



# Versionshinweise zu Sun Java System Message Queue 4.2



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

Teilnr.: 820-5638  
Juli 2008

Copyright 2008 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.

Diese 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 seiner Niederlassungen 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<sup>TM</sup> 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 JEGLICHE 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.

# Inhalt

---

<b>1</b>	<b>Versionshinweise zu Sun Java System Message Queue 4.2</b>	<b>5</b>
	Änderungsprotokoll der Versionshinweise	6
	Installation von bzw. Aufrüsten auf Message Queue 4.2	7
	Message Queue 4.2 - Unterstützte Plattformen und Komponenten	7
	Unterstützte Betriebssystemplattformen	7
	Unterstützung der Systemvirtualisierung	8
	Komponentenabhängigkeiten	8
	Neue Funktionen in Message Queue 4.2 und aktuellen Versionen	10
	Neue Funktionen in Message Queue 4.2	10
	Neue Funktionen in Message Queue 4.1	19
	Neue Funktionen in Message Queue 4.0	23
	Funktion ist in zukünftiger Version veraltet	27
	In Message Queue 4.2 und aktuellen Versionen behobene Fehler	27
	In Message Queue 4.2 behobene Fehler	27
	In Message Queue 4.1 behobene Fehler	29
	Behobene Probleme in Message Queue 4.0	30
	Dokumentationsaktualisierungen in Message Queue 4.2	31
	Kompatibilitätsprobleme	31
	Änderungen am Message Queue 4.2-Dokumentationssatz	32
	Neue Zielmetrik	33
	Automatischer Broker-Start unter Solaris 10 OS	34
	Änderungen an der JMX-API	35
	Unterstützung für DN-Benutzernamensformat für Client-Authentifizierung	40
	Erweiterung zur JAAS-Authentifizierung	41
	Bekannte Probleme und Beschränkungen	41
	Installationsprobleme	41
	Veraltete Passwortoption	47
	Administration/Konfiguration	48

Broker-Probleme .....	49
Broker-Cluster .....	51
JMX-Probleme .....	53
SOAP-Unterstützung .....	53
Dateien für Neuverteilung .....	53
Eingabehilfen für Personen mit Behinderungen .....	54
Problemmeldungen und Feedback .....	54
Sun Java System-Software-Forum .....	54
Java Technology Forum .....	55
Ihre Meinung ist gefragt .....	55
Weitere Quellen von Sun .....	55

# Versionshinweise zu Sun Java System Message Queue 4.2

---

Version 4.2

Teilnr. 820-3701

Diese Versionshinweise enthalten wichtige Informationen, die zum Zeitpunkt der Herausgabe von Sun Java™ System Message Queue 4.2 zur Verfügung standen. Hier finden Sie u. a. Informationen zu neuen Funktionen, Verbesserungen und bekannten Problemen. Lesen Sie dieses Dokument, bevor Sie Message Queue 4.2 verwenden.

Diese Versionshinweise enthalten auch Informationen über die Versionen 4.0 und 4.1 von Message Queue: unter [“Neue Funktionen in Message Queue 4.0“ auf Seite 23](#) bzw. [“Neue Funktionen in Message Queue 4.1“ auf Seite 19](#) finden Sie Informationen zu den in diesen Versionen neu hinzugekommenen Funktionen.

Die neueste Ausgabe dieser Versionshinweise wird auf der Sun Java System Message Queue Dokumentations-Website, unter <http://docs.sun.com/coll/1307.3>, 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](#)
- [“Installation von bzw. Aufrüsten auf Message Queue 4.2“ auf Seite 7](#)
- [“Message Queue 4.2 - Unterstützte Plattformen und Komponenten“ auf Seite 7](#)
- [“Neue Funktionen in Message Queue 4.2 und aktuellen Versionen“ auf Seite 10](#)
- [“Funktion ist in zukünftiger Version veraltet“ auf Seite 27](#)
- [“In Message Queue 4.2 und aktuellen Versionen behobene Fehler“ auf Seite 27](#)
- [“Dokumentationsaktualisierungen in Message Queue 4.2“ auf Seite 31](#)
- [“Bekannte Probleme und Beschränkungen“ auf Seite 41](#)
- [“Dateien für Neuverteilung“ auf Seite 53](#)
- [“Eingabehilfen für Personen mit Behinderungen“ auf Seite 54](#)
- [“Problemmeldungen und Feedback“ auf Seite 54](#)

- [“Ihre Meinung ist gefragt“ auf Seite 55](#)
- [“Weitere Quellen von Sun“ auf Seite 55](#)

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 Revisionsverlauf

Date	Änderungen
Mai 2006	Erste Ausgabe dieses Dokuments für Message Queue 4.0.
Januar 2007	Erste Ausgabe dieses Dokuments für Message Queue 4.1 Beta. Fügt eine Beschreibung der JAAS-Unterstützung hinzu.
April 2007	Zweite Ausgabe dieses Dokuments für Message Queue 4.1 Beta. Fügt eine Hochverfügbarkeitsfunktion hinzu.
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 2008	Erste Entwurfsausgabe dieses Dokuments für Message Queue 4.2. Ergänzt durch neue Funktionen für diese Version.

## Installation von bzw. Aufrüsten auf Message Queue 4.2

Mithilfe des Message Queue 4.2-Installationsprogramms können Sie eine Neuinstallation von Message Queue 4.2 oder eine Aufrüstung von Message Queue 3.6 oder höher durchführen. Das Verfahren und alle anderen Informationen, die für Installation oder Aufrüstung auf den Plattformen Solaris, Linux und Windows relevant sind, werden im Handbuch *Sun Java System Message Queue 4.2 Installation Guide* dokumentiert, der noch nicht für Message Queue 4.2 aktualisiert wurde.

Wenn Sie eine Aufrüstung von einer Message Queue-Version vor 3.6 durchführen möchten, lesen Sie im *Sun Java Enterprise System 5-Aufrüstungshandbuch für UNIX*, *Sun Java Enterprise System 5 Update 1 Upgrade Guide for UNIX* nach.

Lesen Sie auch unter [“Installationsprobleme“](#) auf Seite 41 nach. Dort sind bekannte Probleme und Einschränkungen beschrieben, die bei Installation und Aufrüstung gelten.

## Message Queue 4.2 - Unterstützte Plattformen und Komponenten

In diesem Abschnitt werden folgende Themen in Bezug auf die für Message Queue 4.2 geltenden Systemvoraussetzungen behandelt:

- [“Unterstützte Betriebssystemplattformen“](#) auf Seite 7
- [“Unterstützung der Systemvirtualisierung“](#) auf Seite 8
- [“Komponentenabhängigkeiten“](#) auf Seite 8

### Unterstützte Betriebssystemplattformen

Message Queue 4.2 wird auf den Betriebssystemplattformen Solaris, Linux und Windows 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 4.2 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
Linux	Red Hat Enterprise Linux Advanced Server 3.0, 4.0, 5.0, alle Aktualisierungen, 32- und 64-Bit-Versionen (x86, x64)  Red Hat Enterprise Linux Enterprise Server 3.0, 4.0, 5.0, alle Aktualisierungen, 32- und 64-Bit-Versionen (x86, x64)
Windows	Windows Vista  Windows XP Professional, SP2 ( x86) <sup>1</sup>  Windows 2000 Advanced Server, SP4 (x86) <sup>2</sup>  Windows Server 2003, Standard und Enterprise Edition, SP2, 32- und 64-Bit-Versionen (x86, x64) <sup>3</sup>

<sup>1</sup> Keine Unterstützung für Home, Tablet PC oder Media Center Edition

<sup>2</sup> Keine Unterstützung für Professional oder Server Edition

<sup>3</sup> 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 4.2 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.



**Hinweis** – Mit dem Message Queue-Installationsprogramm können Sie ein(e) bestehende(s) JDK/JRE auswählen, oder die JDK-Version (1.5.0\_15) installieren.

