patch 2. J’ai besoin d’aide pour déterminer de quels bogues il s’agit. Existe-t-il également des corrections de bogues supplémentaires n’incluant pas celles déjà mentionnées ?

TABLEAU 1-7 Bogues résolus dans Message Queue 4.2

Bogue	Description
6581592	Lorsque le programme d’installation ou de désinstallation est exécuté en mode texte (<code>installer -t</code>), l’écran de résumé affiche le répertoire contenant les fichiers journaux ou de synthèse mais ne répertorie pas le nom de ces fichiers.
6585911	L’écran de sélection JDK du programme d’installation inclut de façon incorrecte le JRE groupé avec le programme d’installation et utilisé pour exécuter le programme d’installation.
6587112	L’écran de synthèse du programme d’installation affiche des informations parasites dans les environnements multi-octets.
6587127	Lors de l’exécution du programme d’installation en se référant à un fichier de réponses (<code>installer -a nom_fichier -s</code>), si celui-ci n’existe pas, alors les messages d’erreurs sont incohérents et peu clairs.
6590969	Autorise le nom d’utilisateur dynamique pour l’authentification de connexion du client.
6594381	L’installation des RPM de localisation de Message Queue 4.1 (qui se produit lorsque vous sélectionnez la case “Install Message Queue multilingual packages” sur l’écran Multilingual Packages) échouera s’il existe d’anciennes versions de ces RPM de localisation de Message Queue sur votre système.
6599144	Lorsque vous désinstallez Message Queue 4.2, la page de garde et le programme de désinstallation restent bloqués et les écrans apparaissent vides et grisés sur Java SE 6, alors qu’ils s’exécutent normalement sur Java SE 5.
6615741	Le message délivré dans une session de consommateur transactionnelle annulée n’est pas redélivré si le consommateur d’origine s’est fermé avant l’annulation.
6629922	Le gestionnaire des transactions distribuées ne redélivre pas dans l’ordre les messages au consommateur inactif.
6635130	Échec du courtier à notifier au producteur de messages non persistants de reprendre la production après avoir été interrompu, car la destination a atteint les limites de mémoire ou de messages.
6641117	Le message délivré dans une session de consommateur transactionnelle annulée n’est pas redélivré si le consommateur d’origine s’est fermé après l’annulation.

TABLEAU 1-7 Bogues résolus dans Message Queue 4.2 (Suite)

Bogue	Description
6683897	L'écran de synthèse du programme d'installation de Message Queue rapporte une erreur de configuration bien que la configuration apparaisse réussie : le programme d'insatllation ne peut pas écrire dans <code>/dev/sterr</code> sur certains ordinateurs.
6684069	Dans un cluster de courtiers dans lequel un grand nombre de messages est délivré à un client distant dans la transaction consommateur, la transaction de validation échoue.
6688935	La valeur par défaut du délai d'attente de lecture de Portmapper est trop petite.
6695238	Des applications client C ne peuvent pas se connecter à un courtier installé à un emplacement dont le chemin possède des espaces.
6710168	Le consommateur ne consomme plus de messages si la destination est interrompue deux fois sans reprise entre les pauses.
6710169	L'opération <code>JMX ConsumerManagerMonitor.getConsumerInfo</code> reviens toujours à <code>SESSION_TRANSACTED</code> pour le mode accusé de réception.

Bogues résolus dans Message Queue 4.1

Le tableau suivant décrit les bogues résolus dans Message Queue 4.1.

TABLEAU 1-8 Bogues résolus dans Message Queue 4.1

Bogue	Description
6381703	Les messages distants transactionnels peuvent être validés deux fois si le courtier à l'origine des messages redémarre.
6388049	Impossible de nettoyer une transaction distribuée incomplète.
6401169	Les options de validation et d'annulation de <code>imqcmd</code> n'envoient pas d'invite de confirmation.
6473052	Par défaut, les files d'attente créées automatiquement doivent être alternées. (<code>MaxNumberConsumers = -1</code>).
6474990	Le journal du courtier affiche une <code>ConcurrentModificationException</code> pour la commande <code>imqcmd list dst</code> .
6487413	Fuite de mémoire lorsque le comportement aux limites est <code>REMOVE_OLDEST</code> ou <code>REMOVE_LOWER_PRIORITY</code> .
6488340	Basculement du courtier et attente de la réponse par le client pour en accuser réception.
6502744	Le courtier ne respecte pas la limite par défaut de la file d'attente de message bloqués de 1000 messages.

TABLEAU 1-8 Bogues résolus dans Message Queue 4.1 (Suite)

Bogue	Description
6517341	L'exécution client doit améliorer la logique de reconnexion lorsque le client est connecté à un cluster de courtiers haute disponibilité, en autorisant celui-ci à se reconnecter quelle que soit la valeur de la propriété <code>imqReconnectEnabled</code> .
6528736	Le service de démarrage automatique de Windows (<code>imqbrokersvc</code>) s'arrête brutalement au démarrage.
6561494	Les messages sont transmis au mauvais consommateur lorsque ceux-ci partagent une session.
6567439	Les messages produits dans une transaction de niveau PREPARED sont transmis en désordre s'ils sont validés après le redémarrage du courtier.

Bogues résolus dans Message Queue 4.0

Le tableau suivant décrit les bogues résolus dans Message Queue 4.0.

TABLEAU 1-9 Bogues résolus dans Message Queue 4.0

N° de bogue	Description
4986481	Dans Message Queue 3.5, l'appel de <code>Session.recover</code> peut être bloqué en mode de reconnexion automatique.
4987325	L'indicateur de redistribution a été défini sur <code>false</code> pour les messages redistribués après l'appel de <code>Session.recover</code> .
6157073	Modification du nouveau message de connexion pour inclure le nombre de connexions sur le service, en plus du nombre total de connexions.
6193884	Message Queue envoie un message parasite vers le syslog dans des langues utilisant des caractères non ASCII pour les messages.
6196233	La sélection de messages à l'aide de <code>JMSMessageID</code> ne fonctionne pas.
6251450	<code>ConcurrentModificationException</code> sur <code>connectList</code> durant la fermeture du cluster.
6252763	<code>java.nio.BufferOverflowException</code> dans <code>java.nio.HeapByteBuffer.putLong/Int</code> .
6260076	Le premier message publié après le démarrage est lent avec le stockage Oracle.
6260814	Le sélecteur traitant <code>JMSUserID</code> donne une évaluation toujours <code>false</code> .
6264003	Le navigateur de file d'attente affiche des messages qui font partie des transactions qui n'ont pas été validées.
6271876	Le contrôle de flux de connexions ne fonctionne pas correctement lors de la fermeture d'un consommateur avec des messages non consommés.
6279833	Message Queue ne doit pas autoriser deux courtiers à utiliser les mêmes tables JDBC.

