

Versionshinweise zu Sun GlassFish Message Queue 4.4 Update 1



Teilenr.: 821-1511-10
Dezember 2009

Copyright ©2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Alle Rechte vorbehalten.

Sun Microsystems, Inc. hat Rechte in Bezug auf geistiges Eigentum an der Technologie, die in dem in diesem Dokument beschriebenen Produkt enthalten ist. Im Besonderen und ohne Einschränkung umfassen diese Ansprüche in Bezug auf geistiges Eigentum eines oder mehrere Patente und eines oder mehrere Patente oder Anwendungen mit laufendem Patent in den USA und in anderen Ländern.

Rechte der US-Regierung – Kommerzielle Software. Für bei der Regierung beschäftigte Benutzer gelten die Standardlizenzvereinbarung von Sun Microsystems, Inc. sowie die einschlägigen Bestimmungen des FAR und seiner Ergänzungen.

Dieses Lieferung schließt möglicherweise Materialien ein, die von Fremdanbietern entwickelt wurden.

Teile dieses Produkts können von Berkeley BSD Systems abgeleitet und durch die University of California lizenziert sein. UNIX ist eine eingetragene Marke in den Vereinigten Staaten und anderen Ländern und wird ausschließlich durch die X/Open Company Ltd. lizenziert.

Sun, Sun Microsystems, das Sun-Logo, das Solaris-Logo, das Java-Kaffeetassen-Logo, docs.sun.com, Java und Solaris sind Marken oder eingetragene Marken von Sun Microsystems, Inc. oder dessen Tochterunternehmen in den USA und anderen Ländern. Sämtliche SPARC-Marken werden unter Lizenz verwendet und sind Marken oder eingetragene Marken von SPARC International Inc. in den Vereinigten Staaten und anderen Ländern. Produkte mit der SPARC-Marke basieren auf einer von Sun Microsystems Inc. entwickelten Architektur.

Die grafischen Benutzeroberflächen von OPEN LOOK und Sun™ wurden von Sun Microsystems Inc. für seine Benutzer und Lizenznehmer entwickelt. Sun erkennt die Pionierleistung von Xerox bei der Ausarbeitung und Entwicklung des Konzepts von visuellen oder grafischen Benutzeroberflächen für die Computerindustrie an. Sun ist Inhaber einer einfachen Lizenz von Xerox für die Xerox Graphical User Interface (grafische Benutzeroberfläche von Xerox). Mit dieser Lizenz werden auch die Sun-Lizenznehmer abgedeckt, die grafische OPEN LOOK-Benutzeroberflächen implementieren und sich ansonsten an die schriftlichen Sun-Lizenzvereinbarungen halten.

Produkte, die in dieser Veröffentlichung beschrieben sind, und die in diesem Handbuch enthaltenen Informationen unterliegen den Gesetzen der US-Exportkontrolle und können den Export- oder Importgesetzen anderer Länder unterliegen. Die Verwendung im Zusammenhang mit Nuklear-, Raketen-, chemischen und biologischen Waffen, im nuklear-maritimen Bereich oder durch in diesem Bereich tätige Endbenutzer, direkt oder indirekt, ist strengstens untersagt. Der Export oder Rückexport in Länder, die einem US-Embargo unterliegen, oder an Personen und Körperschaften, die auf der US-Exportausschlussliste stehen, einschließlich (jedoch nicht beschränkt auf) der Liste nicht zulässiger Personen und speziell ausgewiesener Staatsangehöriger, ist strengstens untersagt.

Die DOKUMENTATION WIRD "AS IS" BEREITGESTELLT, UND JEGICHE AUSDRÜCKLICHE ODER IMPLIZITE BEDINGUNGEN, DARSTELLUNGEN UND HAFTUNG, EINSCHLIESSLICH JEGLICHER STILLSCHWEIGENDER HAFTUNG FÜR MARKTFÄHIGKEIT, EIGNUNG FÜR EINEN BESTIMMTEN ZWECK ODER NICHTÜBERTRETUNG WERDEN IM GESETZLICH ZULÄSSIGEN RAHMEN AUSDRÜCKLICH AUSGESCHLOSSEN.

Copyright ©2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivés du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement 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., ou ses filiales, aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

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

Inhalt

1	Versionshinweise	5
	Änderungsprotokoll der Versionshinweise	6
	Info zu Message Queue 4.4 Update 1	7
	Message Queue 4.4 Update 1 - Unterstützte Plattformen und Komponenten	7
	Unterstützte Betriebssystemplattformen	7
	Unterstützung der Systemvirtualisierung	8
	Komponentenabhängigkeiten	8
	Neue Funktionen in Message Queue 4.4 Update 1 und aktuellen Versionen	10
	Neue Funktionen in Message Queue 4.4 Update 1	10
	Neue Funktionen in Message Queue 4.4	11
	Neue Funktionen in Message Queue 4.3	13
	Neue Funktionen in Message Queue 4.2	19
	Neue Funktionen in Message Queue 4.1	23
	Neue Funktionen in Message Queue 4.0	27
	In zukünftigen Versionen veraltete Funktionen	31
	In Message Queue 4.4 Update 1 und aktuellen Versionen behobene Fehler	32
	In Message Queue 4.4 Update 1 behobene Fehler	32
	Behobene Probleme in Message Queue 4.4	34
	In Message Queue 4.3 behobene Fehler	37
	In Message Queue 4.2 behobene Fehler	37
	In Message Queue 4.1 behobene Fehler	39
	Behobene Probleme in Message Queue 4.0	40
	Dokumentationsaktualisierungen in Message Queue 4.4 Update 1	41
	Kompatibilitätsprobleme	41
	Änderungen am Message Queue 4.4 Update 1-Dokumentationssatz	42
	Bekannte Probleme und Beschränkungen	43
	Installationsprobleme	43
	Veraltete Passwortoption	50

Administration/Konfiguration	51
Broker-Probleme	52
Broker-Cluster	54
JMX-Probleme	56
SOAP-Unterstützung	57
Dateien für Neuverteilung	57
Eingabehilfen für Personen mit Behinderungen	57
Problemmeldungen und Feedback	58
Sun GlassFish-Software-Forum	58
Java Technology Forum	58
Ihre Meinung ist gefragt	59
Weitere Quellen von Sun	59

Versionshinweise

Version 4.4 Update 1

Teilenr. 821-1511-10

Diese Versionshinweise enthalten wichtige Informationen, die zum Zeitpunkt der Veröffentlichung von Sun GlassFish Message Queue 4.4 Update 1 verfügbar waren. Hier finden Sie u. a. Informationen zu neuen Funktionen, Verbesserungen und bekannten Problemen. Lesen Sie dieses Dokument, bevor Sie Message Queue 4.4 Update 1 verwenden.

Diese Versionshinweise enthalten auch Informationen über die Versionen 4.4, 4.3, 4.2, 4.1 und 4.0 von Message Queue. In „[Neue Funktionen in Message Queue 4.2](#)“ auf Seite 19, „[Neue Funktionen in Message Queue 4.1](#)“ auf Seite 23 und „[Neue Funktionen in Message Queue 4.0](#)“ auf Seite 27 finden Sie Informationen zu den neuen Funktionen in diesen Versionen.

Die neueste Ausgabe dieser Versionshinweise wird auf der Sun GlassFish Message Queue-Dokumentations-Website unter <http://docs.sun.com/col1/1307.7> bereitgestellt. Besuchen Sie diese Website vor der Installation und Konfiguration Ihrer Software und später regelmäßig, um stets die neuesten Versionshinweise und Produktdokumentationen verfügbar zu haben.

In diesen Versionshinweisen werden die folgenden Themen behandelt:

- „Änderungsprotokoll der Versionshinweise“ auf Seite 6
- „Info zu Message Queue 4.4 Update 1“ auf Seite 7
- „Message Queue 4.4 Update 1 - Unterstützte Plattformen und Komponenten“ auf Seite 7
- „Neue Funktionen in Message Queue 4.4 Update 1 und aktuellen Versionen“ auf Seite 10
- „In zukünftigen Versionen veraltete Funktionen“ auf Seite 31
- „In Message Queue 4.4 Update 1 und aktuellen Versionen behobene Fehler“ auf Seite 32
- „Dokumentationsaktualisierungen in Message Queue 4.4 Update 1“ auf Seite 41
- „Bekannte Probleme und Beschränkungen“ auf Seite 43
- „Dateien für Neuverteilung“ auf Seite 57
- „Eingabehilfen für Personen mit Behinderungen“ auf Seite 57
- „Problemmeldungen und Feedback“ auf Seite 58

- „Ihre Meinung ist gefragt“ auf Seite 59
- „Weitere Quellen von Sun“ auf Seite 59

Diese Dokumentation nimmt Bezug auf URLs zu Produkten von Drittanbietern und bietet weitere relevante Informationen.

Sun ist nicht für die Verfügbarkeit der in diesem Dokument erwähnten Websites anderer Hersteller verantwortlich. Sun haftet nicht für den Inhalt oder Werbung auf diesen Websites oder für die auf diesen Websites angebotenen Produkte und Materialien. Sun übernimmt keine Verantwortung oder Haftung für tatsächliche oder angebliche Schäden oder Verluste, die im Zusammenhang mit den auf diesen Websites angebotenen Informationen, Waren oder Dienstleistungen entstanden sind.

Änderungsprotokoll der Versionshinweise

In der folgenden Tabelle sind die Datumsangaben zu den 4.x-Versionen von Message Queue sowie die in diesem Dokument enthaltenen Änderungen bei den einzelnen Versionen aufgeführt.

TABELLE 1-1 Änderungsprotokoll

Datum	Beschreibung der Änderungen
Dezember 2009	Veröffentlichung dieses Dokuments für Message Queue 4.4 Update 1. Fügt dieser Version neue Funktionen hinzu und beseitigt Probleme, die im vorhergehenden Installationsprogramm auftraten.
Dezember 2009	Zweite Ausgabe dieses Dokuments für Message Queue 4.4. Behebt Fehler und ergänzt Auslassungen.
Oktober 2009	Veröffentlichung dieses Dokuments für Message Queue 4.4. Ergänzt durch neue Funktionen für diese Version.
Mai 2009	Erste Ausgabe dieses Dokuments für Message Queue 4.4 Beta. Ergänzt durch neue Funktionen für diese Version.
Dezember 2008	Veröffentlichung dieses Dokuments für Message Queue 4.3. Ergänzt durch neue Funktionen für diese Version.
August 2008	Veröffentlichung dieses Dokuments für Message Queue 4.2. Ergänzt durch neue Funktionen für diese Version.
September 2007	Dritte Ausgabe dieses Dokuments für Message Queue 4.1. Ergänzt durch eine Beschreibung der Unterstützung für Java Enterprise System Monitoring Framework, korrigierte C-Anschlüsse, Fehlerkorrekturen und andere Funktionen.
April 2007	Zweite Ausgabe dieses Dokuments für Message Queue 4.1 Beta. Fügt eine Hochverfügbarkeitsfunktion hinzu.

TABELLE 1-1 Änderungsprotokoll (Fortsetzung)

Datum	Beschreibung der Änderungen
Januar 2007	Erste Ausgabe dieses Dokuments für Message Queue 4.1 Beta. Fügt eine Beschreibung der JAAS-Unterstützung hinzu.
Mai 2006	Erste Ausgabe dieses Dokuments für Message Queue 4.0.

Info zu Message Queue 4.4 Update 1

Sun GlassFish Message Queue ist ein leistungsfähiger Nachrichtendienst, der ein zuverlässiges, asynchrones Messaging gemäß JMS 1.1 (Java Messaging Specification) bietet. Zusätzlich stellt Message Queue Funktionen bereit, die über JMS hinausgehen, um den Anforderungen großer Bereitstellungen im Unternehmensbereich gerecht zu werden.

Message Queue 4.4 Update 1 ist eine Unterversion, die ein neues Installationsprogramm basierend auf dem plattformunabhängigen Image Packaging System (IPS) bietet, auch als pkg (5) -System bekannt. Zusätzlich enthält Version 4.4 Update 1 zahlreiche Funktionserweiterungen und Fehlerkorrekturen.

Message Queue 4.4 Update 1 - Unterstützte Plattformen und Komponenten

In diesem Abschnitt werden folgende Themen in Bezug auf die für Message Queue 4.4 Update 1 geltenden Systemvoraussetzungen behandelt:

- „Unterstützte Betriebssystemplattformen“ auf Seite 7
- „Unterstützung der Systemvirtualisierung“ auf Seite 8
- „Komponentenabhängigkeiten“ auf Seite 8

Unterstützte Betriebssystemplattformen

Message Queue 4.4 Update 1 wird auf den Betriebssystemplattformen Solaris, Linux, Windows und AIX unterstützt. In [Tabelle 1-2](#) finden Sie die unterstützten Versionen dieser Plattformen. Die für die einzelnen Plattformen geltenden Hardware-Anforderungen finden Sie im *Sun Java System Message Queue 4.3 Installation Guide*

TABELLE 1-2 Unterstützte Plattformversionen

Plattform	Unterstützte Versionen
Solaris	Solaris 9 (SunOS 5.9), alle Aktualisierungen (SPARC, x86)
	Solaris 10 (SunOS 5.10), alle Aktualisierungen (SPARC, x86, x64)

TABELLE 1-2 Unterstützte Plattformversionen (Fortsetzung)

Plattform	Unterstützte Versionen
OpenSolaris	OpenSolaris 2008.11 (x86, x64) OpenSolaris 2009.06 (x86, x64)
Linux	Red Hat Enterprise Linux Advanced Server 4.0, 5.0, alle Aktualisierungen, 32- und 64-Bit-Versionen (x86, x64) Red Hat Enterprise Linux Enterprise Server 4.0, 5.0, alle Aktualisierungen, 32- und 64-Bit-Versionen (x86, x64)
AIX	AIX 6.1
Windows	Windows Vista Windows XP Professional, SP2 (x86) ¹ Windows Server 2003, Standard und Enterprise Edition, SP2, 32- und 64-Bit-Versionen (x86, x64) ² Windows Server 2008 Standard und Enterprise Editions, SP2, 32- und 64-Bit-Versionen (x86, x64) ²

¹ Keine Unterstützung für Home, Tablet PC oder Media Center Edition

² Keine Unterstützung für Web oder Small Business Server Edition

Unterstützung der Systemvirtualisierung

Bei der Systemvirtualisierung handelt es sich um eine Technologie, mit der mehrere Instanzen eines Betriebssystems auf einer gemeinsam genutzten Hardware unabhängig voneinander ausgeführt werden können. Auf der Funktionsebene erkennt die auf einem Betriebssystem in einer virtualisierten Umgebung bereitgestellte Software im Allgemeinen nicht, dass die zugrunde liegende Plattform virtualisiert wurde. Sun testet seine Sun Java System-Produkte auf ausgewählten Systemvirtualisierungs- und Betriebssystemkombinationen, um sicherzustellen, dass diese Produkte in virtualisierten Umgebungen mit zulässiger Größe und Konfiguration weiterhin so arbeiten wie auf nicht virtualisierten Systemen. Weitere Informationen über die Unterstützung von Sun für Sun Java System-Produkte in virtualisierten Umgebungen finden Sie unter <http://docs.sun.com/doc/820-4651>.

Komponentenabhängigkeiten

Neben den plattformspezifischen Anforderungen hängt Message Queue & auch von bestimmten Grundkomponenten ab, die installiert sein müssen, damit Message Queue-Clients entwickelt und ausgeführt werden können. In [Tabelle 1-3](#) werden diese Komponenten beschrieben. Andere Versionen oder Herstellerimplementierungen können auch verwendet werden. Diese sind jedoch von Sun Microsystems nicht getestet und werden daher nicht offiziell unterstützt.

TABELLE 1-3 Erforderliche Unterstützungskomponenten

Komponente	Unterstützt	Unterstützte Versionen ¹
Java Runtime Environment (JRE)	Message Queue-Broker- und Verwaltungswerkzeuge	J2SE™ Runtime Environment 1.5.0 oder höher Java™ SE Runtime Environment 1.6.0
Java Software Development Kit (JDK), Standard Edition	Java-Client-Entwicklung und -Einsatz	J2SE™ Development Kit 1.5.0 oder höher Java SE Development Kit 1.6.0. Beachten Sie, dass für JMX Version 1.6.0_10 oder höher erforderlich ist.

¹ nur Produktversionen von Sun Microsystems

In [Tabelle 1-4](#) finden Sie zusätzliche Komponenten, die Sie installieren können, um weitere Unterstützung für Message Queue-Clients bereitzustellen. Möglicherweise benötigen Sie nicht alle aufgeführten Komponenten. Wenn Sie beispielsweise keinen C-Client schreiben, benötigen Sie den C-Compiler, die C++-Laufzeitbibliothek, NSPR und NSS nicht.