TABELLE 1-3 Erforderliche Unterstützungskomponenten

Komponente	Unterstützt	Unterstützte Versionen
Java Runtime Environment (JRE)	Message Queue-Broker- und Verwaltungswerkzeuge	J2SE™ Runtime Environment 1.5.0_15 oder höher Java™ SE Runtime Environment 1.6.0 (nur Versionen von Sun Microsystems)
Java Software Development Kit (JDK), Standard Edition	Java-Client-Entwicklung und -Einsatz	J2SE™ Development Kit 1.5.0_15 oder höher Java SE Development Kit 1.6.0 (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. Sie benötigen nicht unbedingt 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 Java System Application Server Enterprise Edition, Version 9.1 Update Release 2
Web Server	HTTP/HTTPS	Sun Java System Web Server Enterprise Edition, Version 7.0, Update 2
Datenbank	JDBC-basierter Datenspeicher	HADB, Version 4.4.3.5 Java DB (Apache Derby), Version 10.2.2 MySQL Community Edition, Version 5.0 Oracle10g postgreSQL, Version 8.1 <b>Hinweis</b> – Die PointBase-Datenbank wird nicht mehr unterstützt.
Hochverfügbare Datenbank	Hochverfügbarkeits-Broker-Cluster	HADB, Version 4.4.3.5 MySQL Cluster Edition, Version 5.0 Oracle10g

TABELLE 1-4 Optionale Unterstützungskomponenten (Fortsetzung)

Komponente	Unterstützt	Unterstützte Versionen
Lightweight Directory Access Protocol-(LDAP-)Verzeichnisse	Message Queue – Benutzer-Repository und verwaltete Objekte	Sun Java System 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 <sup>1</sup>
C-Compiler und kompatible C++-Runtime-Bibliothek	Message Queue-C-Clients	<b>Solaris:</b> Sun Studio, Version 11 oder höher, C++-Compiler mit Standardmodus und C-Compiler <b>Linux:</b> gcc/g++, Version 3.2.3 <b>Windows:</b> Microsoft Windows Visual C++, Version 6.0 SP3
Netscape Portable Runtime (NSPR)	Message Queue-C-Clients	Version 4.7-1 <sup>2</sup>
Network Security Services (NSS)	Message Queue-C-Clients	Version 3.11.9-1 <sup>2</sup>

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

<sup>2</sup> Gebündelt als gemeinsam genutztes Paket im Download-Bundle

## Neue Funktionen in Message Queue 4.2 und aktuellen Versionen

Die neuen Funktionen in Message Queue 4.2, 4.1 und 4.0 sind in folgenden Abschnitten beschrieben:

- [“Neue Funktionen in Message Queue 4.2“ auf Seite 10](#)
- [“Neue Funktionen in Message Queue 4.1“ auf Seite 19](#)
- [“Neue Funktionen in Message Queue 4.0“ auf Seite 23](#)

### Neue Funktionen in Message Queue 4.2

Sun Java System 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 die JMS-Spezifikation hinausgehen, um den Anforderungen großer Bereitstellungen im Unternehmensbereich gerecht zu werden.

Message Queue 4.2 ist eine Unterversion, die eine Anzahl von Funktionserweiterungen und Fehlerkorrekturen enthält. In diesem Abschnitt wird die Installation von bzw. die Aufrüstung auf Message Queue 4.2 erläutert und die neuen Funktionen, die in dieser Version enthalten sind, werden beschrieben:

- [“Mehrere Ziele für einen Herausgeber bzw. Abonnenten“](#) auf Seite 11
- [“Schemavalidierung von XML-Payload-Meldungen“](#) auf Seite 13
- [“C-API-Unterstützung für verteilte Transaktionen“](#) auf Seite 15
- [“Unterstützung des Installationsprogramms für Sun Connection-Registrierung“](#) auf Seite 17
- [“Unterstützung für MySQL-Datenbank“](#) auf Seite 19

Informationen zu den in den Versionen Message Queue 4.0 und 4.1 neu hinzugekommenen Funktionen finden Sie in [“Neue Funktionen in Message Queue 4.0“](#) auf Seite 23 bzw. [“Neue Funktionen in Message Queue 4.1“](#) auf Seite 19.

## Mehrere Ziele für einen Herausgeber bzw. Abonnenten

In Message Queue 4.2 kann ein Herausgeber nun 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.

---

Das Format des symbolischen Themenzielnamens besteht aus mehreren Segmenten, bei denen Platzhalterzeichen (\*, \*\*, >) für ein oder mehrere Namenssegmente stehen können. Beispiel: Angenommen Sie verwenden folgendes Benennungsschema für Themenziele:

*Größe* verwendet wird.*Farbe*. *Form*

Dabei können die Themennamensegmente folgende Werte aufweisen:

- *Größe*: large, medium, small, ...
- *Farbe*: red, green, blue, ...
- *Form*: circle, triangle, square, ...

Die Nachrichtenwarteschlange unterstützt folgende Platzhalterzeichen:

- \* steht für ein einzelnes Segment
- \*\* steht für ein oder mehrere Segmente
- > Steht für eine beliebige Anzahl aufeinander folgender Segmente

Sie können daher mehrere Themenziele wie folgt angeben:

`large.*.circle` steht für:

```
large.red.circle
large.green.circle
...
```

`**.*.square` steht für alle Namen, die auf `.square` enden, z. B.:

```
small.green.square
medium.blue.square
...
```

`small.>` steht für alle Zielnamen, die mit `small.` beginnen, z. B.:

```
small.blue.circle
small.red.square
...
```

Um diese Funktion für mehrere Ziele zu verwenden, erstellen Sie Themenziele mit einem ähnlichen Benennungsschema wie oben beschrieben. Die Client-Anwendungen können dann Herausgeber oder Konsumenten mithilfe von symbolischen Zielnamen erstellen. Beispiel:

```
...
String DEST_LOOKUP_NAME = "large.*.circle";
Topic t = (Destination) ctx.lookup(DEST_LOOKUP_NAME);
TopicPublisher myPublisher = mySession.createPublisher(t);
myPublisher.send(myMessage);

...
String DEST_LOOKUP_NAME = "**.*.square";
Topic t = (Destination) ctx.lookup(DEST_LOOKUP_NAME);
TopicSubscriber mySubscriber = mySession.createSubscriber(t);
Message m = mySubscriber.receive();
```

Im ersten Beispiel legt der Broker eine Kopie der Nachricht an einem Ziel ab, das dem symbolischen Namen `large.*.circle` entspricht. Im zweiten Beispiel wird ein Abonnement erstellt, wenn es mindestens ein Ziel gibt, das mit dem symbolischen Namen `**.*.square` übereinstimmt, und der Abonnement erhält Nachrichten aus allen Zielen, die mit diesem symbolischen Namen übereinstimmen. Wenn es keine Ziele gibt, die mit dem symbolischen Namen übereinstimmen, wird der Abonnent erst erstellt, wenn ein solches Ziel vorhanden ist.

Wenn ein Administrator zusätzliche Ziele erstellt, die mit einem symbolischen Namen übereinstimmen, veröffentlichen die Platzhalter-Herausgeber, die mit diesem symbolischen Namen erstellt wurden, anschließend unter diesem Ziel, und die Platzhalter-Abonnenten, die mit diesem symbolischen Namen erstellt wurden, empfangen anschließend Nachrichten von diesem Ziel.

Außerdem melden die Message Queue-Verwaltungswerkzeuge, neben der Gesamtzahl an Herausgebern (Produzenten) und Abonnenten (Konsumenten) für ein Themenziel auch die Anzahl der Herausgeber, bei denen es sich um Platzhalter-Herausgeber handelt (einschließlich der zugehörigen symbolischen Zielnamen) und die Anzahl der Abonnenten, bei denen es sich um Platzhalter-Abonnenten handelt (einschließlich der symbolischen Zielnamen), sofern vorhanden.

## Schemavalidierung von XML-Payload-Meldungen

Diese neue Funktion in Message Queue 4.2 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.)

Client-Anwendungen, die diese neue Funktion verwenden, sollten die Java SE-Version auf JRE 1.5 oder höher aufrüsten.