TABLEAU 1-9 Bogues résolus dans Message Queue 4.0 (Suite)

N° de bogue	Description
6293053	Le courtier maître ne démarre pas correctement si l'adresse IP du système a été modifiée, à moins que le magasin soit nettoyé (via <code>-reset store</code> .)
6294767	Le courtier de Message Queue doit définir <code>SO_REUSEADDR</code> sur les sockets de réseau qu'il ouvre.
6304949	Impossible de définir la propriété <code>ClientID</code> pour <code>TopicConnectionFactory</code> .
6307056	Le journal <code>txn</code> est un goulot d'étranglement des performances.
6320138	L'API C de Message Queue manque de capacités pour déterminer le nom d'une file d'attente à partir d'un en-tête Répondre à.
6320325	Le courtier sélectionne parfois JDK 1.4 avant JDK 1.5 sur Solaris même lorsque les deux versions sont installées.
6321117	L'initialisation de cluster multicourtier émet une <code>java.lang.NullPointerException</code> .
6330053	Le client JMS lève une <code>java.lang.NoClassDefFoundError</code> lors de la validation d'une transaction de l'abonné.
6340250	Prise en charge du type <code>MESSAGE</code> dans l'API C.
6351293	Ajout d'une prise en charge pour la base de données Apache Derby.

Mises à jour de la documentation relative à Message Queue 4.2

Cette section contient des informations sur les mises à jour de la documentation de Message Queue 4.2 :

- [“Problèmes de compatibilité” à la page 30](#)
- [“Modifications dans le jeu de documentation de Message Queue 4.2” à la page 31](#)
- [“Nouvelles métriques de destination” à la page 32](#)
- [“Démarrage automatique de courtier sur le SE Solaris 10” à la page 33](#)
- [“Modifications apportées à JMX API” à la page 34](#)
- [“Prise en charge pour le format du nom d'utilisateur dynamique pour l'authentification de client” à la page 39](#)
- [“Amélioration de l'authentification JAAS” à la page 39](#)

Problèmes de compatibilité

Cette section décrit les problèmes de compatibilité concernant Message Queue 4.2.

Stabilité de l'interface

Sun Java System Message Queue utilise de nombreuses interfaces pouvant être modifiées dans le temps. L'Annexe B, "Stability of Message Queue Interfaces" du *Sun Java System Message Queue 4.2 Administration Guide* classe les interfaces selon leur stabilité. Plus l'interface est stable, plus il y a de chances pour qu'elle ne soit pas modifiée dans les versions à venir d'un produit.

Problèmes liés à la prochaine version principale de Message Queue

La prochaine version principale de Message Queue pourra introduire des changements la rendant incompatible avec des applications actuelles de Message Queue . Ces informations sont fournies dans l'intérêt d'une publication extensive.

- L'emplacement des fichiers individuels installés pour Sun Java System Message Queue peut changer. Cette modification risque de briser les applications existantes qui dépendent de l'emplacement actuel de certains fichiers Message Queue.
- Il se peut que les courtiers Message Queue 3.5 et antérieurs ne puissent pas fonctionner dans un cluster hébergeant des courtiers plus récents.
- Dans les prochaines versions, il est possible que les clients Message Queue ne puissent plus utiliser les versions JDK antérieures à 1.5.
- Dans les prochaines versions, il est possible que les clients Message Queue ne puissent plus utiliser les versions JDK antérieures à la version 1.6.

Modifications dans le jeu de documentation de Message Queue 4.2

Le jeu de documentation de Message Queue 4.2 comprend des mises à jour du jeu de documentation de Message Queue 4.1, comme décrit ci-dessous :

Présentation technique

Le *Sun Java System Message Queue 4.2 Installation Guide* a été mis à jour pour inclure les nouvelles fonctions de Message Queue 4.2 et la structure mise à jour pour les clusters de courtiers haute disponibilité.

Administration Guide

Le *Guide d'administration* a été mis à jour pour inclure les nouvelles fonctions de Message Queue 4.2.

Informations d'installation et de mise à niveau

Le *Sun Java System Message Queue 4.2 Installation Guide* n'a pas été mis à jour pour inclure les nouvelles fonctions de Message Queue 4.2, en particulier la nouvelle fonctionnalité d'enregistrement de Sun Connect du programme d'installation. Ces informations sont fournies dans ces *Notes de version de Message Queue*

Guide du développeur pour Clients Java

Le *Guide du développeur pour Clients Java* n'a pas été mis à jour pour inclure les nouvelles fonctions de Message Queue 4.2. Ces informations sont fournies dans ces *Notes de version de Message Queue*

Guide du développeur pour Clients C

Le *Guide du développeur pour Clients C* n'a pas été mis à jour pour inclure les nouvelles fonctions de Message Queue 4.2. Ces informations sont fournies dans ces *Notes de version de Message Queue*

Guide du développeur pour Clients JMX

n'a pas été mis à jour pour inclure les nouvelles fonctions de Message Queue 4.2. Ces informations sont fournies dans ces *Notes de version de Message Queue*

Nouvelles métriques de destination

Message Queue 4.2 comprend de nouvelles métriques de destination pouvant être utiles lors de la surveillance des destinations dans un cluster de courtiers. Dans un cluster de courtiers, les destinations sont propagées à tous les courtiers du cluster. Toutefois, un message, une fois produit, est stocké dans une destination cible du courtier hébergeant le producteur de messages et n'est envoyé à la destination correspondante sur un autre courtier du cluster que s'il existe un consommateur actif pour cette destination. Il en résulte que les messages stockés dans une destination spécifiées dépendent du courtier dans le cluster hébergeant la destination.

Autrement dit, dans un cluster de courtiers, les messages stockés dans une destination donnée sur un courtier donné du cluster se composent de messages produits directement vers la destination et de messages envoyés à la destination à partir de courtiers distants dans le cluster. L'analyse de l'acheminement et de la livraison des messages dans un cluster de courtiers est parfois utile pour savoir, dans une destination, combien de messages sont locaux (produits localement) et distants (produits à distance).

Le tableau suivant présente deux nouvelles quantités métriques de destination physique introduites dans Message Queue 4.2. Les nouvelles quantités métriques sont disponibles via les commandes `imqcmd list dst`, `imqcmd query dst` et les nouveaux attributs JMX (reportez-vous au [“MBean de contrôle de destination”](#) à la page 36).

TABLEAU 1-10 Métriques de destinations physiques

Quantité métrique	Description	Fichier journal ?	metrics dst Type de métriques	Sujet de métriques
Nb de messages à distance	Nombre actuel de messages stockés dans une mémoire et un magasin persistant produits vers un courtier distant dans un cluster. Ce nombre ne compte pas les messages inclus dans les transactions.	Non	Non disponible ¹	Non disponible
Total d'octets de message à distance	Taille totale actuelle en octets des messages stockés dans une mémoire et un magasin persistant produits vers un courtier distant dans un cluster. Cette valeur n'inclut pas les messages inclus dans les transactions.	Non	Non disponible ¹	Non disponible

¹ Disponible avec la commande `imqcmd query dst`

Démarrage automatique de courtier sur le SE Solaris 10

Cette section décrit la façon de configurer le démarrage automatique de courtier sur les systèmes d'exploitation Solaris 10. Plutôt que d'utiliser un fichier rc pour implémenter un démarrage automatique de courtier quand un ordinateur redémarre, la procédure suivante utilise l'outil de gestion des services Solaris 10 (SMF).

Pour plus d'informations sur l'utilisation de l'outil de gestion des services, reportez-vous à la documentation de Solaris 10.

▼ Implémentation du démarrage automatique de courtier sur Solaris 10 OS

1 Importer le service mqbroker dans le répertoire SMF.

```
# svccfg import /var/svc/manifest/application/sun/mq/mqbroker.xml
```

2 Vérifier que l'importation a été effectuée avec succès en contrôlant l'état du service mqbroker.

```
# svcs mqbroker
```

Le résultat affiché par la commande ressemble à celui présenté ci-dessous.

```
STATE STIME FMRI
disabled 16:22:50 svc:/application/sun/mq/mqbroker:default
```

Le service est initialement affiché comme désactivé.