TABELLE 1-4 Optionale Unterstützungskomponenten

Komponente	Unterstützt	Unterstützte Versionen
Anwendungsserver	HTTP/HTTPS	Sun GlassFish Application Server Enterprise Edition, Version 9.1.1 Sun GlassFish Enterprise Server v2.1, v2.1.1, v3
Web Server	HTTP/HTTPS	Sun GlassFish Web Server Enterprise Edition, Version 7.0, Update 3
Datenbank	JDBC-basierter Datenspeicher	HADB, Versionen 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 und 11g postgresql, Version 8.1 Hinweis – Die PointBase-Datenbank wird nicht mehr unterstützt.
Hochverfügbare Datenbank	Hochverfügbarkeits-Broker-Cluster	HADB, Versionen 4.4.3, 4.5, 4.6 MySQL Cluster Edition, Version 5.1.39-ndb-7.0.9 Oracle 10g und 11g

TABELLE 1-4 Optionale Unterstützungskomponenten *(Fortsetzung)*

Komponente	Unterstützt	Unterstützte Versionen
Lightweight Directory Access Protocol-(LDAP)-Verzeichnisserver	Message Queue – Benutzer-Repository und verwaltete Objekte	Sun GlassFish Directory Server, Version 6.0
Java Naming and Directory Interface (JNDI)	Unterstützung für verwaltete Objekte und LDAP-Benutzer-Repository	JNDI Version 1.2.1 LDAP Service Provider, Version 1.2.2 File System Service Provider, Version 1.2 Beta 3 ¹
C-Compiler und kompatible C++-Runtime-Bibliothek	Message Queue-C-Clients	Solaris: Sun Studio, Version 12 oder höher, C++-Compiler mit Standardmodus und C-Compiler Linux: gcc/g++, Version 3.4.6 Windows: Microsoft Windows Visual Studio, Version 2008 SP1
Netscape Portable Runtime (NSPR)	Message Queue-C-Clients	Version 4.7
Network Security Services (NSS)	Message Queue-C-Clients	Version 3.12.3

¹ Nur Unterstützung für verwaltete Objekte; unterstützt für Entwicklung und Tests, nicht jedoch für die Bereitstellung in einer Produktionsumgebung

Neue Funktionen in Message Queue 4.4 Update 1 und aktuellen Versionen

Die neuen Funktionen in Message Queue 4.4 Update 1 und vorhergehende Versionen in der Message Queue 4.x-Familie sind in folgenden Abschnitten beschrieben:

- „Neue Funktionen in Message Queue 4.4 Update 1“ auf Seite 10
- „Neue Funktionen in Message Queue 4.4“ auf Seite 11
- „Neue Funktionen in Message Queue 4.3“ auf Seite 13
- „Neue Funktionen in Message Queue 4.2“ auf Seite 19
- „Neue Funktionen in Message Queue 4.1“ auf Seite 23
- „Neue Funktionen in Message Queue 4.0“ auf Seite 27

Neue Funktionen in Message Queue 4.4 Update 1

Message Queue 4.4 Update 1 ist eine Unterversion, die eine Anzahl von Funktionserweiterungen und Fehlerkorrekturen enthält. Dieser Abschnitt enthält eine Beschreibung neuer Funktionen in dieser Version:

- „Neues Installationsprogramm“ auf Seite 11
- „Transaktionsprotokollunterstützung für Cluster“ auf Seite 11
- „Prozessinterner Broker“ auf Seite 11

Neues Installationsprogramm

Message Queue 4.4 Update 1 bietet ein neues Multiplattform-Installationsprogramm basierend auf dem pkq (5) -System, auch als IPS oder Image Packaging System bekannt. Informationen zu diesem Installationsprogramm finden Sie unter [Sun GlassFish Message Queue 4.4 Update 1 Installation Guide](#).

Transaktionsprotokollunterstützung für Cluster

Message Queue 4.4 Update 1 fügt einen Transaktionsfortdauermechanismus für dateibasierte Datenspeicher hinzu, der Broker-Cluster unterstützt. Dieser Mechanismus bietet auch andere Funktionen, die unter „Optimizing File-Based Transaction Persistence“ in [Sun GlassFish Message Queue 4.4 Administration Guide](#) beschrieben sind.

Prozessinterner Broker

Message Queue 4.4 Update 1 unterstützt das Ausführen eines Broker von einem Java-Client aus. Ein solcher Broker wird als *prozessinterner* oder *eingebetteter* Broker bezeichnet und auf derselben JVM ausgeführt wie der Java-Client, der ihn erstellt und startet. Weitere Informationen finden Sie unter [Kapitel 6, „Embedding a Message Queue Broker in a Java Client“](#) in [Sun GlassFish Message Queue 4.4 Developer's Guide for Java Clients](#).

Neue Funktionen in Message Queue 4.4

Message Queue 4.4 ist eine Unterversion, die zahlreiche Funktionserweiterungen und Fehlerkorrekturen enthält. Dieser Abschnitt enthält eine Beschreibung neuer Funktionen in dieser Version:

- „JMS-Brückendienst“ auf Seite 11
- „STOMP-Brückendienst“ auf Seite 12
- „Zusätzliche Erweiterungen“ auf Seite 13

JMS-Brückendienst

Da JMS kein Wire-Protokoll zur Kommunikation zwischen Brokern und Clients festlegt, verwendet jeder JMS-Provider (einschließlich Message Queue) ein eigenes, proprietäres Protokoll. Dies verhindert die Interoperabilität zwischen JMS-Providern.

Der JMS-Brückendienst in Message Queue 4.4 schließt diese Lücke, indem er einem Message Queue-Broker ermöglicht, seine Ziele den Zielen externer JMS-Provider zuzuordnen. Dadurch kann der Message Queue-Broker mit Clients des externen JMS-Providers kommunizieren.

Der JMS-Brückendienst unterstützt die Zuordnung von Zielen in externen JMS-Providern, die folgende Voraussetzungen erfüllen:

- Sie sind mit JMS 1.1 kompatibel.
- Sie unterstützen JNDI-Administrationsobjekte.
- Sie verwenden Verbindungen des Typs `javax.jms.ConnectionFactory` oder `javax.jms.XAConnectionFactory`.
- Für transaktionale Zuordnungen unterstützen sie XA-Schnittstellen als Ressourcenmanager.

Viele Open Source- und kommerzielle JMS-Provider erfüllen diese Voraussetzungen. Dadurch wird der JMS-Brückendienst zu einer wirkungsvollen Möglichkeit zur Integration von Message Queue in eine vorhandene Messaging-Umgebung, in der andere JMS-Provider verwendet werden.

Weitere Informationen zum JMS-Brückendienst finden Sie unter:

- Informationen zu Architektur, Unterkomponenten und Funktionen des JMS-Brückendienstes finden Sie unter „[JMS Bridge Service](#)“ in *Sun GlassFish Message Queue 4.4 Technical Overview*.
- Informationen über die Konfiguration und Verwaltung von JMS-Brücken in einem Broker finden Sie unter „[Configuring and Managing JMS Bridge Services](#)“ in *Sun GlassFish Message Queue 4.4 Administration Guide*.

STOMP-Brückendienst

Wie bereits erwähnt legt JMS kein Wire-Protokoll zur Kommunikation zwischen Brokern und Clients fest. Das Open Source-Projekt STOMP (Streaming Text Oriented Messaging Protocol) unter <http://stomp.codehaus.org> definiert ein einfaches Wire-Protokoll, das in beliebiger Sprache geschriebene Clients zur Kommunikation mit beliebigen Messaging-Providern, die das STOMP-Protokoll unterstützen, nutzen können.

Message Queue 4.4 bietet Unterstützung für das STOMP-Protokoll mithilfe des STOMP-Brückendienstes. Der Dienst ermöglicht einem Message Queue-Broker die Kommunikation mit STOMP-Clients.

Weitere Informationen über den STOMP-Brückendienst finden Sie in folgenden Dokumenten:

- Informationen zur Architektur und den Funktionen des STOMP-Brückendienstes finden Sie unter „[STOMP Bridge Service](#)“ in *Sun GlassFish Message Queue 4.4 Technical Overview*.
- Informationen zur Konfiguration und Verwaltung einer STOMP-Brücke in einem Broker finden Sie in „[Configuring and Managing STOMP Bridge Services](#)“ in *Sun GlassFish Message Queue 4.4 Administration Guide*.

Zusätzliche Erweiterungen

Message Queue 4.4 bietet außerdem die folgenden zusätzlichen Erweiterungen:

- „Neue Universal Message Service-Funktionen (UMS)“ auf Seite 13
- „Unterstützung für IPS-Pakete“ auf Seite 13
- „Überwachungsprotokollierungsfunktion wiederhergestellt“ auf Seite 13

Neue Universal Message Service-Funktionen (UMS)

Der UMS bietet jetzt Funktionen, die auf HTTP GET basieren:

- **getBrokerInfo**: ruft Informationen über den Broker ab.
- **getConfig**: ruft Informationen über die UMS-Konfiguration ab.
- **debug**: aktiviert und deaktiviert die Debug-Protokollierung.
- **ping**: kommuniziert mit dem Broker, um zu bestätigen, dass dieser ausgeführt wird.

Informationen zu diesen neuen Funktionen finden Sie unter „Query and utility functions using HTTP GET“ unter <https://mq.dev.java.net/4.4-content/imqums/protocol.html>.

Eine Übersicht über UMS erhalten Sie in „Universal Message Service (UMS)“ auf Seite 14. Die Dokumentation zur UMS-API finden Sie unter <https://mq.dev.java.net/4.4-content/imqums/protocol.html>. Programmierbeispiele in mehreren Sprachen finden Sie unter <https://mq.dev.java.net/4.4-content/imqums/examples/README.html>.

Unterstützung für IPS-Pakete

Message Queue wird nun mit dem quelloffenen Image Packaging System (IPS), das auch als pkg (5)-System bezeichnet wird, verpackt und zur Verteilung bereitgestellt. Diese Verpackungsmethode ermöglicht die Integration von Message Queue in Sun GlassFish Enterprise Server 2.1.1.

Überwachungsprotokollierungsfunktion wiederhergestellt

Message Queue 3.7 bot eine Überwachungsprotokollierungsfunktion, die in Message Queue 4.0 entfernt wurde. Diese Funktion wurde in Message Queue wiederhergestellt. 4.4. Informationen zu dieser Funktion finden Sie unter „Audit Logging“ in *Sun GlassFish Message Queue 4.4 Administration Guide*.

Neue Funktionen in Message Queue 4.3

Message Queue 4.3 war eine Unterversion, die eine Reihe von Funktionserweiterungen und Fehlerkorrekturen enthielt. Dieser Abschnitt enthält eine Beschreibung neuer Funktionen in dieser Version:

- „Universal Message Service (UMS)“ auf Seite 14
- „Unterstützung der AIX-Plattform“ auf Seite 16
- „Neues Zip-basiertes Installationsprogramm“ auf Seite 17
- „Erweiterte Plattformunterstützung“ auf Seite 17
- „Zusätzliche Erweiterungen“ auf Seite 18

Universal Message Service (UMS)

Mit Message Queue 4.3 wird ein neuer Universal Messaging Service (UMS) und eine Messaging-API eingeführt, die den Zugriff auf Message Queue über ein beliebiges HTTP-fähiges Gerät ermöglicht. Auf diese Art können fast alle Anwendungen miteinander kommunizieren und von der Zuverlässigkeit und garantierten Zustellung durch JMS-Messaging profitieren. Außerdem bietet UMS eine erweiterte Skalierbarkeit für JMS-Messaging, wodurch die Anzahl der Messaging-Clients internetähnliche Ausmaße erreichen kann.

Architektur

Die grundlegende UMS-Architektur wird in der folgenden Abbildung gezeigt:

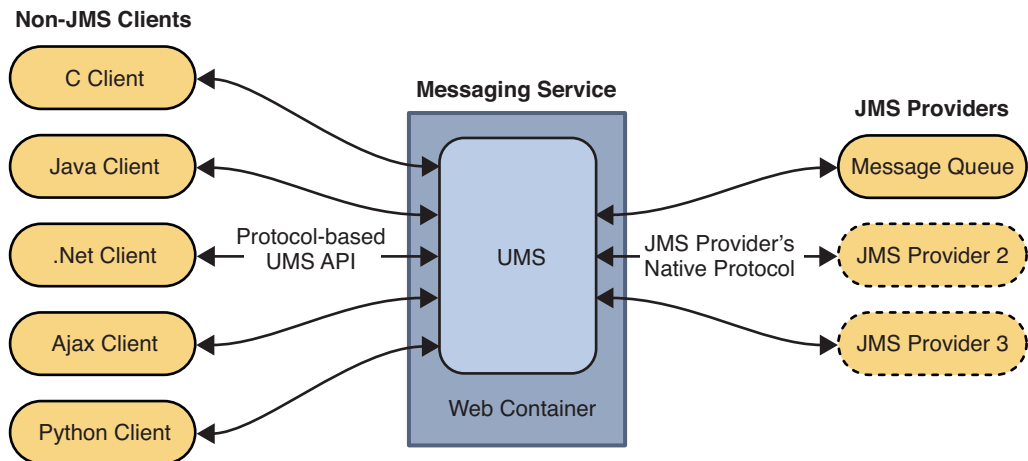


ABBILDUNG 1-1 UMS-Architektur

Der UMS, der auf einem Webserver ausgeführt wird, ist sprachneutral und plattformunabhängig. Der UMS dient als Schnittstelle zwischen einer Client-Anwendung ohne JMS und einem JMS-Provider. Er empfängt Meldungen, die mithilfe der UMS-API versendet werden, wandelt sie in JMS-Meldungen um und erzeugt sie am Ziel im JMS-Provider mithilfe des nativen Protokolls des Providers. Ebenso empfängt er Nachrichten von den Zielen im

JMS-Provider, wandelt sie in Text oder SOAP-Nachrichten um und sendet sie entsprechend den über die UMS-API angegebenen Anforderungen an Clients ohne JMS.

Die einfache, sprachunabhängige, protokollbasierte UMS-API unterstützt sowohl webbasierte als auch nicht webbasierte Anwendungen und kann mit Skript- und Programmiersprachen verwendet werden. Die API wird in zwei Stilen angeboten: als einfache Messaging-API, die ein Protokoll des Typs Representational State Transfer (REST) verwendet, und als XML-Messaging-API, die das Protokoll in einen SOAP-Nachrichten-Header einbettet. In beiden Fällen erfordert jedoch die API nur eine einfache HTTP-Anforderung zum Senden oder Empfangen einer Nachricht.

Die Einfachheit und Flexibilität der UMS-API bedeutet, dass AJAX, .NET, Python, C, Java und viele andere Anwendungen Textnachrichten und/oder SOAP-Nachrichten (mit Anhang) an JMS-Ziele senden oder von JMS-Zielen empfangen können. Beispielsweise können Python-Anwendungen mit .NET-Anwendungen, das iPhone mit Java-Anwendungen usw. kommunizieren.

Für Message Queue 4.3 unterstützt der UMS nur Message Queue als JMS-Provider.

Weitere Funktionen

Der UMS kann nicht nur wie oben beschrieben als einfache Schnittstelle verwendet werden. Er unterstützt Client-Sitzungen mit und ohne Status. Der UMS behält den Sitzungsstatus für die Client-Anwendung über mehrere Dienstanforderungen hinweg bei, wenn dies vom Client gefordert wird. Der UMS kann eine Container-verwaltete Authentifizierung nutzen und/oder zur Authentifizierung von Clients mithilfe des Message Queue-Brokers konfiguriert werden. Der UMS unterstützt außerdem Transaktionen und ermöglicht Client-Anwendungen das Übernehmen und Zurücksetzen von mehreren Dienstanforderungen als einzelne atomare Einheit.

Da der UMS eine große Anzahl von Clients über eine einzelne Verbindung mit dem Message Queue-Broker unterstützen kann, entlastet er die Verbindungsdienste des Brokers und ermöglicht maximale Skalierbarkeit. Außerdem kann die UMS-Kapazität durch horizontale Skalierung erhöht werden und ermöglicht so ein internetähnliches Messaging-Aufkommen.

Aufgrund der Einfachheit der protokollbasierten UMS-API sind auf der Client-Seite keine Client-Bibliotheken erforderlich. Dadurch kann die API zukünftig für zusätzliche JMS-Funktionen ohne Upgrade der Client-Anwendungen erweitert werden.

Verwenden von UMS

Um den UMS zu verwenden, stellen Sie ihn in einem Webcontainer bereit, der Servlet 2.4 oder höhere Spezifikationen unterstützt. Starten Sie den Message Queue-Broker, erstellen Sie die gewünschten Ziele und schreiben Sie eine Messaging-Anwendung, die die UMS-API zum Senden oder Empfangen von Nachrichten verwendet.

Die im Rahmen von Message Queue 4.3 bereitgestellte UMS-Datei `imqums.war` wird, abhängig von der Plattform, an einem der folgenden Speicherorte installiert:

Sie können die `.war`-Datei bei Bedarf umbenennen.

TABELLE 1-5 Speicherort der Datei `imqums.war`

Plattform	Speicherort von <code>imqums.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>

Nachdem Sie die Datei `imqums.war` in einem Webcontainer unter `localhost:port` bereitgestellt haben, finden Sie die UMS-Dokumentation unter:

`http://localhost:port/imqums`

Andernfalls steht Ihnen folgende UMS-Dokumentation zur Verfügung:

- Informationen zur Konfiguration des UMS finden Sie unter <https://mq.dev.java.net/4.3-content/ums/config.html>.
- Die Dokumentation zur UMS-API finden Sie unter <https://mq.dev.java.net/4.3-content/ums/protocol.html>.
- Programmierbeispiele in mehreren Sprachen finden Sie unter <https://mq.dev.java.net/4.3-content/ums/examples/README.html>.

Unterstützte Webcontainer

UMS wird gegenwärtig auf folgenden Webcontainern unterstützt:

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

Unterstützung der AIX-Plattform

Message Queue 4.3 bietet Pakete für die AIX-Plattform und ein entsprechendes Installationsprogramm.

Die AIX-Implementierung von Message Queue unterstützt folgende Software:

- AIX v 6.1 oder höher (frühere Versionen von AIX werden über das Unix/Java Only-Bundle unterstützt)
- DB2-Unterstützung
- IBM XL C/C++ Compiler V9.0

- JDK 1.5 oder höher

Installationsanweisungen finden Sie in Kapitel 4, „AIX Installation“, im *Sun Java System Message Queue 4.3 Installation Guide*.

Auf der AIX-Plattform werden Message Queue-Dateien in einem einzigen Message Queue-Ausgangsverzeichnis, `IMQ_HOME`, installiert. `IMQ_HOME` gibt das Verzeichnis `mqInstallHome/mq` an, wobei `mqInstallHome` dem Ausgangsverzeichnis der Installation entspricht, das Sie bei der Installation des Produkts angeben (standardmäßig `home-directory/MessageQueue`).

Daraus ergibt sich die gleiche Message Queue-Verzeichnisstruktur wie bei der Windows-Plattform (siehe Abschnitt zu Windows in [Anhang A](#), „Distribution-Specific Locations of Message Queue Data“ in *Sun GlassFish Message Queue 4.4 Administration Guide*).

