

Sun GlassFish Message Queue 4.4 Update 1 リリースノート



Part No: 821-1514-11
2009年12月

Sun Microsystems, Inc. (以下米国Sun Microsystems社とします)は、本書に記述されている製品に含まれる技術に関連する知的財産権を所有します。特に、この知的財産権はひとつかそれ以上の米国における特許、あるいは米国およびその他の国において申請中の特許を含んでいることがあります。が、それらに限定されるものではありません。

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者によって開発された素材を含んでいることがあります。

本製品のは、カリフォルニア大学からライセンスされているBerkeleyBSDシステムに基づいていることがあります。は、X/Open Company, Ltd.が独占的にライセンスしている米国ならびに他の国における登録商標です。

un、Sun Microsystems、Sunのロゴマーク、Solarisのロゴマーク、Java CoffeeCupのロゴマーク、docs.sun.com、Javaおよびは、米国およびその他の国における米国Sun Microsystems社の商標、登録商標もしくは、サービスマークです。すべてのSPARCは、米国SPARC International, Inc.のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC商標が付いたは、米国Sun Microsystems社が開発したアーキテクチャに基づくものです。

OPENLOOKおよびSunTMGraphical Userは、米国Sun Microsystems社が自社のユーザおよびライセンス実施権者向けに開発しました。米国Sun Microsystems社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国Xerox社の先駆者としての成果を認めるものです。米国Sun Microsystems社は米国Xerox社からXerox Graphical User Interfaceの非独占的ライセンスを取得しており、このは、OPENLOOKのグラフィカル・ユーザインタフェースを実装するか、またはその他の方法で米国Sun Microsystems社との書面によるライセンス契約を遵守する、米国Sun Microsystems社のライセンス実施権者にも適用されます。

本書で言及されている製品や含まれているは、米国輸出規制法で規制されるものであり、その他の国の輸出入に関する法律の対象となることがあります。核、ミサイル、化学あるいは生物兵器、原子力の海洋輸送手段へのは、直接および間接を問わず厳しく禁止されています。米国が禁輸の対象としているや、限定はされませんが、取引禁止顧客や特別指定国民のリストを含む米国輸出排除リストで指定されているものへの輸出および再輸出は厳しく禁止されています。

本書は、「現状」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されな、明示的であるか黙示的であるかを問わない、なんらの保証も行われなものとします。

目次

1	リリースノート	5
	リリースノート改訂履歴	6
	メッセージキュー 4.4 Update 1 の概要	7
	メッセージキュー 4.4 Update 1 でサポートされるプラットフォームとコンポーネント	7
	オペレーティングシステムプラットフォームのサポート	7
	システムの仮想化のサポート	8
	コンポーネントの依存関係	8
	メッセージキュー 4.4 Update 1 の新機能および最近のリリース	10
	メッセージキュー 4.4 Update 1 の新機能	11
	メッセージキュー 4.4 の新機能	11
	メッセージキュー 4.3 の新機能	14
	メッセージキュー 4.2 の新機能	20
	メッセージキュー 4.1 の新機能	23
	メッセージキュー 4.0 の新機能	27
	今後のリリースで使用されなくなる機能	31
	メッセージキュー 4.4 Update 1 で修正されたバグおよび最近のリリース	32
	メッセージキュー 4.4 Update 1 で修正されたバグ	32
	メッセージキュー 4.4 で修正されたバグ	35
	メッセージキュー 4.3 で修正されたバグ	37
	メッセージキュー 4.2 で修正されたバグ	38
	メッセージキュー 4.1 で修正されたバグ	39
	メッセージキュー 4.0 で修正されたバグ	40
	メッセージキュー 4.4 Update 1 のドキュメントの更新	41
	互換性の問題	41
	メッセージキュー 4.4 Update 1 ドキュメントセットの変更点	42
	既知の問題点と制限事項	43
	インストールに関する情報	43
	使用されなくなったパスワードオプション	50

管理および設定上の問題	51
ブローカの問題	52
ブローカクラスタ	54
JMX の問題	57
SOAP サポート	57
再配布可能なファイル	57
障害を持つユーザー向けのアクセシビリティ機能	58
問題の報告とフィードバックの方法	58
Sun GlassFish Software Forum	58
Java Technology Forum	59
このドキュメントに関するコメント	59
その他の情報	59

リリースノート

Version 4.4 Update 1

Part No. 821-1514-11

このリリースノートには、Sun GlassFish メッセージキュー 4.4 Update 1 のリリース時点で入手できる重要な情報が含まれています。ここでは、新機能、拡張機能、既知の問題、制限事項などについて説明します。メッセージキュー 4.4 Update 1 をご使用になる前に、このリリースノートをお読みください。

このリリースノートには、メッセージキュー 4.4、4.3、4.2、4.1 および 4.0 のリリースに関する情報も含まれています。たとえば、各リリースで導入された機能については、20 ページの「メッセージキュー 4.2 の新機能」、23 ページの「メッセージキュー 4.1 の新機能」、および 27 ページの「メッセージキュー 4.0 の新機能」をそれぞれ参照してください。

これらのリリースノートの最新版は、Sun GlassFish メッセージキュー ドキュメント Web サイト (<http://docs.sun.com/coll/1307.7>) から入手できます。ソフトウェアをインストールおよび設定する前だけでなく、それ以降も定期的にこの Web サイトをチェックして、最新のリリースノートと製品ドキュメントを確認してください。

このリリースノートは、次の節で構成されています。

- 6 ページの「リリースノート改訂履歴」
- 7 ページの「メッセージキュー 4.4 Update 1 の概要」
- 7 ページの「メッセージキュー 4.4 Update 1 でサポートされるプラットフォームとコンポーネント」
- 10 ページの「メッセージキュー 4.4 Update 1 の新機能および最近のリリース」
- 31 ページの「今後のリリースで使用されなくなる機能」
- 32 ページの「メッセージキュー 4.4 Update 1 で修正されたバグおよび最近のリリース」
- 41 ページの「メッセージキュー 4.4 Update 1 のドキュメントの更新」
- 43 ページの「既知の問題点と制限事項」
- 57 ページの「再配布可能なファイル」

- 58 ページの「障害を持つユーザー向けのアクセシビリティ機能」
- 58 ページの「問題の報告とフィードバックの方法」
- 59 ページの「このドキュメントに関するコメント」
- 59 ページの「その他の情報」

このドキュメント内で参照している第三者の URL は、追加の関連情報を提供しません。

このドキュメント内で引用する第三者の Web サイトの可用性について Sun は責任を負いません。こうしたサイトやリソース上の、またはこれらを通じてな、コンテンツ、広告、製品、その他の素材について、Sun は推奨しているわけではなく、Sun はいかなる責任も負いません。こうしたサイトやリソース上の、またはこれらを経由してな、コンテンツ、製品、サービスを利用または信頼したことによって発生した(あるいは発生したと主張される)いかなる損害や損失についても、Sun は一切の責任を負いません。

