

Notes de version de Sun GlassFish Message Queue 4.4 Update 1



Référence : 821-1513-10
Décembre 2009

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. 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 Sun™ 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 sa 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, DÉCLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE À LA QUALITÉ MARCHANDE, À L'APTITUDE À UNE UTILISATION PARTICULIÈRE OU À L'ABSENCE DE CONTREFAÇON.

Table des matières

1	Notes de version	5
	Historique de révision des notes de mise à jour	6
	À propos de Message Queue 4.4 Mise à jour 1	7
	Plates-formes et composants pris en charge par Message Queue 4.4 Mise à jour 1	7
	Prise en charge de la plate-forme du système d'exploitation	7
	Prise en charge de la virtualisation du système	8
	Dépendances entre composants	8
	Nouvelles fonctions de Message Queue 4.4 Mise à jour 1 et versions récentes	10
	Nouvelles fonctions de Message Queue 4.4 Mise à jour 1	11
	Nouvelles fonctions de Message Queue 4.4	11
	Nouvelles fonctions de Message Queue 4.3	14
	Nouvelles fonctions de Message Queue 4.2	20
	Nouvelles fonctions de Message Queue 4.1	23
	Nouvelles fonctions de Message Queue 4.0	27
	Fonctions qui seront désapprouvées dans les versions futures	31
	Bogues résolus dans Message Queue 4.4 Mise à jour 1 et ses versions récentes	32
	Bogues résolus dans Message Queue 4.4 Mise à jour 1	32
	Bogues résolus dans Message Queue 4.4	35
	Bogues résolus dans Message Queue 4.3	37
	Bogues résolus dans Message Queue 4.2	38
	Bogues résolus dans Message Queue 4.1	39
	Bogues résolus dans Message Queue 4.0	40
	Mises à jour de la documentation relative à Message Queue 4.4 Mise à jour 1	41
	Problèmes de compatibilité	41
	Modifications dans le jeu de documentation de Message Queue 4.4 Mise à jour 1	42
	Problèmes connus et restrictions	43
	Problèmes d'installation	43
	Option de mot de passe désapprouvée	49

Problèmes d'administration/de configuration	50
Problèmes relatifs au courtier	52
Clusters de courtiers	53
Problèmes relatifs à JMX	56
Prise en charge de SOAP	56
Fichiers redistribuables	56
Accessibilité des personnes handicapées	57
Comment signaler des problèmes et apporter des commentaires	57
Forum de Sun GlassFish	57
Forum sur la technologie Java	58
Vos commentaires sont les bienvenus	58
Ressources Sun supplémentaires	58

Notes de version

Version 4.4 Mise à jour 1

Numéro de référence 821-1513-10

Ces notes de version contiennent des informations importantes, disponibles au moment de la commercialisation de Sun GlassFish Message Queue 4.4 Mise à jour 1. 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 4.4 Mise à jour 1.

Ces notes de version contiennent également des informations sur les versions 4.0, 4.1, 4.2, 4.3 et 4.4 de Message Queue. Par exemple, reportez-vous aux sections “[Nouvelles fonctions de Message Queue 4.2](#)” à la page 20, “[Nouvelles fonctions de Message Queue 4.1](#)” à la page 23 et “[Nouvelles fonctions de Message Queue 4.0](#)” à la page 27, respectivement, pour obtenir des informations sur les nouvelles fonctions de ces versions.

Vous trouverez la dernière édition de ces notes de version sur le site Web de documentation de Sun GlassFish Message Queue, à l'adresse <http://docs.sun.com/coll/1307.7>. 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
- “[À propos de Message Queue 4.4 Mise à jour 1](#)” à la page 7
- “[Plates-formes et composants pris en charge par Message Queue 4.4 Mise à jour 1](#)” à la page 7
- “[Nouvelles fonctions de Message Queue 4.4 Mise à jour 1 et versions récentes](#)” à la page 10
- “[Fonctions qui seront désapprouvées dans les versions futures](#)” à la page 31
- “[Bogues résolus dans Message Queue 4.4 Mise à jour 1 et ses versions récentes](#)” à la page 32
- “[Mises à jour de la documentation relative à Message Queue 4.4 Mise à jour 1](#)” à la page 41
- “[Problèmes connus et restrictions](#)” à la page 43

- “Fichiers redistribuables” à la page 56
- “Accessibilité des personnes handicapées” à la page 57
- “Comment signaler des problèmes et apporter des commentaires” à la page 57
- “Vos commentaires sont les bienvenus” à la page 58
- “Ressources Sun supplémentaires” à la page 58

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
Décembre 2009	Version de ce document pour Message Queue 4.4 Mise à jour 1. Ajoute de nouvelles fonctions pour cette version et élimine les problèmes d’installation obsolète du programme d’installation précédent.
Décembre 2009	Deuxième version de ce document pour Message Queue 4.4. Corrige les erreurs et les omissions.
Octobre 2009	Version du présent document pour Message Queue 4.4. Ajoute de nouvelles fonctions pour cette version.
Mai 2009	Version initiale de ce document pour Message Queue 4.4 Beta. Ajoute de nouvelles fonctions pour cette version.
Décembre 2008	Version du présent document pour Message Queue 4.3. Ajoute de nouvelles fonctions pour cette version.
Août 2008	Version du présent document pour Message Queue 4.2. Ajoute de nouvelles fonctions pour cette version.
Septembre 2007	Troisième version de ce document pour Message Queue 4.1. Ajoute une description de la prise en charge de Java Enterprise System Monitoring Framework, des ports C fixes, des corrections de bogues et d’autres fonctions.

TABLEAU 1-1 Historique des révisions (Suite)

Date	Description des modifications
Avril 2007	Seconde version de ce document pour Message Queue 4.1 Beta. Ajout de la fonctionnalité de haute disponibilité.
Janvier 2007	Version initiale de ce document pour Message Queue 4.1 Beta. Ajout d'une description concernant la prise en charge de JAAS.
Mai 2006	Version initiale de ce document pour Message Queue 4.0.

À propos de Message Queue 4.4 Mise à jour 1

Sun GlassFish Message Queue est un service de messagerie complet qui offre des fonctions de messagerie asynchrones et fiables conformes à la spécification de messagerie Java (JMS) 1.1. En outre, Message Queue propose des fonctions supplémentaires par rapport à la spécification JMS pour répondre aux besoins de déploiement des entreprises à grande échelle.

Message Queue 4.4 Mise à jour 1 est une version mineure qui fournit un nouveau programme d'installation basé sur le système Image Packaging (IP) indépendant de la plate-forme, également appelé système pkg (5). En outre, la version 4.4 Mise à jour 1 inclut un certain nombre d'améliorations de fonctions et de corrections de bogues.

Plates-formes et composants pris en charge par Message Queue 4.4 Mise à jour 1

Cette section couvre les sujets suivants liés aux conditions requises du système Message Queue 4.4 Mise à jour 1 :

- “Prise en charge de la plate-forme du système d'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 d'exploitation

Message Queue 4.4 Mise à jour 1 est pris en charge par les plates-formes des systèmes d'exploitation Solaris, Linux, Windows et AIX. 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 *Guide d'installation de Sun Java System Message Queue 4.3*

TABLEAU 1-2 Versions de plates-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)
OpenSolaris	OpenSolaris 2008.11 (x86, x64) OpenSolaris 2009.06 (x86, x64)
Linux	Red Hat Enterprise Linux Advanced Server 4.0, 5.0, toutes les mises à jour, version 32- et 64-bit (x86, x64) Red Hat Enterprise Linux Enterprise Server 4.0, 5.0, toutes les mises à jour, versions 32- et 64-bit (x86, x64)
AIX	AIX 6.1
Windows	Windows Vista Windows XP Professional, SP2 (x86) ¹ Windows Server 2003 Standard et Enterprise Editions, SP2, versions 32 et 64 bits (x86, x64) ² Windows Server 2008 Standard et Enterprise Editions, SP2, versions 32 et 64 bits (x86, x64) ²

¹ Pas de prise en charge sur Home, Tablet PC ou Media Center Edition² 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/820-4651>.

Dépendances entre composants

Outre des configurations requises spécifiques à la plate-forme, Message Queue dépend également de certains composants de base à installer pour développer et exécuter les clients Message Queue. Le [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.

TABLEAU 1-3 Composants de prise en charge requis

Composant	Prend en charge	Versions prises en charge ¹
Java Runtime Environment (JRE)	Outils d'administration et de courtier Message Queue	J2SE™ Runtime Environment version 1.5.0 ou supérieure Java™ SE Runtime Environment 1.6.0
Java Software Development Kit (JDK), Standard Edition	Développement et déploiement du client Java	J2SE Development Kit version 1.5.0 ou supérieure Java SE Development Kit 1.6.0. Notez que JMX requiert la version 1.6.0_10 ou supérieure.

¹ Versions de production Sun Microsystems uniquement

Le [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 pas forcément besoin de tous les composants répertoriés : par exemple, si vous n'écrivez pas de client C, vous n'aurez pas besoin du compilateur C, de la bibliothèque runtime C++, de NSPR ni de NSS.