Die Message Queue-Unterstützung für die AIX-Plattform schließt die Unterstützung der Message Queue C-API ein. Anweisungen zum Erstellen und Kompilieren von C-Anwendungen auf der AIX-Plattform finden Sie unter XREF.

Neues Zip-basiertes Installationsprogramm

Mit Message Queue 4.3 wird ein neues Installationsprogramm eingeführt, das eine Zip-basierte Verteilung im Gegensatz zur Verteilung in Form von nativen Paketen ermöglicht. Das Programm installiert die .zip-Pakete für Message Queue auf der AIX-Plattform.

Das neue Installationsprogramm extrahiert die .zip-Dateien für Message Queue in alle Verzeichnisse, auf die Sie Schreibzugriff besitzen. (Sie benötigen keine root-Berechtigung.) Es ermöglicht außerdem das Registrieren Ihrer Message Queue-Installation bei Sun Connection.

Um die Größe der Download-Bundles zu reduzieren, ist die Java Runtime nicht mehr in der zip-basierten Verteilung enthalten (auf den meisten Sites ist sie bereits vorhanden). Deshalb muss für den Befehl `installer` ein JDK oder eine JRE angegeben werden, entweder über die Umgebungsvariable `JAVA_HOME` oder über die Befehlszeile mithilfe der Option `-j`. Beispiel:

```
$ installer -j JDK/JRE-Pfad
```

Dabei entspricht *JDK/JRE-Pfad* dem Pfad des JDK oder der JRE.

Erweiterte Plattformunterstützung

Die folgende aktualisierte Plattformunterstützung wird für Message Queue 4.3 zertifiziert:

- Oracle 11g
- Windows Server 2008

Zusätzliche Erweiterungen

Message Queue 4.3 bietet außerdem die folgenden zusätzlichen Erweiterungen:

- „Neue Verzeichnisstruktur für Windows-Plattform“ auf Seite 18
- „Neue Broker-Eigenschaften“ auf Seite 18
- „Verbesserungen der JMX Administration-API“ auf Seite 18
- „Auflisten dauerhafter Abonnements für Platzhalter-Abonnenten“ auf Seite 19

Neue Verzeichnisstruktur für Windows-Plattform

Die installierte Verzeichnisstruktur für Message Queue auf der Windows-Plattform wurde im Vergleich zu früheren Versionen geändert und an diejenige der AIX-Plattform angepasst. Diese Verzeichnisstruktur wird zukünftig auch für die Solaris- und Linux-Plattformen verwendet. Dies ermöglicht mehrere Installationen auf einem einzelnen Computer und die automatische Aktualisierung von Message Queue über Sun Connection, einem von Sun gehosteten Dienst, mit dem Sie Sun-Hardware und -Software aufzeichnen, organisieren und warten können (siehe „Unterstützung des Installationsprogramms für Sun Connection-Registrierung“ auf Seite 21).

Neue Broker-Eigenschaften

Die nachfolgend beschriebenen neuen Eigenschaften sind bei der Konfiguration eines Brokers verfügbar:

TABELLE 1-6 Broker-Weiterleitungs- und Zustellungseigenschaften

Eigenschaft	Typ	Standardwert	Beschreibung
<code>imq.transaction.producer.maxNumMsgs</code>	Integer	1000	Die maximale Anzahl von Nachrichten, die der Produzent in einer einzelnen Transaktion verarbeiten kann. Es wird empfohlen, einen Wert unter 5000 festzulegen, um eine Überlastung der Ressourcen zu verhindern.
<code>imq.transaction.consumer.maxNumMsgs</code>	Integer	100	Die maximale Anzahl von Nachrichten, die der Konsument in einer einzelnen Transaktion verarbeiten kann. Es wird empfohlen, einen Wert unter 1000 festzulegen, um eine Überlastung der Ressourcen zu verhindern.
<code>imq.persist.jdbc.connection.limit</code>	Integer	5	Die maximale Anzahl von Verbindungen zur Datenbank, die geöffnet werden können.

Verbesserungen der JMX Administration-API

Der JMX-API wurden ein neues Attribut und neue Schlüssel für zusammengesetzte Datumsangaben hinzugefügt:

- Ein `NextMessageID`-Attribut wurde der Zielüberwachungs-MBean hinzugefügt, um die JMS-Nachrichten-ID für die nächste an einen Konsumenten zu liefernde Nachricht anzuzeigen.

- Ein NextMessageID-Schlüssel für zusammengesetzte Datumsangaben wurde der Überwachungs-MBean für den Konsumentenmanager hinzugefügt, um die JMS-Nachrichten-ID der nächsten an den Konsumenten zu liefernden Nachricht anzuzeigen.
- Ein NumMsgsPending-Schlüssel für zusammengesetzte Datumsangaben wurde der Überwachungs-MBean des Konsumentenmanagers hinzugefügt, um die Anzahl von Nachrichten anzuzeigen, die an den Konsumenten gesendet wurden.

Weitere Informationen finden Sie in [Kapitel 3, „Message Queue MBean Reference“ in *Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients*](#).

Auflisten dauerhafter Abonnements für Platzhalter-Abonnenten

Der Befehl zum Auflisten dauerhafter Abonnements:

```
list dur [-d Themenname]
```

wurde erweitert, sodass die Angabe des Themennamens optional ist. Wenn kein Thema angegeben wird, listet der Befehl die dauerhaften Abonnements für alle Themen auf (auch diejenigen, für die Platzhalter-Namenskonventionen verwendet wurden).

Neue Funktionen in Message Queue 4.2

Message Queue 4.2 war eine Unterversion, die eine Reihe neuer Funktionen, einige Funktionserweiterungen und Fehlerkorrekturen enthielt. In diesem Abschnitt werden die neuen Funktionen in der Version 4.2 beschrieben und weitere Referenzen angegeben:

- „Mehrere Ziele für einen Herausgeber bzw. Abonnenten“ auf Seite 19
- „Schemavalidierung von XML-Payload-Meldungen“ auf Seite 20
- „C-API-Unterstützung für verteilte Transaktionen“ auf Seite 20
- „Unterstützung des Installationsprogramms für Sun Connection-Registrierung“ auf Seite 21
- „Unterstützung für MySQL-Datenbank“ auf Seite 21
- „Zusätzliche Erweiterungen“ auf Seite 22

Informationen zu den neuen Funktionen in den Versionen Message Queue 4.1 und 4.0 finden Sie in „[Neue Funktionen in Message Queue 4.1](#)“ auf Seite 23 bzw. „[Neue Funktionen in Message Queue 4.0](#)“ auf Seite 27.

Mehrere Ziele für einen Herausgeber bzw. Abonnenten

Ab Message Queue 4.2 kann ein Herausgeber Nachrichten an mehrere Themenziele veröffentlichen und ein Abonnent kann Nachrichten von mehreren Themenzielen konsumieren. Diese Fähigkeit wird erreicht, indem ein Themenzielname mit Platzhalterzeichen verwendet wird, der für mehrere Ziele steht. Durch die Verwendung dieser

symbolischen Namen können die Administratoren bei Bedarf zusätzliche Themenziele erstellen, die dem Benennungsschema mit Platzhaltern entsprechen. Die hinzugefügten Ziele werden automatisch bei Veröffentlichungen durch Herausgeber und beim Konsum durch Abonnenten berücksichtigt. (Bei Abonnenten sind Themen mit Platzhalterzeichen üblicher als bei Herausgebern.)

Hinweis – Diese Funktion gilt nicht für Warteschlangenziele.

Eine Beschreibung des Formats der symbolischen Themenzielnamen und Verwendungsbeispiele erhalten Sie in „Supported Topic Destination Names“ in *Sun GlassFish Message Queue 4.4 Administration Guide*.

Schemavalidierung von XML-Payload-Meldungen

Diese in Message Queue 4.2 eingeführte Funktion ermöglicht die Validierung des Inhalts einer Text-XML-Nachricht (keine Objektnachrichten) anhand eines XML-Schemas an der Stelle, an der die Nachricht an den Broker gesendet wird. Der Speicherort des XML-Schemas (XSD) wird als Eigenschaft eines Message Queue-Ziels angegeben. Wenn kein XSD-Speicherort angegeben wurde, wird die DTD-Deklaration innerhalb des XML-Dokuments zur Durchführung der DTD-Validierung verwendet. (XSD-Validierung, die Datentyp und Wertebereichsvalidierung beinhaltet, ist strenger als DTD-Validierung.)

Informationen zur Verwendung dieser Funktion finden Sie unter „Schema Validation of XML Payload Messages“ in *Sun GlassFish Message Queue 4.4 Developer’s Guide for Java Clients*.

C-API-Unterstützung für verteilte Transaktionen

Gemäß dem verteilten Transaktionsmodell von X/Open beruht die Unterstützung für Transaktionen auf einem Manager für verteilte Transaktionen, der die von einer oder mehreren Ressourcenmanagern durchgeführten Vorgänge aufzeichnet und verwaltet. In Message Queue 4.2 unterstützt die Message Queue C-API die XA-Schnittstelle (zwischen einem Manager für verteilte Transaktionen und Message Queue als XA-konformen Ressourcenmanager). Dies ermöglicht Message Queue C-API-Clients, die in einer Umgebung ausgeführt werden, in der verteilte Transaktionen verarbeitet werden (wie beispielsweise BEA Tuxedo), sich an verteilten Transaktionen zu beteiligen.

Diese Unterstützung für verteilte Transaktionen besteht aus folgenden neuen C-AP-Funktionen (und neuen Parametern und Fehlercodes), die zur Implementierung der XA-Schnittstellenspezifikation verwendet werden:

```
MQGetXAConnection()  
MQCreateXASession()
```

Wenn eine C-Client-Anwendung im Kontext einer verteilten Transaktion verwendet werden soll, muss sie mithilfe von `MQGetXACONNECTION()` eine Verbindung herstellen und mithilfe von `MQCREATEXASession()` eine Sitzung für Produktion und Konsum von Nachrichten erstellen. Starten, Übernehmen und Zurücksetzen von verteilten Transaktionen wird durch APIs verwaltet, die vom Manager für verteilte Transaktionen bereitgestellt werden.

Einzelheiten zur Verwendung der verteilten Transaktionsfunktionen finden Sie in [„Working With Distributed Transactions“](#) in *Sun GlassFish Message Queue 4.4 Developer's Guide for C Clients*.

Message Queue 4.2 bietet Programmierungsbeispiele, die auf dem Tuxedo-Transaktionsmanager beruhen. Informationen zur Verwendung dieser Beispielprogramme finden Sie in [„Distributed Transaction Sample Programs“](#) in *Sun GlassFish Message Queue 4.4 Developer's Guide for C Clients*.

Hinweis – Die verteilte Transaktionsfunktionalität wird auf Solaris, Linux und Windows-Plattformen unterstützt. Bislang ist sie jedoch nur für die Solaris-Plattform zertifiziert.

Unterstützung des Installationsprogramms für Sun Connection-Registrierung

Das Message Queue-Installationsprogramm wurde erweitert und ermöglicht die Registrierung von Message Queue bei Sun Connection, einem von Sun gehosteten Dienst, mit dem Sie Sun-Hardware und -Software aufzeichnen, organisieren und warten können.

Im Rahmen der Message Queue-Installation können Sie Message Queue bei Sun Connection registrieren lassen. Informationen zur installierten Message Queue, wie beispielsweise Release-Version, Hostname Betriebssystem, Installationsdatum und ähnliche grundlegende Informationen werden sicher an die Sun Connection-Datenbank übertragen. Der Sun Connection-Inventardienst kann Sie bei der Organisation Ihrer Hard- und Software von Sun unterstützen, während der Aktualisierungsdienst Sie über die aktuellsten Sicherheitsfixes, empfohlene Updates und Funktionserweiterungen informieren kann.

Einzelheiten zum Registrieren von Message Queue bei Sun Connection finden Sie im *Sun Java System Message Queue 4.3-Installationshandbuch*.

Unterstützung für MySQL-Datenbank

Ab Message Queue 4.2 werden MySQL-Datenbanken als JDBC-basierte Datenspeicher unterstützt. MySQL Cluster Edition kann als JDBC-Datenbank für einen eigenständigen Broker verwendet werden, und MySQL Cluster Edition kann als hochverfügbarer gemeinsam genutzter Datenspeicher verwendet werden, der für ein erweitertes Broker-Cluster benötigt wird. Informationen zur Konfiguration von Message Queue zur Verwendung von MySQL

finden Sie in „Configuring a JDBC-Based Data Store“ in *Sun GlassFish Message Queue 4.4 Administration Guide* und in „Enhanced Broker Cluster Properties“ in *Sun GlassFish Message Queue 4.4 Administration Guide*.

Zusätzliche Erweiterungen

Zusätzlich zu den oben beschriebenen Funktionen schließt Message Queue 4.2 folgende Erweiterungen ein:

■ Metriken für remote produzierte Nachrichten

Message Queue 4.2 beinhaltet eine neue Zielmetrik, die bei der Überwachung von Zielen in einem Broker-Cluster nützlich sein kann. In einem Broker-Cluster setzen sich die Nachrichten, die in einem bestimmten Ziel auf einem bestimmten Broker im Cluster gespeichert sind, aus Nachrichten zusammen, die direkt für dieses Ziel produziert wurden, und aus Nachrichten, die von Remote-Brokern im Cluster am das Ziel gesendet wurden. Bei der Analyse der Nachrichtenweiterleitung und -zustellung in einem Broker-Cluster ist es zuweilen hilfreich zu wissen, wie viele Nachrichten in einem Ziel lokal (lokal produziert) und wie viele remote (remote produziert) sind.

Message Queue 4.2 verfügt über zwei neue Größen für physische Ziele:

Num messages remote (Anzahl der Remote-Nachrichten) und Total Message bytes remote (Bytezahl Remote-Nachrichten insgesamt). Die neuen metrischen Größen sind über die Befehle `imqcmd list dst` und `imqcmd query dst` (siehe „Viewing Physical Destination Information“ in *Sun GlassFish Message Queue 4.4 Administration Guide*) sowie über neue JMX-Attribute (siehe „Destination Monitor“ in *Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients*) verfügbar.

■ Informationen zu Platzhalter-Produzenten und Platzhalter-Konsumenten

Informationen zur Unterstützung der Verwendung von Platzhalterzeichen in Zielnamen (siehe „Mehrere Ziele für einen Herausgeber bzw. Abonnenten“ auf Seite 19) werden über neue Überwachungsdaten bereitgestellt. Beispielsweise kann die Anzahl der mit einem Ziel verbundenen Platzhalter-Produzenten oder -Konsumenten über den Befehl `imqcmd query dst` abgefragt werden (siehe „Viewing Physical Destination Information“ in *Sun GlassFish Message Queue 4.4 Administration Guide*) und ist auch über neue JMX-Attribute verfügbar (siehe „Destination Monitor“ in *Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients*). Platzhalterinformationen stehen auch über die `ConsumerManager Monitor`- und `ProducerManager Monitor`-MBeans zur Verfügung.

■ Unterstützung für das DN-Benutzernamensformat für Client-Authentifizierung

Ab Message Queue 4.2 wird das DN-Benutzernamensformat bei der Client-Verbindungsauthentifizierung anhand eines LDAP-Benutzer-Repositorys unterstützt. Die Unterstützung beinhaltet folgende neue Broker-Eigenschaft (und Wert):

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

Mit dieser Eigenschaft kann der Broker einen Client-Benutzer anhand eines Eintrags in einem LDAP-Benutzer-Repository authentifizieren, indem aus dem DN-Benutzernamensformat der Wert des durch folgende Eigenschaft angegebenen Attributs extrahiert wird:

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

Der Broker verwendet den Wert des oben angegebenen Attributs als Namen des Benutzers bei Zugriffssteuerungsvorgängen.

Wenn beispielsweise gilt: `imq.user_repository.ldap.uidattr=udi` und wenn ein Benutzername für die Client-Authentifizierung im Format `udi=mquser,ou=People,dc=red,dc=sun,dc=com` vorliegt, dann wird "mquser" zur Durchführung der Zugriffssteuerung extrahiert.

- **Erweiterung der JAAS-Authentifizierung**

Ab Message Queue 4.2 unterstützt die JAAS-Authentifizierung die Authentifizierung durch IP-Adressen sowie durch Benutzernamen.

Neue Funktionen in Message Queue 4.1

Message Queue 4.1 war eine Unterversion, die eine Reihe neuer Funktionen, einige Funktionserweiterungen und Fehlerkorrekturen enthielt. In diesem Abschnitt werden die neuen Funktionen in der Version 4.1 beschrieben und weitere Referenzen für Ihre Verwendung angegeben:

- „Hochverfügbarkeits-Broker-Cluster“ auf Seite 23
- „JAAS-Unterstützung“ auf Seite 25
- „Formatänderung für persistenten Datenspeicher“ auf Seite 25
- „Broker-Umgebungs-Konfiguration“ auf Seite 25
- „Unterstützung für Java ES Monitoring Framework“ auf Seite 25
- „Erweiterte Transaktionsverwaltung“ auf Seite 26
- „Feste Ports für C-Clientverbindungen“ auf Seite 26

Informationen zu den in Message Queue 4.0 hinzugefügten Funktionen finden Sie unter „[Neue Funktionen in Message Queue 4.0](#)“ auf Seite 27.

Hochverfügbarkeits-Broker-Cluster

Ab Message Queue 4.1 ist ein neues, erweitertes Broker-Cluster verfügbar. Im Vergleich zu einem konventionellen Broker-Cluster, das nur *Nachrichtendienst*-Verfügbarkeit bietet (wenn ein Broker ausfällt, ist ein anderer Broker verfügbar, um den Nachrichtendienst bereitzustellen), bietet das erweiterte Broker-Cluster auch *Daten*-Verfügbarkeit (wenn ein Broker ausfällt, stehen seine persistenten Nachrichten und Statusdaten einem anderen Broker zur Verfügung, der sie verwenden kann, um die Nachrichtenzustellung zu übernehmen).