Zur Aktivierung der XML-Schemavalidierung legen sie folgende Eigenschaften für das physische Ziel fest:

TABELLE 1-5 Eigenschaften des physischen Ziels für die XML-Schemavalidierung

Eigenschaft	Typ	Standardwert	Beschreibung
<code>validateXMLSchemaEnabled</code>	Boolesch	<code>false</code>	Ist XML-Schemavalidierung aktiviert?  Wenn dieser Wert auf <code>false</code> (falsch) gesetzt oder gar nicht festgelegt wurde, wird die XML-Schemavalidierung nicht für das Ziel aktiviert.

TABELLE 1-5 Eigenschaften des physischen Ziels für die XML-Schemavalidierung (Fortsetzung)

Eigenschaft	Typ	Standardwert	Beschreibung
XMLSchemaURLList	String	Null	<p>Leerzeichengetrennte Liste der URI-Zeichenfolgen für XML Schema Document (XSD)</p> <p>Die URIs verweisen auf den Speicherort einer oder mehrerer XSDs zur Verwendung mit der XML-Schemavalidierung, sofern aktiviert</p> <p>Verwenden Sie doppelte Leerzeichen um diesen Wert, wenn mehrere URIs angegeben werden.</p> <p>Beispiel:</p> <p>“http://foo/flap.xsd http://test.com/test.xsd”</p> <p>Wenn diese Eigenschaft nicht festgelegt oder NULL ist und die XML_Validierung aktiviert wurde, wird die XML-Validierung mithilfe eines im XML-Dokument angegebenen DTD durchgeführt.</p>
reloadXMLSchemaOnFailure	Boolesch	false	<p>Erneutes Laden des XML-Schemas bei Versagen aktiviert?</p> <p>Wenn dieser Wert auf false (falsch) gesetzt oder gar nicht festgelegt wurde, wird das Schema nicht erneut geladen, wenn die Validierung fehlschlägt.</p>

Wenn die XML-Validierung aktiviert ist, versucht die Message Queue-Client-Laufzeit, eine XML-Nachricht anhand der angegebenen XSDs (oder anhand des DTD, wenn kein XSD angegeben wurde) zu validieren, bevor sie an den Broker gesendet wird. Wenn das angegebene Schema nicht gefunden oder die Nachricht nicht validiert werden kann, wird die Nachricht nicht gesendet, und eine Ausnahme wird ausgelöst.

Die XML-Validierungseigenschaften können mit dem Befehl `imqcmd create dst` bzw. `imqcmd update dst` zum Zeitpunkt der Zielerstellung bzw. -aktualisierung festgelegt werden. Die XML-Validierungseigenschaften sollten festgelegt werden, wenn ein Ziel inaktiv ist: wenn es also keine Konsumenten und Produzenten aufweist und wenn sich keine Nachrichten in dem Ziel befinden.

**Hinweis** – Wenn ein XSD zur Laufzeit nicht zugänglich ist, kann es erforderlich sein, die XMLSchemaURLList zu bearbeiten, während ein Ziel aktiv ist.

Wenn eine der XML-Validierungseigenschaften festgelegt ist, während ein Ziel aktiv ist (beispielsweise, wenn ein Produzent mit dem Ziel verbunden ist), wird die Änderung erst wirksam, wenn der Produzent erneut eine Verbindung zum Broker herstellt. Ebenso gilt: Wenn

infolge einer Änderung der Anwendungsanforderungen ein XSD geändert wird, müssen alle Client-Anwendungen, die auf der Grundlage des geänderten XSD XML-Nachrichten erstellen, erneut eine Verbindung zum Broker herstellen.

Wenn die Eigenschaft `reloadXMLSchemaOnFailure` auf `true` (wahr) gesetzt wird und die XML-Validierung fehlschlägt, versucht die Message Queue-Client-Laufzeit, das XSD erneut zu laden, bevor der erneute Versuch unternommen wird, eine Nachricht zu validieren. Die Client-Laufzeit löst eine Ausnahme aus, wenn die Validierung mithilfe der neu geladenen SxD fehlschlägt.

## 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 Message Queue C-API nun die XA-Schnittstelle (zwischen einem Manager für verteilte Transaktionen und Message Queue als XA-konformen Ressourcenmanager), was es Message Queue C-API-Clients, die in einer Umgebung ausgeführt werden, die verteilte Transaktionen verarbeitet (wie beispielsweise BEA Tuxedo), sich an verteilten Transaktionen zu beteiligen.

Diese Unterstützung für verteilte Transaktionen besteht aus folgenden neuen C-API-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.

## Öffentliche Informationen

Für die X/Open-XA-Schnittstellenspezifikation sind folgende öffentlichen Informationen in Bezug auf den XA-konformen Message Queue-Ressourcenmanager erforderlich:

- Name der `xa_switch_t`-Struktur: `sun_my_xa_switch`
- Name des Ressourcenmanagers: `SUN_RM`
- Die zu verknüpfende MQ C-API-Bibliothek: `mqcrt`
- Zeichenfolge `xa_close` und Format: `none`
- Zeichenfolge `xa_open` und Format: durch „;“ getrennte Name=Wert-Paare

Folgende Name/Wert-Paare werden unterstützt:

TABELLE 1-6 Name/Wert-Paare des Message Queue-Ressourcenmanagers

Name	Wert	Beschreibung	Standard
Adresse	<i>host:port</i>	Der Host:Anschluss des Portmapper-Diensts des Brokers.	localhost:7676
username	string	Der Benutzername für die Herstellung einer Verbindung zum Broker	guest
Passwort	string	Das zum Benutzernamen gehörende Passwort	guest
conntype	TCPor SSL	Der Protokolltyp für die Verbindung mit dem Broker	TCP
trustedhost	true/false	Ob der Broker-Host vertrauenswürdig ist (gilt nur bei conntype=SSL)	true
certdbpath	string	Der vollständige Pfad zu dem Verzeichnis, das das NSS-Zertifikat und die Schlüsseldatenbankdateien enthält	nicht festgelegt
clientid	string	Nur für dauerhafte JMS-Abonnements erforderlich	nicht festgelegt
reconnects	Integer	Die Anzahl der erneuten Verbindungsversuche für den Broker (0 bedeutet: keine erneute Verbindung)	0

## Programmierungsbeispiele

Um eine Anwendung zu programmieren, die verteilte Transaktionen verwendet, erstellen Sie einen serverseitigen Dienst, der in der Umgebung des Transaktionsmanagers ausgeführt wird, und einen clientseitigen CODE, der die Transaktionsmanager-APIs aufruft. Message Queue 4.2 bietet Programmierungsbeispiele, die auf dem Tuxedo-Transaktionsmanager beruhen. Diese Beispiele befinden sich im Beispielprogrammverzeichnis auf den einzelnen Plattformen im Verzeichnis `./C/tuxedo`.