TABLEAU 1-4 Composants de prise en charge en option

Composant	Prend en charge	Versions prises en charge
Serveur d'application	HTTP/HTTPS	Sun GlassFish Application Server Enterprise Edition, version 9.1.1 Sun GlassFish Enterprise Server v2.1, v2.1.1, v3
Serveur web	HTTP/HTTPS	Sun GlassFish Web Server Enterprise Edition, Version 7.0, Mise à jour 3
Base de données	Magasin de données basé sur JDBC	HADB, versions 4.4.3, 4.5, 4.6 Java DB (Apache Derby), Version 10.4 MySQL Community/Enterprise Edition, Version 5.1 Oracle 9.2i, 10g et 11g postgresql, Version 8.1 Remarque – La base de données PointBase n'est plus prise en charge.

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

Composant	Prend en charge	Versions prises en charge
Base de données hautement disponible	Clusters de courtiers haute disponibilité	HADB, versions 4.4.3, 4.5, 4.6 MySQL Cluster Edition, version 5.1.39-ndb-7.0.9 Oracle 10g et 11g
Serveur de répertoires Lightweight Directory Access Protocol (LDAP)	Référentiel utilisateur et objets administrés de Message Queue	Sun GlassFish Directory Server, Version 6.0
JNDI (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 12 ou supérieure, compilateur C++ avec mode standard et compilateur C Linux : gcc/g++, Version 3.4.6 Windows : Microsoft Windows Visual Studio, version 2008 SP1
Netscape Portable Runtime (NSPR)	Clients C de Message Queue	Version 4.7
Network Security Services (NSS)	Clients C de Message Queue	Version 3.12.3

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

Nouvelles fonctions de Message Queue 4.4 Mise à jour 1 et versions récentes

Les nouvelles fonctions de Message Queue 4.4 Mise à jour 1 et des versions précédentes de la gamme Message Queue 4.x sont décrites dans les sections suivantes :

- “Nouvelles fonctions de Message Queue 4.4 Mise à jour 1” à la page 11
- “Nouvelles fonctions de Message Queue 4.4” à la page 11
- “Nouvelles fonctions de Message Queue 4.3” à la page 14
- “Nouvelles fonctions de Message Queue 4.2” à la page 20
- “Nouvelles fonctions de Message Queue 4.1” à la page 23
- “Nouvelles fonctions de Message Queue 4.0” à la page 27

Nouvelles fonctions de Message Queue 4.4 Mise à jour 1

Message Queue 4.4 Mise à jour 1 est une version mineure apportant un certain nombre d'améliorations de fonctions et de corrections de bogues. Cette section décrit les nouvelles fonctions de cette version :

- “Nouveau programme d'installation” à la page 11
- “Prise en charge de la journalisation des transactions des clusters” à la page 11
- “Courtier en cours” à la page 11

Nouveau programme d'installation

Message Queue 4.4 Mise à jour 1 fournit un nouveau programme d'installation multiplate-formes basé sur le système `pkq(5)`, également appelé IPS ou Image Packaging System. Pour plus d'informations sur ce programme d'installation, reportez-vous au document *Sun GlassFish Message Queue 4.4 Update 1 Installation Guide*.

Prise en charge de la journalisation des transactions des clusters

Message Queue 4.4 Mise à jour 1 ajoute un mécanisme de persistance des transactions destiné aux magasins de données basés sur les fichiers et qui prend en charge les clusters de courtiers. Ce mécanisme fournit d'autres fonctionnalités décrites à la section “[Optimizing File-Based Transaction Persistence](#)” du *Sun GlassFish Message Queue 4.4 Administration Guide*.

Courtier en cours

Message Queue 4.4 Mise à jour 1 prend en charge l'exécution d'un courtier au sein d'un client Java. Ce courtier, appelé courtier *en cours* ou *intégré*, s'exécute dans la même JVM que le client Java qui le crée et le démarre. Pour plus d'informations, reportez-vous au [Chapitre 6](#), “[Embedding a Message Queue Broker in a Java Client](#)” du *Sun GlassFish Message Queue 4.4 Developer's Guide for Java Clients*.

Nouvelles fonctions de Message Queue 4.4

Message Queue 4.4 est une version mineure apportant un certain nombre d'améliorations de fonctions et de corrections de bogues. Cette section décrit les nouvelles fonctions de cette version :

- “Service de pont JMS” à la page 12
- “Service de pont STOMP” à la page 12
- “Améliorations supplémentaires” à la page 13

Service de pont JMS

La spécification JMS ne définissant pas un protocole câblé pour la communication entre les courtiers et les clients, chaque fournisseur JMS (y compris ceux de Message Queue) a défini et utilise son propre protocole propriétaire. Cette situation a mené à une non-interopérabilité entre les fournisseurs JMS.

Le service de pont JMS dans Message Queue 4.4 met fin à cette désynchronisation en activant un courtier Message Queue destiné à mapper ses destinations à celles de fournisseurs JMS externes. Concrètement, ce mappage permet au courtier de Message Queue de communiquer avec des clients du fournisseur JMS externe.

Le service de pont JMS prend en charge les destinations de mappage des fournisseurs JMS externes :

- Compatibles JMS1.1
- Prenant en charge les objets administratifs JNDI
- Utilisant des fabriques de connexion de type `javax.jms.ConnectionFactory` ou `javax.jms.XAConnectionFactory`
- Dans le cadre d'un mappage transactionnel, prenant en charge les interfaces XA comme un gestionnaire de ressources

De nombreux fournisseurs JMS open source et commerciaux répondent à ces exigences, de sorte que le service de pont JMS permet d'intégrer efficacement Message Queue dans un environnement de messagerie existant qui utilise d'autres fournisseurs JMS.

Pour plus d'informations sur le service de pont JMS, voir :

- Pour plus d'informations sur l'architecture, les sous-composants et les capacités du service de pont JMS, reportez-vous à la section “[JMS Bridge Service](#)” du *Sun GlassFish Message Queue 4.4 Technical Overview*.
- Pour plus d'informations sur la configuration et la gestion des ponts JMS dans un courtier, reportez-vous à la section “[Configuring and Managing JMS Bridge Services](#)” du *Sun GlassFish Message Queue 4.4 Administration Guide*.

Service de pont STOMP

Comme indiqué précédemment, la spécification JMS ne définit pas de protocole câblé pour la communication entre les courtiers et les clients. Le projet open source STOMP (Streaming Text Oriented Messaging Protocol), disponible à la page <http://stomp.codehaus.org>, définit un protocole câblé simple que les clients écrits dans n'importe quel langage peuvent utiliser pour communiquer avec un fournisseur de messagerie prenant en charge le protocole STOMP.

Message Queue 4.4 prend en charge le protocole STOMP via le service de pont STOMP. Ce service permet à un courtier Message Queue de communiquer avec les clients STOMP.

Pour plus d'informations sur le service de pont STOMP :

- Pour plus d'informations sur l'architecture et les capacités du service de pont STOMP, reportez-vous à la section “STOMP Bridge Service” du *Sun GlassFish Message Queue 4.4 Technical Overview*.
- Pour plus d'informations sur la configuration et la gestion d'un pont STOMP dans un courtier, reportez-vous à la section “Configuring and Managing STOMP Bridge Services” du *Sun GlassFish Message Queue 4.4 Administration Guide*.

Améliorations supplémentaires

Les améliorations supplémentaires suivantes sont également disponibles dans Message Queue 4.4 :

- “Nouvelles fonctions UMS (Universal Message Service)” à la page 13
- “Package IPS pris en charge” à la page 13
- “Fonction de journalisation d'audit réintégré” à la page 14

Nouvelles fonctions UMS (Universal Message Service)

Le service UMS fournit maintenant des fonctions qui utilisent HTTP GET pour offrir plusieurs services :

- **getbrokerinfo**: récupérer des informations sur le courtier.
- **getconfiguration** : récupérer les informations relatives à la configuration UMS.
- **debug** : activer ou désactiver l'enregistrement de débogage sur le serveur UMS.
- **ping** : communiquer avec le courtier pour confirmer qu'il est en cours d'exécution.

Pour plus d'informations sur ces nouvelles fonctions, reportez-vous à la section “Query and utility functions using HTTP GET” de la page <https://mq.dev.java.net/4.4-content/imqums/protocol.html>.

Pour une présentation des UMS, reportez-vous à la section “Universal Message Service (UMS)” à la page 14. Pour la documentation de l'API UMS, consultez la page <https://mq.dev.java.net/4.4-content/imqums/protocol.html>. Pour obtenir des exemples de programmation dans plusieurs langages, reportez-vous à la page <https://mq.dev.java.net/4.4-content/ums/examples/README.html>.

Package IPS pris en charge

Message Queue est désormais fourni pour distribution avec le système IPS (Image Packaging System, système de conditionnement d'image) open source, également appelé système pkg (5). Cette méthode de conditionnement a été ajoutée afin que Message Queue puisse être intégré à Sun GlassFish Enterprise Server 2.1.1.

Fonction de journalisation d'audit réintégré

Message Queue 3.7 fournissait une fonction de journalisation d'audit qui a été supprimée dans la version 4.0. Cette fonction a été réintégré dans Message Queue 4.4. Pour plus d'informations sur cette fonction, reportez-vous à la section “[Audit Logging](#)” du *Sun GlassFish Message Queue 4.4 Administration Guide*.