Die Hochverfügbarkeitsimplementierung ab Message Queue 4.1 verwendet einen gemeinsam genutzten JDBC-basierten Datenspeicher: Anstatt dass jedes Broker-Cluster seinen eigenen persistenten Datenspeicher besitzt, verwenden alle Broker im Cluster dieselbe JDBC-konforme Datenbank. Wenn ein bestimmter Broker ausfällt, übernimmt ein anderer Broker innerhalb des Clusters die Nachrichtenzustellung für den ausgefallenen Broker. Dabei verwendet der Failover-Broker Daten und Statusinformationen aus dem gemeinsam genutzten Datenspeicher. Messaging-Clients des ausgefallenen Brokers stellen eine neue Verbindung zum Failover-Broker her, der ununterbrochenen Nachrichtendienst bietet.

Der freigegebene JDBC-basierte Speicher, der in der Message Queue 4.1-Hochverfügbarkeitsimplementierung verwendet wurde, muss ebenfalls hochverfügbar sein. Wenn Sie keine hochverfügbare Datenbank besitzen oder wenn eine unterbrechungsfreie Nachrichtenzustellung für Sie nicht von hoher Wichtigkeit ist, können Sie weiterhin konventionelle Cluster verwenden, die Dienstverfügbarkeit ohne Datenverfügbarkeit bieten.

Um ein erweitertes Message Queue 4.1-Broker-Cluster zu konfigurieren, geben Sie für jeden Broker im Cluster folgende Broker-Eigenschaften an:

- *Eigenschaften für die Clustermitgliedschaft*, die angeben, dass der Broker Teil eines erweiterten Broker-Clusters ist, und die ID des Clusters und des Brokers innerhalb des Clusters festlegen
- *Eigenschaften für hochverfügbare Datenbanken*, die das persistente Datenmodell (JDBC), den Namen des Datenbankherstellers und herstellerspezifische Konfigurationseigenschaften angeben
- *Eigenschaften für Fehlererkennung und Failover*, die angeben, wie ein Broker-Ausfall ermittelt und mithilfe eines Failover-Brokers behandelt wird

Um die Implementierung mit dem erweiterten Broker-Cluster verwenden zu können, müssen Sie folgende Schritte ausführen:

1. Installieren einer hochverfügbaren Datenbank
2. Installieren der .jar-Datei des JDBC-Treibers
3. Erstellen des Datenbankschemas für den hochverfügbaren persistenten Datenspeicher
4. Festlegen der Hochverfügbarkeitseigenschaften für jeden Broker im Cluster
5. Starten der einzelnen Broker im Cluster

Eine Beschreibung der Konzepte des erweiterten Broker-Clusters und ein Vergleich mit konventionellen Clustern finden Sie in [Kapitel 4, „Broker Clusters“ in *Sun GlassFish Message Queue 4.4 Technical Overview*](#). Die Vorgehensweise und Referenzinformationen zu erweiterten Broker-Clustern finden Sie in [Kapitel 10, „Configuring and Managing Broker Clusters“ in *Sun GlassFish Message Queue 4.4 Administration Guide*](#) und in [„Cluster Configuration Properties“ in *Sun GlassFish Message Queue 4.4 Administration Guide*](#).

Wenn Sie eine hochverfügbare Datenbank zusammen mit Message Queue 4.0 verwendet haben und nun zu einem erweiterten Broker-Cluster wechseln möchten, können Sie mit dem

Datenbank-Manager-Dienstprogramm (`imqdbmgr`) die Konvertierung in einen gemeinsam genutzten persistenten Datenspeicher durchführen. Weitere Informationen zu Bekannten Problemen und Einschränkungen finden Sie unter „[Broker-Cluster](#)“ auf Seite 54.

JAAS-Unterstützung

Neben den dateibasierten und den LDAP-basierten integrierten Authentifizierungsmechanismen wurde in Message Queue 4.1 auch die Unterstützung für den Java Authentication and Authorization Service (JAAS), aufgenommen, sodass Sie einen externen Authentifizierungsmechanismus mit dem Broker verbinden können, um Message Queue-Clients zu authentifizieren.

Eine Beschreibung der Informationen, die ein Broker einem JAAS-konformen Authentifizierungsdienst zur Verfügung stellt und eine Erläuterung der Konfiguration des Brokers zur Verwendung eines solchen Diensts finden Sie in „[Using JAAS-Based Authentication](#)“ in *Sun GlassFish Message Queue 4.4 Administration Guide*.

Formatänderung für persistenten Datenspeicher

In Message Queue 4.1 wurde der JDBC-basierten Datenspeicher dahingehend geändert, dass erweiterte Broker-Cluster unterstützt werden. Aus diesem Grund wird das Format des JDBC-basierten Datenspeichers auf Version 410 erhöht. Die Formatversionen 350, 370 und 400 werden automatisch zu Version 410 migriert.

Beachten Sie, dass das Format des dateibasierten persistenten Datenspeichers auf Version 370 verbleibt, da daran keine Änderungen vorgenommen wurden.

Broker-Umgebungs-Konfiguration

Die Eigenschaft `IMQ_DEFAULT_EXT_JARS` wurde in die Umgebungsconfigurationsdatei von Message Queue 4.1, `imqenv.conf`, aufgenommen. Sie können diese Eigenschaft festlegen, um die Pfadnamen externer `.jar`-Dateien anzugeben, die beim Start des Brokers in `CLASSPATH` aufgenommen werden sollen. Wenn Sie diese Eigenschaft verwenden, um den Speicherort von externen `.jar`-Dateien anzugeben, müssen Sie diese Dateien nicht mehr in das Verzeichnis `lib/ext` kopieren. Externe `.jar`-Dateien können sich auf JDBC-Treiber oder JAAS-Anmeldemodule beziehen. Die folgende Beispieleigenschaft gibt den Speicherort von JDBC-Treibern an.

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

Unterstützung für Java ES Monitoring Framework

In Message Queue 4.1 wurde Unterstützung für das Sun Java Enterprise System (Java ES) Monitoring Framework aufgenommen, mit dem Java ES-Komponenten mithilfe einer üblichen grafischen Oberfläche überwacht werden können. Diese Oberfläche wird durch die webbasierte Sun Java System Monitoring Console implementiert. Administratoren können mit der Konsole

Leistungsstatistiken anzeigen, Regeln für die automatische Überwachung erstellen und Alarme bestätigen. Wenn Sie Message Queue gemeinsam mit anderen Java ES-Komponenten ausführen, ist es möglicherweise praktischer, eine einzige Oberfläche zur Verwaltung all dieser Komponenten zu verwenden.

Informationen zur Verwendung des Java ES Monitoring Framework zur Überwachung von Message Queue finden Sie unter XREF.

Erweiterte Transaktionsverwaltung

Zuvor konnte ein Administrator nur für Transaktionen mit dem Status PREPARED ein Rollback ausführen. Das heißt, dass beim nicht ordnungsgemäßen Beenden einer Sitzung, die Teil einer verteilten Transaktion war, für die Transaktion ein Status beibehalten wurde, der nicht durch einen Administrator bereinigt werden konnte. In Message Queue 4.1 können Sie mit dem Befehlsdienstprogramm (`imqcmd`) Transaktionen bereinigen (durch ein Rollback zurücksetzen), die folgende Statuswerte aufweisen: STARTED, FAILED, INCOMPLETE, COMPLETE und PREPARED.

Um zu ermitteln, ob für eine bestimmte Transaktion ein Rollback durchgeführt werden kann (insbesondere, wenn diese nicht den Status PREPARED aufweist), bietet das Befehlsdienstprogramm zusätzliche Daten als Teil der `imqcmd query txn`-Ausgabe: Es gibt die ID der Verbindung an, mit der die Transaktion gestartet wurde, sowie den Zeitpunkt, an dem die Transaktion erstellt wurde. Mithilfe dieser Informationen kann ein Administrator ermitteln, ob für die Transaktion ein Rollback durchgeführt werden muss. Im Allgemeinen sollte der Administrator ein zu frühes Rollback für eine Transaktion vermeiden.

Feste Ports für C-Clientverbindungen

In Message Queue 4.1 können C-Clients nun ebenso wie Java-Clients eine Verbindung mit einem festen Broker-Anschluss herstellen und müssen nicht mehr eine Verbindung zu einem Anschluss herstellen, der dynamisch vom Portmapper-Dienst des Brokers zugewiesen wurde. Feste Anschlussverbindungen sind nützlich, wenn Sie versuchen, eine Firewall zu überwinden, oder wenn Sie den Portmapper-Dienst aus einem anderen Grund umgehen müssen.

Zur Konfiguration einer festen Anschlussverbindung müssen Sie sowohl den Broker als auch die Laufzeit des C-Client (beide Enden der Verbindung) konfigurieren. Wenn Sie Ihren Client z. B. über `ssljms` mit Anschluss1756 verbinden möchten, führen Sie die folgenden Schritte aus.

- Legen Sie auf der Client-Seite folgende Eigenschaften fest:
`MQ_SERVICE_PORT_PROPERTY=1756`
`MQ_CONNECTION_TYPE_PROPERTY=SSL`
- Legen Sie auf der Broker-Seite die Eigenschaft `imq.serviceName.protocolType.port` wie folgt fest:
`imq.ssljms.tls.port=1756`

Hinweis – Die Verbindungseigenschaft `MQ_SERVICE_PORT_PROPERTY` wurde durch Backporting auf Message Queue 3.7 Update 2 übertragen.

Neue Funktionen in Message Queue 4.0

Message Queue 4.0 war eine Unterversion, die auf die Unterstützung von Application Server 9 PE beschränkt war. Sie beinhaltet einige neue Funktionen sowie einige Funktionserweiterungen und Fehlerbehebungen. Dieser Abschnitt enthält eine Beschreibung neuer Funktionen in dieser Version:

- „Unterstützung für die JMX Administration-API“ auf Seite 27
- „Protokollierung zur Client-Laufzeit“ auf Seite 27
- „Verbindungseignissenbenachrichtigungs-API“ auf Seite 28
- „Erweiterungen der Broker-Administration“ auf Seite 28
- „Anzeigen von Informationen zu einem JDBC-basierten Datenspeicher“ auf Seite 29
- „JDBC-Anbieter-Unterstützung“ auf Seite 29
- „Formatänderungen für persistenten Datenspeicher“ auf Seite 30
- „Zusätzliche Nachrichteneigenschaften“ auf Seite 30
- „SSL-Unterstützung“ auf Seite 30



Achtung – Eine der kleinen, jedoch wichtigen Änderungen, die mit Version 4.0 eingeführt wurden, ist die Tatsache, dass die Befehlszeilenoption nun nicht mehr zur Angabe eines Passworts verwendet werden kann. Ab dieser Version müssen Sie alle Passwörter in einer Datei speichern, wie in „[Veraltete Passwortoption](#)“ auf Seite 50 beschrieben, oder Sie bei Aufforderung eingeben.

Unterstützung für die JMX Administration-API

Für die Konfiguration und Überwachung von Message Queue-Brokern wurde in Übereinstimmung mit der JMX-Spezifikation (Java Management Extensions) eine neue API hinzugefügt. Mithilfe dieser API haben Sie die Möglichkeit, Broker-Funktionen programmatisch in einer Java-Client-Anwendung zu konfigurieren und zu überwachen. In früheren Versionen von Message Queue war der Zugriff auf diese Funktionen ausschließlich über die Administrationsdienstprogramme der Befehlszeile oder die Administrationskonsole möglich.

Weitere Informationen finden Sie im [Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients](#).

Protokollierung zur Client-Laufzeit

In Message Queue 4.0 wurde Unterstützung für Client-Laufzeitprotokollierung von verbindungs- und sitzungsbezogenen Ereignissen eingeführt.

Weitere Informationen zur Client-Laufzeitprotokollierung und ihrer Konfiguration finden Sie im Java Dev Guide auf Seite 137.

Verbindungsereignisbenachrichtigungs-API

In Message Queue 4.0 wurde eine Ereignisbenachrichtigungs-API eingeführt, mit der die Client-Laufzeit eine Anwendung über Änderungen im Verbindungsstatus informieren kann. Verbindungsereignisbenachrichtigungen ermöglichen es einem Message Queue-Client, das Schließen und erneute Verbinden von Ereignissen zu überwachen und dem Benachrichtigungstyp und Verbindungsstatus entsprechende Aktionen auszuführen. Wenn es beispielsweise zu einem Failover kommt und der Client mit einem anderen Broker erneut verbunden wird, wird in einer Anwendung möglicherweise der Transaktionsstatus bereinigt, um mit einer neuen Transaktion fortzufahren.

Informationen zu Verbindungsereignissen und zum Erstellen einer Ereignisüberwachungsfunktion finden Sie im Java Dev Guide auf Seite 96.

Erweiterungen der Broker-Administration

In Message Queue 4.0 wurden ein neuer Unterbefehl und mehrere Befehlsoptionen zum Befehlsdienstprogramm (`imqcmd`) hinzugefügt, um es Administratoren zu ermöglichen, einen Broker stillzulegen (zu inaktivieren), einen Broker nach einem angegebenen Zeitraum herunterzufahren, eine Verbindung zu vernichten oder Java-Systemeigenschaften (z. B. Verbindungsbezogene Eigenschaften) festzulegen.

- Bei der Stilllegung (Inaktivierung) des Brokers wird dieser in einen inaktiven Zustand versetzt, in dem Nachrichten entfernt werden können, bevor der Broker heruntergefahren oder neu gestartet wird. Zu einem Broker, der inaktiviert wird, können keine neuen Verbindungen erstellt werden. Um den Broker zu inaktivieren, geben Sie einen Befehl ähnlich dem folgenden ein.

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

- Um den Broker nach einem bestimmten Intervall herunterzufahren, geben Sie einen Befehl ähnlich dem folgenden ein. (Das Zeitintervall gibt die Anzahl an Sekunden bis zum Herunterfahren des Brokers an.)

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

Wenn Sie ein Zeitintervall angeben, protokolliert der Broker eine Meldung, die darauf hinweist, wann er heruntergefahren wird. Zum Beispiel:

Broker wird in 29 Sekunden (29996 Millisekunden) heruntergefahren.

Während der Broker auf das Herunterfahren wartet, wird sein Verhalten folgendermaßen beeinflusst.

- Administrative JMS-Verbindungen werden weiterhin akzeptiert.
- Neue JMS-Verbindungen werden nicht akzeptiert.
- Bestehende JMS-Verbindungen werden weiterhin funktionieren.

- Der Broker kann nicht die Funktionen für einen anderen Broker in einem erweiterten Broker-Cluster übernehmen.
- Das Dienstprogramm `imqcmd` wird nicht gesperrt, es wird die Anforderung zum Herunterfahren des Brokers senden und umgehend zurückgeben.
- Um eine Verbindung zu löschen, geben Sie einen Befehl ähnlich dem folgenden ein.

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

 Verwenden Sie den Befehl `imqcmd list cxn` oder `imqcmd query cxn`, um die Verbindungs-ID abzurufen.
- Zur Angabe einer Systemeigenschaft über den Befehl `imqcmd` verwenden Sie die neue `-D`-Option. Dies ist nützlich, um die Werkseinstellungen der JMS-Verbindung oder verbindungsbezogene Java-Systemeigenschaften außer Kraft zu setzen. Beispiel:

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

Vollständige Informationen über die Syntax des Befehls `imqcmd` finden Sie in [Kapitel 16](#), „[Command Line Reference](#)“ in *Sun GlassFish Message Queue 4.4 Administration Guide*.

Anzeigen von Informationen zu einem JDBC-basierten Datenspeicher

In Message Queue 4.0 wurde ein neuer Unterbefehl `query` zum Datenbank-Manager-Dienstprogramm, `imqdbmgr`, hinzugefügt. Dieser Unterbefehl wird zum Anzeigen von Informationen zu einem JDBC-basierten Datenspeicher (u. a. Datenbankversion, Datenbankbenutzer und ob die Datenbanktabellen erstellt wurden) verwendet.

Im Folgenden finden Sie ein Beispiel für die Informationen, die über diesen Befehl angezeigt werden.

```
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.
```

JDBC-Anbieter-Unterstützung

In Message Queue 4.0 wird nun Apache Derby Version 10.1.1 als JDBC-basierter Datenspeicheranbieter unterstützt.

Formatänderungen für persistenten Datenspeicher

Message Queue 4.0 führte Änderungen am JDBC-basierten Datenspeicher zur Optimierung und zur Unterstützung zukünftiger Erweiterungen ein. Aus diesem Grund wurde das Format des JDBC-basierten Datenspeichers auf Version 400 erhöht. Beachten Sie, dass die Version des dateibasierten Datenspeichers Version 370 bleibt, da daran keine Änderungen vorgenommen wurden.

Zusätzliche Nachrichteneigenschaften

Message Queue 4.0 fügte zwei neue Eigenschaften hinzu, die für alle Nachrichten festgelegt werden, die in die Warteschlange für nicht zugestellte Nachrichten eingestellt werden.

- `JMS_SUN_DMQ_PRODUCING_BROKER` zeigt dem Broker an, wo die Nachricht generiert wurde.
- `JMS_SUN_DMQ_DEAD_BROKER` zeigt dem Broker an, wer die Nachricht als nicht zugestellt gekennzeichnet hat.

SSL-Unterstützung

Ab Message Queue 4.0 ist der Standardwert für die Werkseinstellung der Clientverbindung `imqSSLIsHostTrusted` auf `false` gesetzt. Wenn die Anwendung jedoch von dem vorherigen Standardwert `true` abhängt, müssen Sie die Eigenschaft neu konfigurieren und ausdrücklich auf `true` setzen.

Möglicherweise vertrauen Sie dem Host, wenn der Broker so konfiguriert ist, dass er selbst signierte Zertifikate verwendet. In diesem Fall sollten Sie zusätzlich zur Angabe, dass die Verbindung einen SSL-basierten Verbindungsdienst verwenden soll (über die `imqConnectionType`-Eigenschaft), die Eigenschaft `imqSSLIsHostTrusted` auf `"true"` setzen.