リリースノート改訂履歴

次の表は、メッセージキュー製品のすべての 4.x リリースの日付と、各リリースに関連するこのドキュメントの変更内容を示しています。

表 1-1 改訂履歴

改訂日	変更点の説明
2009 年 12 月	メッセージキュー 4.4 Update 1 のこのドキュメントのリリース。このリリースの新機能が追加されています。また、以前のインストールプログラムに関する古いインストールの問題が削除されています。
2009 年 12 月	メッセージキュー 4.4 のこのドキュメントの 2 番目のリリース。誤記と情報の不足が修正されています。
2009 年 10 月	メッセージキュー 4.4 のこのドキュメントのリリース。このリリースの新機能が追加されています。
2009 年 5 月	メッセージキュー 4.4 Beta のこのドキュメントの初回リリース。このリリースの新機能が追加されています。
2008 年 12 月	メッセージキュー 4.3 のこのドキュメントのリリース。リリースの新機能が追加されています。
2008 年 8 月	メッセージキュー 4.2 のこのドキュメントのリリース。リリースの新機能が追加されています。
2007 年 9 月	メッセージキュー 4.1 のこのドキュメントの 3 番目のリリース。Java Enterprise System Monitoring Framework のサポート、固定 C ポート、バグ修正、およびその他の機能に関する説明が追加されています。

改訂日	変更点の説明
2007年4月	メッセージキュー 4.1 Beta のこのドキュメントの2番目のリリース。高可用性の機能が追加されています。
2007年1月	メッセージキュー 4.1 Beta のこのドキュメントの初回リリース。JAAS サポートの説明が追加されています。
2006年5月	メッセージキュー 4.0 のこのドキュメントの初回リリース。

メッセージキュー 4.4 Update 1 の概要

Sun GlassFish メッセージキューは、多くの機能を備えるメッセージサービスで、Java Messaging Specification (JMS) 1.1 に準拠する信頼性の高い非同期メッセージングを提供します。メッセージキューでは、JMS 仕様を超える機能も用意され大規模企業のシステム配備のニーズにも対応できるようになっています。

メッセージキュー 4.4 Update 1 は、プラットフォームに依存しない Image Packaging System (IPS) に基づいた新しいインストールプログラムを提供するマイナーリリースです。IPS は、pkg(5) システムとも呼ばれています。また、version 4.4 Update 1 には多数の機能拡張とバグ修正も含まれています。

メッセージキュー 4.4 Update 1 でサポートされるプラットフォームとコンポーネント

この節は、メッセージキュー 4.4 Update 1 システム要件に関する次のトピックで構成されています。

- [7 ページの「オペレーティングシステムプラットフォームのサポート」](#)
- [8 ページの「システムの仮想化のサポート」](#)
- [8 ページの「コンポーネントの依存関係」](#)

オペレーティングシステムプラットフォームのサポート

メッセージキュー 4.4 Update 1 は、Solaris、Linux、Windows、および AIX オペレーティングシステムのプラットフォームでサポートされます。表 1-2 に、サポートされる各プラットフォームのバージョンを示します。各プラットフォームのハードウェア要件については、『『Sun Java System Message Queue 4.3 Installation Guide』』を参照してください。

表 1-2 サポートされるプラットフォームのバージョン

プラットフォーム	サポートされるバージョン
Solaris	Solaris 9 (SunOS 5.9)、すべての更新版 (SPARC、x86) Solaris 10 (SunOS 5.10)、すべての更新版 (SPARC、x86、x64)
OpenSolaris	OpenSolaris 2008.11 (x86、x64) OpenSolaris 2009.06 (x86、x64)
Linux	Red Hat Enterprise Linux Advanced Server 4.0、5.0、すべての更新版、32 および 64 ビットバージョン (x86、x64) Red Hat Enterprise Linux Enterprise Server 4.0、5.0、すべての更新版、32 および 64 ビットバージョン (x86、x64)
AIX	AIX 6.1
Windows	Windows Vista Windows XP Professional、SP2 (x86) ¹ Windows Server 2003 Standard Edition、Enterprise Edition、SP2、32 および 64 ビットバージョン (x86、x64) ² Windows Server 2008 Standard Edition、Enterprise Edition、SP2、32 および 64 ビットバージョン (x86、x64) ²

¹ Home Edition、Tablet PC Edition、Media Center Edition はサポートされません

² Web Edition、Small Business Server Edition はサポートされません

システムの仮想化のサポート

システムの仮想化は、複数のオペレーティングシステム (OS) インスタンスを共有ハードウェア上で個別に実行できるようにするテクノロジーです。機能的にいうと、仮想化された環境でホストされる OS に配備されたソフトウェアは、通常はベースとなるプラットフォームが仮想化されていることを認識しません。Sun では、精選されたシステムの仮想化と OS の組み合わせについて、その Sun Java System 製品のテストを行なっています。これは、Sun Java System 製品が、適切な規模と構成の仮想化環境でも、仮想化されていないシステム上の場合と同様に機能することを効果的に確認するためのテストです。仮想化された環境での Sun Java System 製品に対する Sun のサポートの詳細は、<http://docs.sun.com/doc/820-4651> を参照してください。

コンポーネントの依存関係

プラットフォーム固有の要件のほかに、メッセージキューは特定の基本コンポーネントにも依存します。メッセージキュークライアントを開発して実行する場合

は、それらのコンポーネントをインストールしてください。表 1-3 に、それらのコンポーネントを示します。ほかのバージョンまたはベンダー製品も実装できますが、それらは Sun Microsystems によるテストが実施されていないため、公式にはサポートされません。

表 1-3 必須のサポートコンポーネント

コンポーネント	サポート対象	サポートされるバージョン ¹
Java Runtime Environment (JRE)	メッセージキューブローカおよび管理ツール	J2SE™ Runtime Environment 1.5.0 以降 Java™ SE Runtime Environment 1.6.0
Java Software Development Kit (JDK), Standard Edition	Java クライアントの開発および配備	J2SE Development Kit 1.5.0 以降 Java SE Development Kit 1.6.0。JMX ではバージョン 1.6.0_10 以降が必要です。

¹ Sun Microsystems 製品バージョンのみ

表 1-4 に、メッセージキュークライアントのサポートを追加するためにインストールできるその他のコンポーネントを示します。一覧に示されているコンポーネントのすべてが必要とは限りません。たとえば、C クライアントを作成しない場合は、C コンパイラ、C++ 実行時ライブラリ、NSPR、または NSS は必要ありません。