Nouvelles fonctions de Message Queue 4.3

Message Queue 4.3 était une version mineure apportant un certain nombre d'améliorations de fonctions et des corrections de bogues. Cette section décrit les nouvelles fonctions incluses dans cette version :

- “[Universal Message Service \(UMS\)](#)” à la page 14
- “[Prise en charge de la plate-forme AIX](#)” à la page 17
- “[Nouveau programme d'installation à partir de zip](#)” à la page 18
- “[Prise en charge d'une plate-forme étendue](#)” à la page 18
- “[Améliorations supplémentaires](#)” à la page 18

Universal Message Service (UMS)

Message Queue 4.3 introduit un nouveau service UMS (Universal Messaging Service, service de messagerie universelle) et une API de messagerie qui permet d'accéder à Message Queue à partir de n'importe quel appareil compatible http. Par conséquent, la plupart des applications peuvent communiquer avec toute autre application et profiter de la fiabilité et de l'envoi garanti de messages JMS. En outre, le service UMS offre une évolutivité améliorée pour la messagerie JMS, permettant d'augmenter le nombre de clients de messagerie à l'échelle d'Internet.

Architecture

La figure suivante illustre l'architecture UMS de base :

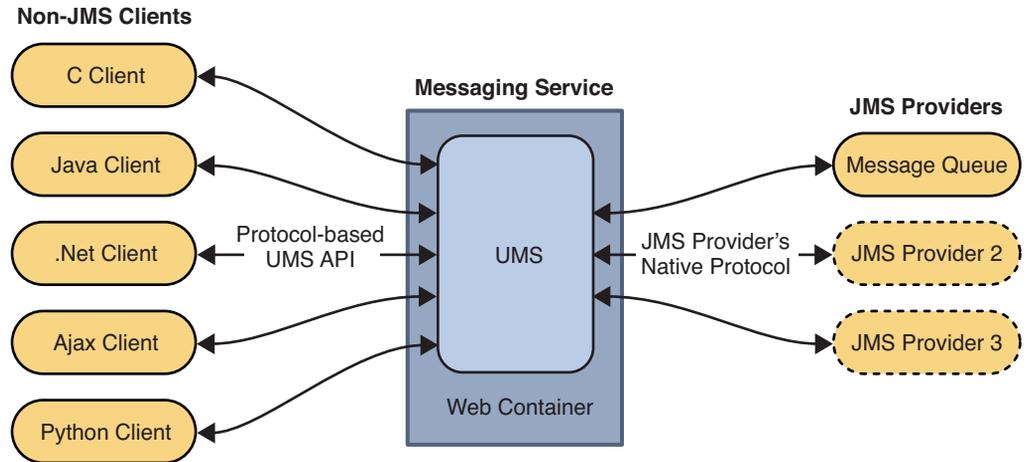


FIGURE 1-1 Architecture UMS

Le service UMS, qui s'exécute sur un serveur Web, est un langage neutre et indépendant de la plate-forme. Le service UMS sert de pont entre une application cliente non JMS et un fournisseur JMS. Il reçoit les messages envoyés via l'API UMS, les transforme en messages JMS, puis les envoie aux différentes destinations dans le fournisseur JMS par le biais du protocole natif du fournisseur. De même, il extrait les messages des destinations du fournisseur JMS, les transforme en texte ou en messages SOAP, puis envoie les messages aux clients non JMS comme demandé par les clients via l'API UMS.

L'API UMS, basée sur un protocole simple, non dépendant d'un langage, prend en charge les applications basées sur le Web et non basées sur le Web, et peut être utilisée avec les langages de script et de programmation. L'API est proposée dans deux styles : une API de messagerie simple qui utilise un protocole de type REST (Representational State Transfer) et une API de messagerie XML qui incorpore le protocole dans les en-têtes de messages SOAP. Dans les deux cas, cependant, l'API ne requiert qu'une seule requête http pour envoyer ou recevoir un message.

La simplicité et la flexibilité de l'API UMS signifie que AJAX, .NET, Python, C, Java et bien d'autres applications peuvent envoyer un message texte et/ou des messages SOAP (avec pièce jointe) aux destinations JMS ou recevoir de messages de destinations JMS. Par exemple, les applications Python peuvent communiquer avec les applications .NET, les applications iPhone avec les applications Java, etc.

Pour Message Queue 4.3, le service UMS prend uniquement en charge Message Queue en tant que fournisseur JMS.

Autres fonctions

Le service UMS est plus que le simple pont décrit ci-dessus. Il prend en charge les sessions client avec ou sans état. Si le client le demande, le service UMS conservera l'état de la session pour l'application client sur plusieurs requêtes de service. Le service UMS peut utiliser l'authentification par conteneur, être configuré pour authentifier les clients à l'aide du courtier Message Queue, ou les deux. Le service UMS prend également en charge les transactions, permettant ainsi aux applications clientes de valider ou répéter plusieurs requêtes de service comme une seule unité atomique.

Le service UMS pouvant prendre en charge un grand nombre de clients sur une connexion unique avec le courtier Message Queue, il facilite la charge des services de connexion du courtier, pour une évolutivité optimale. En outre, la capacité UMS peut être augmentée par mise à l'échelle horizontale, ce qui permet des charges de messagerie de niveau Internet.

Côté client, en raison de la simplicité de l'API UMS basée sur protocole, aucun client de bibliothèque n'est requis. Par conséquent, l'API peut être étendue à l'avenir pour mettre en œuvre des fonctions JMS sans mettre à niveau les applications clientes.

Utilisation du service UMS

Pour utiliser le service UMS, vous devez le déployer dans un conteneur Web prenant en charge les spécifications Servlet 2.4 ou version ultérieure, démarrer le courtier Message Queue, créer les destinations appropriées et écrire une application de messagerie qui utilise l'API UMS pour envoyer ou recevoir des messages.

Le fichier UMS `imqms.war`, contenu dans la distribution de Message Queue 4.3, est installé à l'emplacement suivant, en fonction de la plate-forme :

Vous pouvez renommer le fichier `.war`.

TABLEAU 1-5 Emplacement du fichier `imqms.war`

Plate-forme	Emplacement du fichier <code>imqms.war</code>
Solaris	<code>/usr/share/lib/imq</code>
Linux	<code>/opt/sun/mq/share/lib</code>
AIX	<code>IMQ_HOME/lib</code>
Windows	<code>IMQ_HOME\lib</code>

Après avoir déployé le fichier `imqms.war` dans un conteneur Web à `localhost:port`, vous pouvez trouver la documentation UMS disponible à l'adresse suivante :

`http://localhost:port/imqms`

Sinon, vous pouvez trouver la documentation UMS comme suit :

- Pour plus d'informations sur la configuration du service UMS, reportez-vous à la page <https://mq.dev.java.net/4.3-content/ums/config.html>.
- Pour la documentation de l'API UMS, consultez la page <https://mq.dev.java.net/4.3-content/ums/protocol.html>.
- Pour obtenir des exemples de programmation dans plusieurs langages, reportez-vous à la page <https://mq.dev.java.net/4.3-content/ums/examples/README.html>.

Conteneurs Web pris en charge

Le service UMS est actuellement pris en charge sur les conteneurs Web suivants :

- Sun GlassFish Enterprise Server, Version 2.1 et Version 3 Prelude
- Tomcat, Versions 5.5 et 6.0

Prise en charge de la plate-forme AIX

Message Queue 4.3 fournit des packages pour la plate-forme AIX et un programme d'installation pour les installer.

L'implémentation AIX de Message Queue prend en charge les logiciels suivants :

- AIX v 6.1 ou version supérieure (les versions précédentes de AIX sont prises en charge via l'ensemble Unix/Java uniquement)
- Prise en charge de DB2
- IBM XL C/C++ V9.0
- JDK 1.5 ou supérieur

Pour obtenir des instructions sur la procédure d'installation, reportez-vous au Chapter 4, "AIX Installation," du document *Sun GlassFish Message Queue 4.3 Installation Guide*.

Sur la plate-forme AIX, les fichiers Message Queue sont installés sous un répertoire de base Message Queue unique, `IMQ_HOME`. `IMQ_HOME` indique le répertoire `mqInstallHome/mq`, où `mqInstallHome` est le répertoire d'installation de base que vous avez spécifié lors de l'installation du produit (par défaut, `home-directory /MessageQueue`).

La structure du répertoire de Message Queue qui en résulte est identique à celle de la plate-forme Windows (reportez-vous à la section Windows de l'[Annexe A](#), "Distribution-Specific Locations of Message Queue Data" du *Sun GlassFish Message Queue 4.4 Administration Guide*).

La prise en charge de Message Queue pour la plate-forme AIX comprend la prise en charge de la C-API de Message Queue. Pour obtenir des instructions sur la création et la compilation d'applications C sur la plate-forme AIX, reportez-vous à XREF.

Nouveau programme d'installation à partir de zip

Message Queue 4.3 introduit un nouveau programme d'installation pour les distributions basées sur des zips, par opposition aux distributions de packages natifs. Le programme d'installation permet d'installer les nouvelles distributions .zip de Message Queue pour la plate-forme AIX.

Le nouveau programme d'installation extrait les fichiers .zip de Message Queue vers tout répertoire pour lequel vous disposez de droits d'accès en écriture (il n'est pas nécessaire de disposer de droits d'accès de superutilisateur), et il vous permet également d'enregistrer votre installation de Message Queue avec Sun Connection.

Pour réduire la taille de téléchargement des bundles, Java Runtime n'est plus inclus dans le zip de la distribution (la plupart des sites en sont déjà équipés). Par conséquent, la commande `install` nécessite de spécifier un JDK ou un JRE, soit à l'aide de la variable d'environnement `JAVA_HOME`, soit à l'aide de l'option `-j` de la ligne de commande, comme suit :

```
$ install -j JDK/JRE-path
```

où *JDK/JRE-path* correspond au chemin d'accès au JDK ou JRE spécifié.