3 Active le service mqbroker.

```
# svcadm enable svc:/application/sun/mq/mqbroker:default
```

L'activation du service mqbroker lancera le processus imqbrokerd. Un redémarrage relancera par la suite le courtier.

4 Configurer le service mqbroker pour passer tout argument désiré à la commande imqbrokerd.

La propriété options/server_args est utilisée pour passer des arguments à imqbrokerd. Par exemple, pour ajouter -loglevel DEBUGHIGH, procédez comme suit :

```
# svccfg
svc:> select svc:/application/sun/mq/mqbroker
svc:/application/sun/mq/mqbroker> setprop options/server_args=\"-loglevel DEBUGHIGH\"
svc:/application/sun/mq/mqbroker> exit
```

Modifications apportées à JMX API

Message Queue prend en charge l'API d'extensions de gestion de Java (JMX) pour configurer et contrôler les fonctions de courtier par le biais d'un programme à partir d'une application client Message Queue. Message Queue 4.2 comprend les extensions pour l'API JMX servant à prendre en charge de nouvelles fonctions et fonctionnalités de la version. Les nouveaux attributs JMX, opérations et/ou clés de conversion sont définis pour les Mbeans suivants :

- “MBean de contrôle du gestionnaire de consommateurs” à la page 34
- “MBean de configuration de destinations” à la page 35
- “MBean de configuration du gestionnaire de destinations” à la page 36
- “MBean de contrôle de destination” à la page 36
- “MBean de contrôle du gestionnaire des producteurs” à la page 38

MBean de contrôle du gestionnaire de consommateurs

Les attributs, opérations et clés de conversion des tableaux suivants prennent en charge la fonction décrite dans [“Destinations multiples pour un éditeur ou un abonné”](#) à la page 11.

Le nom de l'attribut suivant est défini comme des constantes statiques dans la classe d'utilitaires `com.sun.messaging.jms.management.server.ConsumerAttributes` .

TABLEAU 1-11 Attributs de contrôle du gestionnaire de consommateurs

Nom	Type	Paramétrable ?	Description
NumWildcardConsumers	Entier	Non	Nombre de consommateurs de messages génériques associés au courtier

Les noms des opérations suivantes sont définis comme des constantes statiques dans la classe d'utilitaires `com.sun.messaging.jms.management.server.ConsumerAttributes`.

TABLEAU 1-12 Opérations de contrôle du gestionnaire de consommateurs

Nom	Paramètres	Type de résultat	Description
<code>getConsumerWildcards</code>	none	String[]	Chaînes génériques utilisées par les consommateurs actuels associés au courtier
<code>getNumWildcardConsumers</code>	Chaîne générique	Entier	Nombre de consommateurs actuels associés au courtier et utilisant la chaîne générique spécifiée

Les clés de conversion suivantes sont définies comme des constantes statiques dans la classe d'utilitaires `com.sun.messaging.jms.management.server.ConsumerInfo`.

TABLEAU 1-13 Clés de conversion pour des informations de consommateurs de messages

Nom	Type de valeur	Description
DestinationNames	String[]	Noms de destination correspondant aux génériques utilisés par des consommateurs génériques Pour des destinations de sujet uniquement.
Caractère générique	Booléenne	Consommateur générique ? Pour des destinations de sujet uniquement.

MBean de configuration de destinations

Les attributs dans le tableau suivant prennent en charge la fonction décrite dans [“Validation des schémas des messages de charge utile XML” à la page 13](#).

Les noms des attributs suivants sont définis comme des constantes statiques dans la classe d'utilitaires `com.sun.messaging.jms.management.server.DestinationAttributes`.

TABLEAU 1-14 Attributs de configuration de destinations

Nom	Type	Paramétrable ?	Description
ValidateXMLSchemaEnabled	Booléenne	Oui	<p>Validation du schéma XML activée ?</p> <p>Si l'option est définie non définie ou définie sur false, alors la validation du schéma XML n'est pas activée pour la destination.</p>
XMLSchemaURLList	Chaîne	Oui	<p>Liste des chaînes URI du document (XSD) du schéma XML séparées par des espaces</p> <p>Les URI pointent vers l'emplacement d'un ou plusieurs XSD à utiliser pour la validation du schéma XML, si activée.</p> <p>Mettez cette valeur entre guillemets si plusieurs URI sont spécifiés.</p> <p>Exemple :</p> <pre>"http://foo/flap.xsd http://test.com/test.xsd"</pre> <p>Si cette propriété n'est pas définie ou null et si la validation XML est activée, la validation XML est effectuée à l'aide d'un DTD spécifié dans le document XML.</p>
ReloadXMLSchemaOnFailure	Booléenne	Oui	<p>Rechargement du schéma XML en cas d'échec activé ?</p> <p>Si l'option est définie non définie ou définie sur false, alors le schéma n'est pas rechargé en cas d'échec de la validation.</p>

MBean de configuration du gestionnaire de destinations

Les nouveaux attributs de MBean de configuration de destinations, ci-dessus prenant en charge la nouvelle fonction [“Validation des schémas des messages de charge utile XML”](#) à la page 13, peuvent être utilisés pour la création d'une destination à l'aide de l'opération créer du MBean de configuration du gestionnaire des destinations.

MBean de contrôle de destination

Le premier ensemble d'attributs dans le tableau suivant prend en charge la fonction décrite dans [“Destinations multiples pour un éditeur ou un abonné”](#) à la page 11 et le second ensemble d'attributs prend en charge l'amélioration décrite dans [“Nouvelles métriques de destination”](#) à la page 32.

Les noms des attributs suivants sont définis comme des constantes statiques dans la classe `utilitaires.com.sun.messaging.jms.management.server.DestinationAttributes`.

TABLEAU 1–15 Attributs de contrôle de destinations

Nom	Type	Paramétrable ?	Description
NumWildcards	Entier	Non	Nombre actuel de producteurs et de consommateurs de messages génériques associés à la destination Pour des destinations de sujet uniquement.
NumWildcardProducers	Entier	Non	Nombre actuel de producteurs de messages génériques associés à la destination Pour des destinations de sujet uniquement.
NumWildcardConsumers	Entier	Non	Nombre actuel de consommateurs de messages génériques associés à la destination Pour des destinations de sujet uniquement.
NumMsgsRemote	Long	Non	Nombre actuel de messages stockés dans une mémoire et un magasin persistant produits vers un courtier distant dans un cluster. Ce nombre ne compte pas les messages inclus dans les transactions.
TotalMsgBytesRemote	Long	Non	Taille totale actuelle en octets de messages stockés dans une mémoire et un magasin persistant produits vers un courtier distant dans un cluster. Cette valeur ne comprend pas les messages inclus dans les transactions.

Les opérations dans le tableau suivant prennent en charge la fonction décrite dans [“Destinations multiples pour un éditeur ou un abonné” à la page 11](#).

Les noms des opérations suivantes sont définis comme des constantes statiques dans la classe `d'utilitairescom.sun.messaging.jms.management.server.DestinationOperations`.

TABLEAU 1–16 Opérations de contrôle des destinations

Nom	Paramètres	Type de résultat	Description
getWildcards	none	String[]	Les chaînes génériques utilisées par les consommateurs et producteurs actuels associés à la destination Pour des destinations de sujet uniquement.
getConsumerWildcards	none	String[]	Chaînes génériques utilisées par les consommateurs actuels associés à la destination Pour des destinations de sujet uniquement.