表 1-4 任意のサポートコンポーネント

コンポーネント	サポート対象	サポートされるバージョン
アプリケーションサーバー	HTTP/HTTPS	Sun GlassFish Application Server Enterprise Edition, Version 9.1.1 Sun GlassFish Enterprise Server v2.1、v2.1.1、v3
Web サーバー	HTTP/HTTPS	Sun GlassFish Web Server Enterprise Edition, Version 7.0, Update 3
データベース	JDBC ベースのデータストア	HADB, Versions 4.4.3、4.5、4.6 Java DB (Apache Derby), Version 10.4 MySQL Community/Enterprise Edition, Version 5.1 Oracle 9.2i、10g、および 11g postgresql, Version 8.1 注 - PointBase データベースは、現在はサポートされていません。

表 1-4 任意のサポートコンポーネント (続き)

コンポーネント	サポート対象	サポートされるバージョン
高可用性データベース	高可用性ブローカクラス タ	HADB, Versions 4.4.3、4.5、4.6 MySQL Cluster Edition, Version 5.1.39-ndb-7.0.9 Oracle 10g および 11g
Lightweight Directory Access Protocol (LDAP) ディレクトリサーバー	メッセージキュー ユーザーリポジトリおよび 管理対象オブジェクト	Sun GlassFish Directory Server, Version 6.0
Java Naming and Directory Interface (JNDI)	管理対象オブジェクトの サポートと LDAP ユーザーリポジトリ	JNDI Version 1.2.1 LDAP Service Provider, Version 1.2.2 File System Service Provider, Version 1.2 Beta 3 ¹
C 言語のコンパイラおよび同等の C++ ランタイムライブラリ	メッセージキュー C クラ イアント	Solaris: Sun Studio, Version 12 以降、C++ コン パイラ (標準モード) および C コンパイラ Linux: gcc/g++, Version 3.4.6 Windows: Microsoft Windows Visual Studio, Version 2008 SP1
Netscape Portable Runtime (NSPR)	メッセージキュー C クラ イアント	Version 4.7
Network Security Services (NSS)	メッセージキュー C クラ イアント	Version 3.12.3

¹ 管理対象オブジェクトのサポートのみ。開発とテストでサポートされますが、本稼働環境での開発ではサポートされません

メッセージキュー 4.4 Update 1 の新機能および最近のリリース

メッセージキュー 4.4 Update 1 の新機能と、メッセージキュー 4.x ファミリのこれまでのリリースの新機能を、次の節に示します。

- [11 ページの「メッセージキュー 4.4 Update 1 の新機能」](#)
- [11 ページの「メッセージキュー 4.4 の新機能」](#)
- [14 ページの「メッセージキュー 4.3 の新機能」](#)
- [20 ページの「メッセージキュー 4.2 の新機能」](#)
- [23 ページの「メッセージキュー 4.1 の新機能」](#)
- [27 ページの「メッセージキュー 4.0 の新機能」](#)

メッセージキュー 4.4 Update 1 の新機能

メッセージキュー 4.4 Update 1 は、いくつかの機能拡張とバグ修正が施されたマイナーリリースです。この節では、このリリースに含まれる新しい機能を説明します。

- 11 ページの「新しいインストールプログラム」
- 11 ページの「クラスタのトランザクションログのサポート」
- 11 ページの「インプロセスブローカ」

新しいインストールプログラム

メッセージキュー 4.4 Update 1 では、pkg(5) システムに基づいた新しい複数プラットフォーム用のインストーラが提供されています。pkg システムは、IPS (Image Packaging System) と呼ばれています。このインストーラについては、『[Sun GlassFish Message Queue 4.4 Update 1 Installation Guide](#)』を参照してください。

クラスタのトランザクションログのサポート

メッセージキュー 4.4 Update 1 では、ブローカクラスタをサポートするファイルベースのデータストアに対して、トランザクションの持続メカニズムが追加されました。このメカニズムは、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「[Optimizing File-Based Transaction Persistence](#)」で説明されているように、ほかの機能も提供します。

インプロセスブローカ

メッセージキュー 4.4 Update 1 は、Java クライアント内からのブローカの実行をサポートします。これらのブローカは「インプロセスブローカ」または「組み込みブローカ」と呼ばれ、ブローカを作成および起動した Java クライアントと同じ JVM で動作します。詳細は、『[Sun GlassFish Message Queue 4.4 Developer's Guide for Java Clients](#)』の第 6 章「[Embedding a Message Queue Broker in a Java Client](#)」を参照してください。

メッセージキュー 4.4 の新機能

メッセージキュー 4.4 は、多数の機能拡張とバグ修正を含むマイナーリリースです。この節では、このリリースに含まれる新しい機能を説明します。

- 12 ページの「JMSブリッジサービス」
- 12 ページの「STOMPブリッジサービス」
- 13 ページの「その他の拡張機能」

JMSブリッジサービス

JMS の仕様では、ブローカとクライアント間の通信に使用するワイヤープrotocol が定義されていないため、メッセージキューを含む各 JMS プロバイダは、独自のプロトコルを定義して使用しています。この状況により、JMS プロバイダ間の相互運用性が失われてきました。

メッセージキュー 4.4 の JMS ブリッジサービスは、メッセージキューブローカが自身の送信先を外部 JMS プロバイダにある送信先にマッピングできるようにすることで、相互運用性の問題を解消します。実際にこのマッピングにより、メッセージキューブローカは外部 JMS プロバイダのクライアントと通信できるようになります。

JMS ブリッジサービスは、次のような外部 JMS プロバイダにある送信先のマッピングをサポートします。

- JMS 1.1 に準拠している
- JNDI 管理オブジェクトをサポートしている
- `javax.jms.ConnectionFactory` または `javax.jms.XAConnectionFactory` タイプの接続ファクトリを使用している
- トランザクションマッピングで、XA インタフェースをリソースマネージャーとしてサポートしている

オープンソースおよび商用の JMS プロバイダの多くは、これらの要件を満たしています。したがって JMS ブリッジサービスは、ほかの JMS プロバイダを使用している既存のメッセージング環境にメッセージキューを統合するための効果的な方法となります。

JMS ブリッジサービスの詳細は、次の情報を参照してください。

- JMS ブリッジサービスのアーキテクチャー、サブコンポーネント、および機能については、『[Sun GlassFish Message Queue 4.4 Technical Overview](#)』の「[JMS Bridge Service](#)」を参照してください。
- ブローカで JMS ブリッジを設定および管理する方法については、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「[Configuring and Managing JMS Bridge Services](#)」を参照してください。