Prise en charge d'une plate-forme étendue

La prise en charge de la plate-forme mise à jour suivante sera certifiée pour Message Queue 4.3 :

- Oracle 11g
- Windows Server 2008

Améliorations supplémentaires

Les améliorations supplémentaires suivantes sont incluses dans Message Queue 4.3 :

- [“Nouvelle structure de répertoires sur les plates-formes Windows”](#) à la page 18
- [“Nouvelles propriétés de courtier”](#) à la page 19
- [“Améliorations de l'API d'administration JMX”](#) à la page 19
- [“Liste des abonnements durables pour les abonnés de messages génériques”](#) à la page 19

Nouvelle structure de répertoires sur les plates-formes Windows

La structure de répertoires installée pour Message Queue sur la plate-forme Windows a été modifiée par rapport aux versions précédentes pour la rendre identique à celle de la plate-forme AIX. Cette structure de répertoires sera également appliquée aux plates-formes Solaris et Linux dans le futur, afin de faciliter les installations multiples sur seul ordinateur et la mise à jour automatique de Message Queue via Sun Connection, un service hébergé par Sun qui vous aide à suivre, organiser et entretenir le matériel et les logiciels Sun (reportez-vous à la section [“Prise en charge du programme d'installation pour l'enregistrement à Sun Connection”](#) à la page 21).

Nouvelles propriétés de courtier

Les nouvelles propriétés suivantes sont disponibles pour la configuration d'un courtier :

TABLEAU 1-6 Routage du courtier et propriétés de distribution

Propriétés	Type	Valeur par défaut	Description
<code>imq.transaction.producer.maxNumMsgs</code>	Entier	1000	Nombre maximal de messages qu'un producteur peut traiter en une seule transaction. Il est recommandé que la valeur soit inférieure à 5 000 pour éviter d'épuiser les ressources.
<code>imq.transaction.consumer.maxNumMsgs</code>	Entier	100	Nombre maximal de messages qu'un consommateur peut traiter en une seule transaction. Il est recommandé que la valeur soit inférieure à 1 000 pour éviter d'épuiser les ressources.
<code>imq.persist.jdbc.connection.limit</code>	Entier	5	Nombre maximal de connexions qui peuvent être ouvertes à la base de données.

Améliorations de l'API d'administration JMX

Un nouvel attribut et des clés de données composites ont été ajoutés à l'API JMX, comme suit :

- Un attribut `NextMessageID` a été ajouté au MBean de contrôle de destination afin de fournir l'ID de message JMS du prochain message envoyé à un consommateur.
- Une clé `NextMessageID` pour date composite a été ajoutée au MBean de contrôle du gestionnaire de consommateurs pour fournir l'ID de message JMS du prochain message envoyé au consommateur.
- Une clé `NumMsgsPending` pour date composite a été ajoutée au MBean de contrôle du gestionnaire de consommateurs pour fournir le nombre de messages distribués au consommateur.

Pour de plus amples informations, reportez-vous au [Chapitre 3, “Message Queue MBean Reference”](#) du *Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients*.

Liste des abonnements durables pour les abonnés de messages génériques

La commande permettant de dresser une liste des abonnements durables :

```
list dur [-d topicName]
```

a été améliorée afin que la spécification du nom de la rubrique soit facultative. Si la rubrique n'est pas spécifiée, la commande crée une liste de l'ensemble des abonnements durables pour toutes les rubriques (y compris ceux comportant des conventions d'attribution de noms avec des caractères génériques)

Nouvelles fonctions de Message Queue 4.2

Message Queue 4.2 était une version mineure apportant un certain nombre d'améliorations de fonctions et des corrections de bogues. Cette section décrit les nouvelles fonctions de la version 4.2 et fournit de plus amples références destinées à votre usage :

- “Destinations multiples pour un éditeur ou un abonné” à la page 20
- “Validation des schémas des messages de charge utile XML” à la page 20
- “Prise en charge de C-API pour les transactions distribuées” à la page 21
- “Prise en charge du programme d'installation pour l'enregistrement à Sun Connection” à la page 21
- “Prise en charge de la base de données MySQL” à la page 22
- “Améliorations supplémentaires” à la page 22

Pour de plus amples informations sur les fonctions introduites dans Message Queue 4.1 et 4.0, reportez-vous respectivement aux sections “Nouvelles fonctions de Message Queue 4.1” à la page 23 et “Nouvelles fonctions de Message Queue 4.0” à la page 27.

Destinations multiples pour un éditeur ou un abonné

Dans Message Queue 4.2, un éditeur peut 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 des noms de destination de sujet symboliques et d'autres exemples de leur utilisation sont décrits à la section “Supported Topic Destination Names” du *Sun GlassFish Message Queue 4.4 Administration Guide*.

Validation des schémas des messages de charge utile XML

Cette fonctionnalité, introduite dans Message Queue 4.2, permet de valider le contenu d'un message XML texte (pas objet) 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.)

Pour plus d'informations concernant l'utilisation de cette fonction, reportez-vous à la section [“Schema Validation of XML Payload Messages”](#) du *Sun GlassFish Message Queue 4.4 Developer's Guide for Java Clients*.

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, la C-API de Message Queue prend désormais en charge l'interface XA (entre un gestionnaire de transactions distribuées et Message Queue en tant que gestionnaire de ressources conforme à XA), permettant aux clients de la 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.

Pour plus de détails sur l'utilisation des fonctions de transaction distribuée, reportez-vous à la section [“Working With Distributed Transactions”](#) du *Sun GlassFish Message Queue 4.4 Developer's Guide for C Clients*.

Message Queue 4.2 fournit des exemples de programmation sur la base du gestionnaire de transactions Tuxedo. Pour plus d'informations sur l'utilisation de ces programmes échantillons, reportez-vous à la section [“Distributed Transaction Sample Programs”](#) du *Sun GlassFish Message Queue 4.4 Developer's Guide for C Clients*.

Remarque – La fonctionnalité de transaction distribuée est prise en charge sous Solaris, Linux et Windows. Cependant, à ce jour, elle a uniquement été certifiée sur la plate-forme Solaris.

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 mettre à jour votre matériel et vos logiciels Sun.

Dans le cadre 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 transmises de façon sécurisée à la base de données de Sun Connection. Le service d'inventaire de Sun Connection peut vous aider à organiser vos logiciels et votre 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 de fonctions.

Pour plus d'informations sur l'enregistrement de Message Queue avec Sun Connection, reportez-vous au *Sun Java System Message Queue 4.3 Installation Guide*.

Prise en charge de la base de données MySQL

Message Queue 4.2 a introduit la prise en charge de la base de données MySQL en tant que magasin de données basé sur JDBC. MySQL Cluster Edition peut être utilisé comme base de données JDBC pour un courtier autonome et comme magasin de données partagées hautement disponible nécessaire à un cluster de courtier amélioré. Pour de plus amples informations sur la configuration de Message Queue pour utiliser MySQL, reportez-vous aux sections [“Configuring a JDBC-Based Data Store”](#) du *Sun GlassFish Message Queue 4.4 Administration Guide* du [“Enhanced Broker Cluster Properties”](#) du *Sun GlassFish Message Queue 4.4 Administration Guide*.

Améliorations supplémentaires

En plus des fonctions décrites ci-dessus, Message Queue 4.2 inclut les améliorations suivantes :

■ Métriques de messages produites à distance

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 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).

Deux nouvelles quantités métriques de destination physique sont incluses dans Message Queue 4.2 :

Nb de messages à distance et Nbre total d'octets de message à distance. Les nouvelles quantités métriques sont disponibles via les commandes `imqcmd list dst` et `imqcmd query dst` (voir la section [“Viewing Physical Destination Information”](#) du *Sun GlassFish Message Queue 4.4 Administration Guide*) et via de nouveaux attributs JMX (voir la section [“Destination Monitor”](#) du *Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients*).

■ Informations relatives aux producteurs et consommateurs de messages génériques

Des informations relatives à la prise en charge de l'utilisation de caractères génériques dans les noms de destination (voir la section [“Destinations multiples pour un éditeur ou un abonné” à la page 20](#)) sont fournies par le biais de nouvelles données de contrôle. Par exemple, le nombre de producteurs ou consommateurs de messages génériques associé à une destination est accessible via la commande `imqcmd query dst` (voir la section [“Viewing Physical Destination Information” du *Sun GlassFish Message Queue 4.4 Administration Guide*](#)) et via de nouveaux attributs JMX (voir la section [“Destination Monitor” du *Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients*](#)). Les informations relatives aux messages génériques sont également disponibles via les MBeans de contrôle du gestionnaire de consommateurs et de contrôle du gestionnaire des producteurs.

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

Message Queue 4.2 a introduit la prise en charge du format de 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**

Message Queue 4.2 a introduit l'authentification JAAS par adresse IP ainsi que par nom d'utilisateur.

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 24](#)
- [“Prise en charge de JAAS” à la page 25](#)
- [“Modification du format du magasin de données persistant” à la page 25](#)
- [“Configuration de l'environnement du courtier” à la page 25](#)
- [“Prise en charge de Java ES Monitoring Framework” à la page 26](#)

- “Gestion améliorée des transactions” à la page 26
- “Ports fixes pour les connexions de clients C” à la page 26

Pour obtenir des informations sur les nouvelles fonctions de Message Queue 4.0, reportez-vous à la section “Nouvelles fonctions de Message Queue 4.0” à la page 27.