TABLEAU 1-16 Opérations de contrôle des destinations (Suite)

Nom	Paramètres	Type de résultat	Description
<code>getProducerWildcards</code>	none	String[]	Les chaînes génériques utilisées par les producteurs actuels associés à la destination Pour des destinations de sujet uniquement.
<code>getNumWildcardConsumers</code>	Chaîne générique	Entier	Nombre de consommateurs actuels associés à la destination et utilisant la chaîne générique spécifiée Pour des destinations de sujet uniquement.
<code>getNumWildcardProducers</code>	Chaîne générique	Entier	Nombre de producteurs actuels associés à la destination et utilisant la chaîne générique spécifiée Pour des destinations de sujet uniquement.

MBean de contrôle du gestionnaire des producteurs

Les attributs, opérations et clés de conversion dans les tableaux ci-dessous prennent en charge la fonction décrite dans [“Destinations multiples pour un éditeur ou un abonné”](#) à la page 11.

Le nom de l'attribut suivant est défini comme des constantes statiques dans la classe `d'utilitaires.com.sun.messaging.jms.management.server.ProducerAttributes`.

TABLEAU 1-17 Attributs de contrôle du gestionnaire de producteurs

Nom	Type	Paramétrable ?	Description
<code>NumWildcardProducers</code>	Entier	Non	Nombre de producteurs de messages génériques associés au courtier

Les noms des opérations suivantes sont définis comme des constantes statiques dans la classe `d'utilitaires.com.sun.messaging.jms.management.server.ProducerAttributes`.

TABLEAU 1-18 Opérations de contrôle du gestionnaire de producteurs

Nom	Paramètres	Type de résultat	Description
<code>getProducerWildcards</code>	none	String[]	Chaînes génériques utilisées par les producteurs actuels associés au courtier
<code>getNumWildcardProducers</code>	Chaîne générique	Entier	Nombre de producteurs actuels associés au courtier et utilisant la chaîne générique spécifiée

Les clés de conversion suivantes sont définies comme des constantes statiques dans la classe `d'utilitaires.com.sun.messaging.jms.management.server.ProducerInfo`.

TABLEAU 1–19 Clés de conversion pour des informations de producteurs de messages

Nom	Type de valeur	Description
DestinationNames	String[]	Noms de destination correspondant aux génériques utilisés par des producteurs génériques Pour des destinations de sujet uniquement.
Caractère générique	Booléenne	Producteur générique ? Pour des destinations de sujet uniquement.

Prise en charge pour le format du nom d'utilisateur dynamique pour l'authentification de client

Message Queue 4.2 prend en charge le format du nom d'utilisateur dynamique dans l'authentification de connexion client par rapport au répertoire utilisateur LDAP. La prise en charge implique la nouvelle propriété suivante de courtier (et sa valeur) :

```
imq.user_repository.ldap.usrformat=dn
```

Cette propriété permet au courtier d'authentifier un utilisateur client par rapport à une entrée dans un répertoire utilisateur LDAP en extrayant du format du nom d'utilisateur dynamique la valeur de l'attribut spécifiée par la propriété suivante :

```
imq.user_repository.ldap.uidattr
```

Le courtier utilise la valeur de l'attribut ci-dessus comme nom de l'utilisateur dans les opérations de contrôle d'accès.

Par exemple, si `imq.user_repository.ldap.uidattr=udi` et un nom d'utilisateur dynamique d'authentification client est au format `udi=mquser,ou=People,dc=red,dc=sun,dc=com`, alors "mquser" doit être extrait pour effectuer un contrôle d'accès.

Amélioration de l'authentification JAAS

L'authentification JAAS Message Queue 4.2 prend en charge l'authentification par l'adresse IP ainsi que par le nom d'utilisateur dynamique.

Problèmes connus et restrictions

Cette section contient une liste des problèmes connus concernant Message Queue 4.2. Les domaines suivants du produit sont abordés :

- “Problèmes d'installation” à la page 40
- “Option de mot de passe désapprouvée” à la page 46
- “Problèmes d'administration/de configuration” à la page 47
- “Problèmes relatifs au courtier” à la page 48
- “Clusters de courtiers” à la page 49
- “Problèmes relatifs à JMX” à la page 51
- “Prise en charge de SOAP” à la page 51

Pour obtenir une liste des bogues actuels, de leur état et de leurs solutions, les membres de Java Developer Connection™ peuvent consulter la « Bug Parade » sur le site Web de Java Developer Connection. Avant de signaler tout nouveau bogue, merci de consulter cette page. Bien que tous les bogues de Message Queue n'y soient pas répertoriés, il est préférable de consulter cette page pour savoir si un problème a déjà été signalé.

<http://bugs.sun.com/bugdatabase/index.jsp>

Remarque – L'adhésion à Java Developer Connection est gratuite, mais elle requiert une inscription. Pour savoir comment devenir membre de Java Developer Connection, consultez la page Web « For Developers » de Sun.

Pour signaler un nouveau bogue ou soumettre une demande d'amélioration, envoyez un e-mail à l'adresse suivante : imq-feedback@sun.com.

Problèmes d'installation

Cette section décrit les problèmes liés à l'installation de Message Queue version 4.2.

Enregistrement du produit et Java ES

Message Queue 4.2, comme Message Queue 4.1, est installé par un programme d'installation relativement nouveau qui installe et met à niveau les composants partagés de Java Enterprise System (Java ES) requis par Message Queue; par exemple, JDK, NSS, JavaHelp etc

Le nouveau programme d'installation Message Queue et l'ancien programme d'installation de Java ES utilisés pour installer les versions précédentes de Message Queue, ne partagent pas le même registre de produit. Si une version de Message Queue, précédemment installée avec Java ES, est supprimée et mise à niveau vers Message Queue 4.2 par le programme d'installation de Message Queue, le registre de produit Java ES peut alors se trouver en état incohérent. Ainsi, si

le programme de désinstallation de Java ES est exécuté, il est possible qu'il supprime par accident Message Queue 4.2 et les composants partagés dont il dépend même s'il ne les a pas installés.

Le meilleur moyen de mettre à niveau le logiciel Message Queue installé par le programme d'installation de Java ES est de

1. Supprimer Message Queue et ses composants partagés avec le programme de désinstallation de Java ES
2. d'installer Message Queue 4.2 avec le programme d'installation de Message Queue .

Installation sous Windows

Lorsque vous installez Message Queue sous Windows, veuillez prendre en compte les limitations suivantes.