Dieses Verzeichnis beinhaltet eine README-Datei, die erläutert, wie Tuxedo zur Verwendung des Message Queue-Ressourcenmanagers eingerichtet wird und wie die folgenden Beispielprogramme in der Tuxedo-Umgebung erstellt werden können:



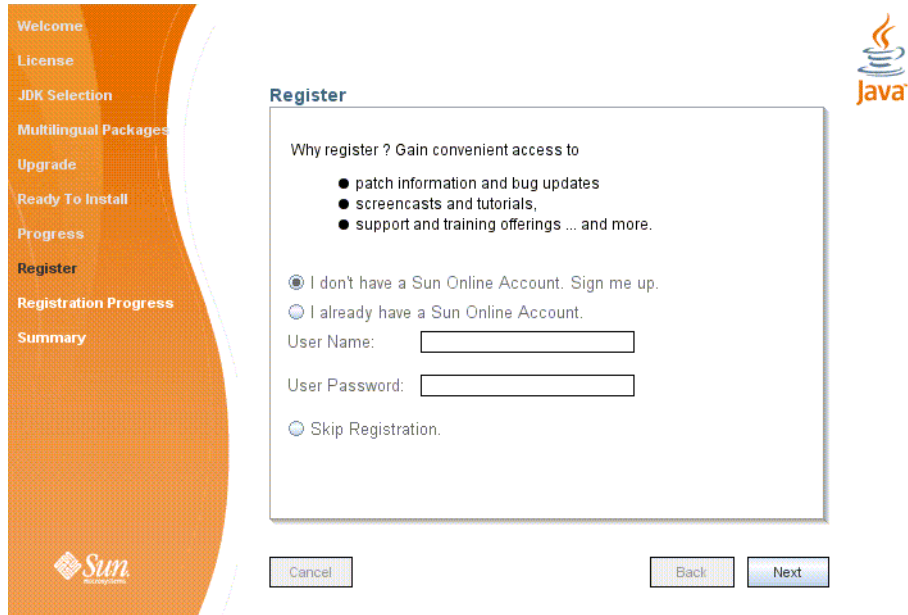
Beispielprogramm	Beschreibung
<code>jmsserver.c</code>	Implementiert Tuxedo-Dienste, die Nachrichten mithilfe von Message Queue senden und empfangen.
<code>jmssclient_sender.c</code>	Tuxedo-Client, der den Nachrichtenproduktionsdienst im Programm <code>jmsserver.c</code> verwendet.
<code>jmssclient_receiver.c</code>	Tuxedo-Client, der den Nachrichtenempfangsdienst im Programm <code>jmsserver.c</code> verwendet.
<code>async_jmsserver.c</code>	Implementiert einen Tuxedu-Dienst, der asynchron Nachrichten mithilfe von Message Queue konsumiert.
<code>jmssclient_async_receiver.c</code>	Tuxedo-Client, der den asynchronen Nachrichtenempfangsdienst im Programm <code>jmsserver.c</code> verwendet.

## Unterstützung des Installationsprogramms für Sun Connection-Registrierung

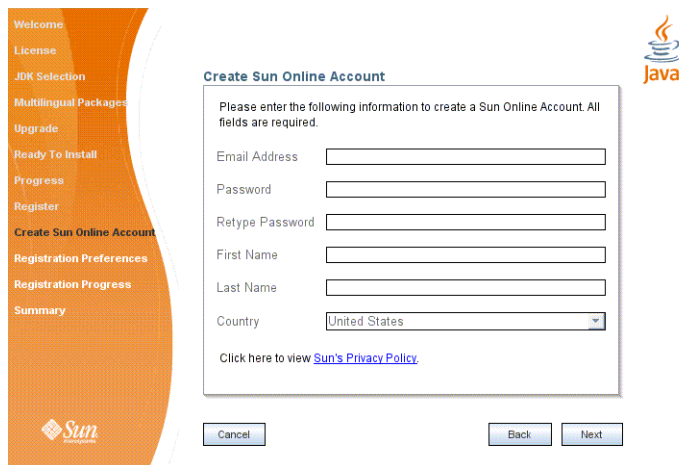
Das Message Queue-Installationsprogramm wurde erweitert und ermöglicht nun die Registrierung von Message Queue bei Sun Connection, einem von Sun gehosteten Dienst, mit dem Sie Sun-Hardware und -Software aufzeichnen, organisieren und warten können.

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.

Der folgende Bildschirm des Installationsprogramms wurde für die Sun Connection-Registrierung zu Message Queue 4.2 hinzugefügt:



Für die Registrierung müssen Sie ein Sun Online-Konto besitzen bzw. erstellen. Wenn Sie noch nicht über ein Konto verfügen, zeigt das Installationsprogramm folgenden Bildschirm zur Erstellung eines Sun Online-Kontos an:



Sie können mithilfe der oben angegebenen Bildschirm Message Queue während der Installation erstellen oder bis zum Abschluss der Installation warten und das Installationsprogramm wie folgt im reinen Registrierungsmodus ausführen:

```
# installer -r
```

Für den reinen Registrierungsmodus muss Message Queue 4.2 bereits installiert sein und zeigt nur die Bildschirme des Installationsprogramms an, die sich auf die Registrierung beziehen.

## Unterstützung für MySQL-Datenbank

Message Queue 4.2 unterstützt MySQL-Datenbank als JDBC-basierter Datenspeicher. 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 einen Hochverfügbarkeits-Broker-Cluster benötigt wird. Informationen zur Konfiguration von Message Queue für die Verwendung von MySQL finden Sie unter [“Configuring a JDBC-Based Data Store“](#) in *Sun Java System Message Queue 4.2 Administration Guide* sowie unter [“High-Availability Cluster Properties“](#) in *Sun Java System Message Queue 4.2 Administration Guide*.

## 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 19
- [“JAAS-Unterstützung“](#) auf Seite 21
- [“Formatänderung für persistenten Datenspeicher“](#) auf Seite 21
- [“Broker-Umgebungs-Konfiguration“](#) auf Seite 21
- [“Unterstützung für Java ES Monitoring Framework“](#) auf Seite 21
- [“Erweiterte Transaktionsverwaltung“](#) auf Seite 22
- [“Feste Ports für C-Clientverbindungen“](#) auf Seite 22

Informationen zu den in Message Queue 4.0 hinzugefügten Funktionen finden Sie unter [“Neue Funktionen in Message Queue 4.0“](#) auf Seite 23.

## Hochverfügbarkeits-Broker-Cluster

In Message Queue 4.1 kamen Hochverfügbarkeits-Broker-Cluster neu hinzu. Im Vergleich zu konventionellen Broker Clustern, die nur *Nachrichtendienst*-Verfügbarkeit (wenn ein Broker ausfällt, ist ein anderer verfügbar, um den Nachrichtendienst bereitzustellen) bieten, bieten Hochverfügbarkeits-Broker-Cluster auch *Daten*-Verfügbarkeit (wenn ein Broker ausfällt, stehen seine persistenten Meldungen und Statusdaten einem anderen Broker zur Verfügung, der sie verwenden kann, um die Nachrichtenzustellung zu übernehmen).

Die seit Message Queue 4.1 verfügbare Hochverfügbarkeits-Implementierung verwendet einen freigegebenen 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 Nachrichtenweiterleitung und Zustellung 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 einen Message Queue 4.1-Hochverfügbarkeits-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 Hochverfügbarkeits-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 herstellereigenschaftenspezifische 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 Hochverfügbarkeits-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 im Zusammenhang mit Hochverfügbarkeits-Broker-Clustern und einen Vergleich mit konventionellen Clustern finden Sie in [Kapitel 4, "Broker Clusters"](#) in *Sun Java System Message Queue 4.2 Technical Overview*. Die Vorgehensweise und Referenzinformationen zu Hochverfügbarkeits-Broker-Clustern finden Sie in [Kapitel 8, "Managing Broker Clusters"](#) in *Sun Java System Message Queue 4.2 Administration Guide* und unter ["Cluster Configuration Properties"](#) in *Sun Java System Message Queue 4.2 Administration Guide*.

Wenn Sie eine hochverfügbare Datenbank zusammen mit Message Queue 4.0 verwendet haben und nun zu einem Hochverfügbarkeits-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 51](#).

## 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 steht und eine Erläuterung der Konfiguration des Brokers zur Verwendung eines solchen Diensts finden Sie unter [“Using JAAS-Based Authentication“ in \*Sun Java System Message Queue 4.2 Administration Guide\*](#).

## Formatänderung für persistenten Datenspeicher

Message Queue 4.1 änderte den JDBC-basierten Datenspeicher dahingehend, dass Hochverfügbarkeits-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 auf die Version 410 erhöht.