Clusters de courtiers haute disponibilité

Message Queue 4.1 a introduit un nouveau cluster de courtiers amélioré. 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 améliorés assurent eux la disponibilité des *données* (si un courtier échoue, ses messages persistants et données d'état sont disponibles pour qu'un autre courtier puisse le relayer et délivrer les messages).

L'implémentation de haute disponibilité introduite dans Message Queue 4.1 utilise un magasin de données basé sur JDBC partagé : plutôt que tous les courtiers d'un cluster de courtiers disposent de leur propre magasin de données persistantes, tous les courtiers du cluster partagent la même base de données compatible JDBC. Si un courtier particulier échoue, un autre courtier du cluster prend sa relève pour acheminer et livrer ses messages. 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 amélioré pour 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 de courtiers amélioré, 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 détecté et géré à l'aide d'un courtier de substitution.

L'implémentation de clusters de courtiers améliorés 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 courtiers améliorés et obtenir une comparaison de ceux-ci par rapport aux clusters conventionnels, reportez-vous au [Chapitre 4, “Broker Clusters” du *Sun GlassFish Message Queue 4.4 Technical Overview*](#). Pour obtenir des informations procédurales et référentielles sur les clusters de courtiers améliorés, reportez-vous au [Chapitre 10, “Configuring and Managing Broker Clusters” du *Sun GlassFish Message Queue 4.4 Administration Guide*](#) et à la section “Cluster Configuration Properties” du [Sun GlassFish Message Queue 4.4 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 amélioré, 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 53](#) 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 d'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 la section [“Using JAAS-Based Authentication” du *Sun GlassFish Message Queue 4.4 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 améliorés. Pour cette raison, le format du magasin de données basé sur JDBC est passé à la version 410. Les versions aux formats 350, 370 et 400 sont automatiquement migrées vers la version 410.

Veuillez noter que le format du magasin de données persistant basé sur les fichiers reste à la version 370 car aucun changement n'a été apporté 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 commune. 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 des 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 dont l'état est l'un des suivants : STARTED, FAILED, INCOMPLETE, COMPLETE et PREPARED.

Pour vous aider à déterminer si une transaction particulière peut être annulée ou pas (en particulier son état n'est pas PREPARED), l'utilitaire Commande fournit des données supplémentaires dans la sortie `imqcmd query txn` : il fournit l'ID de connexion pour la connexion à l'origine de la transaction et indique l'heure à laquelle la transaction a été créée. À l'aide de ces informations, un administrateur peut alors décider d'annuler ou non 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 27
- “Journalisation à l'exécution client” à la page 28
- “API de notification des événements de connexion” à la page 28
- “Améliorations de l'administration du courtier” à la page 28
- “Affichage des informations concernant le magasin de données basé sur JDBC” à la page 29
- “Prise en charge du fournisseur JDBC” à la page 30
- “Modifications du format du magasin de données persistant” à la page 30
- “Propriétés supplémentaires des messages” à la page 30
- “Prise en charge de SSL” à la page 30



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 49 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 GlassFish Message Queue 4.4 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 améliorés.
- 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 16, “Command Line Reference”](#) du *Sun GlassFish Message Queue 4.4 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 la version de la 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.

```
imqdbmgr query
```

```
[04/Oct/2005:15:30:20 PDT] Using plugged-in persistent store:
    version=400
    brokerid=Mozart1756
    database connection url=jdbc:oracle:thin:@Xhome:1521:mqdb
    database user=scott
Running in standalone mode.
Database tables have already been created.
```