- Le programme d'installation n'ajoute pas d'entrées pour Message Queue dans le menu Démarrer > Programmes. (*Bogue 6567258*)
Solution : pour démarrer la console d'administration, utilisez la ligne de commande comme décrit dans la section [“Starting the Administration Console”](#) du *Sun Java System Message Queue 4.2 Administration Guide*.
- Le programme d'installation n'ajoute pas le répertoire `IMQ_HOME\mq\bin` à la variable d'environnement `PATH`. (*Bogue 6567197*)
Solution : les utilisateurs doivent ajouter cette entrée à leur variable d'environnement `PATH` ou fournir un nom de chemin complet lorsqu'ils invoquent les utilitaires de Message Queue (`IMQ_HOME\mq\bin\ commande`).
- Le programme d'installation n'ajoute pas d'entrée dans le registre Windows pour indiquer que Message Queue a été installé. (*Bogue 6586389*)
- Lorsqu'il est exécuté en mode silencieux avec un fichier de réponse, le programme d'installation réapparaît immédiatement. L'installation s'effectue mais l'utilisateur n'a pas la possibilité de savoir quand l'installation silencieuse est réellement exécutée. (*Bogue 6586560*)
- Toute tentative d'exécuter le programme d'installation en mode texte (`installer -t`) sur Windows provoque un message d'erreur qui s'affiche en anglais même si le programme d'installation est exécuté dans une autre langue que l'anglais. Le mode texte n'est pas pris en charge sous Windows. (*Bogue 6594142*)
- Le programme d'installation n'installe pas par défaut Message Queue sur le même lecteur que celui sur lequel le système d'exploitation est installé. (*Bogue 6673511*)
- Pour l'installation et la désinstallation sur Windows, il n'existe aucun fichier `.bat` que l'utilisateur puisse exécuter, et celui-ci ne peut pas non plus désinstaller en utilisant les programmes Ajouter/Supprimer dans le panneau de configuration de Windows. (*Bogue 6673417*)

- Sur Windows Vista, vous ne pouvez pas installer Message Queue dans C:\Program Files à moins de procéder à l'installation à partir de l'invite de commande en tant qu'administrateur. (Bogue 6701661)
Solution : Pour installer à partir de l'invite de commande en tant qu'administrateur :
 1. Démarrer→Programmes→Accessoires→Invite de commande.
 2. Cliquez avec le bouton droit sur l'invite de commande.
 3. Sélectionnez Exécuter en tant qu'administrateur.
 4. Changez le répertoire vers l'image d'installation de Message Queue 4.2.
 5. Exécutez `installer.vbs`.
- Lorsque le programme de désinstallation est exécuté en mode de simulation (`uninstaller -n`), il exécute de façon incorrecte une désinstallation. (Bogue 6719051)
Solution : exécuter une installation silencieuse à l'aide de la commande suivante :
`uninstaller -s`
- La chaîne "Install Home" sur la page d'accueil du programme d'installation n'est pas localisée. (Bogue 6592491)

Installation sous Solaris

- Lorsque le programme d'installation est exécuté en mode de simulation (`installer -n`), l'écran de synthèse affiche des messages d'erreur, ainsi qu'un état d'installation « Incomplet ». Cela est incorrect et peut prêter à confusion ; l'installation en mode de simulation n'installe aucun élément sur le système mais crée uniquement un fichier de réponse utilisable par la suite pour exécuter une installation silencieuse. (Bogue 6594351)
- Le programme d'installation n'exécute pas l'enregistrement Sun Connection lorsque l'exécution se fait en mode silencieux avec un fichier de réponse (`installer -a filename -s`). (Bogue 6710268)
- Lors de l'exécution du programme d'installation en mode texte, et lors de la saisie d'un nom d'utilisateur ou d'un mot de passe pour l'enregistrement à Sun Connect ou la création d'un compte en ligne, vous ne pouvez pas corriger ces entrées avec la touche d'effacement arrière. (Bogue 6673460)
Solution \~: Utilisez les touches `Ct r l +H` au lieu de la touche d'effacement arrière, ou utilisez un émulateur de terminal différent comme `dt term` ou `xterm`.
- L'écran de mise à niveau du programme d'installation ne rapporte pas toujours correctement la version installée existante de Message Queue ou du moteur du programme d'installation. (Bogue 6679765)
- Lors de l'utilisation du programme d'installation en mode texte et de la tentative d'enregistrement de Sun Connection avec un nom d'utilisateur et un mot de passe invalides, le programme d'installation affiche "enregistrement impossible", envoie une exception de pointeur null et se ferme. (Bogue 6666365)

Installation sous Linux

Les problèmes suivants affectent le processus d'installation sur une plate-forme Linux.

- Sur le panneau de sélection du JDK, la liste déroulante n'affiche qu'un seul élément. Il est donc difficile de sélectionner tout autre JDK dans la liste. (*Bogue 6584735*)
- Si le JDK est actif et que l'utilisateur sélectionne « Installer le JDK par défaut » à l'écran de sélection du JDK, le programme d'installation tente toujours de l'installer et signale qu'il n'est pas en mesure d'installer le package. L'installation se termine correctement malgré ce problème. (*Bogue 6581310*)
- Si la version de JDK actuellement installée est ultérieure à JDK 1.5.0_15 (la version installée normalement par le programme d'installation Message Queue), alors le programme de désinstallation de Message Queue ne peut pas trouver le répertoire par défaut `IMQ_JAVAHOME` et renvoie une erreur. (*Bogue 6673415*)

Solution : Installez JDK 1.5 manuellement comme suit avant d'exécuter le programme de désinstallation de Message Queue.

```
# cd installImage/Product/UNIX/LINUX/X86/2.4/Packages
```

```
# rpm -i --force jdk-1.5.0_15-linux- arch.rpm
```

où *arch* est soit `i586` ou `amd64`.

- Lorsque le programme d'installation est exécuté en mode de simulation (`installer -n`), l'écran de synthèse affiche des messages d'erreur, ainsi qu'un état d'installation « Incomplet ». Cela est incorrect et peut prêter à confusion ; l'installation en mode de simulation n'installe aucun élément sur le système mais crée uniquement un fichier de réponse qui peut être utilisé par la suite pour exécuter l'installation silencieuse. (*Bogue 6594351*)

Installation sur toutes les plates-formes

Ces problèmes affectent le processus d'installation sur toutes les plates-formes.

- L'écran Prêt à installer affiche le nom du produit comme « mq » plutôt que comme Sun Java System Message Queue 4.2. (*Bogue 6650841*)
- Lorsque le programme d'installation est en cours d'exécution pour l'installation de Message Queue 4.2.1 et que l'écran de progression est affiché, le bouton Annuler est actif. Si vous sélectionnez ce bouton à ce stade, l'installation sera incomplète ou endommagée. (*Bogue 6595578*)
- L'écran de synthèse du programme d'installation contient certains liens qui, lorsque vous cliquez dessus, lancent un visualiseur de page de synthèse ou de journal. Si vous fermez la fenêtre du visualiseur en utilisant le bouton « X » au lieu du bouton Fermer, il sera alors impossible de la faire réapparaître. (*Bogue 6587138*)

Solution : Utilisez le bouton Fermer pour fermer la fenêtre.

- Lorsqu'un système dispose d'anciennes versions de Message Queue et de NSS/NSPR, l'écran de mise à niveau du programme d'installation signale uniquement Message Queue comme ayant besoin d'une mise à niveau ; il n'est pas mentionné que NSS/NSPR doivent être mis à

niveau également. Tous les logiciels pertinents seront néanmoins mis à niveau (comme indiqué par l'écran Prêt à installer qui présente les informations correctes). (*Bogue 6580696*)

- La liste des JDK sur l'écran de sélection JDK est active même lorsque l'option "Choisir un JDK" n'est pas sélectionnée. (*Bogue 6650874*)

Anomalies de version dans le programme d'installation

Le programme d'installation affiche en opaque les informations de version de Message Queue. (*Bogue 6586507*)

Sur la plate-forme Solaris, reportez-vous au tableau suivant pour déterminer la Message Queue version affichée par le programme d'installation.