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 Umgebungsdatei 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 nun mit dem Befehlsdienstprogramm (`imqcmd`) Transaktionen bereinigen (in einem 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 stellt die Verbindungs-ID für die Verbindung bereit, welche die Transaktion gestartet hat, und gibt die Uhrzeit an, zu welcher 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 JMX Administration API“ auf Seite 23
- “Protokollierung zur Client-Laufzeit“ auf Seite 23
- “Verbindungseignissenachrichtigungs-API“ auf Seite 24
- “Erweiterungen der Broker-Administration“ auf Seite 24
- “Anzeigen von Informationen zu einem JDBC-basierten Datenspeicher“ auf Seite 25
- “JDBC-Anbieter-Unterstützung“ auf Seite 25
- “Formatänderungen für persistenten Datenspeicher“ auf Seite 26
- “Zusätzliche Nachrichteneigenschaften“ auf Seite 26
- “SSL-Unterstützung“ auf Seite 26



---

**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 47 beschrieben, oder Sie bei Aufforderung eingeben.

---

### Unterstützung für 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 Java System Message Queue 4.2 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 Hochverfügbarkeits-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 zur Syntax des Befehls `imqcmd` finden Sie in [Kapitel 13, "Command Line Reference"](#) in *Sun Java System Message Queue 4.2 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] Verwenden des integrierten Persistenzspeichers:
    Version=400
    Broker-ID=Mozart1756
    Datenbankverbindung url=jdbc:oracle:thin:@Xhome:1521:mqdb
    Datenbankbenutzer=scott
```

Ausführen im Standalone-Modus.

Datenbanktabellen wurden bereits erstellt.

## 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 bei Message Queue 4.0 die Version des dateibasierten Datenspeichers weiterhin 370 ist, da keine Änderungen daran 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 Clientanwendungsname
```

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

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

## Funktion ist in zukünftiger Version veraltet

Nachrichtenbasierte Überwachung, mit der Sie einen Broker und seine Ziele mithilfe von Metrikinformationen überwachen können, die in Metrik-Themenziele geschrieben sind, wird in zukünftigen Versionen nicht mehr verwendet.

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 JMX Administration API“ auf Seite 23](#)). 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.

## In Message Queue 4.2 und aktuellen Versionen behobene Fehler

Message Queue 4.2 beinhaltet neue Fehlerkorrekturen und berücksichtigt auch Fehler, die in den Versionen Message Queue 4.1 und Message Queue 4.0 behoben wurden.

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

- [“In Message Queue 4.2 behobene Fehler“ auf Seite 27](#)
- [“In Message Queue 4.1 behobene Fehler“ auf Seite 29](#)
- [“Behobene Probleme in Message Queue 4.0“ auf Seite 30](#)

## In Message Queue 4.2 behobene Fehler

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

**Hinweis** – Die folgende Tabelle sollte die nennenswerten Fehler enthalten, die in 4.1 Patch 1 und 4.1 Patch 2 behoben wurden. Ich brauche Hilfe bei der Ermittlung dieser Fehler. Gibt es darüber hinaus weitere Fehlerkorrekturen, die die soeben erwähnten nicht beinhalten?

TABELLE 1–7 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 unkar.
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 Deisnallation 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.

TABELLE 1-7 In Message Queue 4.2 behobene Fehler (Fortsetzung)

Fehler	Beschreibung
6683897	Der Zusammenfassungsbildschirm des Message Queue-Installationsprogramms meldet einen Konfigurationsfehler, obwohl die Konfiguration erfolgreich durchgeführt worden zu sein scheint: Installationsprogramm kann auf einigen Computern nicht <code>/dev/sterr</code> schreiben.
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 1-8 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.

TABELLE 1–8 In Message Queue 4.1 behobene Fehler (Fortsetzung)

Fehler	Beschreibung
6517341	Die Client-Laufzeit muss die Logik für die Verbindungswiederherstellung verbessern, sobald der Client mit einem Hochverfügbarkeits-Broker-Cluster verbunden ist, indem der Client die Verbindung unabhängig vom Wert der Eigenschaft <code>imqReconnectEnabled</code> wiederherstellen kann.
6528736	Automatischer Windows-Startdienst ( <code>imqbrokersvc</code> ) 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 1–9 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>JMSUserID</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.

TABELLE 1-9 Behobene Probleme in Message Queue 4.0 (Fortsetzung)

Fehlernummer	Beschreibung
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> )
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 Leistungsentpass.
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 4.2

Dieser Abschnitt enthält Informationen zu Dokumentationsaktualisierungen in Message Queue 4.2:

- “Kompatibilitätsprobleme“ auf Seite 31
- “Änderungen am Message Queue 4.2-Dokumentationssatz“ auf Seite 32
- “Neue Zielmetrik“ auf Seite 33
- “Automatischer Broker-Start unter Solaris 10 OS“ auf Seite 34
- “Änderungen an der JMX-API“ auf Seite 35
- “Unterstützung für DN-Benutzernamensformat für Client-Authentifizierung“ auf Seite 40
- “Erweiterung zur JAAS-Authentifizierung“ auf Seite 41

## Kompatibilitätsprobleme

In diesem Abschnitt werden bekannte Kompatibilitätsprobleme in Bezug auf Message Queue 4.2 beschrieben.

## Schnittstellenstabilität

Sun Java System 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 Java System Message Queue 4.2 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 Java System 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 4.2-Dokumentationssatz

Der Message Queue 4.2-Dokumentationssatz enthält Aktualisierungen für den Message Queue 4.1-Dokumentationssatz, wie unten beschrieben:

### Technischer Überblick

Das Handbuch *Sun Java System Message Queue 4.2 Installation Guide* wurde aktualisiert und berücksichtigt nun die neuen Funktionen in Message Queue 4.2 und ein aktualisiertes Framework für Hochverfügbarkeits-Broker-Cluster.

### Administration Guide

Das Handbuch *Administration Guide* wurde aktualisiert und berücksichtigt nun die neuen Funktionen in Message Queue 4.2.



## Installations- und Upgradeinformationen

Das Handbuch *Sun Java System Message Queue 4.2 Installation Guide* wurde nicht zur Berücksichtigung der neuen Funktionen in Message Queue 4.2, insbesondere der Sun Connect-Registrierungsfunktion im Installationsprogramm, aktualisiert. Diese Informationen sind in den vorliegenden *Message Queue - Versionshinweise* enthalten

## Developer's Guide for Java Clients

Das Handbuch *Developer's Guide for Java Clients* wurde nicht zur Berücksichtigung der neuen Funktionen in Message Queue 4.2 aktualisiert. Diese Informationen sind in den vorliegenden *Message Queue - Versionshinweise* enthalten

## Developer's Guide for C Clients

Das Handbuch *Developer's Guide for C Clients* wurde nicht zur Berücksichtigung der neuen Funktionen in Message Queue 4.2 aktualisiert. Diese Informationen sind in den vorliegenden *Message Queue - Versionshinweise* enthalten

## Developer's Guide for JMX Clients

wurde nicht zur Berücksichtigung der neuen Funktionen in Message Queue 4.2 aktualisiert. Diese Informationen sind in den vorliegenden *Message Queue - Versionshinweise* enthalten

## Neue Zielmetrik

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 werden Ziele an alle Broker im Cluster weitergegeben. Nachrichten werden bei ihrer Erstellung jedoch im Ziel des Home-Brokers des Nachrichtenproduzenten gespeichert und nur dann an das entsprechende Ziel auf einem anderen Brokers im Cluster gesendet, wenn ein aktiver Konsument für dieses Ziel vorhanden ist. In Folge sind die in einem angegebenen Ziel gespeicherten Nachrichten von dem Broker in dem Cluster abhängig, in dem sich das Ziel befindet.

Anders ausgedrückt: 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.

In der folgenden Tabelle sind zwei neue metrische Größen für physische Ziele in Message Queue 4.2 dargestellt. Die neuen metrischen Größen sind über die Befehle `mqcmd list dst` und `mqcmd query dst` sowie über die neuen JMX-Attribute verfügbar (siehe ["Zielüberwachungs-MBean"](#) auf Seite 37).