Prise en charge du fournisseur JDBC

Dans Message Queue 4.0, Apache Derby Version 10.1.1 est désormais pris en charge 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 JDBC est passé à 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
```

Fonctions qui seront désapprouvées dans les versions futures

Les fonctions suivantes seront désapprouvées dans une version ultérieure :

- **Surveillance basée sur les messages**

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 la section [“Prise en charge de l'API d'administration JMX” à la page 27](#)). 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.

- **Programme d'installation en mode texte**

Le mode texte du programme d'installation de Message Queue (`install -t`) sera supprimé sur toutes les plates-formes de système d'exploitation. En mode texte, le texte simple s'affiche dans la fenêtre du terminal pour simuler l'apparence du mode d'interface graphique (IG). Le mode IG et le mode silencieux seront toujours pris en charge.

- **Plates-formes prises en charge**

Windows 2000 et Red Hat Linux 3 ne seront plus pris en charge dans les versions futures.

- **Adaptateur de ressources JMSRA**

L'adaptateur de ressources de Message Queue, `imqjmsra.rar`, généralement référencé comme JMSRA, sera remplacé dans une version future de Message Queue par un nouvel adaptateur de ressources. JMSRA est utilisé pour intégrer Message Queue au serveur Sun Java System Application Server.

Le nouvel adaptateur de ressources, qui va combiner les fonctions existantes de JMSRA avec les fonctions d'autres adaptateurs de ressources JMS Sun, fourniront une prise en charge spécialisée pour Message Queue, ainsi que d'autres fournisseurs, dans un environnement de

serveur Java EE 5 Application Server. À ce titre, elle sera utilisée pour intégrer Message Queue à un serveur Sun GlassFish Enterprise Server et à la suite Sun Java Application Composite Platform Suite (Java CAPS).

Bogues résolus dans Message Queue 4.4 Mise à jour 1 et ses versions récentes

Message Queue 4.4 Mise à jour 1 inclut de nouveaux correctifs de bogues et intègre également les bogues qui ont été résolus dans les versions précédentes de la famille Message Queue 4.x.

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

- [“Bogues résolus dans Message Queue 4.4 Mise à jour 1” à la page 32](#)
- [“Bogues résolus dans Message Queue 4.4” à la page 35](#)
- [“Bogues résolus dans Message Queue 4.3” à la page 37](#)
- [“Bogues résolus dans Message Queue 4.2” à la page 38](#)
- [“Bogues résolus dans Message Queue 4.1” à la page 39](#)
- [“Bogues résolus dans Message Queue 4.0” à la page 40](#)

Bogues résolus dans Message Queue 4.4 Mise à jour 1

Le tableau suivant décrit les bogues résolus dans Message Queue. Certains de ces problèmes sont marqués "(OpenMQ)", ce qui indique que le problème a été résolu dans le projet open source [Open Message Queue](#) sur lequel Sun GlassFish Message Queue est basé.

TABLEAU 1-7 Bogues résolus dans Message Queue 4.4 Mise à jour 1

Bogue	Description
6590909	La MDB de mode DIRECT ne se connecte pas au courtier distant lorsque la liste d'adresses est remplacée.
6616704	La mémoire du courtier s'accroît lorsque de nombreux consommateurs sont créés au cours d'une session.
6745761	XAResource.isSameRM() doit renvoyer true lorsque deux connexions sont utilisées dans le même TX XA (avec JMSJCA).
6745763	XAResource.isSameRM() doit renvoyer true lorsque deux connexions sont utilisées dans le même TX XA (en mode JMSRA DIRECT).
6745768	XAResource.isSameRM() doit renvoyer true lorsque deux connexions sont utilisées dans le même TX XA (JMSRA LOCAL/REMOTE).
6760450	La mémoire de messages est corrompue si la machine est redémarrée sans arrêter l'instance MQ (GF).

TABLEAU 1-7 Bogues résolus dans Message Queue 4.4 Mise à jour 1 (Suite)

Bogue	Description
6766241	UMS : l'exemple AJAX SendMsg.html utilise /ums comme racine de contexte par défaut. Il doit utiliser /imqums.
6766852	DirectXAResource traduit le statut CONFLICT du courtier par "TxID is already in use".
6799428	Les messages non persistants/non durables déposés dans DMQ ne peuvent pas être consommés mais peuvent être parcourus.
6799428	Les messages non persistants/non durables déposés dans DMQ ne peuvent pas être consommés mais peuvent être parcourus.
6809353	HA openmq 4.3 avec posgresql (8.1) ne fonctionne pas (imqbrokerd ne peut pas démarrer).
6809750	Le pool de connexions (de JMSRA) pour la connexion IDClient ne fonctionne pas.
6812198	Une exception Classcast est générée lors d'un contrôle réalisé à l'aide de métriques de rubriques MQ.
6832000	La connexion JDBC reapExcessConnection MQ s'exécute avec une rotation de processeur élevée.
6833109	L'exemple d'application JMX MQClusterMonitor génère une exception sur AIX avec JDK6.
6835420	La valeur par défaut de NoGCDefault n'est pas calculée correctement. Cela peut causer une opération GC excessive lorsque la mémoire est insuffisante.
6852018	Le message d'erreur "Impossible d'ajouter un consommateur {0} durable. Aucun IDClient n'a été défini lors de la connexion." est trompeur
6856991	Une exception NullPointerException générée après le redémarrage du courtier annule une transaction PREPARED de consommateur durable.
6874125	AVERTISSEMENT : MQJMSRA_DC2001: connectionId=555670328604044289;_destroy(): appelé sur une connexion...
6878945	RFE : JMSBridge : autorise la spécification d'un nom d'utilisateur/mot de passe pour créer une connexion à partir de la fabrique de connexions.
6881493	Les destinations temporaires d'administration ne doivent pas être stockées pour le courtier HA.
6881753	RFE JMSBridge : autorise le balisage de chaque message avec le nom jmsbridge avant de procéder au transfert vers la cible.
6884673	Le courtier MQ 4.4 ne parvient pas à établir de connexion de cluster avec le courtier MQ 3.7/3.6.
6886390	Les messages Persist/Txn publiés et transférés à DMQ peuvent provoquer des erreurs "mq.sys.dmq not found" lors de leur consommation à partir de DMQ.
6886515	Une exception AccessControlException est générée lors de l'utilisation de JMX pour supprimer une destination dans un courtier intégré.

TABLEAU 1-7 Bogues résolus dans Message Queue 4.4 Mise à jour 1 (Suite)

Bogue	Description
6890628	La définition de la propriété de courtier "imq.autocreate.destination.isLocalOnly=true" n'a aucun effet.
6891615	Le sélecteur ne fonctionne pas toujours lors de l'exécution du courtier 4.3 dans glassfish.
6891624	Le nombre de messages 'Remote' peut dépasser celui des 'Count' dans 'imqcmd list dst'.
6891629	Besoin d'un message convivial lorsqu'une exception arithmétique se produit dans le sélecteur.
6891717	Si ifimq.transaction.autorollback=true, l'accusé de réception d'une transaction PREPARED qui doit être automatiquement annulé n'a pas été supprimé, ce qui entraîne l'exception TransactionAckExistEx.
6891802	Le message "[B4061]:Can not use Transaction ID..currently in use" s'affiche au redémarrage du courtier après réception de l'accusé d'une transaction distante de reprise.
6892512	Fuite de mémoire : les destinations temporaires ne sont pas supprimées de la connexion lorsque tempDest.delete() est appelé.
6895040	Si le courtier maître possède une destination temporaire, le courtier esclave ne parvient pas à récupérer uidprefix au démarrage après expiration du délai de verrouillage d'uidprefix.
6896230	Il est possible qu'un nouveau consommateur créé dans le courtier maître pendant qu'il redémarre après la synchronisation avec les esclaves, ne se propage pas partout.
6896764	La méthode equals de TransactionAcknowledgement est incorrecte.
6898355	Le verrouillage de la reprise est redéfinie lors de l'initialisation du gestionnaire de clusters au redémarrage du courtier sans attendre la fin de la reprise.
6901405	RFE : consigne les informations sur le fournisseur JDBC et sur les propriétés du fournisseur si spécifié.
16 (OpenMQ)	Le sélecteur ne fonctionne pas toujours lors de l'exécution du courtier 4.3 dans glassfish.
17 (OpenMQ)	HA openmq 4.3 avec postgresql (8.1) ne fonctionne pas (imqbrokerd ne peut pas démarrer).
22 (OpenMQ)	Le programme d'installation fait référence à un fichier binaire qui n'existe pas et par conséquent échoue.
25 (OpenMQ)	Fuite de mémoire lors de la création de TemporaryTopic.
29 (OpenMQ)	Isolement du courtier
30 (OpenMQ)	Le nombre de messages 'Remote' peut dépasser celui des 'Count' dans 'imqcmd list dst'.
31 (OpenMQ)	Besoin d'un message convivial lorsqu'une exception arithmétique se produit dans le sélecteur.
32 (OpenMQ)	Correction pour dépassements Int-> Long
33 (OpenMQ)	Programme d'installation OpenMQ : une erreur "Invalid SwiXML Descriptor" se produit lorsqu'il est exécuté dans un environnement linguistique japonais.

Bogues résolus dans Message Queue 4.4

Le tableau suivant décrit les bogues résolus dans Message Queue 4.4.

TABEAU 1-8 Bogues résolus dans Message Queue 4.4

Bogue	Description
6242247	Le cluster MQ avec courtier maître démarre et se bloque si les deux courtiers sur une même machine possèdent le même nom
6760937	Le courtier ne se reconnecte pas à la base de données en cas de redémarrage
6763252	Le courtier doit consigner un message plus clair que l'exception NPE lorsque le système accuse réception d'un message qui a été expiré/supprimé.
6765410	Le courtier maître envoie les intérêts locaux 2 fois, ce qui entraîne l'exception esclave Abonnement durable déjà actif
6796506	Le message PREPARED distant n'est pas renvoyé après restauration en cas de délai d'attente lors de la réception de la réponse PREPARED distante
6807708	TemporaryDestination.delete échoue si le courtier maître n'est pas en cours d'exécution.
6812037	RFE : fait passer MQ_CALLBACK_RUNTIME_ERROR à afterMessageDelivery si MQMessageListenerFunc renvoie une erreur.
6812755	Le message de journal de niveau FINE doit être WARNING si les appels de before/afterMessageDelivery renvoient une erreur.
6816023	L'exception Message.setStringProperty() n'affiche pas le nom de propriété dans l'exception de caractère non admis.
6819095	RFE : le cluster doit prendre en charge la définition de la taille du tampon de flux d'entrée/sortie et de TcpNoDelay.
6820585	'Imqcmd list txn' n'affiche pas les transactions de cluster COMMITTED en attente de l'achèvement du courtier distant.
6820588	Une transaction de cluster qui consomme à la fois les messages locaux et distants conserve l'état COMMITTED dans l'état d'attente.
6821639	NPE sur l'annulation/validation de la transaction pendant la récupération AS pour le mode MQRA-DIRECT
6823364	RFE : mise à niveau du compilateur C-API vers Sun Studio 12 sous Solaris.
6829113	Une exception ConcurrentModificationException est générée lorsque le délai d'annulation de Tuxedo TM est plus long que la transaction en raison d'une charge importante.
6832197	L'accusé de réception distant non transactionnel ne doit pas attendre la réponse distante si le client ne demande pas d'accusé de réception pour un accusé de réception.

TABLEAU 1-8 Bogues résolus dans Message Queue 4.4 (Suite)

Bogue	Description
6834735	Un message du journal imprécis "Unexpected Broker Internal Error" s'affiche lorsque le délai d'attente de Tuxedo TM est plus long que la transaction en état START.
6836364	L'abonné de message générique ne reçoit pas les messages distants si son sujet est créé avant l'abonné
6836691	Une exception "HA(JCAPS):msg already been removed" se produit à la réception après l'annulation du récepteur XA, puis envoie un message.
6836749	HA(JCAPS):ack existe dans l'exception du magasin après la réception d'une annulation durable, puis la validation d'un message
6837671	HA(JCAPS):endless redistribue un message validé lors d'une annulation XAResourceImpl.rollback après un envoi réussi.
6839193	RFE : mise à niveau du compilateur C++ vers Visual Studio 2008 SP1.
6845625	Le courtier entre dans un état de mémoire faible lorsque des consommateurs distants sont créés/fermés de façon répétée.
6852207	Une NPE levée par l'envoi d'un message au courtier distant génère le message "unable to process message" à la lecture du paquet de message.
6853822	Un message d'exception imprécis "Cannot perform operation END_TRANSACTION" s'affiche lors de la fin d'une transaction FAILED.
6854142	Les messages "Waiting for cluster connection" et "Closed cluster connection" s'affichent pour le courtier distant toutes les 3 minutes.
6858121	Un AVERTISSEMENT imprécis 'Unknow transaction' s'affiche dans le journal du courtier dans 'imqcmd list txn' si la transaction distante n'existe pas.
6858488	La transaction COMMITTED n'est pas supprimé du courtier de base de transactions si le courtier participant à distance a supprimé sa transaction COMMITTED.