TABLEAU 1-20 Traduction de la chaîne de version

Version telle qu'affichée par le programme d'installation sur le SE Solaris	Version correspondante de Message Queue
4.2.0.0	4.2
4.1.0.2	4.1 Patch 2
4.1.0.1	4.1 Patch 1
4.1.0.0	4.1
3.7.2.1	3.7 UR2 Patch 1
3.7.0.2	3.7 UR2
3.7.0.1	3.7 UR1
3.6.0.0	3.6
3.6.0.4	3.6 SP4
3.6.0.3	3.6 SP3
3.6.0.2	3.6 SP2
3.6.0.1	3.6 SP1

Remarque – Pour les versions de patch jusqu'au 3.6 SP4 (par exemple, 3.6 SP4 Patch 1), la chaîne de version affichée par le programme d'installation reste la même. Vous devez exécuter la commande `imqbrokerd -version` pour déterminer la version exacte.

Sur la plate-forme Linux, le nombre de versions affiché par le programme d'installation se fait sous la forme suivante.

majorReleaseNumber . minorReleaseNumber - someNumber

Par exemple, 3.7–22. Ce numéro signale uniquement qu'il s'agit de l'une des versions 3.7 sans spécifier laquelle. Pour déterminer la version Message Queue installée, exécutez la commande :

```
imqbrokerd -version.
```

Problèmes liés à la localisation

Les bogues suivants sont liés aux problèmes de localisation.

- Lorsque le programme d'installation est exécuté en mode texte (`installer -t`), dans une langue non anglaise, les caractères multioctet apparaissent corrompus. (*Bogue 6586923*)
- Sur l'écran de progression de l'installation, la barre de progression affiche des caractères inconnus. L'infobulle est codée en dur pour les langues non anglaises. (*Bogue 6591632*)
- Le mode Texte (`installer -t`) n'est pas pris en charge sous Windows. L'exécution du programme d'installation en mode texte, sous Windows, entraîne l'affichage d'un message d'erreur. Ce message n'est pas localisé lorsque le programme d'installation est exécuté en langues non anglaises. (*Bogue 6594142*)
- L'écran relatif à la licence du programme d'installation affiche le texte correspondant en anglais, quelle que soit la langue d'exécution du programme. (*Bogue 6592399*)
Solution : Pour accéder à des fichiers de licence localisés, consultez le fichier `LICENSE_MULTILANGUAGE.pdf`.
- L'aide relative à l'utilisation du programme d'installation n'est pas localisée. (*Bogue 6592493*)
- La chaîne « None » apparaissant sur la page HTML de synthèse du programme d'installation est codée en dur en anglais. (*Bogue 6593089*)
- Lorsque le programme d'installation est exécuté dans un environnement linguistique allemand, l'écran d'accueil n'affiche pas le texte complet qui apparaît pour les autres langues. (*Bogue 6592666*)
- La chaîne « Install Home » apparaissant sur la page d'accueil de l'installation n'est pas localisée. Elle s'affiche en anglais même si le programme d'installation est exécuté en langues non anglaises. (*Bogue 6592491*)
- Lorsque le programme d'installation est exécuté en mode texte (`installer -t`), les choix de réponse en anglais « Yes » et « No » sont utilisés quelle que soit la langue d'exécution choisie. (*Bogue 6593230*)
- L'infobulle du bouton Parcourir à l'écran de sélection du JDK est codée en dur en anglais. (*Bogue 6593085*)

Option de mot de passe désapprouvée

Dans les versions précédentes de Message Queue, vous aviez la possibilité d'utiliser l'option `-p` ou `-password` pour spécifier un mot de passe, de manière interactive, pour les commandes suivantes : `imqcmd`, `imqbrokerd` et `imdbmgr`. À partir de la version 4.0, ces options ont été désapprouvées.

À la place, vous pouvez créer un fichier mots de passe spécifiant les mots de passe pertinents et référencer le fichier mots de passe à l'aide de l'option de commande `-passfile`, ou saisir simplement un mot de passe lorsque vous y êtes invité par la commande.

Un fichier de mots de passe peut contenir un ou plusieurs des mots de passe énumérés ci-dessous :

- Un mot de passe de keystore utilisé pour ouvrir le keystore SSL. Utilisez la propriété `imq.keystore.password` pour spécifier ce mot de passe.
- Un mot de passe de référentiel LDAP utilisé pour se connecter, de manière sécurisée, à l'aide d'un répertoire LDAP si la connexion n'est pas anonyme. Utilisez la propriété `imq.user_repository.ldap.password` pour spécifier ce mot de passe.
- Un mot de passe de base de données JDBC utilisé pour se connecter à une base de données compatible JDBC. Utilisez la propriété `imq.persist.jdbc.vendorName.password` pour spécifier ce mot de passe. Le composant *nomFournisseur* du nom de la propriété est une variable spécifiant le fournisseur de la base de données. Vous avez le choix entre `hadb`, `derby`, `pointbase`, `oracle` ou `mysql`.
- Un mot de passe pour la commande `imqcmd` (en vue d'effectuer des tâches d'administration du courtier). Utilisez la propriété `imq.imqcmd.password` pour spécifier ce mot de passe.

Dans l'exemple suivant, le mot de passe pour la base de données JDBC est défini dans le fichier mots de passe sur `abracadabra`.

```
imq.persist.jdbc.mysql.password=abracadabra
```

Vous pouvez utiliser un fichier de mots de passe de l'une des façons suivantes.

- Configurer le courtier pour utiliser le fichier de mots de passe en paramétrant les propriétés suivantes dans son fichier `config.properties`.

```
imq.passfile.enabled=true
imq.passfile.dirpath=passwordFileDirectory
imq.passfile.name=passwordFileName
```

- Utiliser l'option `-passfile` de la commande pertinente, par exemple :

```
imqbrokerd -passfile passwordFileName
```

Problèmes d'administration/de configuration

Les problèmes suivants sont liés à l'administration et à la configuration de Message Queue

- Sur les plate-formes Windows, le pare-feu Windows intégré, activé par défaut, doit être configuré manuellement avec une règle de pare-feu qui permet au courtier d'accepter les connexions entrantes des clients. (*Bogue 6675595*)

1. Double-cliquez sur le pare-feu Windows dans le panneau de configuration

Cliquez sur Continuer sur la boîte de dialogue de configuration du compte utilisateur pour ouvrir la boîte de dialogue Paramétrage du pare-feu Windows.

2. Dans la boîte de dialogue de paramétrage du pare-feu Windows, cliquez sur l'onglet Exceptions.
3. Cliquer sur Ajouter programme.
4. Dans la boîte de dialogue Ajouter un programme, sélectionnez `java.exe` et cliquez sur Naviguer.

Windows identifie le processus de courtier comme système binaire de Java Platform SE. Par conséquent, localisez `java.exe` utilisé par le courtier (habituellement sur `jdk1.5.0_15\jre\bin\java.exe`).

5. Cliquez sur Changement de portée.
6. Dans la boîte de dialogue Changement de portée, sélectionnez "Tout ordinateur (y compris ceux situés sur Internet."
7. Cliquez sur OK.
8. Dans la boîte de dialogue Ajouter un programme, cliquez sur OK.
9. Dans la boîte de dialogue Paramétrer le pare-feu Windows, cliquez sur OK.