STOMPブリッジサービス

すでに説明したように、JMS 仕様にはブローカとクライアント間の通信に使用するワイヤープrotocol が定義されていません。STOMP (Streaming Text Oriented Messaging Protocol) オープンソースプロジェクト (<http://stomp.codehaus.org>) は、任意の言語で記述されたクライアントが STOMP プロトコルをサポートするメッセージングプロバイダとの通信に使用できる、単純なワイヤープrotocol を定義します。

メッセージキュー 4.4 は、STOMP ブリッジサービスを通して STOMP プロトコルのサポートを提供します。このサービスにより、メッセージキューブローカは STOMP クライアントと通信できます。

STOMP ブリッジサービスの詳細は、次の情報を参照してください。

- STOMP ブリッジサービスのアーキテクチャーと機能については、『[Sun GlassFish Message Queue 4.4 Technical Overview](#)』の「[STOMP Bridge Service](#)」を参照してください。
- ブローカで STOMP ブリッジを設定および管理する方法については、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「[Configuring and Managing STOMP Bridge Services](#)」を参照してください。

その他の拡張機能

メッセージキュー 4.4 には、次の拡張機能も追加されています。

- 13 ページの「[新しい Universal Message Service \(UMS\) の機能](#)」
- 13 ページの「[IPS パッケージのサポート](#)」
- 14 ページの「[再導入された監査ログ機能](#)」

新しい Universal Message Service (UMS) の機能

UMS には、HTTP GET を使用して次のようなサービスを提供する機能が追加されました。

- **getBrokerInfo**: ブローカに関する情報を取得します。
- **getConfiguration**: UMS 設定に関する情報を取得します。
- **debug**: UMS サーバーのデバッグロギング機能をオンまたはオフにします。
- **ping**: ブローカと通信し、動作を確認します。

これらの新機能については、<https://mq.dev.java.net/4.4-content/imqums/protocol.html> の「[Query and utility functions using HTTP GET](#)」を参照してください。

UMS の概要については、14 ページの「[Universal Message Service \(UMS\)](#)」を参照してください。UMS API のドキュメントについては、<https://mq.dev.java.net/4.4-content/imqums/protocol.html> を参照してください。各言語のプログラミング例については、<https://mq.dev.java.net/4.4-content/imqums/examples/README.html> を参照してください。

IPS パッケージのサポート

現在のメッセージキューは、オープンソースの Image Packaging System (IPS) を使用して配布用にパッケージされています。IPS は、pkg(5) システムとも呼ばれます。このパッケージ方法は、メッセージキューを Sun GlassFish Enterprise Server 2.1.1 に統合するために追加されました。

再導入された監査ログ機能

メッセージキュー 3.7 で提供された監査ログ機能は、メッセージキュー 4.0 で削除されました。この機能がメッセージキュー 4.4 で再導入されました。この機能については、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「[Audit Logging](#)」を参照してください。

メッセージキュー 4.3 の新機能

メッセージキュー 4.3 は、多数の機能拡張とバグ修正を含むマイナーリリースです。この節では、このリリースに含まれる新しい機能を説明します。

- 14 ページの「[Universal Message Service \(UMS\)](#)」
- 17 ページの「[AIX プラットフォームのサポート](#)」
- 18 ページの「[新しい ZIP ベースのインストーラ](#)」
- 18 ページの「[プラットフォームサポートの拡張](#)」
- 18 ページの「[その他の拡張機能](#)」

Universal Message Service (UMS)

メッセージキュー 4.3 では、HTTP 対応デバイスから Message Queue にアクセスするための、新しい UMS (Universal Message Service) とメッセージング API が導入されました。これにより、ほとんどのアプリケーションがほかのアプリケーションと通信できるようになり、JMS メッセージングの信頼性と保証された配信による利益を受けられるようになりました。また、UMS は JMS メッセージングのスケラビリティを拡張し、メッセージングクライアントの数がインターネットクラスの規模に達する場合でも対応できます。

アーキテクチャー

基本的な UMS のアーキテクチャーを次の図に示します。

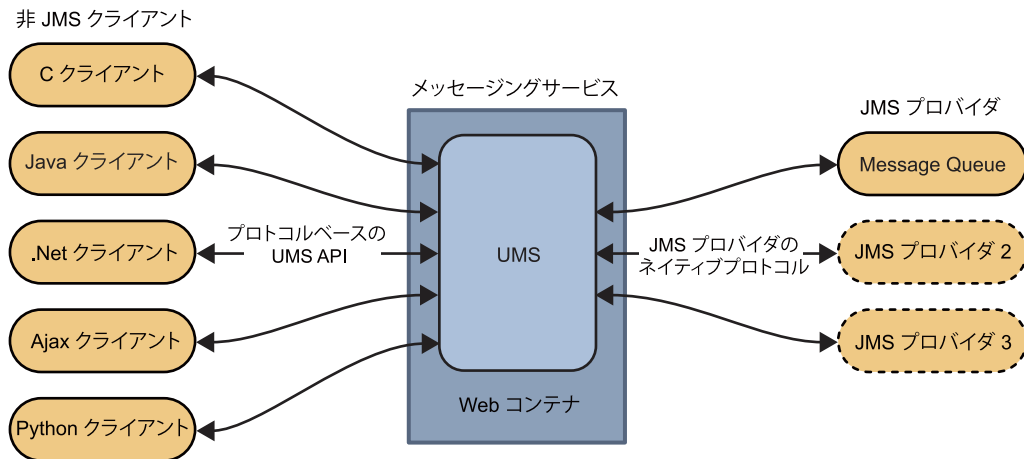


図 1-1 UMS のアーキテクチャー

UMS は Web サーバーで動作し、言語的に中立でプラットフォームに依存しません。UMS は、非 JMS クライアントアプリケーションと JMS プロバイダ間のゲートウェイとして機能します。UMS API を使用して送信されたメッセージを受信し、受信したメッセージを JMS メッセージに変換して、JMS プロバイダのネイティブプロトコルを使用してプロバイダ内の送信先に送信します。同様に、JMS プロバイダの送信先からメッセージを受信してテキストまたは SOAP メッセージに変換し、UMS API を通してクライアントから要求されたとおりに、非 JMS クライアントにメッセージを送信します。

単純で言語に依存しないプロトコルベースの UMS API は、Web ベースのアプリケーションと非 Web ベースのアプリケーションの両方をサポートし、スクリプトおよびプログラミング言語の両方で使用できます。API は、REST (Representational State Transfer) 形式のプロトコルを使用する単純なメッセージング API と、SOAP メッセージヘッダーにプロトコルを埋め込む XML メッセージング API の、2 つの形式で提供されます。どちらの場合も、API は単一の http 要求だけを使用して、メッセージの送信または受信を行います。