6858905	ConcurrentModificationException dans Consumer.destroyConsumer
6861362	RFE : JMSBridge : prend en charge le mappage automatique de la destination cible vers la destination Message source.getJMSDestination source.
6861528	RFE : JMSBridge : autorise un message de branche MessageTransformer.transformer() vers une autre destination dans la cible.
6861653	Les informations de transactions de cluster excessives envoyées à COMMIT sont incomplètes dans le courtier distant, en cas de charge de transactions importante.
6862413	Message de journal imprécis "mq://xxx.xxx.xx:pppp/ ..." is reachable within 60 seconds".
6863867	Une exception MissingResourceException est générée lors du redémarrage du courtier HA s'il possède l'état COMMITTED en attente d'un courtier distant qui ne fonctionne pas.

TABLEAU 1-8 Bogues résolus dans Message Queue 4.4 (Suite)

Bogue	Description
6867596	Une transaction PREPARED récupérée après le redémarrage du courtier retrouve l'état PREPARED si le courtier redémarre à nouveau.
6868525	Une exception NullPointerException est générée lors du transfert d'une destination temporaire vers le courtier distant lors de l'établissement de lien.
6868578	Certains éléments broadcast/unicast ne possèdent pas l'état vérifié si un lien a été établi avec un courtier distant, ce qui interfère avec le protocole de transfert en cours et allonge le temps nécessaire à l'établissement du lien.
6871612	Les messages HA:log "Cant notify transaction.completion.." s'affichent lors de la consommation des messages distants si le courtier en attente ne fonctionne pas.
6886391	L'exception NullPointerException est générée lorsque le système accuse réception du message si celui-ci a déjà été supprimé.

Bogues résolus dans Message Queue 4.3

Le tableau suivant décrit les bogues résolus dans Message Queue 4.3.

TABLEAU 1-9 Bogues résolus dans Message Queue 4.3

Bogue	Description
6634033	Le protocole de cluster ne propage pas valeur de <code>imqConsumerFlowLimit</code> pour les courtiers distants lorsqu'un client est créé.
6713012	Si un consommateur sur un courtier de cluster est détruit en même temps qu'un courtier distant est redémarré, certains messages risquent de ne pas être transmis.
6727555	Message du journal du courtier « Octets max par msg dépassés » : les valeurs de taille réelle du message et le nombre maximal d'octets par message ont été commutés.
6737404	Les métriques JMX doivent fournir le nombre de messages répartis à partir de destinations (rubriques et files d'attente) mais encore non livrés aux consommateurs.
6740568	Le courtier renvoie une exception lorsqu'il consomme un trop grand nombre de messages dans une transaction unique.
6758524	La commande permettant de créer une liste d'abonnements durables (<code>imqcmd list dur -d "foo.*"</code>) n'accepte pas les caractères génériques dans le nom de la destination.
6758952	En raison du paramètre <code>imq.portmapper.hostname=localhost</code> , les courtiers ne peuvent pas établir la connexion dans un cluster.
6758817	En raison du paramètre <code>imq.cluster.hostname=localhost</code> (non recommandé), les courtiers sur différentes machines ne peuvent pas établir la connexion dans un cluster.

Bogues résolus dans Message Queue 4.2

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

TABEAU 1-10 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.
6683897	Erreur de configuration des rapports sur l'écran de résumé du programme d'installation de Message Queue même si la configuration semble se terminer avec succès : le programme d'installation ne peut pas écrire pour <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.

TABLEAU 1-10 Bogues résolus dans Message Queue 4.2 (Suite)

Bogue	Description
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 JMX ConsumerManagerMonitor.getConsumerInfo revient toujours à SESSION_TRANSACTED 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 1-11 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 imqcmd n'envoient pas d'invite de confirmation.
6473052	Par défaut, les files d'attente créées automatiquement doivent être alternées. (MaxNumberConsumers = -1).
6474990	Le journal du courtier affiche une ConcurrentModificationException pour la commande imqcmd list dst.
6487413	Fuite de mémoire lorsque le comportement aux limites est REMOVE_OLDEST ou REMOVE_LOWER_PRIORITY.
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 messages bloqués de 1 000 messages.
6517341	L'exécution client doit améliorer la logique de reconnexion lorsque le client est connecté à un cluster de courtiers amélioré, en autorisant celui-ci à se reconnecter quelle que soit la valeur de la propriété imqReconnectEnabled.
6528736	Le service de démarrage automatique de Windows (imqbrokersvc) s'arrête brutalement au démarrage.
6561494	Les messages sont transmis au mauvais consommateur lorsque ceux-ci partagent une session.

TABLEAU 1-11 Bogues résolus dans Message Queue 4.1 (Suite)

Bogue	Description
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 1-12 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.
6293053	Le courtier maître ne démarre pas correctement si l'adresse IP du système a été modifiée, à moins que le magasin ne 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.

TABLEAU 1-12 Bogues résolus dans Message Queue 4.0 (Suite)

N° de bogue	Description
6320138	La C-API 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 MESSAGE dans la C-API.
6351293	Ajout d'une prise en charge pour la base de données Apache Derby.

Mises à jour de la documentation relative à Message Queue 4.4 Mise à jour 1

Cette section contient des informations sur les mises à jour de la documentation de Message Queue 4.4 Mise à jour 1 :

- [“Problèmes de compatibilité” à la page 41](#)
- [“Modifications dans le jeu de documentation de Message Queue 4.4 Mise à jour 1” à la page 42](#)

Problèmes de compatibilité

Cette section décrit les problèmes de compatibilité concernant Message Queue 4.4 Mise à jour 1.

Stabilité de l'interface

Sun GlassFish Message Queue utilise de nombreuses interfaces pouvant être modifiées dans le temps. L'Annexe B, “Stability of Message Queue Interfaces” du *Sun GlassFish Message Queue 4.4 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 pourrait 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 GlassFish Message Queue peut changer. Cette modification risque de briser les applications existantes qui dépendent de l'emplacement actuel de certains fichiers de 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 4.4 Mise à jour 1

Le jeu de documentation de Message Queue 4.4 Mise à jour 1 comprend des mises à jour du jeu de documentation de Message Queue 4.4, comme décrit ci-dessous :

Présentation technique

Le document [Sun GlassFish Message Queue 4.4 Technical Overview](#) reflète les nouvelles fonctions de Message Queue 4.4.

Guide d'administration

Le document [Sun GlassFish Message Queue 4.4 Administration Guide](#) inclut des résolutions de bogues mineurs et reflète les nouvelles fonctions de Message Queue 4.4.

Guide du développeur pour Clients Java

Le document [Sun GlassFish Message Queue 4.4 Developer's Guide for Java Clients](#) inclut des résolutions de bogues mineurs.

Guide du développeur pour Clients C

Le document [Sun GlassFish Message Queue 4.4 Developer's Guide for C Clients](#) inclut des résolutions de bogues mineurs.

Guide du développeur pour Clients JMX

Le document [Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients](#) reflète les nouveaux nom de produit et numéro de version de Message Queue.

Problèmes connus et restrictions

Cette section contient une liste des problèmes connus concernant Message Queue 4.4 Mise à jour 1. Les domaines suivants du produit sont abordés :

- “Problèmes d'installation” à la page 43
- “Option de mot de passe désapprouvée” à la page 49
- “Problèmes d'administration/de configuration” à la page 50
- “Problèmes relatifs au courtier” à la page 52
- “Clusters de courtiers” à la page 53
- “Problèmes relatifs à JMX” à la page 56
- “Prise en charge de SOAP” à la page 56

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 4.4 Mise à jour 1.

Enregistrement du produit et Java ES

Message Queue 4.4, tout comme Message Queue 4.2 et 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 utilisé 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, puis que Message Queue & 4.4 est installé via le programme d'installation de

Message Queue, le registre de produit Java ES peut alors se trouver dans un état incohérent. Ainsi, si le programme de désinstallation de Java ES est exécuté, il est possible qu'il supprime par accident Message Queue 4.4 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. Installer le programme d'installation de Message Queue pour installer Message Queue 4.4.

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 4.3. (*Bogue 6650841*)
- Lorsque le programme d'installation est en cours d'exécution pour l'installation de Message Queue 4.3 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.

- L'exécution du programme d'installation en mode d'enregistrement seul (`installer -r`) après avoir effectué une installation en mode silencieux au cours de laquelle l'enregistrement a été ignoré entraîne un échec de l'enregistrement avec une erreur « Fin prématurée de fichier ». (*Bogue 6767988*)
- Lors de l'exécution du programme d'installation de Message Queue sur un ordinateur sur lequel le JDK n'est pas installé, le message d'erreur suivant s'affiche : « Racine incorrecte dans la clé de registre HKLM\SOFTWARE\JavaSoft\Java Runtime Environment\CurrentVersion ». (*Bogue 6764358*)

Solution : installez le JDK avant de lancer le programme d'installation.

- Le répertoire mqInstallHome est créé par le programme d'installation de Message Queue avant que vous ne cliquiez sur le bouton Installer sur l'écran Prêt pour l'installation. (*Bogue 6595590*)

Installation sous Windows

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

- La structure de répertoires installée de Message Queue 4.3 sur la plate-forme Windows est différente de celle des versions précédentes. Reportez-vous à la section “Installed Directory Structure” du *Sun Java System Message Queue 4.3 Installation Guide*.
- 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 à la section “Starting the Administration Console” du *Sun GlassFish Message Queue 4.4 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)
- Le programme d'installation n'ajoute pas le courtier de Message Queue en tant que service Windows.
Solution : ajoutez manuellement le courtier de Message Queue en tant que service Windows à l'aide de la commande `imqsvcadm`.
- Si le JDK n'est pas installé, le programme d'installation renvoie l'erreur suivante : « Racine incorrecte dans la clé de registre `HKLM\SOFTWARE\JavaSoft\Java Runtime Environment\CurrentVersion` ». (Bogue 6764358)\par
Solution : si cette erreur s'affiche, installez un JDK et poursuivez.
- 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 en mode silencieux est réellement exécutée. (Bogue 6586560)
- Le programme d'installation installe Message Queue sur `C:\` même si le système d'exploitation est installé sur un autre lecteur. (Bogue 6673511)
- Pour l'installation et la désinstallation sous Windows, il n'existe aucun fichier `.bat` que l'utilisateur puisse exécuter, et celui-ci ne peut pas non plus effectuer une désinstallation par le biais de Ajouter/Supprimer des programmes dans le panneau de configuration de Windows. (Bogue 6673417)
- Sous Windows Vista, vous ne pouvez pas installer Message Queue sous `C:\Program Files`, sauf si vous procédez à l'installation à partir d'une 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 : exécutez 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*)
- Le programme de désinstallation basé sur un zip de Message Queue se bloque sous Windows 2003. (*Bogue 6764370*)

Solution : supprimez manuellement le répertoire `mqInstallHome`.

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 qui peut être utilisé par la suite pour exécuter l'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*)

Installation sous Linux

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

- Sous Red Hat Linux 5, la bibliothèque `compat-libstdc++` nécessaire à l'exécution des applications clientes C n'est pas incluse dans la distribution de Message Queue et n'est donc n'est pas installée par le programme d'installation de Message Queue. Si vous développez et exécutez des clients C, vous devez installer cette bibliothèque manuellement.

Le rpm `compat-libstdc++` se trouve généralement sur le support d'installation de la version Linux que vous utilisez. Elle peut être installée à l'aide de la commande suivante :