Um Clientanwendungen sicher auszuführen, wenn der Broker selbst signierte Zertifikate verwendet, verwenden Sie beispielsweise einen ähnlichen Befehl wie den folgenden.

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

Um das Befehlsdienstprogramm (`imqcmd`) sicher zu verwenden, wenn der Broker selbstsignierte Zertifikate verwendet, müssen Sie einen Befehl wie den folgenden (zur Auflistung der Konnektorservices) verwenden.

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

In zukünftigen Versionen veraltete Funktionen

Die folgenden Funktionen sind in zukünftigen Versionen veraltet:

- **Nachrichtenbasierte Überwachung**

Die nachrichtenbasierte Überwachung nutzt den konfigurierbaren Metrics Message Producer, um Metrikdaten in JMS-Nachrichten zu schreiben, die anschließend in Abhängigkeit vom Typ der in den Nachrichten enthaltenen Metrikinformationen an Metrik-Themenziele gesendet werden. Der Zugriff auf diese Metrikinformationen erfolgt dann durch Schreiben einer Client-Anwendung, die das entsprechende Metrik-Themenziel abonniert, die zugehörigen Nachrichten konsumiert und die Daten wie festgelegt verarbeitet.

Die nachrichtenbasierte Überwachungsfunktion wurde durch die JMX Administration-API ersetzt, die in MQ 4.0 implementiert wurde (siehe „[Unterstützung für die JMX Administration-API](#)“ auf Seite 27). Die JMX-API ist umfassender (sie enthält mehr Metrikdaten als an die Themenziele gesendet werden) und beruht auf dem Industriestandard JMX.

Es gibt keinen zwingenden Grund zur Verwendung der nachrichtenbasierten Überwachung, nun da Message Queue die JMX-API unterstützt. Informationen zur nachrichtenbasierten Überwachung bleiben in der Message Queue-Dokumentation erhalten, bis die Funktion formal verworfen wurde.

- **Textbasiertes Installationsprogramm**

Der Textmodus des Message Queue-Installationsprogramms (`installer -t`) wird für alle Betriebssystemplattformen entfernt. Im Textmodus wird einfacher Text in einem Terminal-Fenster angezeigt, um den GUI-Modus (GUI, grafische Benutzeroberfläche) zu simulieren. Der GUI-Modus und der automatische Modus werden weiterhin unterstützt.

- **Plattformunterstützung**

Windows 2000 und Red Hat Linux 3 werden in zukünftigen Versionen nicht mehr unterstützt.

- **JMSRA-Ressourcenadapter**

Der Message Queue-Ressourcenadapter `mqjmsra.rar`, üblicherweise als JMSRA bezeichnet, wird in zukünftigen Message Queue-Versionen durch einen neuen Ressourcenadapter ersetzt. JMSRA wird zur Integration von Message Queue in den Sun Java System Application Server verwendet.

Der neue Ressourcenadapter, der die vorhandenen JMSRA-Funktionen mit denjenigen anderer Sun JMS-Ressourcenadapter kombiniert, bietet spezielle Unterstützung für Message Queue und andere Provider in einer Java EE 5 Application Server-Umgebung. Er dient zur Integration von Message Queue in Sun GlassFish Enterprise Server und in Sun Java Composite Application Platform Suite (Java CAPS).

In Message Queue 4.4 Update 1 und aktuellen Versionen behobene Fehler

Message Queue 4.4 Update 1 beinhaltet neue Fehlerkorrekturen und berücksichtigt auch Fehler, die in den vorhergehenden Versionen der Message Queue 4.x-Familie behoben wurden.

In den folgenden Abschnitten werden Fehler aufgeführt, die in den jeweiligen Versionen behoben wurden:

- „In Message Queue 4.4 Update 1 behobene Fehler“ auf Seite 32
- „Behobene Probleme in Message Queue 4.4“ auf Seite 34
- „In Message Queue 4.3 behobene Fehler“ auf Seite 37
- „In Message Queue 4.2 behobene Fehler“ auf Seite 37
- „In Message Queue 4.1 behobene Fehler“ auf Seite 39
- „Behobene Probleme in Message Queue 4.0“ auf Seite 40

In Message Queue 4.4 Update 1 behobene Fehler

In der folgenden Tabelle sind die in Message Queue 4.4 Update 1 behobenen Fehler aufgelistet. Einige dieser Probleme werden mit ?(OpenMQ)“ gekennzeichnet, was darauf hinweist, dass das Problem im Open Source-Projekt [Open Message Queue](#), auf dem Sun GlassFish Message Queue basiert, behoben wurde.

TABELLE 1-7 In Message Queue 4.4 Update 1 behobene Fehler

Fehler	Beschreibung
6590909	DIRECT-Modus MDB stellt keine Verbindung mit dem Remote-Broker her, wenn die Adressliste überschrieben wird.
6616704	Vergrößerung des Broker-Speichers bei Erstellung vieler Konsumenten innerhalb einer Sitzung
6745761	XAResource.isSameRM() sollte ?true“ (wahr) ausgeben, wenn zwei Verbindungen in derselben XA-Transaktion (mit JMSJCA) verwendet werden.
6745763	XAResource.isSameRM() sollte ?true“ (wahr) ausgeben, wenn zwei Verbindungen in derselben XA-Transaktion (JMSRA DIRECT-Modus) verwendet werden.
6745768	XAResource.isSameRM() sollte ?true“ (wahr) ausgeben, wenn zwei Verbindungen in derselben XA-Transaktion (JMSRA LOCAL/REMOTE) verwendet werden.
6760450	Beschädigung des Nachrichtenspeichers beim Neustart des Computers ohne Anhalten der MQ (GF)-Instanz
6766241	UMS: Das AJAX-Beispiel SendMsg.html verwendet /ums als standardmäßiges Kontext-Root. Es sollte /imqums verwenden.

TABELLE 1-7 In Message Queue 4.4 Update 1 behobene Fehler (Fortsetzung)

Fehler	Beschreibung
6766852	DirectXAResource übersetzt den vom Broker empfangenen Status CONFLICT in "TxID wird bereits verwendet".
6799428	Nicht dauerhafte Nachrichten/Nicht dauerhaft in DMQ hinterlegte Nachrichten können nicht verarbeitet aber durchsucht werden.
6799428	Nicht dauerhafte Nachrichten/Nicht dauerhaft in DMQ hinterlegte Nachrichten können nicht verarbeitet aber durchsucht werden.
6809353	openmq 4.3 HA mit postgresql (8.1) funktioniert nicht (imqbrokerd kann nicht starten).
6809750	Verbindungspooling (von JMSRA) für clientId-Verbindung funktioniert nicht.
6812198	Classcast-Ausnahmefehler bei Überwachung mithilfe von MQ-Themenmetriken
6832000	MQ reapExcessConnection JDBC-Verbindung führt zu HIGH CPU-Schleife.
6833109	MQClusterMonitor JMX-Beispielanwendung gibt Ausnahmefehler bei AIX mit JDK6 aus.
6835420	Standardwert von NoGCDefault falsch berechnet. Kann bei geringer Speicherkapazität eine übermäßige GC-Ausführung verursachen.
6852018	Fehlermeldung "Dauerhafter Konsument {0} kann nicht hinzugefügt werden. Keine Client-ID für Verbindung festgelegt." ist irreführend.
6856991	NullPointerException nach Broker-Neustart, dann Zurücksetzen einer dauerhaften Konsumenten-Transaktion mit dem Status PREPARED.
6874125	WARNUNG: MQJMSRA_DC2001: connectionId=555670328604044289:_destroy():called on a connection...
6878945	RFE: JMSBridge: Angabe eines Benutzernamen/Passworts für die Erstellung einer Verbindung von einer Connection Factory zulassen
6881493	Temporäre Admin-Ziele sollten für HA-Broker nicht gespeichert werden.
6881753	RFE JMSBridge: Kennzeichnung jeder Nachricht mit dem Namen jmsbridge vor der Übertragung zum Ziel zulassen
6884673	MQ 4.4-Broker konnte keine Clusterverbindung zu MQ 3.7/3.6-Broker herstellen.
6886390	Von Persist/Txn veröffentlichte Nachrichten, die an DMQ übermittelt wurden, können den Fehler "mq.sys.dmq nicht gefunden" hervorrufen, wenn sie aus DMQ konsumiert werden.
6886515	AccessControlException bei Verwendung von JMX zum Löschen eines Ziels in einem eingebetteten Broker
6890628	Die Einstellung der Broker-Eigenschaft auf "imq.autocreate.destination.isLocalOnly=true" hat keine Auswirkung.
6891615	Selektor funktioniert nicht immer bei Ausführung von Broker 4.3 in Glassfish.

TABELLE 1-7 In Message Queue 4.4 Update 1 behobene Fehler <i>(Fortsetzung)</i>	
Fehler	Beschreibung
6891624	'Remote'-Nachrichten können in 'imqcmd list dst' höher als 'Count' sein.
6891629	Benutzerfreundliche Nachricht erforderlich, wenn arithmetischer Ausnahmefehler in Selektor auftritt
6891717	ifimq.transaction.autorollback=true,autorollback PREPARED-Bestätigung nicht gelöscht, was zur Ausnahme TransactionAckExistEx führt
6891802	"[B4061]:Transaktions-ID kann nicht verwendet werden nicht verwenden" bei Broker-Neustart nach Übernahme der Remote-Transaktions-Bestätigung
6892512	Nicht freigegebener Speicher: Temporäre Ziele werden nicht aus der Verbindung entfernt, wenn tempDest.delete() aufgerufen wird.
6895040	Wenn der Masterbroker ein temporäres Ziel hat, bekommt der Slave-Broker ein UID-Präfix bei Start nach UID-Präfix-Zeitüberschreitungen.
6896230	Neuer Konsument während Neustart des Masterbrokers erstellt, sync/w-Slaves werden danach möglicherweise nicht an alle propagiert.
6896764	Gleich-Methode bei TransactionAcknowledgement ist nicht korrekt
6898355	Übernahme-Sperre in cluster managr init bei Broker-Neustart zurückgesetzt, ohne den Abschluss der Übernahme abzuwarten
6901405	RFE: Protokoll JDBC Herstellerinformationen und Herstellereigenschaften, falls angegeben
16 (OpenMQ)	Selektor funktioniert nicht immer bei Ausführung von Broker 4.3 in Glassfish.
17 (OpenMQ)	openmq 4.3 HA mit postgresql (8.1) funktioniert nicht (imqbrokerd kann nicht starten).
22 (OpenMQ)	Installationsprogramm referenziert nicht vorhandene Binärdaten und schlägt fehl.
25 (OpenMQ)	Nicht freigegebener Speicher bei Erstellung eines temporären Themas.
29 (OpenMQ)	Broker-Isolation
30 (OpenMQ)	'Remote'-Nachrichten können in 'imqcmd list dst' höher als 'Count' sein.
31 (OpenMQ)	Benutzerfreundliche Nachricht bei arithmetischem Ausnahmefehler erforderlich tritt im Selektor auf
32 (OpenMQ)	korrigieren für int-> lange Overflows
33 (OpenMQ)	OpenMQ-Installationsprogramm: Fehler "Ungültiger SwiXML-Deskriptor" bei Ausführung unter dem Gebietsschema ja (Japanisch)

Behobene Probleme in Message Queue 4.4

In der folgenden Tabelle sind die in Message Queue 4.4 behobenen Fehler aufgelistet.

TABELLE 1-8 Behobene Probleme in Message Queue 4.4

Fehler	Beschreibung
6242247	MQ-Cluster mit Masterbroker wird gestartet und reagiert nicht, wenn beide Broker sich auf demselben Gerät befinden und denselben Namen haben.
6760937	Broker stellt keine neue Verbindung mit der Datenbank nach einem Neustart her.
6763252	Broker sollte eine aussagekräftigere Nachricht als NPE protokollieren, wenn eine Nachricht bestätigt wird, die abgelaufen ist oder entfernt wurde.
6765410	Wenn der Masterbroker lokale Interessen zwei Mal sendet, tritt ein Ausnahmefehler auf und der Slave meldet, dass das dauerhafte Abonnement bereits aktiv ist.
6796506	Remote-Nachricht mit dem Status PREPARED wird nach dem Rollback nicht erneut zugestellt, wenn ein Timeout beim Empfang der Remote-Antwort mit dem Status PREPARE auftritt.
6807708	TemporaryDestination.delete schlägt fehl, wenn Masterbroker nicht ausgeführt wird.
6812037	RFE: MQ_CALLBACK_RUNTIME_ERROR an afterMessageDelivery weiterleiten, wenn MQMessageListenerFunc einen Fehler ausgibt.
6812755	Die Nachricht der Protokollebene FINE sollte WARNING sein, wenn Rückrufe vor/nach MessageDelivery Fehlermeldung ausgeben.
6816023	Ausnahmefehler Message.setStringProperty() zeigt beim Ausnahmefehler für unzulässige Zeichen den Namen der Eigenschaft nicht an.
6819095	RFE: Cluster sollte die Einstellung der Puffergröße des Input-/Output-Stream und von TcpNoDelay unterstützen.
6820585	'imqcmd list txn' zeigt keine Cluster-Transaktionen mit dem Status COMMITTED an, die auf Abschluss durch Remote-Broker warten.
6820588	Eine Cluster-Transaktion, die sowohl lokale als auch Remote-Nachrichten konsumiert, bleibt im Wartestatus COMMITTED.
6821639	NPE bei Rollback-/Übernahmetransaktion während AS-Wiederherstellung für MQRA-DIRECT-Modus.
6823364	RFE: C-API-Compiler für Sun Studio 12 bei Solaris aktualisieren
6829113	ConcurrentModificationException bei Transaktionszeitüberschreitung bei Tuxedo TM-Rollback bei starker Belastung
6832197	Nicht transaktionale Remote-Bestätigung sollte nicht auf Remote-Antwort warten, wenn der Client nicht nach Bestätigung der Bestätigung fragt.
6834735	Protokoll-Nachricht "Unerwarteter interner Broker-Fehler", wenn Tuxedo TM im START-Status für eine Transaktion eine Zeitüberschreitung feststellt
6836364	Platzhalter-Abonnent empfängt Remote-Nachricht nicht, wenn deren Thema vor dem Abonnenten erstellt wurde.

TABELLE 1-8 Behobene Probleme in Message Queue 4.4 (Fortsetzung)	
Fehler	Beschreibung
6836691	HA(JCAPS):Nachricht wurde bereits entfernt Ausnahmefehler bei Erhalt nach XA Receiver-Rollback, dann Nachricht übermitteln
6836749	HA(JCAPS):ack in Speicherausnahme bei Empfang nach Rollback eines dauerhaften Abonnements und anschließender Übergabe einer Nachricht.
6837671	HA(JCAPS):endlose erneute Sendung einer übermittelten Nachricht bei XAResourceImpl.rollback nach erfolgreicher Übermittlung
6839193	RFE: C++-Compiler auf Visual Studio 2008 SP1 aktualisieren
6845625	Broker hat zu wenig Speicherplatz zur Verfügung, wenn Remote-Konsumenten wiederholt erstellt/geschlossen werden.
6852207	NPE beim Senden der Nachricht an Remote-Broker bewirkt, dass beim Lesen von msg pkt der Fehler "Nachricht kann nicht verarbeitet werden" auftritt.
6853822	Ausnahmefehler "Vorgang END_TRANSACTION kann nicht durchgeführt werden" beim Beenden einer Transaktion mit Status FAILED.
6854142	"Warten auf Cluster-Verbindung" "Cluster-Verbindung geschlossen" alle 3 Minuten an den Remote-Broker
6858121	WARNUNG 'Unbekannte Transaktion' in Broker-Protokoll auf 'imqcmd list txn', wenn Remote-Transaktion vorhanden ist
6858488	Transaktion mit Status COMMITTED nicht vom Transaktions-Startbroker entfernt, wenn Remote-Teilnehmer-Broker seine Transaktion mit Status COMMITTED entfernt hat.
6858905	ConcurrentModificationException für Consumer.destroyConsumer.
6861362	RFE: JMSBridge: automatische Zuordnung des Ziels zur Quelle Message.getJMSDestination unterstützen
6861528	RFE: JMSBridge: Weiterleitung der Nachricht an einen anderen Speicherort im Ziel MessageTransformer.transform() zulassen
6861653	Info übermäßige Cluster-Transaktion an COMMIT unvollständig Remote-Broker heruntergefahren wegen hohem Transaktionsaufkommen
6862413	Verwirrende Protokoll-Nachricht "mq://xxx.xxx.xx.xx:pppp/ ..." ist innerhalb von 60 Sekunden verfügbar"
6863867	MissingResourceException auf HA-Broker startet neu, wenn COMMITTED von einem heruntergefahrenen Remote-Broker aussteht.
6867596	Transaktion mit Status PREPARED wieder hergestellt nach Broker-Neustart zurückkehren zu Status PREPARED, wenn Broker erneut neu startet
6868525	NullPointerException beim Weiterleiten des temporären Ziels an Remote-Broker bei Link-Erstellung

TABELLE 1-8 Behobene Probleme in Message Queue 4.4 (Fortsetzung)

Fehler	Beschreibung
6868578	Manche broadcast/unicast-Elemente prüfen nicht, ob eine hergestellte Verknüpfung einen Konflikt mit Link-Handshake aufweist und zu einer Unterbrechung der Verknüpfung führt.
6871612	HA-Protokoll-Nachrichten des Typs ?Kann transaction.completion nicht melden“beim Konsumieren von Remote-Nachrichten, wenn ausstehender Broker heruntergefahren
6886391	NullPointerException beim Bestätigen der Nachricht, wenn Nachricht bereits entfernt wurde

In Message Queue 4.3 behobene Fehler

In der folgenden Tabelle sind die in Message Queue 4.3 behobenen Fehler aufgelistet.