TABELLE 1–10 Metriken für physische Ziele

Metrische Größe	Beschreibung	metrics		Metrikthema
		Protokolldatei	Metriktyp	
Num messages remote (Anzahl Remote-Nachrichten)	Anzahl der aktuell im Arbeitsspeicher und persistenten Speicher gespeicherten Nachrichten, die für einen Remote-Broker in einem Cluster produziert wurden. In diesem Wert sind Nachrichten in Transaktionen nicht enthalten.	Nein	Nicht verfügbar <sup>1</sup>	Nicht verfügbar
Total message bytes remote (Bytezahl Remote-Nachrichten insgesamt)	Gesamtgröße (in Byte) der aktuell im Arbeitsspeicher und persistenten Speicher gespeicherten Nachrichten, die für einen Remote-Broker in einem Cluster produziert wurden. In diesem Wert sind Nachrichten in Transaktionen nicht enthalten.	Nein	Nicht verfügbar <sup>1</sup>	Nicht verfügbar

<sup>1</sup> Verfügbar mit dem Befehl `imqcmd query dst`

## Automatischer Broker-Start unter Solaris 10 OS

In diesem Abschnitt wird die Konfiguration des automatischen Broker-Starts beim Betriebssystem Solaris 10 beschrieben. Anstatt eine rc-Datei zur Implementierung des automatischen Broker-Starts beim Neustart eines Computers zu implementieren, nutzt das folgende Verfahren die Solaris 10 Service Management Facility (SMF).

Weitere Informationen zur Verwendung der Service Management Facility finden Sie in der Dokumentation zu Solaris 10.

### ▼ So implementieren Sie den automatischen Broker-Start unter dem Betriebssystem Solaris 10:

#### 1 Importieren Sie den mqbroker-Dienst in das SMF-Repository.

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

## 2 Vergewissern Sie sich, dass der Importvorgang erfolgreich war, indem Sie den Status des mqbroker-Diensts überprüfen.

```
# svcs mqbroker
```

Die Ausgabe ähnelt Folgendem:

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

Der Dienst wird ursprünglich als deaktiviert angezeigt.

## 3 Aktivieren Sie den mqbroker-Dienst.

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

Durch die Aktivierung des mqbroker-Diensts wird der Prozess imqbrokerd gestartet. Anschließend wird der Broker durch einen Neustart des Computers neu gestartet.

## 4 Konfigurieren Sie den mqbroker-Dienst so, dass alle gewünschten Argumente an den Befehl imqbrokerd weitergeleitet werden.

Die Eigenschaft options/server\_args wird zur Weiterleitung von Argumenten an imqbrokerd verwendet. Gehen Sie beispielsweise wie folgt vor, um -loglevel DEBUGHIGH hinzuzufügen:

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

# Änderungen an der JMX-API

Message Queue unterstützt die Java Management Extensions-(JMX-)API zur programmatischen Konfiguration und Überwachung von Broker-Funktionen aus einer Message Queue-Client-Anwendung. Message Queue 4.2 beinhaltet Erweiterungen an der JMX-API zur Unterstützung neuer Funktionen und Funktionalitäten in der Version. Neue JMX-Attribute, Vorgänge und/oder Nachschlageschlüssel sind für folgende Mbeans definiert:

- “ConsumerManager-Überwachungs-MBean“ auf Seite 35
- “Zielkonfigurations-MBean“ auf Seite 36
- “Zielmanager-Konfigurations-MBean“ auf Seite 37
- “Zielüberwachungs-MBean“ auf Seite 37
- “ProducerManager-Überwachungs-MBean“ auf Seite 39

## ConsumerManager-Überwachungs-MBean

Die Attribute, Vorgänge und Nachschlageschlüssel in folgenden Tabellen unterstützen die in “Mehrere Ziele für einen Herausgeber bzw. Abonnenten“ auf Seite 11 beschriebene Funktion.

Der Name des folgenden Attributs ist als statische Konstante in der Dienstprogrammklasse `com.sun.messaging.jms.management.server.ConsumerAttributes` definiert.

TABELLE 1-11 ConsumerManager-Überwachungsattribute

Name	Typ	Festlegbar?	Beschreibung
<code>NumWildcardConsumers</code>	Integer	Nein	Anzahl der Platzhalter-Nachrichtenkonsumenten, die dem Broker zugeordnet sind

Die Namen der folgenden Vorgänge sind als statische Konstanten in der Dienstprogrammklasse `com.sun.messaging.jms.management.server.ConsumerOperations` definiert.

TABELLE 1-12 ConsumerManager-Überwachungsvorgänge

Name	Parameter	Ergebnistyp	Beschreibung
<code>getConsumerWildcards</code>	none	String[]	Wildcard-Zeichenfolgen die von den aktuellen Konsumenten verwendet werden, die dem Broker zugeordnet sind
<code>getNumWildcardConsumers</code>	wildcard-String	Integer	Anzahl der aktuellen Konsumenten, die dem Broker zugeordnet sind und die angegebene Platzhalterzeichenfolge verwenden

Folgende Nachschlageschlüssel sind als statische Konstanten in der Dienstprogrammklasse `com.sun.messaging.jms.management.server.ConsumerInfo` definiert.

TABELLE 1-13 Nachschlageschlüssel für Nachrichtenkonsumenteninformationen

Name	Werttyp	Beschreibung
<code>DestinationNames</code>	String[]	Zielnamen, die mit von Platzhalter-Konsumenten verwendeten Platzhaltern übereinstimmen. Nur für Themenziele.
<code>Wildcard</code>	Boolean	Platzhalter-Konsument? Nur für Themenziele.

## Zielkonfigurations-MBean

Die Attribute in der folgenden Tabelle unterstützen die in [“Schemavalidierung von XML-Payload-Meldungen“](#) auf Seite 13 beschriebene Funktion.

Die Namen der folgenden Attribute sind als statische Konstanten in der Dienstprogrammklasse `com.sun.messaging.jms.management.server.DestinationAttributes` definiert.

TABELLE 1-14 Zielkonfigurationsattribute

Name	Typ	Festlegbar?	Beschreibung
ValidateXMLSchemaEnabled	Boolean	Ja	<p>Ist XML-Schemavalidierung aktiviert?</p> <p>Wenn dieser Wert auf false (falsch) gesetzt oder gar nicht festgelegt wurde, wird die XML-Schemavalidierung nicht für das Ziel aktiviert.</p>
XMLSchemaURLList	String	Ja	<p>Leerzeichengetrennte Liste der URI-Zeichenfolgen für XML Schema Document (XSD)</p> <p>Die URIs verweisen auf den Speicherort einer oder mehrerer XSDs zur Verwendung mit der XML-Schemavalidierung, sofern aktiviert</p> <p>Verwenden Sie doppelte Leerzeichen um diesen Wert, wenn mehrere URIs angegeben werden.</p> <p>Beispiel:</p> <pre>“http://foo/flap.xsd http://test.com/test.xsd”</pre> <p>Wenn diese Eigenschaft nicht festgelegt oder NULL ist und die XML-Validierung aktiviert wurde, wird die XML-Validierung mithilfe eines im XML-Dokument angegebenen DTD durchgeführt.</p>
ReloadXMLSchemaOnFailure	Boolean	Ja	<p>Erneutes Laden des XML-Schemas bei Versagen aktiviert?</p> <p>Wenn dieser Wert auf false (falsch) gesetzt oder gar nicht festgelegt wurde, wird das Schema nicht erneut geladen, wenn die Validierung fehlschlägt.</p>

## Zielmanager-Konfigurations-MBean

Die oben angegebenen neuen Zielkonfigurations-MBean-Attribute, die die neue Funktion, [“Schemavalidierung von XML-Payload-Meldungen“ auf Seite 13](#) unterstützen, können zum Erstellen eines Ziels mithilfe des Vorgangs create des Zielmanager-Konfigurations-MBean verwendet werden.

## Zielüberwachungs-MBean

Der erste Attributsatz in der folgenden Tabelle unterstützt die in [“Mehrere Ziele für einen Herausgeber bzw. Abonnenten“ auf Seite 11](#) beschriebene Funktion und der zweite Attributsatz unterstützt die in [“Neue Zielmetrik“ auf Seite 33](#) beschriebene Erweiterung.