```
rpm -ivh compat-libstdc++-x.x.x.x.rpm
```

où x représente le numéro de version.

Pour vérifier que la bibliothèque a bien été installée, utilisez la commande suivante :

```
rpm -qa | grep compat-libstdc++
```

- Sous Red Hat Linux 5, les clients C peuvent échouer avec une erreur `PR_LOAD_LIBRARY_ERROR` (*bogue 6885978*).

Sous Red Hat Linux 5, les clients C peuvent échouer en affichant le message suivant :

```
"Preparing for NSS initialization ..."
"Initializing NSS ..."
"Could not connect to broker because 'PR_LOAD_LIBRARY_ERROR' (-5977)."
```

`producer(): Error: PR_LOAD_LIBRARY_ERROR`

Cette erreur survient parce que les bibliothèques NSS/NSPR ne sont pas accessibles.

Pour résoudre ce problème, définissez la variable d'environnement `LD_LIBRARY_PATH` de façon à inclure le chemin des bibliothèques NSS/NSPR, `imq_home/nss/lib`.

- 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 » sur 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` soit `amd64`.

- Lorsque le programme d'installation est exécuté en mode de simulation (`install -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*)

Anomalies de version dans le programme d'installation

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

Plate-forme Solaris

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

TABLEAU 1-13 Traduction de la chaîne de version

Version telle qu'affichée par le programme d'installation sur le SE Solaris	Version de Message Queue correspondante
4.4.1.0	4.4 Update 1
4.4.0.0	4.4
4.3.0.0	4.3
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.

Plate-forme Linux

Sur la plate-forme Linux, le numéro de version 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 (`install -t`), dans une langue non anglaise, les caractères multi-octets 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*)
- 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 aux 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 (`install -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 de mots de passe spécifiant les mots de passe pertinents et référencer le fichier de 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.

- Configurez 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
```

- Utilisez 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 plates-formes Windows, vous devez ajouter manuellement le courtier de Message Queue en tant que service Windows à l'aide de la commande `imqsvcadm`. Le programme d'installation n'effectue pas cela pour vous.
- Sur les plates-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 dans 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 un 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 plates-formes Windows, les commandes `imqadmin` et `imqobjmgr` 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\imqadmin
```

- 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 Java Runtime 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*)
- Sur les plates-formes Linux exécutant SELinux, la commande `pkg` du centre de mises à jour échoue (*bogue 6892062*).

Solution : il s'agit d'un problème connu lié au centre de mises à jour (https://updatecenter2.dev.java.net/issues/show_bug.cgi?id=1211). La commande suivante permet d'activer pkg pour permettre son fonctionnement sous SELinux avec activation de la mise en application :

```
# chcon -f -t textrel_shlib_t $IMAGE/pkg/vendor-packages/OpenSSL/crypto.so
```

Problèmes relatifs au courtier

Les problèmes suivants concernent le courtier de Message Queue.

- Les clients Message Queue 4.4 reçoivent un avertissement incompréhensible lors de la connexion aux courtiers Message Queue 3.7 (*bogue 6899886*).

Lorsqu'un client Message Queue 4.4 se connecte à un courtier Message Queue 3.7, le client reçoit un message d'avertissement qui se présente comme suit :

```
WARNING [I500]: Caught JVM exception: ...
[C4036]: A broker error occurred. :[505] bad version ...
```

Cet avertissement de "version incorrecte" indique que le client doit se reconnecter au courtier à un niveau de protocole inférieur.

- Lors de l'utilisation d'un magasin de données JDBC, le mot de passe de la base de données est stocké dans du texte en clair (*bogue 6691717*).

Solution : sécurisez le fichier de mot de passe contenant le mot de passe de la base de données comme indiqué dans la section "Password Files" du *Sun GlassFish Message Queue 4.4 Administration Guide*.

- 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 Mo envoie une exception SQL « Packet de requête trop grand ». (*Bogue 6682815*)

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

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

- 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 au fichier `config.properties` du courtier (voir la section “[Enhanced Clusters: JDBC Configuration Properties](#)” du *Sun GlassFish Message Queue 4.4 Administration Guide*).

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

- Lors de l'utilisation du pilote 9i (JDBC 9.2.0.x) d'Oracle, le courtier renvoie une exception « Échec de persistance de la propriété... ». (*Bogue 6626825*)

Solution : utilisez le pilote 10g (JDBC 10.2.0.x) d'Oracle, pour lequel le courtier est optimisé.

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

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

Solution : ajoutez la valeur de propriété suivante au fichier `config.properties` du courtier :

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

- Lorsque vous utilisez JVM IBM sur AIX, le courtier s'exécute parfois dans des conditions de mémoire faible ou RED sans raison apparente (*bogue 6899526*).

Solution : utilisez la dernière version de JVM IBM (Java Runtime 1.6.0 IBM Corporation ou version supérieure) et donnez la valeur `imqbrokerd` à l'option GC de JVM IBM suivante :

```
# imqbrokerd -vmargs -Xgcpolicy:gencon
```

Clusters de courtiers

Les problèmes suivants affectent les clusters de courtiers.

- Le courtier à haute disponibilité avec le magasin de données de clusters MySQL ne parvient pas à redémarrer s'il se termine de façon anormale (*bogue 6896877*).

Solution : il s'agit d'un problème connu lié au cluster MySQL (<http://bugs.mysql.com/bug.php?id=47955>). Une correction destinée à résoudre ce problème a été appliquée aux versions MySQL 5.1.39-ndb-6.3.28, 5.1.39-ndb-7.0.9 et 5.1.39-ndb-7.1.0.

- 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 amélioré, l'exécution client tentera de se reconnecter jusqu'à ce qu'elle y parvienne (elle ignore la valeur de l'attribut de connexion par défaut `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 4.3, tous les courtiers doivent avoir la version 3.5 ou ultérieure.
- Les courtiers de Message Queue 4.1, 4.2 et 4.3 ne peuvent pas interopérer dans un cluster par défaut avec les courtiers de Message Queue 3.6 ou 3.7 car la valeur par défaut de `imq.autocreate.queue.maxNumActiveConsumers` a été modifiée entre ces versions. (Bogue 6716400)

Solution : vérifiez que tous les courtiers ont la même valeur ou modifiez la valeur de `imq.autocreate.file.maxNumActiveConsumers`, généralement appliquée par la modification de la configuration de Message Queue 4.1, 4.2 et 4.3 afin de correspondre à celle utilisée par les courtiers de la version 3.6 ou 3.7 (par défaut, de la valeur de -1 à 1, valeur par défaut de la version précédente).

- Pour ajouter un courtier de Message Queue 4.3 (ou 4.x) à un cluster de courtiers de Message Queue 3.x, un courtier maître doit être en cours d'exécution. (Bogue 6763796)
- Lors de la conversion d'un cluster conventionnel en cluster amélioré, vous pouvez utiliser l'utilitaire de gestionnaire de bases de données de Message Queue (`imqdbmgr`) pour convertir un magasin de données basé sur JDBC autonome existant en magasin de données basé sur JDBC partagé comme indiqué à la section "[Cluster Conversion: JDBC-Based Data Store](#)" du *Sun GlassFish Message Queue 4.4 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 hasstore`, peut échouer. Le message « nombre de verrous trop élevé » s'affiche 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 *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 tableau épuisé ». (*Bogue 6550483*)

Pour de plus amples informations, reportez-vous à la section « HADB Problems » du *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 été initiée. (*Bogue 4951010*)

Solution : le consommateur recevra les messages dès qu'il se connecte. Les messages seront renvoyés à un autre consommateur si la connexion du consommateur reste 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]:
Message acknowledgement failed from
mq://129.145.130.95:7677/?instName=a&brokerSessionUID=3209681167602264320:
  ackStatus = NOT_FOUND(404)\
  Reason = Update remote transaction state to COMMITTED(6):
transaction 3534784765719091968 not found, the transaction
may have already been committed.
  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 Notify commit 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:
Update remote transaction state to COMMITTED(6):
transaction 3534784765719091968 not found, the transaction may have already
been committed.:
com.sun.messaging.jmq.jmsserver.util.BrokerException: Update remote transaction
state to COMMITTED(6): transaction 3534784765719091968 not found, the transaction
may have already been committed.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 renvoie 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 GlassFish Message Queue 4.4 Mise à jour 1 contient l'ensemble de fichiers ci-dessous pouvant être utilisés et distribués librement sous forme binaire :

<code>fscontext.jar</code>	<code>jaxm-api.jar</code>
<code>imq.jar</code>	<code>jms.jar</code>
<code>imqjmx.jar</code>	<code>libmqcrt.sl (HP-UX)</code>
<code>imqxm.jar</code>	<code>libmqcrt.so (UNIX)</code>
<code>imqums.war</code>	<code>mqcrt1.dll (Windows)</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 à l'adresse <http://sun.com/software/javaenterprisesystem/get.html>.

Pour obtenir des informations sur l'engagement de Sun en matière d'accessibilité, consultez la page Web <http://sun.com/access>.

Comment signaler des problèmes et apporter des commentaires

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

- Services de support logiciel Sun en ligne disponibles à la page <http://www.sun.com/service/sunone/software>.

Ce site est lié à la base de connaissances 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.
- Les étapes détaillées des méthodes utilisées pour reproduire le problème.
- Les journaux des erreurs ou core dumps éventuels.

Forum de Sun GlassFish

Accédez au forum de Sun GlassFish 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, accédez au site <http://docs.sun.com>, puis cliquez sur Envoyer des 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*, et son numéro de référence est 821-1513-10.

Ressources Sun supplémentaires

Vous pouvez obtenir des informations utiles concernant Sun GlassFish 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>