UMS API の単純性と柔軟性により、AJAX、.NET、Python、C、Java、およびその他の多くのアプリケーションで、JMS 送信先にテキストメッセージや SOAP (添付ファイルを含む) メッセージを送信したり、JMS 送信先からメッセージを受信することができます。たとえば、Python アプリケーションは .NET アプリケーションと通信でき、iPhone は Java アプリケーションと通信することができます。

メッセージキュー 4.3 では、UMS は JMS プロバイダとしてメッセージキューのみをサポートします。

その他の機能

UMS は、先に述べた単純なゲートウェイとして機能するだけではありません。ステートレスクライアントセッションだけでなくステートフルセッションもサポートします。クライアントから要求された場合、UMS はクライアントアプリケーションのセッション状態を複数のサービス要求間で維持します。UMS でコンテナ管理による認証を使用したり、Message Queue ブローカとクライアントを認証するように設定することもできます。UMS はトランザクションもサポートするため、クライアントアプリケーションは複数のサービス要求を単一の不可分な単位としてコミットまたはロールバックすることができます。

UMS は Message Queue ブローカとの単一の接続上で多数のクライアントをサポートできるため、スケーラビリティが最大となるように、ブローカの接続サービス上の負荷を軽減します。また、UMS の能力を水平方向に拡大することで、インターネットクラスの規模のメッセージング負荷に対応することもできます。

プロトコルベースの UMS API は単純であるため、クライアント側にクライアントライブラリは必要ありません。したがって、将来的に API を拡張して、クライアントアプリケーションをアップグレードすることなく、JMS の追加機能を実装することができます。

UMS の使用法

UMS を使用するには、Servlet 2.4 以降の仕様をサポートする Web コンテナに UMS を配備し、Message Queue ブローカを起動します。次に、適切な送信先を作成し、UMS API を使用してメッセージを送受信するメッセージングアプリケーションを作成します。

メッセージキュー 4.3 の配布に含まれる UMS の `imqums.war` ファイルは、プラットフォームごとに次の場所にインストールされます。

`.war` ファイルの名前は、必要に応じて変更できます。

表 1-5 imqums.war ファイルの場所

プラットフォーム	imqums.war ファイルの場所
Solaris	<code>/usr/share/lib/imq</code>
Linux	<code>/opt/sun/mq/share/lib</code>
AIX	<code>IMQ_HOME/lib</code>
Windows	<code>IMQ_HOME\lib</code>

`imqums.war` を `localhost:port` で Web コンテナに配備した場合、UMS のドキュメントは次の場所で参照できます。

`http://localhost:port/imqums`

または、次の場所でも UMS のドキュメントを参照できます。

- UMS の設定については、<https://mq.dev.java.net/4.3-content/ums/config.html> を参照してください。
- UMS API のドキュメントについては、<https://mq.dev.java.net/4.3-content/ums/protocol.html> を参照してください。
- 各言語のプログラミング例については、<https://mq.dev.java.net/4.3-content/ums/examples/README.html> を参照してください。

サポートされる Web コンテナ

現在、UMS は次の Web コンテナでサポートされます。

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

AIX プラットフォームのサポート

メッセージキュー 4.3 には、AIX プラットフォームのパッケージと、これらをインストールするためのインストーラが用意されています。

メッセージキューの AIX 実装では、次のソフトウェアがサポートされます。

- AIX v6.1 以降 (これより前のバージョンは、Unix/Java のみのバンドルでサポートされます)
- DB2 サポート
- IBM XL C/C++ Compiler V9.0
- JDK 1.5 以降

インストールの手順については、『『Sun Java System Message Queue 4.3 Installation Guide』』の「「AIX Installation」」を参照してください。

AIX プラットフォームでは、メッセージキューのファイルはメッセージキューのホームディレクトリ (IMQ_HOME) 以下にだけインストールされます。IMQ_HOME は、*mqInstallHome/mq* ディレクトリを表します。*mqInstallHome* は、製品のインストール時に指定したインストールのホームディレクトリです (デフォルトでは、*home-directory/MessageQueue*)。

結果として、メッセージキューのディレクトリ構造は Windows プラットフォームの場合と同じになります。『『Sun GlassFish Message Queue 4.4 Administration Guide』』の付録 A 「Distribution-Specific Locations of Message Queue Data」の Windows の節を参照してください。

メッセージキューの AIX プラットフォームのサポートには、メッセージキュー C-API のサポートも含まれます。AIX プラットフォームで C アプリケーションをビルドおよびコンパイルする手順については、XREF を参照してください。

新しい ZIP ベースのインストーラ

メッセージキュー 4.3 では、ネイティブパッケージ配布ではなく、ZIP ベースの配布用に新しいインストーラが導入されました。インストーラは、メッセージキューの新しい ZIP 形式の配布を AIX プラットフォームにインストールするために使用します。

新しいインストーラは、メッセージキューの .ZIP ファイルをユーザーが書き込み可能な (root 特権を必要としない) 任意のディレクトリに展開し、メッセージキューのインストールを Sun Connection に登録できるようにします。

ダウンロードバンドルのサイズを最小化するために、ZIP ベースの配布には Java ランタイムが含まれていません (ほとんどのサイトでは、すでにインストールされているはずです)。したがって、`installer` コマンドでは、`JAVA_HOME` 環境変数を使用するか、次のようにコマンド行に `-j` オプションを使用して、JDK または JRE を指定する必要があります。

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

JDK/JRE-path は、指定する JDK または JRE のパスです。

プラットフォームサポートの拡張

メッセージキューでは、次の更新されたプラットフォームサポートが保証されません。

- Oracle 11g
- Windows Server 2008

その他の拡張機能

メッセージキューには、次の拡張機能も追加されています。

- [18 ページの「Windows プラットフォームでの新しいディレクトリ構造」](#)
- [19 ページの「新しいブローカのプロパティー」](#)
- [19 ページの「JMX 管理 API の機能強化」](#)
- [19 ページの「ワイルドカードサブスクリバの永続サブスクリプションの一覧表示」](#)

Windows プラットフォームでの新しいディレクトリ構造

Windows プラットフォームでは、メッセージキューのインストールディレクトリの構造が、AIX プラットフォームでのディレクトリ構造に一致するように以前のバージョンから変更されました。このディレクトリ構造は、今後 Solaris および Linux プラットフォームでも採用される予定で、単一のコンピュータへの複数のインストールや Sun Connection を利用したメッセージキューの自動更新が容易になります。Sun Connection は、Sun のハードウェアとソフトウェアの追跡、構成、および維持を支援するために Sun が提供するサービスです。21 ページの「インストーラでの Sun Connection 登録のサポート」を参照してください。