Die Namen der folgenden Attribute sind als statische Konstanten in der Dienstprogrammklasse `com.sun.messaging.jms.management.server.DestinationAttributes` definiert.

TABELLE 1–15 Zielüberwachungsattribute

Name	Typ	Festlegbar?	Beschreibung
<code>NumWildcards</code>	Integer	Nein	Aktuelle Anzahl an Platzhalter-Nachrichtenproduzenten und Platzhalter-Nachrichtenkonsumenten, die dem Ziel zugeordnet sind  Nur für Themenziele.
<code>NumWildcardProducers</code>	Integer	Nein	Aktuelle Anzahl an Platzhalter-Nachrichtenproduzenten, die dem Ziel zugeordnet sind  Nur für Themenziele.
<code>NumWildcardConsumers</code>	Integer	Nein	Aktuelle Anzahl an Platzhalter-Nachrichtenkonsumenten, die dem Ziel zugeordnet sind  Nur für Themenziele.
<code>NumMsgsRemote</code>	Long	Nein	Aktuelle Anzahl der im Arbeitsspeicher und persistenten Speicher gespeicherten Nachrichten, die für einen Remote-Broker in einem Cluster produziert wurden. In diesem Wert sind Nachrichten in Transaktionen nicht enthalten.
<code>TotalMsgBytesRemote</code>	Long	Nein	Gesamtgröße (in Byte) der aktuell im Arbeitsspeicher und persistenten Speicher gespeicherten Nachrichten, die für einen Remote-Broker in einem Cluster produziert wurden. In diesem Wert sind Nachrichten in Transaktionen nicht enthalten.

Die Vorgänge in der folgenden Tabelle unterstützen die in [“Mehrere Ziele für einen Herausgeber bzw. Abonnenten“](#) auf Seite 11 beschriebene Funktion.

Die Namen der folgenden Vorgänge sind als statische Konstanten in der Dienstprogrammklasse `com.sun.messaging.jms.management.server.DestinationOperations` definiert.

TABELLE 1-16 Zielüberwachungsvorgänge

Name	Parameter	Ergebnistyp	Beschreibung
getWildcardcards	none	String[]	Platzhalterzeichenketten, die von aktuellen Konsumenten und Produzenten verwendet werden, die dem Ziel zugeordnet sind  Nur für Themenziele.
getConsumerWildcardcards	none	String[]	Wildcard-Zeichenfolgen die von den aktuellen Konsumenten verwendet werden, die dem Ziel zugeordnet sind  Nur für Themenziele.
getProducerWildcardcards	none	String[]	Wildcard-Zeichenfolgen die von den aktuellen Produzenten verwendet werden, die dem Ziel zugeordnet sind  Nur für Themenziele.
getNumWildcardConsumers	wildcard-String	Integer	Anzahl der aktuellen Konsumenten, die dem Ziel zugeordnet sind und die angegebene Platzhalterzeichenfolge verwenden  Nur für Themenziele.
getNumWildcardProducers	wildcard-String	Integer	Anzahl der aktuellen Produzenten, die dem Ziel zugeordnet sind und die angegebene Platzhalterzeichenfolge verwenden  Nur für Themenziele.

## ProducerManager-Überwachungs-MBean

Die Attribute, Vorgänge und Nachschlageschlüssel in den unten stehenden Tabellen unterstützen die in [“Mehrere Ziele für einen Herausgeber bzw. Abonnenten“](#) auf Seite 11 beschriebene Funktion.

Der Name des folgenden Attributs ist als statische Konstante in der Dienstprogrammklasse `com.sun.messaging.jms.management.server.ProducerAttributes` definiert.

TABELLE 1-17 ProducerManager-Überwachungsattribute

Name	Typ	Festlegbar?	Beschreibung
NumWildcardProducers	Integer	Nein	Anzahl der Platzhalter-Nachrichtenproduzenten, die dem Broker zugeordnet sind

Die Namen der folgenden Vorgänge sind als statische Konstanten in der Dienstprogrammklasse `com.sun.messaging.jms.management.server.ProducerOperations` definiert.

TABELLE 1-18 ProducerManager-Überwachungsvorgänge

Name	Parameter	Ergebnistyp	Beschreibung
<code>getProducerWildcards</code>	<code>none</code>	<code>String[]</code>	Wildcard-Zeichenfolgen die von den aktuellen Produzenten verwendet werden, die dem Broker zugeordnet sind
<code>getNumWildcardProducers</code>	<code>wildcard-String</code>	<code>Integer</code>	Anzahl der aktuellen Produzenten, die dem Broker zugeordnet sind und die angegebene Platzhalterzeichenfolge verwenden

The following lookup keys are defined as static constants in the utility class `com.sun.messaging.jms.management.server.ProducerInfo`.

TABELLE 1-19 Nachschlageschlüssel für Nachrichtenproduzenteninformationen

Name	Werttyp	Beschreibung
<code>DestinationNames</code>	<code>String[]</code>	Zielnamen, die mit von Platzhalter-Produzenten verwendeten Platzhaltern übereinstimmen.  Nur für Themenziele.
<code>Wildcard</code>	<code>Boolean</code>	Platzhalter-Produzent?  Nur für Themenziele.

## Unterstützung für DN-Benutzernamensformat für Client-Authentifizierung

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

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

Mit dieser Eigenschaft kann der Brokert 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 zur JAAS-Authentifizierung

Die JAAS-Authentifizierung von Message Queue 4.2 unterstützt Authentifizierung durch IP-Adressen sowie durch Benutzernamen.

## Bekannte Probleme und Beschränkungen

In diesem Abschnitt werden die bekannten Probleme mit Message Queue 4.2 aufgeführt. Die folgenden Produktbereiche werden besprochen:

- "Installationsprobleme" auf Seite 41
- "Veraltete Passwortoption" auf Seite 47
- "Administration/Konfiguration" auf Seite 48
- "Broker-Probleme" auf Seite 49
- "Broker-Cluster" auf Seite 51
- "JMX-Probleme" auf Seite 53
- "SOAP-Unterstützung" auf Seite 53

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 4.2 beschrieben.

## Produktregistrierung und Java ES

Message Queue 4.2 wird ebenso wie Message Queue 4.1 von einem relativ neuen Installationsprogramm installiert, das auch zur Installation und Aufrüstung 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 mithilfe des Message Queue-Installationsprogramms auf Message Queue 4.2 aufgerüstet wird, ist der Status der Java ES-Produktregistrierung möglicherweise inkonsistent. Dies hat zur Folge, dass beim Ausführen des Java ES-Deinstallationsprogramms eventuell Message Queue 4.2 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 mit dem Java ES-Deinstallationsprogramm Message Queue sowie die gemeinsam genutzten Komponenten.
2. Installieren Sie mit dem Message Queue-Installationsprogramm Message Queue 4.2.

## Installation unter Windows

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

- 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 Java System Message Queue 4.2 Administration Guide](#) beschrieben.

- Das Installationsprogramm fügt das Verzeichnis `IMQ_HOME\mq\bin` nicht zur `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`) aufrufen.