TABELLE 1-9 In Message Queue 4 behobene Fehler 3

Fehler	Beschreibung
6634033	Cluster-Protokoll gibt den Wert von <code>imqConsumerFlowLimit</code> nicht an Remote-Broker weiter, wenn ein Client erstellt wird.
6713012	Wenn das Löschen eines Konsumenten auf einem Broker in einem Cluster und der Neustart eines Remote-Brokers auf den gleichen Zeitpunkt fallen, kann dies dazu führen, dass einige Nachrichten nicht zugestellt werden.
6727555	In der Broker-Protokollmeldung "Max bytes per msg exceeded" (Maximale Bytezahl pro Nachricht überschritten) sind die tatsächliche Größe der Nachricht und der Wert für max bytes per message vertauscht.
6737404	Die JMX-Metrik muss die Anzahl der Nachrichten angeben, die von den Zielen (Themen und Warteschlangen) gesendet wurden, jedoch den Konsumenten noch zugestellt werden müssen.
6740568	Broker gibt einen Ausnahmefehler aus, wenn zu viele Nachrichten in einer einzigen Transaktion konsumiert werden.
6758524	Der Befehl zum Auflisten der dauerhaften Abonnements (<code>imqcmd list dur -d "foo.*"</code>) akzeptiert keine Platzhalterzeichen im Zielnamen.
6758952	Die Einstellung <code>imq.portmapper.hostname=localhost</code> führt dazu, dass Broker keine Verbindung zu einem Cluster herstellen können.
6758817	Die Einstellung <code>imq.cluster.hostname=localhost</code> (nicht empfohlen) führt dazu, dass Broker auf verschiedenen Geräten keine Verbindung zu einem Cluster herstellen können.

In Message Queue 4.2 behobene Fehler

In der folgenden Tabelle sind die in Message Queue 4.2 behobenen Fehler aufgelistet.

TABELLE 1–10 In Message Queue 4.2 behobene Fehler

Fehler	Beschreibung
6581592	Wenn das Installations- bzw. Deinstallationsprogramm im Textmodus ausgeführt wird (<code>installer -t</code>), wird auf der Zusammenfassungsseite zwar das Verzeichnis mit den Protokoll-/Zusammenfassungsdateien angezeigt, die Namen dieser Dateien werden jedoch nicht aufgelistet.
6585911	Auf dem Bildschirm zur JDK-Auswahl des Installationsprogramms ist die JRE fälschlicherweise mit dem Installationsprogramm gebündelt und wird zur Ausführung des Installationsprogramms verwendet.
6587112	Auf der Zusammenfassungsseite des Installationsprogramms werden bei Multi-Byte-Gebietsschemata bedeutungslose Zeichen angezeigt.
6587127	Wenn bei der Ausführung des Installationsprogramms durch Verweis auf eine Antwortdatei (<code>installer -a filename -s</code>) die Antwortdatei nicht vorhanden ist, sind die Fehlermeldungen inkonsistent und unklar.
6590969	Lässt das DN-Benutzernamensformat bei der Client-Verbindungsauthentifizierung zu.
6594381	Installation von Message Queue 4.1-Lokalisierungs-RPMs (erfolgt bei Aktivierung des Kontrollkästchens "Mehrsprachige Message Queue-Pakete installieren+++ " im Bildschirm Mehrsprachige Pakete+++) schlägt fehl, wenn ältere Versionen von Message Queue-Lokalisierungs-RPMs im System vorhanden sind.
6599144	Bei der Deinstallation von Message Queue 4.2 reagieren bei Java SE 6 der Begrüßungsbildschirm und das Deinstallationsprogramm nicht mehr und die Bildschirme erscheinen leer und grau, unter Java SE 5 funktionieren sie jedoch.
6615741	Nachrichten, die in einer transaktionalen Konsumentensitzung zugestellt werden, die in einem Rollback zurückgesetzt wird werden nicht erneut zugestellt, wenn der ursprüngliche Konsument die Sitzung vor dem Rollback geschlossen hat.
6629922	Behandlungsroutine für verteilte Transaktionen stellt Nachricht an inaktiven Konsumenten nicht in richtiger Reihenfolge erneut zu.
6635130	Broker informiert Produzent nicht über nicht-persistente Nachrichten, die die Produktion wieder aufnehmen, nachdem eine Pause erfolgte, da da beim Ziel Grenzwerte für Arbeitsspeicher oder Nachrichten erreicht wurden.
6641117	Nachrichten, die in einer transaktionalen Konsumentensitzung zugestellt werden, die in einem Rollback zurückgesetzt wird werden nicht erneut zugestellt, wenn der ursprüngliche Konsument die Sitzung nach dem Rollback geschlossen hat.
6683897	Der Zusammenfassungsbildschirm des Message Queue-Installationsprogramms meldet in einen Konfigurationsfehler, dass das Installationsprogramm auf einigen Computern nicht in <code>/dev/sterr</code> schreiben kann, obwohl die Konfiguration erfolgreich durchgeführt worden zu sein scheint.

TABELLE 1–10 In Message Queue 4.2 behobene Fehler (Fortsetzung)

Fehler	Beschreibung
6684069	In Broker-Clustern, bei denen eine große Anzahl an Nachrichten in einer Konsumententransaktion an einen Remote-Client zugestellt werden, schlägt die Übernahmetransaktion (Commit) fehl.
6688935	Standardwert für Portmapper-Lesezeitüberschreitung ist zu klein.
6695238	C-Client-Anwendungen können keine Verbindung zu einem Broker herstellen, der an einem Speicherort installiert ist, dessen Pfad Leerzeichen enthält.
6710168	Konsument konsumiert keine Nachrichten mehr, wenn das Ziel zweimal angehalten wird, ohne dass eine Wiederaufnahme zwischen den Pausen erfolgt.
6710169	JMX-Vorgang <code>ConsumerManagerMonitor.getConsumerInfo</code> gibt als Bestätigungsmodus stets <code>SESSION_TRANSACTED</code> aus.

In Message Queue 4.1 behobene Fehler

In der folgenden Tabelle sind die in Message Queue 4.1 behobenen Fehler aufgelistet.

TABELLE 1–11 In Message Queue 4.1 behobene Fehler

Fehler	Beschreibung
6381703	Transaktionale Remote-Nachrichten können zweifach übermittelt werden, wenn der Broker neu gestartet wird, von dem diese Nachrichten stammen.
6388049	Nicht abgeschlossene verteilte Transaktion kann nicht bereinigt werden.
6401169	Für die Übergabe- und Rollback-Optionen für <code>imqcmd</code> wird keine Bestätigungsaufforderung angezeigt.
6473052	Standard für automatisch erstellte Warteschlangen sollte Round Robin sein. (<code>MaxNumberConsumers = -1</code>).
6474990	Broker-Protokoll zeigt <code>ConcurrentModificationException</code> für <code>imqcmd list dst</code> -Befehl.
6487413	Speicherleck, wenn Verhalten bei Begrenzungen <code>REMOVE_OLDEST</code> oder <code>REMOVE_LOWER_PRIORITY</code> ist.
6488340	Broker reagiert nicht, und Client wartet auf Antwort zur Bestätigung.
6502744	Broker beachtet nicht das Standardlimit von 1000 Nachrichten in der Warteschlange für nicht zugestellte Nachrichten.
6517341	Die Client-Laufzeit muss die Logik für die Verbindungswiederherstellung verbessern, sobald der Client mit einem erweiterten Broker-Cluster verbunden ist, indem dem Client ermöglicht wird, die Verbindung unabhängig vom Wert der Eigenschaft <code>imqReconnectEnabled</code> wiederherzustellen.

TABELLE 1-11 In Message Queue 4.1 behobene Fehler (Fortsetzung)

Fehler	Beschreibung
6528736	Automatischer Windows-Startdienst (<i>imqbrokersvc</i>) stürzt während des Startvorgangs ab.
6561494	Nachrichten werden an den falschen Konsumenten zugestellt, wenn beide eine Sitzung teilen.
6567439	Produzierte Nachrichten in einer PREPARED-Transaktion werden nicht ordnungsgemäß zugestellt, wenn diese nach dem Neustart des Brokers gesendet werden.

Behobene Probleme in Message Queue 4.0

In der folgenden Tabelle sind die in Message Queue 4.0 behobenen Probleme aufgelistet.

TABELLE 1-12 Behobene Probleme in Message Queue 4.0

Fehlernummer	Beschreibung
4986481	In Message Queue 3.5 kann sich die aufgerufene <code>Session.recover</code> -Methode im Modus für die automatische Neuverbindung aufhängen.
4987325	Erneut versendetes Flag war für die neu versendeten Nachrichten auf <code>false</code> gesetzt, nachdem die <code>Session.recover</code> -Methode aufgerufen wurde.
6157073	Ändern der neuen Verbindungsmeldung, um die Anzahl an Verbindungen mit dem Dienst zusätzlich zur Gesamtzahl der Verbindungen hinzuzufügen.
6193884	Message Queue gibt unleserliche Nachrichten im <code>syslog</code> von Länderinformationen aus, die für Nachrichten keine ASCII-Zeichen verwenden.
6196233	Nachrichtenauswahl mithilfe von <code>JMSMessageID</code> funktioniert nicht.
6251450	<code>ConcurrentModificationException</code> für <code>connectList</code> während Clusterbeendigung.
6252763	<code>java.nio.BufferOverflowException</code> in <code>java.nio.HeapByteBuffer.putLong/Int</code> .
6260076	Erste Nachrichtenveröffentlichung nach Start bei Verwendung von Oracle-Speicher erfolgt langsam.
6260814	Auswahlverarbeitung auf <code>JMSXUserID</code> wird immer als <code>false</code> ausgewertet.
6264003	Der Warteschlangenbrowser zeigt Nachrichten an, die Bestandteil von Transaktionen sind, die noch nicht verarbeitet wurden.
6271876	Verbindungsdatensteuerung arbeitet nicht ordnungsgemäß, wenn ein Verbraucher mit nicht verarbeiteten Nachrichten geschlossen wird.
6279833	Message Queue sollte nicht zulassen, dass zwei Broker dieselben JDBC-Tabellen verwenden.
6293053	Master-Broker wird nicht ordnungsgemäß gestartet, wenn die IP-Adresse des Systems geändert wird, es sei denn, der Speicher wurde geleert (mithilfe von <code>-reset store</code>).

TABELLE 1–12 Behobene Probleme in Message Queue 4.0 (Fortsetzung)

Fehlernummer	Beschreibung
6294767	Message Queue-Broker muss <code>SO_REUSEADDR</code> auf dem Netzwerksocket setzen, der geöffnet wird.
6304949	<code>ClientID</code> -Eigenschaft für <code>TopicConnectionFactory</code> kann nicht festgelegt werden.
6307056	<code>Dastxn</code> -Protokoll führt zu einem Leistungsengpass.
6320138	Message Queue C-API kann den Namen einer Warteschlange für einen <code>Reply-To-Header</code> nicht ermitteln.
6320325	Der Broker wählt unter Solaris gelegentlich JDK 1.4 anstelle JDK 1.5, selbst wenn beide Versionen installiert sind.
6321117	Die Initialisierung eines Multibroker-Clusters führt zu einer <code>java.lang.NullPointerException</code> .
6330053	Der JMS-Client gibt <code>java.lang.NoClassDefFoundError</code> zurück, wenn eine Transaktion vom Abonnenten übergeben wird.
6340250	Unterstützung für <code>MESSAGE</code> -Typ in C-API.
6351293	Unterstützung für Apache Derby-Datenbank hinzufügen.

Dokumentationsaktualisierungen in Message Queue 4.4 Update 1

Dieser Abschnitt enthält Informationen zu Dokumentationsaktualisierungen in Message Queue 4.4 Update 1:

- „Kompatibilitätsprobleme“ auf Seite 41
- „Änderungen am Message Queue 4.4 Update 1-Dokumentationssatz“ auf Seite 42

Kompatibilitätsprobleme

In diesem Abschnitt werden bekannte Kompatibilitätsprobleme in Bezug auf Message Queue 4.4 Update 1 beschrieben.

Schnittstellenstabilität

Sun GlassFish Message Queue verwendet zahlreiche Schnittstellen, die sich mit der Zeit ändern können. Eine Klassifizierung der Schnittstellen nach ihrer Stabilität finden Sie in [Anhang B, „Stability of Message Queue Interfaces“](#) in *Sun GlassFish Message Queue 4.4 Administration Guide*. Je stabiler eine Schnittstelle, desto weniger wahrscheinlich ist eine Änderung in den nachfolgenden Produktversionen.

Mögliche Probleme hinsichtlich der nächsten Hauptversion von Message Queue

Die nächste Hauptversion von Message Queue umfasst eventuell Änderungen, die eine Inkompatibilität mit den aktuellen Message Queue-Client-Anwendungen verursachen können. Diese Informationen werden im Interesse einer vollständigen Offenlegung angegeben.

- Die Speicherorte einzelner Dateien, die als Bestandteil von Sun GlassFish Message Queue installiert werden, kann sich ändern. Dadurch können in vorhandenen Anwendungen Fehler auftreten, die vom aktuellen Speicherort bestimmter Message Queue-Dateien abhängen.
- Broker der Version Message Queue 3.5 oder niedrigere Versionen können in einem Cluster mit neueren Brokern möglicherweise nicht mehr eingesetzt werden.
- In zukünftigen Versionen ist die Verwendung von früheren JDK-Versionen als 1.5 in Message Queue-Clients eventuell nicht mehr möglich.
- In zukünftigen Versionen ist die Verwendung von früheren JDK-Versionen als 1.6 in Message Queue-Clients eventuell nicht mehr möglich.

Änderungen am Message Queue 4.4 Update 1-Dokumentationssatz

Der Message Queue 4.4 Update 1-Dokumentationssatz enthält Aktualisierungen für den Message Queue 4.4-Dokumentationssatz, wie unten beschrieben:

Technischer Überblick

Im *Sun GlassFish Message Queue 4.4 Technical Overview* werden neue Funktionen in Message Queue 4.4 beschrieben.

Administration Guide

Der *Sun GlassFish Message Queue 4.4 Administration Guide* enthält Korrekturen geringfügiger Fehler und beschreibt neue Funktionen in Message Queue 4.4.

Developer's Guide for Java Clients

Der *Sun GlassFish Message Queue 4.4 Developer's Guide for Java Clients* enthält Korrekturen geringfügiger Fehler.

Developer's Guide for C Clients

Der *Sun GlassFish Message Queue 4.4 Developer's Guide for C Clients* enthält Korrekturen geringfügiger Fehler.

Developer's Guide for JMX Clients

Der *Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients* enthält Produktnamen und Versionsnummer von Message Queue.

Bekannte Probleme und Beschränkungen

In diesem Abschnitt werden die bekannten Probleme mit Message Queue 4.4 Update 1 aufgeführt. Die folgenden Produktbereiche werden besprochen:

- „Installationsprobleme“ auf Seite 43
- „Veraltete Passwortoption“ auf Seite 50
- „Administration/Konfiguration“ auf Seite 51
- „Broker-Probleme“ auf Seite 52
- „Broker-Cluster“ auf Seite 54
- „JMX-Probleme“ auf Seite 56
- „SOAP-Unterstützung“ auf Seite 57

Eine Liste der aktuellen Fehler, deren Status und Umgehungsmöglichkeiten finden Sie als Mitglied der Java Developer Connection™ in der Bug Parade auf der Java Developer Connection-Website. Prüfen Sie die Informationen auf dieser Seite, bevor Sie einen neuen Fehler melden. Auch wenn nicht alle Message Queue-Fehler aufgelistet sind, ist diese Seite ein guter Ausgangspunkt, wenn Sie feststellen möchten, ob ein Problem bekannt gegeben wurde.

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

Hinweis – Die Mitgliedschaft bei der Java Developer Connection ist kostenlos, es ist jedoch eine Registrierung erforderlich. Auf der Sun Java-Webseite wird beschrieben, wie Sie Mitglied bei der Java Developer Connection werden.

Wenn Sie einen neuen Fehler melden oder eine Funktionsanfrage einreichen möchten, senden Sie eine E-Mail an imq-feedback@sun.com.

Installationsprobleme

In diesem Abschnitt werden die Installationsprobleme zu Message Queue Version 4.4 Update 1 beschrieben.

Produktregistrierung und Java ES

Message Queue 4.4 wird wie Message Queue 4.2 und 4.1 von einem relativ neuen Installationsprogramm installiert, das auch zur Installation und Aktualisierung der gemeinsam genutzten Komponenten von Java Enterprise System (Java ES) dient, die von Message Queue benötigt werden, z. B. JDK, NSS, JavaHelp usw.

Das neue Message Queue-Installationsprogramm und das ältere Java ES-Installationsprogramm, das zur Installation früherer Message Queue-Versionen verwendet wurden, verwenden nicht dieselbe Produktregistrierung. Wenn eine Message Queue-Version, die mit dem Java ES-Installationsprogramm installiert wurde, entfernt und dann Message Queue 4.4 mithilfe des Message Queue-Installationsprogramms installiert wird, ist der Status der Java ES-Produktregistrierung möglicherweise inkonsistent. Dies hat zur Folge, dass beim Ausführen des Java ES-Deinstallationsprogramms eventuell Message Queue 4.4 und die benötigten gemeinsamen Komponenten versehentlich entfernt werden, obwohl sie nicht von diesem Programm installiert wurden.

Zur Aktualisierung der Message Queue-Software, die mit dem JES-Installationsprogramm installiert wurde, gehen Sie am besten wie folgt vor.

1. Entfernen Sie Message Queue sowie die gemeinsam genutzten Komponenten mit dem Java ES-Deinstallationsprogramm.
2. Installieren Sie Message Queue 4.4 mit dem Message Queue-Installationsprogramm.

Installation auf allen Plattformen

Diese Probleme betreffen die Installation auf allen Plattformen.