新しいブローカのプロパティ

ブローカの設定に、次の新しいプロパティを使用できます。

表 1-6 ブローカのルーティングと配信のプロパティ

プロパティ	タイプ	デフォルト値	説明
<code>imq.transaction.producer.maxNumMsgs</code>	整数	1000	プロデューサが単一のトランザクションで処理できるメッセージの最大数。リソースを使い果たさないように、この値は 5000 未満にすることを推奨します。
<code>imq.transaction.consumer.maxNumMsgs</code>	整数	100	コンシューマが単一のトランザクションで処理できるメッセージの最大数。リソースを使い果たさないように、この値は 1000 未満にすることを推奨します。
<code>imq.persist.jdbc.connection.limit</code>	整数	5	データベースに対して開くことができる接続の最大数。

JMX 管理 API の機能強化

新しい属性と複合データキーが、次のように JMX API に追加されました。

- `NextMessageID` 属性は、送信先監視 MBean に追加され、コンシューマに配信される次のメッセージの JMS メッセージ ID を示します。
- 複合データの `NextMessageID` キーは、コンシューママネージャーの監視 MBean に追加され、コンシューマに配信される次のメッセージの JMS メッセージ ID を示します。
- 複合データの `NumMsgsPending` キーは、コンシューママネージャーの監視 MBean に追加され、コンシューマにディスパッチされているメッセージの数を示します。

詳細は、『[Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients](#)』の第 3 章「[Message Queue MBean Reference](#)」を参照してください。

ワイルドカードサブスクリバの永続サブスクリプションの一覧表示

永続サブスクリプションを一覧表示するコマンドは次のとおりです。

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

このコマンドは、トピック名の指定を省略できるように拡張されています。トピックを指定しない場合、コマンドはすべてのトピック (ワイルドカード命名規則によるトピックを含む) について、すべての永続サブスクリプションを表示します。

メッセージキュー 4.2 の新機能

メッセージキュー 4.2 は、いくつかの新機能、機能拡張、およびバグ修正を実装したマイナーリリースです。ここでは 4.2 リリースの新機能について説明するとともに、詳細な情報の参照先を示します。

- 20 ページの「パブリッシャーまたはサブスクライバの複数の送信先」
- 20 ページの「XML ペイロードメッセージのスキーマ検証」
- 21 ページの「分散トランザクションの C-API サポート」
- 21 ページの「インストーラでの Sun Connection 登録のサポート」
- 22 ページの「MySQL データベースのサポート」
- 22 ページの「その他の拡張機能」

メッセージキュー 4.1 および 4.0 で導入された機能については、それぞれ、23 ページの「メッセージキュー 4.1 の新機能」および 27 ページの「メッセージキュー 4.0 の新機能」を参照してください。

パブリッシャーまたはサブスクライバの複数の送信先

メッセージキュー 4.2 では、パブリッシャーは複数のトピック送信先にメッセージを発行でき、サブスクライバは複数のトピック送信先からメッセージを消費できます。この機能は、トピック送信先の名前にワイルドカード文字を使用して、複数の送信先を表すことにより実現されます。そのような記号名を使用することにより、管理者は、必要な場合に、ワイルドカードのネーミングスキームに整合する追加のトピック送信先を作成できます。送信先が追加されると自動的に、パブリッシャーはその送信先に発行し、サブスクライバはその送信先から消費するようになります。ワイルドカードトピックのサブスクライバの方が、パブリッシャーよりも一般的です。

注-この機能は、キュー送信先には適用されません。

記号によるトピック送信先名の形式と使用例については、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「Supported Topic Destination Names」を参照してください。

XML ペイロードメッセージのスキーマ検証

この機能はメッセージキュー 4.2 で導入され、メッセージがブローカに送信された時点で、テキスト(オブジェクトではない)の XML メッセージの XML スキーマを検証できます。XML スキーマ(XSD)の場所は、Message Queue 送信先のプロパティーとして指定されます。XSD の場所が指定されていない場合は、XML ドキュメント内の DTD 宣言を使用して DTD 検証が実行されます。データ型および値の範囲の検証を含む XSD 検証は、DTD 検証よりも厳格です。

この機能の使用法については、『[Sun GlassFish Message Queue 4.4 Developer's Guide for Java Clients](#)』の「[Schema Validation of XML Payload Messages](#)」を参照してください。

分散トランザクションの C-API サポート

X/Open 分散トランザクションモデルに従って、分散トランザクションのサポートは、1つ以上のリソースマネージャーで実行される操作の追跡と管理を行う分散トランザクションマネージャーに依存します。メッセージキュー 4.2 では、Message Queue C-API で XA インタフェース (分散トランザクションマネージャーと、XA 準拠のリソースマネージャーとしての Message Queue の間のインタフェース) がサポートされます。それにより、BEA Tuxedo などの分散トランザクション処理環境で実行される Message Queue C-API クライアントが、分散トランザクションに参加できます。

この分散トランザクションのサポートは、次に示す、XA インタフェース仕様を実装するための新しい C-API 関数 (および新しいパラメータとエラーコード) から成り立ちます。

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

C クライアントアプリケーションを分散トランザクションのコンテキストで使用する場合は、MQGetXAConnection() を使用して接続を取得し、MQCreateXASession() を使用して、メッセージを生成および消費するためのセッションを作成します。すべての分散トランザクションの開始、コミット、およびロールバックは、分散トランザクションマネージャーが提供する API によって管理されます。

分散トランザクション機能の使用法については、『[Sun GlassFish Message Queue 4.4 Developer's Guide for C Clients](#)』の「[Working With Distributed Transactions](#)」を参照してください。

メッセージキュー 4.2 には、Tuxedo トランザクションマネージャーに基づくプログラミング例が用意されています。これらのサンプルプログラムの使用法については、『[Sun GlassFish Message Queue 4.4 Developer's Guide for C Clients](#)』の「[Distributed Transaction Sample Programs](#)」を参照してください。

注-分散トランザクション機能は、Solaris、Linux、および Windows プラットフォームでサポートされますが、現時点で保証されているのは Solaris プラットフォームのみです。

インストーラでの Sun Connection 登録のサポート

Message Queue インストーラが、メッセージキューを Sun Connection に登録できるように拡張されました。Sun Connection は、Sun のハードウェアとソフトウェアの追跡、構成、および維持を支援するために Sun が提供するサービスです。

メッセージキューのインストール中に、メッセージキューを Sun Connection に登録するかどうかを選択できます。インストールした Message Queue に関する情報、たとえば、リリースバージョン、ホスト名、オペレーティングシステム、インストール日などの基本情報は、Sun Connection のデータベースに安全に転送されます。Sun Connection のインベントリサービスは、Sun のハードウェアとソフトウェアの構成に役立ちます。また、更新サービスでは、最新のセキュリティー修正、推奨される更新、および機能拡張に関する情報を得ることができます。