- Das Installationsprogramm fügt keine Einträge zur Windows-Registrierung hinzu, die darauf hinweisen, dass Message Queue installiert ist. (*Fehler 6586389*)
- 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*)

- Beim Versuch, das Installationsprogramm im Textmodus (`installer -t`) unter Windows auszuführen, wird eine Fehlermeldung in englischer Sprache ausgegeben, auch wenn das Installationsprogramm in einer anderen Sprache als Englisch ausgeführt wird. Der Textmodus wird unter Windows nicht unterstützt. (*Fehler 6594142*)
- Das Installationsprogramm installiert Message Queue nicht standardmäßig auf demselben Laufwerk, auf dem das Betriebssystem 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 kann Message Queue nicht unter `C:\Programme` installiert werden, es sei denn die Installation erfolgt als Administrator über eine Eingabeaufforderung. (*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*)

## Installation unter Solaris

- Wenn das Installationsprogramm im Testlaufmodus (`installer -n`) ausgeführt wird, werden auf der Zusammenfassungsseite mehrere Fehlermeldungen angezeigt, und dass der Installationsstatus nicht abgeschlossen wurde. 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*)

- Wenn das Installationsprogramm im Textmodus ausgeführt wird, kann bei der Eingabe von Benutzernamen oder Passwort für die Sun Connect-Registrierung oder zur Erstellung eines Online-Kontos der Benutzername bzw. das Kennwort nicht mit der Rücktaste korrigiert werden. (Fehler 6673460)  
Workaround: Verwenden Sie statt der Rücktaste die Tastenkombination `Strg-H`, oder verwenden Sie einen anderen Terminal-Emulator wie `dtterm` oder `xterm`.
- Der Upgrade-Bildschirm des Installationsprogramms gibt nicht immer die bestehende installierte Version von Message Queue oder des Installationsmoduls korrekt an. (Fehler 6679765)
- Bei Verwendung des Installationsprogramms im Textmodus und dem Versuch, die Sun Connection-Registrierung mit ungültigem Benutzernamen und Passwort durchzuführen, zeigt das Installationsprogramm ein Dialogfeld an, das besagt, dass die Registrierung nicht möglich ist, gibt eine NULL-Zeiger-Ausnahme aus, und wird beendet. (Fehler 6666365)

## Installation unter Linux

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

- 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)

## 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 4.2 angezeigt. (*Fehler 6650841*)
- Solange das Installationsprogramm die Installation von Message Queue 4.2 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 sich auf einem Computersystem ältere Versionen von Message Queue und NSS/NSPR befinden, listet der Upgrade-Bildschirm des Installationsprogramm nur Message Queue als Element auf, für das ein Upgrade erforderlich ist. Es wird nicht darauf hingewiesen, dass auch NSS und NSPR aufgerüstet werden müssen. Es werden dennoch alle relevanten Softwarekomponenten aufgerüstet (wie im Bildschirm ReadyToInstall angegeben, der die richtigen Informationen anzeigt). (*Fehler 6580696*)
- Liste der JDKs im JDK-Auswahlbildschirm ist aktiv, selbst wenn die Option zur Auswahl eines JDK nicht ausgewählt wurde. (*Fehler 6650874*)

## Versionsanomalien im Installationsprogramm

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

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

TABELLE 1–20 Übersetzung der Versionszeichenfolge

Vom Installation unter Solaris OS angezeigte Version	Zugehörige Message Queue-Version
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

TABELLE 1–20 Übersetzung der Versionszeichenfolge (Fortsetzung)

Vom Installation unter Solaris OS angezeigte Version	Zugehörige Message Queue-Version
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.

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

*Hauptversionsnummer . Unterversionsnummer - bedeutungslose Zahl*

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)
- Der Textmodus (`installer -t`) wird unter Windows nicht unterstützt. Beim Ausführen des Installationsprogramms im Textmodus unter Windows wird eine Fehlermeldung angezeigt. Diese Meldung ist nicht lokalisiert, wenn das Installationsprogramm mit einem anderen Gebietsschema als Englisch ausgeführt wird. (Fehler 6594142)

- 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 `-p`- oder `-password`-Option 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 Befehloption `-passwordfile` 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=Passwortdateiverzeichnis  
imq.passfile.name=Passwortdateiname
```

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

## Administration/Konfiguration

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

- 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` and `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*)

## Broker-Probleme

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

- 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 die Fehlermeldung ausgegeben, dass das Paket für die Abfrage zu groß ist (SQLException). (*Fehler 6682815*)

*Umgehung:* Starten sie den MySQL-Server mit der Option `--max_allowed_packet` von mehr als dem Standardwert 1 MB. Verwenden Sie beispielsweise folgenden Wert:

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

- 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) (MESSAGE_ID)
```

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

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

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

## Broker-Cluster

Folgende Probleme betreffen Broker-Cluster.

- 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 Hochverfügbarkeits-Broker-Cluster verbunden ist, versucht die Client-Laufzeit solange, eine Verbindung herzustellen, bis sie Erfolg hat (der Wert im Verbindungsfactory-Attribut `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 Warteschlangesenden und Meldungen von jeder beliebigen Warteschlange im Cluster konsumieren. Die Einschränkung betrifft nur das Durchsuchen der Warteschlange.
- In einem konventionellen Cluster, die Broker mit Version 4.2 enthält, müssen alle Broker Version 3.5 oder höher aufweisen.
- Message Queue 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* Ändern Sie bei den Message Queue 4.2- und 4.1-Brokern `imq.autocreate.queue.maxNumActiveConsumers` vom Standardwert `-1` auf den Standardwert der vorherigen Version, `1`.

- Bei der Konvertierung aus einem konventionellen Cluster in einen Hochverfügbarkeits-Cluster können Sie mit dem Datenbank-Manager-Dienstprogramm von Message Queue (`imqdbmgr`) einen bestehenden eigenständigen JDBC—basierten Datenspeicher in einen gemeinsam genutzten Hochverfügbarkeits-Datenspeicher konvertieren, wie in [“Converting a Standalone Data Store to a Shared Data Store“](#) in *Sun Java System Message Queue 4.2 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 hasstore` 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: Kein Tabellen-Arbeitsspeicher mehr verfügbar." (*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 werden die Meldungen an eine entfernte Verbindung, die noch nicht gestartet wurde, in die Warteschlange gestellt. (*Fehler 4951010*)

*Umgehung:* Der Konsument erhält die Meldungen, sobald die Verbindung gestartet wurde. Die Nachrichten werden an einen anderen Konsument gesendet, wenn die Verbindung beendet wird.

- 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]:
Nachrichtenbestätigung fehlgeschlagen aus
mq://129.145.130.95:7677/?instName=a&brokerSessionUID=3209681167602264320:
  ackStatus = NOT_FOUND(404)\
  Ursache = Aktualisieren von Remote-Transaktionsstatus auf COMMITTED(6):
Transaktion 3534784765719091968 nicht gefunden, die Transaktion
wurde möglicherweise bereits übermittelt.
  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 Benachrichtigung zur Übermittlung von Transaktion
[8-129.145.130.95(95:fd:93:91:ec:a0)-33220-1185481094690,
[consumer:3534784765719133952, type=NONE]]
TUID=3534784765719091968 erhielt Antwort:
com.sun.messaging.jmq.jmsserver.util.BrokerException:
Aktualisieren von Remote-Transaktionsstatus auf COMMITTED(6):
  Transaktion 3534784765719091968 nicht gefunden, die Transaktion wurde möglicherweise
  bereits übermittelt:
com.sun.messaging.jmq.jmsserver.util.BrokerException: Aktualisieren von Remote-Transaktionsstatus
auf COMMITTED(6): Transaktion 3534784765719091968 nicht gefunden, die Transaktion wurde möglicherweise
bereits übermittelt.
```

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.
- Die SOAP-Entwicklung hängt von mehreren Dateien ab: `SUNWjaf`, `SUNWjmail`, `SUNWxsrt` und `SUNWjxap`. 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 `mail.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 Java System Message Queue 4.2 enthält die folgenden Dateisätze, die Sie verwenden und im Binärformat verteilen können:

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

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 Anwendungsversionen finden Sie unter: <http://sun.com/software/javaenterprisesystem/get.html>.

Informationen zum Einsatz von Sun für Eingabehilfen erhalten Sie unter <http://sun.com/access>.

## Problemmeldungen und Feedback

Wenn Sie mit Sun Java System 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 Java System-Software-Forum

Unter der folgenden Adresse steht ein Sun Java System 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 bemht sich um eine stetige Verbesserung der Dokumentationen und ist deshalb an Ihrer Meinung und Ihren Anregungen interessiert.

Wenn Sie Kommentare abgeben möchten, rufen Sie die Seite <http://docs.sun.com> und klicken Sie auf "Kommentare senden". 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 zu Sun Java System Message Queue 4.2, die Teilenummer lautet 820-3701.

## Weitere Quellen von Sun

Nützliche Informationen über Sun Java System 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>