- Im Bildschirm für die Installationsbereitschaft wird der Produktname als "mq" und nicht als Sun Java System Message Queue 4.3 angezeigt. (*Fehler 6650841*)
- Solange das Installationsprogramm die Installation von Message Queue 4.3 ausführt und die Seite mit der Fortschrittsanzeige angezeigt wird, ist die Schaltfläche "Abbrechen" aktiv. Das Klicken auf die Schaltfläche "Abbrechen" führt zu einer unvollständigen oder abgebrochenen Installation. (*Fehler 6595578*)
- Auf der Installationsübersichtsseite werden mehrere Verknüpfungen angezeigt, über die per Mausklick ein Viewer für Protokolle oder Übersichtsseiten angezeigt werden kann. Wenn Sie dieses Viewer-Fenster über die Schaltfläche "X" anstatt über die Schaltfläche "Schließen" schließen, kann dieses Viewer-Fenster nicht erneut geöffnet werden. (*Fehler 6587138*)

Umgehung: Verwenden Sie die Schaltfläche "Schließen", um das Fenster zu schließen.

- Wenn Sie das Installationsprogramm nach einer automatischen Installation ohne Registrierung im Modus nur zur Registrierung (`installer -r`) ausführen, wird eine Fehlermeldung über ein vorzeitiges Dateiende ausgegeben und die Registrierung schlägt fehl. (*Fehler 6767988*)
- Wenn Sie das Installationsprogramm für eine Message Queue-Installation auf einem Computer ausführen, auf dem kein JDK installiert ist, wird die folgende Fehlermeldung angezeigt: "Invalid root in registry key 'HKLM\SOFTWARE\JavaSoft\Java Runtime Environment\CurrentVersion'." (Ungültiger Stamm in Registrierungsschlüssel "HKLM\SOFTWARE\JavaSoft\Java Runtime Environment\CurrentVersion"). (*Fehler 6764358*)

Umgehung: Installieren Sie das JDK, bevor Sie das Installationsprogramm ausführen.

- Das Verzeichnis `mqInstallHome` wird vom Message Queue-Installationsprogramm installiert, bevor Sie im Bildschirm für die Installationsbereitschaft auf die Schaltfläche zum Installieren klicken. (Fehler 6595590)

Installation unter Windows

Wenn Sie Message Queue unter Windows installieren, sollten Sie die folgenden Einschränkungen beachten.

- Die installierte Verzeichnisstruktur von Message Queue 4.3 auf Windows-Plattformen unterscheidet sich von derjenigen vorheriger Versionen. Weitere Informationen finden Sie in „Installed Directory Structure“ im *Sun Java System Message Queue 4.3 Installation Guide*.
- Das Installationsprogramm fügt unter Start > Programme keine neuen Einträge für Message Queue hinzu. (Fehler 6567258)

Umgehung: Um die Administrationskonsole zu starten, verwenden Sie die Befehlszeile wie in „Starting the Administration Console“ in *Sun GlassFish Message Queue 4.4 Administration Guide* beschrieben.

- Das Installationsprogramm fügt dem Verzeichnis `IMQ_HOME\mq\bin` keine PATH-Umgebungsvariable hinzu. (Fehler 6567197)
Umgehung: Benutzer müssen entweder diesen Eintrag zur PATH-Umgebungsvariable hinzufügen oder einen vollständigen Pfadnamen angeben, wenn sie die Message Queue-Dienstprogramme (`IMQ_HOME\mq\bin\Befehl`) aufgerufen.
- Das Installationsprogramm fügt keine Einträge zur Windows-Registrierung hinzu, die darauf hinweisen, dass Message Queue installiert ist. (Fehler 6586389)
- Das Installationsprogramm fügt den Message Queue-Broker nicht als Windows-Dienst hinzu.
Umgehung: Fügen Sie den Message Queue-Broker manuell mit dem Befehl `imqsvcadm` als Windows-Dienst hinzu.
- Wenn kein JDK installiert ist, gibt das Installationsprogramm folgenden Fehler aus: "Invalid root in registry key HKLM\SOFTWARE\JavaSoft\Java Runtime Environment\CurrentVersion" (Ungültiger Stamm in Registrierungsschlüssel HKLM\SOFTWARE\JavaSoft\Java Runtime Environment\CurrentVersion). (Fehler 6764358)\par
Umgehung: Installieren Sie bei diesem Fehler ein JDK, und fahren Sie fort.
- Bei Installation im automatischen Modus mit einer Antwortdatei wird das Installationsprogramm sofort wieder angezeigt. Die Installation wird zwar durchgeführt, jedoch weiß der Benutzer nicht, wann die automatische Installation tatsächlich abgeschlossen ist. (Fehler 6586560)
- Das Installationsprogramm installiert Message Queue auf `C:\`, obwohl das Betriebssystem auf einem anderen Laufwerk installiert ist. (Fehler 6673511)

- Für die Installation und Deinstallation unter Windows gibt es keine .bat-Dateien, die der Benutzer ausführen kann, und der Benutzer kann auch die Deinstallation nicht über die Option Software in der Windows-Systemsteuerung deinstallieren. (*Fehler 6673417*)
- Unter Windows Vista können Sie Message Queue nur im Verzeichnis C:\Programme installieren, wenn Sie die Installation mit Administratorrechten über die Eingabeaufforderung ausführen. (*Fehler 6701661*)

Umgehung: So führen Sie die Installation als Administrator über eine Befehlszeile durch:

1. Start→Programme→Zubehör→Eingabeaufforderung.
 2. Klicken Sie mit der rechten Maustaste auf Eingabeaufforderung.
 3. Wählen Sie Als Administrator ausführen.
 4. Wechseln Sie in das Verzeichnis des Message Queue 4.2-Installationsimage.
 5. Führen Sie `installer.vbs` aus.
- Wenn sich das Deinstallation im Testlaufmodus (`uninstaller -n`) befindet, führt es fälschlicherweise eine Deinstallation durch. (*Fehler 6719051*)
Umgehung: Führen Sie eine automatische Installation mit folgendem Befehl durch:
`uninstaller -s`
 - Die Zeichenfolge "Install Home" auf der Startseite des Installationsprogramms ist nicht lokalisiert (*Fehler 6592491*)
 - Das Deinstallationsprogramm für eine zip-basierte Message Queue-Installation reagiert unter Windows 2003 nicht. (*Fehler 6764370*)
Umgehung: Entfernen Sie das Verzeichnis `mqInstallHome` manuell.

Installation unter Solaris

- Wenn das Installationsprogramm im Testlaufmodus (`installer -n`) ausgeführt wird, werden auf der Zusammenfassungsseite mehrere Fehlermeldungen angezeigt, und der Installationsstatus wird als nicht abgeschlossen angegeben. Dies ist falsch und irreführend. Bei einem Testlauf werden keine Komponenten auf dem System installiert, es wird lediglich die Antwortdatei erstellt, die anschließend für die automatische Installation verwendet werden kann. (*Fehler 6594351*)
- Das Installationsprogramm führt keine Sun Connection-Registrierung durch, wenn die Ausführung im automatischen Modus mit einer Antwortdatei (`installer -a filename -s`) erfolgt. (*Fehler 6710268*)

Installation unter Linux

Die folgenden Probleme betreffen die Installation auf einer Linux-Plattform:

- Unter Redhat Linux 5 ist die Bibliothek `compat-libstdc++`, die zum Ausführen von C-Client-Anwendungen benötigt wird, nicht in der Message Queue-Verteilung enthalten und wird daher nicht vom Message Queue-Installationsprogramm installiert. Wenn Sie C-Clients entwickeln und ausführen, müssen Sie diese Bibliothek manuell installieren.

Den `compat-libstdc++`-RPM finden Sie normalerweise auf dem Installationsdatenträger der verwendeten Linux-Version. Er kann über folgenden Befehl installiert werden:

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

Dabei entspricht `x` der Versionsnummer.

Mit folgendem Befehl überprüfen Sie, ob die Bibliothek erfolgreich installiert wurde:

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

- Unter Red Hat Linux 5 können C-Clients mit der Fehlermeldung `PR_LOAD_LIBRARY_ERROR` (*Fehler 6885978*) fehlschlagen.

Unter Red Hat Linux 5 können C-Clients fehlschlagen. Es wird in etwa folgende Nachricht angezeigt:

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

Dieser Fehler tritt auf, weil kein Zugriff auf die NSS/NSPR-Bibliotheken besteht.

Um dieses Problem zu beheben, stellen Sie die Umgebungsvariable `LD_LIBRARY_PATH` so ein, dass sie den Pfad zu den NSS/NSPR-Bibliotheken enthält: `IMQ_HOME/nss/lib`.

- Auf dem Bildschirm zur JDK-Auswahl wird in der Scroll-Liste nur ein Element angezeigt. Dies macht es schwierig, weitere JDK in der Liste auszuwählen. (*Fehler 6584735*)
- Wenn das JDK aktuell ist, und der Benutzer auf der Seite für die JDK-Auswahl die Option zum Installieren des Standard-JDK wählt, fährt das Installationsprogramm mit den Installationsversuchen fort und gibt anschließend die Meldung aus, dass das Paket nicht installiert werden kann. Die Installation wird trotz dieses Problems erfolgreich abgeschlossen. (*Fehler 6581310*)
- Wenn das aktuell installierte JDK eine spätere Version aufweist als `JDK 1.5.0_15` (die normalerweise vom Message Queue-Installationsprogramm installierte Version), kann das Message Queue-Deinstallationsprogramm das standardmäßige `IMQ_JAVAHOME`-Verzeichnis nicht finden und gibt einen Fehler zurück. (*Fehler 6673415*)

Umgehung: Installieren Sie JDK 1.5 wie folgt manuell, bevor Sie das Message Queue-Deinstallationsprogramm ausführen.

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

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

Dabei ist `arch` entweder `i586` oder `amd64`.

- Wenn das Installationsprogramm im Testlaufmodus (`installer -n`) ausgeführt wird, werden auf der Zusammenfassungsseite mehrere Fehlermeldungen angezeigt, und der Installationsstatus wird als nicht abgeschlossen angegeben. Dies ist falsch und irreführend. Bei einem Testlauf werden keine Komponenten auf dem System installiert, es wird lediglich die Antwortdatei erstellt, die anschließend für die automatische Installation verwendet werden kann. (*Fehler 6594351*)

Versionsanomalien im Installationsprogramm

Die Anzeige der Message Queue-Versionsinformationen im Installationsprogramm ist nicht transparent. (*Fehler 6586507*)

Solaris-Plattform

Bei der Solaris-Plattform können Sie aus der folgenden Tabelle die vom Installationsprogramm angezeigte Message Queue-Version bestimmen.

TABELLE 1-13 Übersetzung der Versionszeichenfolge

Vom Installation unter Solaris OS angezeigte Version	Entsprechender Message Queue-Release
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

Hinweis – Für Patch-Versionen für 3.6 SP4 (z. B. 3.6 SP4 Patch 1) werden im Installationsprogramm dieselben Zeichenfolgen für die Versionen angezeigt. Sie müssen den Befehl `imqbrokerd -version` ausführen, um die genaue Version zu bestimmen.

Linux-Plattform

Unter der Linux-Plattform weist die vom Installationsprogramm angezeigte Versionsnummer folgendes Format auf.

majorReleaseNumber . minorReleaseNumber - someNumber

Zum Beispiel könnte sie 3.7–22 lauten. Dies gibt lediglich an, dass es sich um eine der 3.7-Versionen handelt, jedoch nicht, um welche genau. Zur Ermittlung der installierten Message Queue-Version müssen Sie folgenden Befehl ausführen:

```
imqbrokerd -version
```

Lokalisierungsprobleme

Die folgenden Probleme beziehen sich auf die Lokalisierung.

- Wenn das Installationsprogramm im Textmodus ausgeführt wird (`installer -t`), werden bei anderen Gebietsschemata als Englisch Multibyte-Zeichen unleserlich angezeigt. (*Fehler 6586923*)
- Auf der Seite mit dem Installationsfortschritt zeigt der Fortschrittbalken merkwürdige Zeichen an. Die QuickInfo bei anderen Gebietsschemata als Englisch ist hartcodiert. (*Fehler 6591632*)
- Auf der Lizenzvereinbarungsseite des Installationsprogramms wird die Lizenzvereinbarung auf Englisch angezeigt, unabhängig davon, mit welchem Gebietsschema das Installationsprogramm ausgeführt wird. (*Fehler 6592399*)
Umgehung: Informationen für den Zugriff auf die lokalisierten Lizenzdateien finden Sie in der Datei `LICENSE_MULTILANGUAGE.pdf`.
- Der Hilfetext zur Verwendung des Installationsprogramms ist nicht lokalisiert. (*Fehler 6592493*)
- Die Zeichenfolge "None" auf der HTML-Zusammenfassungsseite des Installationsprogramms ist in englischer Sprache hartcodiert. (*Fehler 6593089*)
- Wenn das Installationsprogramm mit dem deutschen Gebietsschema ausgeführt wird, wird auf der Begrüßungsseite nicht der vollständige Text wie bei anderen Gebietsschemata angezeigt. (*Fehler 6592666*)
- Die Zeichenfolge "Install Home" ist auf der entsprechenden Seite im Installationsprogramm nicht lokalisiert. Sie wird selbst dann auf Englisch angezeigt, wenn das Installationsprogramm mit einem anderen als dem englischen Gebietsschema ausgeführt wird. (*Fehler 6592491*)

- Wenn das Installationsprogramm im Textmodus ausgeführt wird (`installer -t`), lautet die englischen Antwortauswahl "Yes" und "No", unabhängig von dem Gebietsschema, in dem das Installationsprogramm ausgeführt wird. (*Fehler 6593230*)
- Auf der Installationsprogrammseite für die JDK-Auswahl ist die QuickInfo für die Schaltfläche zum Durchsuchen in der englischen Version hartcodiert. (*Fehler 6593085*)

Veraltete Passwortoption

In Vorgängerversionen von Message Queue konnten Sie die Option `-p` oder `-password` verwenden, um ein Passwort für die folgenden Befehle interaktiv anzugeben: `imqcmd`, `imqbrokerd` und `imdbmgr`. Mit Version 4.0 wurden diese Optionen verworfen.

Stattdessen können Sie eine Passwortdatei erstellen, die die relevanten Passwörter angibt, und mit der Befehlsoption `-passfile` auf die Passwortdatei verweisen oder einfach ein Passwort eingeben, wenn Sie von dem Befehl dazu aufgefordert werden.

Eine Passwortdatei kann mindestens eins der im Folgenden aufgelisteten Passwörter enthalten.

- Ein Schlüsselspeicherpasswort zum Öffnen des SSL-Schlüsselspeichers. Legen Sie dieses Passwort über die Eigenschaft `imq.keystore.password` fest.
- Ein LDAP-Repository-Passwort für die sichere Verbindung mit einem LDAP-Verzeichnis, wenn die Verbindung nicht anonym ist. Legen Sie dieses Passwort über die Eigenschaft `imq.user_repository.ldap.password` fest.
- Ein JDBC-Datenbankpasswort für die Verbindung zu einer JDBC-kompatiblen Datenbank. Legen Sie dieses Passwort über die Eigenschaft `imq.persist.jdbc.vendorName.password` fest. Die `vendorName`-Komponente des Eigenschaftsnamen ist eine Variable, die den Datenbankanbieter angibt. Zur Auswahl stehen `hadb`, `derby`, `pointbase`, `oracle` oder `mysql`.
- Ein Passwort für den `imqcmd`-Befehl (zum Ausführen von Broker-Administrationsaufgaben). Legen Sie dieses Passwort über die Eigenschaft `imq.imqcmd.password` fest.

Im folgenden Beispiel ist in der Passwortdatei als Passwort für die JDBC-Datenbank `abracadabra` festgelegt.

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

Passwortdateien können auf folgende Weisen verwendet werden:

- Konfigurieren Sie den Broker zur Verwendung der Passwortdatei, indem Sie folgende Eigenschaften in der Datei `config.properties` des Brokers festlegen.

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

- Verwenden Sie die Option `-passfile` des relevanten Befehls, beispielsweise:
`mqbrokerd -passfile Passwortdateiname`

Administration/Konfiguration

Die folgenden Probleme beziehen sich auf die Administration und Konfiguration von Message Queue.

- Auf Windows-Plattformen müssen Sie den Message Queue-Broker manuell mit dem Befehl `mqsvcadm` als Windows-Dienst hinzufügen. Dieser Vorgang wird nicht vom Installationsprogramm ausgeführt.
- Bei Windows-Plattformen muss die integrierte Windows-Firewall, die standardmäßig aktiviert ist, mit einer Firewall-Regel dahingehend manuell konfiguriert werden, dass der Broker eingehende Verbindungen von Clients akzeptiert. (*Fehler 6675595*)
 1. Doppelklicken Sie in der Systemsteuerung auf die Windows-Firewall
 Sie müssen im Dialogfeld für die Benutzerkontensteuerung auf Fortsetzen klicken, um das Einstellungsdialogfeld der Windows-Firewall zu öffnen.
 2. Klicken Sie im Einstellungsdialogfeld der Windows-Firewall auf die Registerkarte Ausnahmen.
 3. Klicken Sie auf Programm hinzufügen.
 4. Wählen Sie im Dialogfeld Programm hinzufügen den Eintrag `java.exe` aus und klicken Sie auf Durchsuchen.
 Windows identifiziert den Broker-Prozess als Java Platform SE-Binärdatei. Suchen Sie daher die Datei `java.exe`, die vom Broker verwendet wird (normalerweise unter `jdk1.5.0_15\jre\bin\java.exe`).
 5. Klicken Sie auf Bereich ändern.
 6. Wählen Sie im Dialogfeld Bereich ändern folgende Option aus: "Alle Computer (einschließlich der im Internet)".
 7. Klicken Sie auf OK.
 8. Klicken Sie im Dialogfeld Programm hinzufügen auf OK.
 9. Klicken Sie im Einstellungsdialogfeld der Windows-Firewall auf OK.
- Auf Windows-Plattformen lösen die Befehle `mqadmin` und `mqobjmgr` einen Fehler aus, wenn der `CLASSPATH` doppelte Anführungszeichen enthält. (*Fehler 5060769*)
Umgehung: Öffnen Sie ein Fenster mit einer Eingabeaufforderung und heben Sie die Festlegung von `CLASSPATH` auf:
`set classpath=`
 Führen Sie anschließend den gewünschten Befehl im selben Eingabeaufforderungsfenster aus, beispielsweise:

`mqInstallHome\mq\bin\mqadmin`

- Die Option `-javahome` in Solaris- und Windows-Skripts (alle Versionen) funktioniert nicht, wenn der bereitgestellte Wert ein Leerzeichen enthält. (*Fehler 4683029*)

Die Option `javahome` wird von den Message Queue-Befehlen und -Programmen verwendet, um eine alternative Java 2-kompatible Runtime anzugeben. Der Pfadname zur alternativen Java-Runtime darf jedoch keine Leerzeichen enthalten. Nachfolgend werden einige Beispiele für Pfade mit Leerzeichen genannt:

Windows: `C:\jdk 1.4`

Solaris: `/work/java 1.4`

Umgehung: Installieren Sie die Java-Runtime an einem Speicherort oder unter einem Pfad, der keine Leerzeichen enthält.