- Sur les plate-formes Windows, les commandes `mqadmin` et `mqobjmgr` envoient une erreur lorsque CLASSPATH contient des guillemets. (*Bogue 5060769*)

Solution : Ouvrez une fenêtre d'invite de commande et désactivez CLASSPATH:

```
set classpath=
```

Exécutez ensuite la commande souhaitée, la même fenêtre d'invite de commande, par exemple :

```
mqInstallHome\mq\bin\mqadmin
```

- L'option `-javahome` dans tous les scripts Solaris et Windows ne fonctionne pas si la valeur fournie contient un espace. (*Bogue 4683029*)

L'option `javahome` est utilisée par les commandes et utilitaires de Message Queue pour spécifier une autre exécution Java 2 compatible à utiliser. Cependant, le nom de chemin vers l'exécution Java alternative ne doit pas contenir d'espace. Voici quelques exemples de chemins contenant des espaces :

Windows : `C:/jdk 1.4`

Solaris : /work/java 1.4

Solution : Installez l' exécuteur Java à un emplacement ou un chemin ne contenant pas d' espace.

- L'attribut `imqQueueBrowserMaxMessagesPerRetrieve` spécifie le nombre maximal de messages pouvant être récupérés en une seule fois par l'exécution client lors de la navigation dans une file d'attente. L'attribut affecte la façon dont les messages en file d'attente sont regroupés pour être délivrés à l'exécution client, mais cela n'affecte pas le nombre total de messages faisant l'objet de la navigation. L'attribut affecte uniquement le mécanisme de navigation, cela n'affecte pas la livraison de messages en file d'attente. (*Bogue 6387631*)

Problèmes relatifs au courtier

Les problèmes suivants concernent le courtier de Message Queue.

- Le courtier devient inaccessible lorsque le magasin de données persistant ouvre trop de destinations. (*Bogue 4953354*)

Solution : Ce problème est dû au fait que le courtier atteint la limite du descripteur de fichier ouvert définie pour le système. Sur Solaris et Linux, utilisez la commande `ulimit` pour augmenter cette limite.

- Les consommateurs sont orphelins lorsqu'une destination est supprimée. (*Bogue 5060787*)

Les consommateurs actifs sont orphelins lorsqu'une destination est supprimée. Une fois orphelins, ils ne peuvent plus recevoir de messages (même si la destination est recréée).

- Lorsqu'un client JMS utilisant le service de connexion HTTP met brutalement fin à la connexion (en utilisant, par exemple, `Ctrl+C`), le courtier met environ une minute avant de libérer la connexion client et toutes les ressources associées.

Si une autre instance du client est démarrée dans la minute d'attente en essayant d'utiliser le même ID client, la même souscription durable ou file d'attente, il est possible que celle-ci reçoive une exception « L'ID client est déjà utilisé ». Il ne s'agit pas d'un vrai problème, mais d'un effet secondaire du processus de fin décrit précédemment. Si un client est démarré après un délai d'environ une minute, tout doit fonctionner correctement.

- Lors de l'utilisation d'une base de données MySQL pour un magasin de données, le stockage de messages supérieurs à 1 MB envoie un message "Packet de requête trop grand..." `SQLException`. (*Bogue 6682815*)

Solution : Démarrez le serveur MySQL avec l'option `--max_allowed_packet` paramétrée sur une valeur supérieure à 1 MB par défaut. Par exemple, utilisez la valeur suivante :

```
--max_allowed_packet=60M
```

- Lors de l'utilisation de la base de données Java DB pour un magasin de données, le stockage d'un message envoie un message "Verrouillage impossible à obtenir dans le délai requis" `SQLException`. (*Bogue 6691394*)

Solution : Ajouter la valeur de propriété suivante dans le fichier `config.properties` du courtier.:

```
imq.persist.jdbc.derby.table.MYCONSTATE41.index.IDX2=CREATE INDEX &(index)
ON $(name) (MESSAGE_ID)
```

- Lors de l'utilisation d'une base de données MySQL pour un magasin de données partagées hautement disponible, un mécanisme est nécessaire pour configurer le moteur de stockage de MySQL comme NDBCLUSTER. (*Bogue 6691394*)

Solution : Ajoutez la valeur de propriété suivante dans le fichier `config.properties` du courtier.:

```
imq.persist.jdbc.mysql.tableoption=ENGINE=NDBCLUSTER
```

Clusters de courtiers

Les problèmes suivants affectent les clusters de courtiers.

- Seuls les clusters de courtiers complètement connectés sont pris en charge par cette version. Autrement dit, tous les courtiers d'un cluster doivent communiquer directement avec tous les autres. Si vous essayez de connecter les courtiers dans un cluster conventionnel à l'aide de l'argument de ligne de commande `imqbrokerd -cluster`, assurez-vous que tous les courtiers du cluster sont bien inclus.
- Si un client est connecté à un courtier dans un cluster de courtiers haute disponibilité, l'exécution client tentera de se reconnecter jusqu'à ce qu'elle y parvienne (elle ignore la valeur de l'attribut d'usine de connexion `imqAddressListIterations`.)
- Un client ne peut naviguer que dans les contenus des files d'attente situées sur le courtier qui l'héberge. Il peut toutefois continuer d'envoyer des messages vers les files d'attente ou de consommer les messages provenant des files d'attente sur n'importe quel courtier du cluster, la restriction ne s'appliquant qu'à la navigation.
- Dans un cluster conventionnel avec des courtiers de version 4.2, tous les courtiers doivent avoir la version 3.5 ou ultérieure.
- Les courtiers de Message Queue 4.2 et 4.1 ne peuvent pas interopérer dans un cluster par défaut avec des courtiers Message Queue 3.7 ou 3.6 en raison de la valeur par défaut de `imq.autocreate.queue.maxNumActiveConsumers` modifiée entre ces versions. (*Bogue 6716400*)

Solution : remplacez la valeur par défaut de la chaîne des courtiers de Message Queue 4.2 et 4.1, `imq.autocreate.queue.maxNumActiveConsumers (-1)` par la valeur par défaut de la version précédente (1).

- Lors de la conversion d'un cluster conventionnel en cluster haute disponibilité, vous pouvez utiliser l'utilitaire du Gestionnaire de base de données de Message Queue (`imqdbmgr`) pour convertir un magasin de données basé sur un JDBC autonome en un magasin de données

haute disponibilité partagé. Cette procédure est documentée dans “[Converting a Standalone Data Store to a Shared Data Store](#)” du *Sun Java System Message Queue 4.2 Administration Guide*

- Un courtier utilisant HADB ne peut pas gérer des messages supérieurs à 10 Mo. (*Bogue 6531734*)
- Ce processus de conversion, effectué à l'aide de la commande `imqdbmgr upgrade hastore`, peut échouer, en affichant le message « nombre de verrous trop élevé » si le magasin contient plus de 10 000 messages. (*Bogue 6588856*)

Solution Utilisez la commande suivante pour augmenter le nombre de verrous.

```
hadbm set NumberOfLocks=<desiredNumber>
```

Pour de plus amples informations, reportez-vous à la section « HADB Problems » du manuel *Sun Java System Application Server 9.1 Enterprise Edition Troubleshooting Guide*.

- Si plus de 500 messages distants sont validés dans une transaction, le courtier peut renvoyer l'erreur « HADB-E-12815 : espace de mémoire de table épuisé. » (*Bogue 6550483*)

Pour de plus amples informations, reportez-vous à la section « HADB Problems » du manuel *Sun Java System Application Server 9.1 Enterprise Edition Troubleshooting Guide*.