Sun Connection へのメッセージキューの登録については、『『Sun Java System Message Queue 4.3 Installation Guide』』を参照してください。

MySQL データベースのサポート

メッセージキュー 4.2 では、JDBC ベースのデータストアとして MySQL データベースのサポートが導入されました。スタンドアロンブローカ用の JDBC データベースとして、MySQL Cluster Edition を使用できます。MySQL Cluster Edition は、拡張ブローカクラスタに必要な高可用性共有データストアとして使用することもできます。MySQL を使用するためのメッセージキューの設定については、『Sun GlassFish Message Queue 4.4 Administration Guide』の「Configuring a JDBC-Based Data Store」および『Sun GlassFish Message Queue 4.4 Administration Guide』の「Enhanced Broker Cluster Properties」を参照してください。

その他の拡張機能

先に述べた機能に加え、メッセージキュー 4.2 では次の拡張機能が導入されました。

- リモートで生成されたメッセージのメトリック

メッセージキュー 4.2 には、ブローカクラスタでの送信先の監視に役立つ新しい送信先メトリックが含まれています。ブローカクラスタで、クラスタ内の特定のブローカ上の特定の送信先に格納されるメッセージには、クラスタ内のリモートブローカからその送信先に送信されるメッセージだけでなく、直接その送信先に対して生成されたメッセージもあります。ブローカクラスタでのメッセージのルーティングと配信を分析する際に、送信先のメッセージのうち、ローカルで生成されたメッセージの数と、リモートで生成されたメッセージの数を調べると訳に立つことがあります。

メッセージキュー 4.2 では、2つの新しい物理送信先メトリック量として Num messages remote と Total Message bytes remote が導入されました。新しいメトリック量は、`imqcmd list dst` コマンドと `imqcmd query dst` コマンド (『Sun GlassFish Message Queue 4.4 Administration Guide』の「Viewing Physical Destination Information」を参照)、および JMX 属性 (『Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients』の「Destination Monitor」を参照) を通して使用できます。

- ワイルドカードプロデューサとワイルドカードコンシューマの情報

送信先名でのワイルドカード文字の使用をサポートするための情報 (20 ページの「パブリッシャーまたはサブスクリバの複数の送信先」を参照) が、新しい監視データを通して提供されます。たとえば、送信先に関連付けられたワイルドカードプロデューサまたはワイルドカードコンシューマの数は、`imqcmd query dst` コマンド (『Sun GlassFish Message Queue 4.4 Administration Guide』の「Viewing Physical Destination Information」を参照) と、新しい JMX 属性 (『Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients』の「Destination Monitor」を参照) を通して使用できます。また、ワイルドカード情報は、コンシューママネージャーの監視 MBean およびプロデューサマネージャーの監視 MBean を通しても使用できます。

■ Support for DN Username Format for Client Authentication

メッセージキュー 4.2 では、LDAP ユーザーリポジトリによるクライアント接続認証で DN ユーザー名形式のサポートが導入されました。そのため、次の新しいブローカプロパティ (および値) が追加されています。

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

このプロパティにより、ブローカは、次のプロパティで指定された属性の値を DN ユーザー名形式から抽出することにより、LDAP ユーザーリポジトリのエントリと照合してクライアントユーザーを認証します。

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

ブローカは、アクセス制御操作で、前述の属性の値をユーザー名として使用します。

たとえば、`imq.user_repository.ldap.uidattr=udi` と指定したときに、クライアント認証ユーザー名の形式が `udi=mquser,ou=People,dc=red,dc=sun,dc=com` の場合、アクセス制御の実行時に「mquser」が抽出されます。

■ JAAS Authentication Enhancement

メッセージキュー 4.2 の JAAS 認証では、ユーザー名による認証に加えて、IP アドレスによる認証のサポートも導入されました。

メッセージキュー 4.1 の新機能

メッセージキュー 4.1 は、いくつかの新機能、機能拡張、およびバグ修正を実装したマイナーリリースです。ここでは 4.1 リリースの新機能について説明するとともに、詳細な情報の参照先を示します。

- 24 ページの「高可用性ブローカクラスタ」
- 25 ページの「JAAS サポート」
- 25 ページの「持続データストアの形式の変更」
- 25 ページの「ブローカの環境設定」
- 26 ページの「Java ES Monitoring Framework のサポート」
- 26 ページの「拡張されたトランザクション管理」
- 26 ページの「C クライアント接続の固定ポート」

Message Queue 4.0 で導入された機能については、27 ページの「メッセージキュー 4.0 の新機能」を参照してください。

高可用性ブローカクラスタ

メッセージキュー 4.1 では、新しい拡張ブローカクラスタが導入されました。従来のブローカクラスタでは、ブローカで障害が発生した場合に別のブローカがメッセージングサービスを提供する「メッセージングサービス」の可用性のみが提供されていました。拡張ブローカクラスタでは、ブローカで障害が発生した場合に持続メッセージと状態データを使用して別のブローカがメッセージ配信を継承できる、「データ」の可用性も提供されます。

メッセージキュー 4.1 で導入された高可用性の実装は、JDBC ベースの共有データストアを使用します。つまり、ブローカクラスタの各ブローカに固有の持続データストアを用意する代わりに、クラスタ内のすべてのブローカで同じ JDBC 準拠のデータベースを使用します。特定のブローカで障害が発生すると、そのブローカのメッセージ配信はメッセージクラスタ内の別のブローカによって継承されます。そのときに、継承ブローカは、共有データストアのデータおよび状態情報を使用します。障害が発生したブローカのメッセージングクライアントは継承ブローカに再接続するため、メッセージングサービスは中断されません。

メッセージキュー 4.1 の高可用性実装に使用される JDBC ベースの共有ストアは、それ自体も高可用性ストアでなければなりません。高可用性データベースを持っていない場合、またはメッセージ配信が中断されないことを重要視しない場合は、データの可用性がなくサービスの可用性のみを提供する従来のクラスタを引き続き使用できます。

メッセージキュー 4.1 の拡張ブローカクラスタを設定するには、クラスタ内のブローカごとに、次のブローカプロパティを指定します。

- クラスタメンバーシッププロパティ。ブローカが拡張ブローカクラスタ内にあることと、クラスタの ID およびクラスタ内のブローカの ID を指定します。
- 高可用性データベースプロパティ。持続データモデル (JDBC)、データベースベンダーの名前、およびベンダー固有の設定プロパティを指定します。
- 障害検出および継承プロパティ。ブローカ障害の検出および継承ブローカによる処理の方法を指定します。

拡張ブローカクラスタ実装を使用するには、次の操作を実行します。

1. 高可用性データベースをインストールします。
2. JDBC ドライバの .jar ファイルをインストールします。
3. 高可用性持続データストア用のデータベーススキーマを作成します。
4. クラスタ内のブローカごとに、高可用性プロパティを設定します。
5. クラスタ内の各ブローカを起動します。