- Das Attribut `mqQueueBrowserMaxMessagesPerRetrieve` legt die maximale Anzahl an Nachrichten fest, die von der Client-Runtime in einem Schritt abgerufen werden können, wenn die Inhalte eines Warteschlangenziels durchsucht werden. Das Attribut beeinflusst, wie die Nachrichten in der Warteschlange gestapelt werden, um an die Client-Laufzeit zugestellt zu werden, es hat jedoch keine Auswirkungen auf die Gesamtzahl der durchsuchten Nachrichten. Das Attribut beeinflusst nur den Durchsuchungsmechanismus, nicht die Nachrichtenzustellung. (*Fehler 6387631*)
- Auf einer Linux-Plattform, die SELinux ausführt, schlägt der Update Center-Befehl `pkg fehl` (*Fehler 6892062*)

Umgehung: Dieser Fehler wird von einem bekannten Problem im Update Center verursacht (https://updatecenter2.dev.java.net/issues/show_bug.cgi?id=1211). Verwenden Sie den folgenden Befehl, damit `pkg` auf SELinux mit aktivierter Durchsetzung funktioniert:

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

Broker-Probleme

Die nachfolgend beschriebenen Probleme beziehen sich auf den Message Queue-Broker.

- Message Queue 4.4-Clients erhalten eine unklare Warnung beim Verbinden mit Message Queue 3.7-Brokern (*Fehler 6899886*)

Wenn ein Message Queue 4.4-Client eine Verbindung zu einem Message Queue- 3.7-Broker aufbaut, erhält der Client eine Warnung folgender Art:

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

Diese `bad version`-Warnung weist darauf hin, dass der Client auf einer niedrigeren Protokollebene erneut mit dem Broker verbunden werden sollte.

- Bei Verwendung eines JDBC-Datenspeichers wird das Datenbankpasswort in Klartext gespeichert (*Fehler 6691717*)

Umgehung: Sichern Sie die Passwortdatei, die das Datenbankpasswort enthält, wie unter „Password Files“ in *Sun GlassFish Message Queue 4.4 Administration Guide* beschrieben.

- Wenn der persistente Datenspeicher zu viele Zielstandorte öffnet, kann auf den Broker nicht mehr zugegriffen werden. (*Fehler 4953354*)

Umgehung: Diese Bedingung wird vom Broker verursacht, der das Deskriptor-Limit für die offenen Dateien im System erreicht. Unter Solaris und Linux erhöhen Sie das Dateideskriptor-Limit mit dem Befehl `ulimit`.

- Konsumenten verweisen, wenn ein Zielstandort vernichtet wird. (*Fehler 5060787*)

Aktive Konsumenten verweisen, wenn ein Zielstandort gelöscht wird. Ein verwaister Konsument erhält keine Meldungen mehr (auch dann nicht, wenn der Zielstandort neu erstellt wird).

- Wird ein JMS-Client bei Verwendung des HTTP-Verbindungsdiensts plötzlich beendet (z. B. über `Strg-C`), benötigt der Broker etwa eine Minute, bevor die Clientverbindung und alle damit zusammenhängenden Ressourcen freigegeben werden.

Wird innerhalb dieses Zeitraums eine weitere Instanz des Clients gestartet, die versucht, dieselbe Client-ID, Warteschlange oder dasselbe dauerhafte Abonnement zu verwenden, wird möglicherweise ein Ausnahmefehler "Client-ID wird bereits verwendet" ausgegeben. Dies stellt jedoch kein Problem dar, es handelt sich lediglich um eine Nebenwirkung des vorangehend beschriebenen Beendigungsvorgangs. Wenn der Client nach etwa einer Minute gestartet wird, sollte kein Fehler gemeldet werden.

- Bei Verwendung der MySQL-Datenbank für einen Datenspeicher wird bei der Speicherung von Nachrichten mit mehr als 1 MB ein SQL-Ausnahmefehler ausgegeben und gemeldet, dass das Paket für die Abfrage zu groß sei. (*Fehler 6682815*)

Umgehung: Starten Sie den MySQL-Server, wobei für die Option `--max_allowed_packet` mehr als dem Standardwert (1 MB) eingestellt sein muss. Verwenden Sie beispielsweise folgenden Wert:

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

- Bei Verwendung der MySQL-Datenbank für einen hochverfügbaren gemeinsam genutzten Datenspeicher wird ein Mechanismus benötigt, um das MySQL-Speichermodul als NDBCLUSTER zu konfigurieren. (*Fehler 6691394*)

Umgehung: Fügen Sie den folgenden Eigenschaftswert der Datei `config.properties` des Brokers hinzu (siehe „Enhanced Clusters: JDBC Configuration Properties“ in *Sun GlassFish Message Queue 4.4 Administration Guide*)

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

- Bei Verwendung des Oracle-Treibers 9i (JDBC 9.2.0.x) gibt der Broker den Ausnahmefehler "Failed to persist property..." (Keine Persistenz für Eigenschaft möglich) aus. (*Fehler 6626825*)

Umgehung: Verwenden Sie den Treiber 10g (JDBC 10.2.0.x) von Oracle, für den der Broker optimiert ist.

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

- Bei Verwendung der Java DB-Datenbank für einen Datenspeicher wird beim Speichern einer Nachricht die Fehlermeldung ausgegeben, dass innerhalb der angeforderten Zeit keine Sperre bezogen werden konnte (SQLException). (*Fehler 6691394*)

Umgehung: Fügen Sie den folgenden Eigenschaftswert zur `config.properties`-Datei des Brokers hinzu:

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

- Bei Verwendung von IBM JVM auf AIX hat der Broker manchmal scheinbar ohne Grund wenig oder kaum Speicherplatz zur Verfügung (*Fehler 6899526*)

Umgehung: Verwenden Sie die neueste Version von IBM JVM (Java Runtime 1.6.0 IBM Corporation oder höher) und stellen Sie die folgende IBM JVM GC-Option auf `imqbrokerd` ein:

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

Broker-Cluster

Folgende Probleme betreffen Broker-Cluster.

- High Availability Broker mit MySQL Cluster-Datenspeicher kann nicht neu starten, wenn er nicht ordnungsgemäß beendet wurde (*Fehler 6896877*)

Umgehung: Dieses Problem wird durch ein bekanntes Problem im MySQL Cluster verursacht (<http://bugs.mysql.com/bug.php?id=47955>). Dieses Problem wurde in den MySQL-Versionen 5.1.39-ndb-6.3.28, 5.1.39-ndb-7.0.9 und 5.1.39-ndb-7.1.0 behoben.

- In dieser Version werden lediglich vollständig verbundene Broker-Cluster unterstützt. Dies bedeutet, dass jeder Broker in einem Cluster direkt mit allen anderen Brokern im Cluster kommunizieren muss. Wenn Sie Broker mithilfe des Befehlszeilenarguments `imqbrokerd -cluster` zu einem konventionellen Cluster verbinden, müssen Sie darauf achten, dass alle Broker im Cluster enthalten sind.
- Wenn ein Client mit einem Broker in einem erweiterten Broker-Cluster verbunden ist, versucht die Client-Laufzeit, solange eine Verbindung herzustellen, bis sie Erfolg hat (der Wert des Verbindungsfactory-Attributs `imqAddressListIterations` wird ignoriert).
- Ein Client kann nur die Inhalte von Warteschlangen durchsuchen, die sich auf seinem Home-Broker befinden. Der Client kann weiterhin Meldungen an eine beliebige Warteschlangensenden und Meldungen von jeder beliebigen Warteschlange im Cluster konsumieren. Die Einschränkung betrifft nur das Durchsuchen der Warteschlange.
- In einem konventionellen Cluster, das Broker der Version 4.3 enthält, müssen alle Broker Version 3.5 oder höher aufweisen.

- Message Queue 4.3-, 4.2- und 4.1-Broker können standardmäßig nicht in einem Cluster mit Message Queue 3.7- oder 3.6-Brokern interagieren, da sich der Standardwert von `imq.autocreate.queue.maxNumActiveConsumers` zwischen diesen Versionen geändert hat. (Fehler 6716400)

Umgehung: Stellen Sie sicher, dass der Wert von

`imq.autocreate.queue.maxNumActiveConsumers` bei allen Brokern gleich ist.

Normalerweise erfolgt dies, indem die Konfiguration von Message Queue 4.3, 4.2 und 4.1 an diejenige angepasst wird, die von Brokern der Version 3.7 oder 3.6 verwendet wird (standardmäßig wird der Wert -1 zum Standardwert 1 der vorhergehenden Version geändert).

- Um einen Broker für Message Queue 4.3 (oder 4.x) einem Broker-Cluster für Message Queue 3.x hinzuzufügen, muss mindestens ein Masterbroker ausgeführt werden. (Fehler 6763796)
- Bei der Konvertierung aus einem konventionellen Cluster in einen erweiterten Cluster können Sie mit dem Datenbank-Manager-Dienstprogramm von Message Queue (`imqdbmgr`) einen bestehenden eigenständigen JDBC-basierten Datenspeicher in einen gemeinsam genutzten JDBC-Datenspeicher konvertieren, wie in „[Cluster Conversion: JDBC-Based Data Store](#)“ in *Sun GlassFish Message Queue 4.4 Administration Guide* dokumentiert.
- Ein Broker kann bei Verwendung von HADB ausschließlich Nachrichten bis zu einer Größe von 10 MB verarbeiten. (Fehler 6531734)
- Die Umwandlung in einen HADB-Speicher mit dem Befehl `imqdbmgr upgrade hastore` kann mit der Fehlermeldung "zu viele Sperren gesetzt" fehlschlagen, wenn der Speicher mehr als 10.000 Nachrichten enthält. (Fehler 6588856)

Umgehung: Verwenden Sie den folgenden Befehl, um die Anzahl an Sperren zu erhöhen.

```
hadbm set NumberOfLocks=<gewünschte Anzahl>
```

Weitere Informationen finden Sie unter "HADB Problems" im *Sun Java System Application Server 9.1 Enterprise Edition Troubleshooting Guide*.

- Wenn mehr als 500 Remote-Nachrichten in einer Transaktion übermittelt werden, gibt der Broker möglicherweise den Fehler "HADB-E-12815: Table memory space exhausted" (Kein Tabellenspeicher verfügbar) aus. (Fehler 6550483)

Weitere Informationen finden Sie unter "HADB Problems" im *Sun Java System Application Server 9.1 Enterprise Edition Troubleshooting Guide*.

- In einem Broker-Cluster stellt ein Broker die Nachrichten an eine entfernte Verbindung, die noch nicht geöffnet wurde, in eine Warteschlange. (Fehler 4951010)

Umgehung: Der Konsument erhält die Nachrichten, sobald die Verbindung geöffnet wurde. Die Nachrichten werden an einen anderen Konsumenten gesendet, wenn die Verbindung des Konsumenten geschlossen bleibt.

- Wenn ein Konsument mehr als eine Nachricht von einem Remote-Broker in einer Transaktion empfängt, ist es möglich, dass die folgende Fehlermeldung für den Broker protokolliert wird. Diese Fehlermeldung ist nicht kritisch und kann ignoriert werden:

```
[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
```

Diese Meldung wird im Protokoll erfasst, wenn der Nachrichten-Startbroker für spätere Nachrichten in der Transaktion über die Übermittlung benachrichtigt wurde, sobald die `imq.txn.reapLimit`-Eigenschaft im Vergleich zur Anzahl an Remote-Nachrichten in einer Transaktion gering ist. (*Fehler 6585449*)

Umgehung: Um diese Meldung zu verhindern, erhöhen Sie den Wert der Eigenschaft `imq.txn.reapLimit`.

JMX-Probleme

Auf der Windows-Plattform gibt die `getTransactionInfo`-Methode der Überwachungs-MBean für den Transaktionsmanager Transaktionsinformationen zurück, die eine falsche Transaktionserstellungszeit enthalten. (*Fehler 6393359*)

Umgehung: Verwenden Sie stattdessen die `getTransactionInfoByID`-Methode der Überwachungs-MBean für den Transaktionsmanager.

SOAP-Unterstützung

Sie sollten zwei Probleme in Verbindung mit der SOAP-Unterstützung beachten.

- Seit der Veröffentlichung von Message Queue Version 4.0 werden SOAP-verwaltete Objekte nicht mehr unterstützt.
- SOAP-Entwicklung hängt von mehreren Dateien ab: `SUNWjaf`, `SUNWjmail`, `SUNWxprt` und `SUNWjaxp`. In Message Queue Version 4.1 sind diese Dateien nur verfügbar, wenn Sie Message Queue mit JDK Version 1.6.0 oder höher ausführen.
- Bisher verwies die SAAJ 1.2-Implementierungs-Datei `.jar` direkt auf `mail.jar`. In SAAJ 1.3 wurde dieser Verweis gelöscht, daher muss sich die Datei `mail.jar` in Message Queue-Clients ausdrücklich in `CLASSPATH` befinden.

Dateien für Neuverteilung

Sun GlassFish Message Queue 4.4 Update 1 enthält die folgenden Dateisätze, die Sie verwenden und im Binärformat verteilen können:

<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>

Außerdem können Sie die Dateien `LICENSE` und `COPYRIGHT` ebenfalls neu verteilen.

Eingabehilfen für Personen mit Behinderungen

Um Eingabehilfen zu erhalten, die seit der Veröffentlichung dieses Dokuments auf den Markt gekommen sind, lesen Sie Abschnitt 508 der Produktbewertungen (bei Sun auf Anfrage erhältlich), um die für Sie geeignete Version zu ermitteln. Aktualisierte Versionen von Anwendungen finden Sie unter <http://sun.com/software/javaenterprisesystem/get.html>.

Informationen über das Engagement von Sun bezüglich Eingabehilfen von finden Sie unter <http://sun.com/access>.

Problemmeldungen und Feedback

Wenn Sie mit Sun GlassFish Message Queue Probleme haben, wenden Sie sich an die Kundenunterstützung von Sun. Dazu stehen Ihnen folgende Möglichkeiten zur Verfügung:

- Online-Softwaresupport von Sun unter <http://www.sun.com/service/sunone/software>. Diese Site bietet neben Links zur Knowledge Base, zum Online Support Center und zum ProductTracker auch Wartungsprogramme und Support-Kontaktnummern.
- Wenden Sie sich per Telefon an Sun. Verwenden Sie hierzu die auf Ihrem Wartungsvertrag angegebene Telefonnummer.

Damit wir Sie bestmöglich bei der Problembeseitigung unterstützen können, sollten Sie folgende Informationen zur Hand haben, wenn Sie unser Support-Team kontaktieren:

- Beschreibung des Problems, einschließlich der Situation, in der das Problem auftrat, sowie seine Auswirkungen auf Ihre Arbeit.
- Computertyp, Betriebssystem- und Produktversion, u. a. Patches und andere Softwareanwendungen, die das Problem verursacht haben könnten.
- Detaillierte Schritte zu den von Ihnen verwendeten Methoden, um das Problem zu reproduzieren.
- Sämtliche Fehlerprotokolle oder Kernspeicherauszüge.

Sun GlassFish-Software-Forum

Unter der folgenden Adresse steht ein Sun GlassFish Message Queue-Forum zur Verfügung:

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

Wir freuen uns über Ihre Teilnahme.

Java Technology Forum

Im Java Technology Forum finden Sie möglicherweise ein für Sie interessantes JMS-Forum.

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

Ihre Meinung ist gefragt

Sun bemüht sich um eine stetige Verbesserung der Dokumentationen und ist deshalb an Ihrer Meinung und Ihren Anregungen interessiert.

Zum Mitteilen Ihrer Kommentare rufen Sie <http://docs.sun.com> auf und klicken dort auf "Send Comments". Geben Sie auf dem Onlineformular den Namen und die Bestellnummer der Dokumentation an. Die Teilenummer ist eine sieben- bis neunstellige Zahl, die auf der Titelseite der Buches oder oben auf der Dokumentation angegeben ist. Der Titel des vorliegenden Buches lautet beispielsweise *Versionshinweise*, die Teilenummer lautet 821-1511-10.

Weitere Quellen von Sun

Nützliche Informationen über Sun GlassFish finden Sie unter den folgenden Internetadressen:

- Dokumentation
<http://docs.sun.com/prod/java.sys>
- Profi-Services
<http://www.sun.com/service/sunps/sunone>
- Softwareprodukte und Services
<http://www.sun.com/software>
- Softwaresupport
<http://www.sun.com/service/sunone/software>
- Support und Knowledge Base
<http://www.sun.com/service/support/software>
- Support und Schulung
<http://training.sun.com>
- Beratung und Profi-Services
<http://www.sun.com/service/sunps/sunone>
- Informationen für Entwickler
<http://developers.sun.com>
- Sun-Entwicklersupport
<http://www.sun.com/developers/support>
- Softwareschulungen
<http://www.sun.com/software/training>