- Dans un cluster de courtiers, un courtier place des messages en file d'attente sur une connexion distante qui n'a pas démarré (*Bogue 4951010*)

Solution : Le consommateur recevra les messages dès que la connexion est initiée. Les messages seront renvoyés à un autre consommateur si la connexion du consommateur est fermée.

- Lorsque plusieurs messages sont consommés à partir d'un courtier distant dans une transaction, il est possible que le message d'erreur suivant soit consigné vers le courtier. Celui-ci n'est pas important et peut donc être ignoré :

```
[26/Jul/2007:13:18:27 PDT] WARNING [B2117]:
Échec de la réception du message à partir de
mq://129.145.130.95:7677/?instName=a&brokerSessionUID=3209681167602264320:
  ackStatus = NOT_FOUND(404)\
  Raison = Mise à jour de l'état de la transaction distante sur COMMITTED(6) :
transaction 3534784765719091968 introuvable, celle-ci a déjà dû être validée.
  AckType = MSG_CONSUMED
  MessageBrokerSession = 3209681167602264320
  TransactionID = 3534784765719091968
  SysMessageID = 8-129.145.130.95(95:fd:93:91:ec:a0)-33220-1185481094690
  ConsumerUID = 3534784765719133952\par
```

```
[26/Jul/2007:13:18:27 PDT] WARNING Notifier la validation d'une transaction
[8-129.145.130.95(95:fd:93:91:ec:a0)-33220-1185481094690,
[consumer:3534784765719133952, type=NONE]]
TUID=3534784765719091968 got response:
com.sun.messaging.jmq.jmsserver.util.BrokerException:
```

Mise à jour de l'état de la transaction distante sur COMMITTED(6) :
transaction 3534784765719091968 introuvable, celle-ci a déjà dû être validée.:
com.sun.messaging.jmq.jmsserver.util.BrokerException: Mise à jour de la transaction distante sur
COMMITTED(6) : transaction 3534784765719091968 introuvable, celle-ci a déjà dû
être validée.r

Ce message est consigné lors de la notification de validation au courtier de base du message pour les prochains messages de la transaction lorsque la valeur de la propriété `imq.txn.reapLimit` est faible par rapport au nombre de messages distants contenus dans une transaction. (*Bogue 6585449*)

Solution : pour ne pas recevoir ce message, augmentez la valeur de la propriété `imq.txn.reapLimit`.

Problèmes relatifs à JMX

Sur les plates-formes Windows, la méthode `getTransactionInfo` du MBean de contrôle du gestionnaire de transactions retourne des informations de transaction comportant des heures de création incorrectes. (*Bogue 6393359*)

Solution : utilisez plutôt la méthode `getTransactionInfoByID` du MBean de contrôle du gestionnaire de transactions.

Prise en charge de SOAP

Deux problèmes principaux à prendre en compte pour la prise en charge de SOAP.

- À partir de la version 4.0 de Message Queue, la prise en charge des objets administrés par SOAP a été interrompue.
- Le développement de SOAP dépend de plusieurs fichiers : `SUNWjaf`, `SUNWjmail`, `SUNWxsrt` et `SUNWjaxp`. Dans la version 4.1 de Message Queue, ces fichiers sont uniquement disponibles lorsque Message Queue est exécuté avec JDK version 1.6.0 ou version supérieure.
- Précédemment, l'implémentation de SAAJ 1.2 `.jar` référençait directement `mail.jar`. Dans SAAJ 1.3, cette référence a été supprimée ; par conséquent, les clients Message Queue doivent explicitement placer `mail.jar` dans `CLASSPATH`.

Fichiers redistribuables

Sun Java System Message Queue 4.2 contient l'ensemble de fichiers ci-dessous pouvant être utilisés et distribués librement sous forme binaire :

<code>fscontext.jar</code>	<code>jms.jar</code>
<code>imq.jar</code>	<code>libmqcrt.so (HPUX)</code>
<code>imqjmx.jar</code>	<code>libmqcrt.so (UNIX)</code>
<code>imqxm.jar</code>	<code>mqcrt1.dll (Windows)</code>
<code>jaas.jar</code>	

Vous pouvez également redistribuer les fichiers LICENSE et COPYRIGHT.

Accessibilité des personnes handicapées

Pour obtenir la liste des fonctions d'accessibilité mises à disposition depuis la publication de ce média, consultez les évaluations de produit de la Section 508, disponibles sur demande auprès de Sun, afin de déterminer les versions les mieux adaptées au déploiement des solutions accessibles. Des versions à jour des applications sont disponibles sur le site <http://sun.com/software/javaenterprisesystem/get.html>.

Pour plus d'informations sur l'engagement de Sun en matière d'accessibilité, consultez le site suivant : <http://sun.com/access>.

Comment signaler des problèmes et apporter des commentaires

Si vous rencontrez des problèmes avec Sun Java System Message Queue, contactez le service clientèle Sun de l'une des manières suivantes :

- Services de support logiciel en ligne : <http://www.sun.com/service/sunone/software>.
Ce site est lié à la base de savoir de Sun, au centre de support en ligne et à ProductTracker, de même qu'à des programmes de maintenance et des numéros d'assistance téléphonique.
- Le numéro de téléphone indiqué sur votre contrat de maintenance.

Afin que nous puissions vous aider au mieux à résoudre vos problèmes, munissez-vous des informations suivantes lorsque vous contactez le support :

- Description du problème, notamment les conditions dans lesquelles le problème se produit et sa répercussion sur l'opération effectuée.
- Le type de machine, les versions du système d'exploitation et du produit, y compris les patches et autres logiciels pouvant avoir un lien avec le problème.

- Étapes détaillées des méthodes utilisées pour reproduire le problème.
- Journaux des erreurs ou core dumps éventuels.

Forum de Sun Java System

Accédez au forum de Sun Java System Message Queue à partir de l'adresse suivante :

<http://swforum.sun.com/jive/forum.jspa?forumID=24>

Votre participation est la bienvenue.

Forum sur la technologie Java

Il existe un forum JMS au sein des forums sur la technologie Java qui peut être utile.

<http://forum.java.sun.com>

Vos commentaires sont les bienvenus

Dans le souci d'améliorer notre documentation, nous vous invitons à nous faire parvenir vos commentaires et vos suggestions.

Pour nous faire part de vos commentaires, rendez-vous sur le site <http://docs.sun.com>, puis cliquez sur Envoyer commentaires. Dans le formulaire en ligne, indiquez le titre et le numéro de référence du document. La référence est un numéro composé de sept ou neuf chiffres figurant sur la page de garde du manuel ou en haut du document. Par exemple, le titre de ce document est Notes de version de Sun Java System Message Queue 4.2, et son numéro de référence est 820-3701.

Ressources Sun supplémentaires

Vous pouvez obtenir des informations utiles concernant Sun Java System sur les sites Internet suivants :

- Documentation
<http://docs.sun.com/prod/java.sys>
- Services professionnels
<http://www.sun.com/service/sunps/sunone>
- Produits et services logiciels

- <http://www.sun.com/software>
- Services de support logiciel
<http://www.sun.com/service/sunone/software>
- Base de connaissances et support
<http://www.sun.com/service/support/software>
- Services de support et de formation Sun
<http://training.sun.com>
- Services professionnels et de conseil
<http://www.sun.com/service/sunps/sunone>
- Informations pour les développeurs
<http://developers.sun.com>
- Services de support pour les développeurs Sun
<http://www.sun.com/developers/support>
- Formation sur les logiciels
<http://www.sun.com/software/training>