拡張ブローカクラスタの概念に関する説明、および従来のクラスタとの比較については、『[Sun GlassFish Message Queue 4.4 Technical Overview](#)』の第4章「[Broker Clusters](#)」を参照してください。拡張ブローカクラスタの手続きおよび参照に関する情報については、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の第10章「[Configuring and Managing Broker Clusters](#)」および『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「[Cluster Configuration Properties](#)」を参照してください。

メッセージキュー 4.0 で高可用性データベースを使用していた場合、拡張ブローカクラスタに切り替えるには、データベースマネージャユーティリティ (`imqdbmgr`) を使用して共有持続データストアに変換します。54 ページの「[ブローカクラスタ](#)」で、既知の問題および制限事項についても参照してください。

JAAS サポート

メッセージキュー 4.1 では、ファイルベースおよびLDAP ベースの組み込み認証機構に加えて、JAAS (Java Authentication and Authorization Service) サポートも導入されています。JAAS を使用すると、外部の認証機構をブローカに接続して Message Queue クライアントを認証できます。

ブローカによって JAAS 準拠の認証サービスで利用可能になる情報についての説明、およびそれらのサービスを使用するようにブローカを設定する方法については、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「[Using JAAS-Based Authentication](#)」を参照してください。

持続データストアの形式の変更

メッセージキュー 4.1 は、JDBC ベースのデータストアが拡張ブローカクラスタをサポートするように変更されました。このため、JDBC ベースのデータストアの形式がバージョン 410 に更新されました。バージョン 350、370、および 400 の形式は、バージョン 410 に自動的に移行されます。

ファイルベースの持続データストアの形式は、何も変更されていないので、Version 370 のままです。

ブローカの環境設定

メッセージキュー 4.1 の環境設定ファイル `imqenv.conf` に、プロパティー `IMQ_DEFAULT_EXT_JARS` が追加されました。このプロパティーを設定して、ブローカが起動するときに `CLASSPATH` に含まれる外部 `.jar` ファイルのパス名を指定できます。このプロパティーを使用して外部 `.jar` ファイルの場所を指定した場合は、これらのファイルを `lib/ext` ディレクトリにコピーする必要はなくなります。外部 `.jar` ファイルは、JDBC ドライバまたは JAAS ログインモジュールを参照できます。次のサンプルプロパティーは、JDBC ドライバの場所を指定します。

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

Java ES Monitoring Framework のサポート

メッセージキュー 4.1 では、Sun Java Enterprise System (Java ES) Monitoring Framework のサポートが導入されました。Monitoring Framework は、一般的なグラフィカルインタフェースを使用して Java ES コンポーネントを監視できるようにします。このインタフェースは、Sun Java System Monitoring Console と呼ばれる Web ベースのコンソールによって実装されます。管理者はこのコンソールを使用して、パフォーマンス統計の表示、自動監視のためのルールの作成、アラームの確認などを行えます。ほかの Java ES コンポーネントと一緒に Message Queue を実行している場合は、1 つのインタフェースを使用してすべてのコンポーネントを管理するほうが便利なことがあります。

Java ES Monitoring Framework による Message Queue の監視については、XREF を参照してください。

拡張されたトランザクション管理

以前は、管理上、PREPARED 状態のトランザクションだけをロールバックすることができました。つまり、分散トランザクションの一部であるセッションが正常に終了しなかった場合、そのトランザクションは、管理者がクリーンアップできない状態のままになりました。メッセージキュー 4.1 では、コマンドユーティリティー (imqcmd) を使用して、STARTED、FAILED、INCOMPLETE、COMPLETE、および PREPARED の状態のトランザクションをクリーンアップ (ロールバック) することができます。

特定のトランザクションをロールバックできるかどうかを判断できるように (特に、PREPARED 以外の状態の場合)、コマンドユーティリティーは imqcmd query txn の出力の一部として追加データを表示します。このデータには、トランザクションを開始した接続の接続 ID と、トランザクションが作成された時刻が示されます。管理者は、この情報を使用して、トランザクションをロールバックする必要があるかどうかを決定できます。一般に、管理者は早計にトランザクションをロールバックすることを避けるとよいでしょう。

C クライアント接続の固定ポート

メッセージキュー 4.1 では、C クライアントは、Java クライアントと同様に、ブローカのポートマッパーサービスで動的に割り当てられるポートではなく、固定ブローカポートに接続できるようになりました。固定ポート接続は、ファイアウォールを経由する場合、または他の何らかの理由でポートマッパーサービスをバイパスする必要がある場合に役立ちます。

固定ポート接続を設定するには、ブローカと C クライアントランタイムの両方 (接続の両端) を設定する必要があります。たとえば、ssljms を介してクライアントをポート 1756 に接続する場合は、次のように操作します。

- クライアント側で、次のプロパティを設定します。

```
MQ_SERVICE_PORT_PROPERTY=1756
```

MQ_CONNECTION_TYPE_PROPERTY=SSL

- ブローカ側で、`imq.serviceName.protocolType.port` プロパティを次のように設定します。

```
imq.ssljms.tls.port=1756
```

注 - MQ_SERVICE_PORT_PROPERTY 接続プロパティは、Message Queue 3.7 Update 2 にバックポートされています。

メッセージキュー 4.0 の新機能

メッセージキュー 4.0 は、Application Server 9 PE のサポートに限定されたマイナーリリースです。このリリースでは、いくつかの新機能、機能拡張、およびバグ修正が実装されています。この節では、このリリースの新機能について説明します。

- 27 ページの「JMX 管理 API のサポート」
- 28 ページの「クライアントランタイムログ」
- 28 ページの「接続イベント通知 API」
- 28 ページの「ブローカ管理の拡張機能」
- 29 ページの「JDBC ベースのデータストアに関する情報の表示」
- 30 ページの「JDBC プロバイダのサポート」
- 30 ページの「持続データストアの形式の変更」
- 30 ページの「追加のメッセージプロパティ」
- 30 ページの「SSL サポート」



注意 - version 4.0 には、軽度とはいえ、状況によって十分な対応が必要になる変更が導入されました。その1つとして、パスワードを指定するコマンド行オプションが使用されなくなったことが挙げられます。今後は、50 ページの「[使用されなくなったパスワードオプション](#)」で説明するように、すべてのパスワードをファイルに保存するか、または要求されたときに入力します。

JMX 管理 API のサポート

Message Queue 4.0 には、Java Management Extensions (JMX) 仕様に準拠して、メッセージキューブローカを設定および監視するための新しい API が追加されました。この API を使用すると、Java アプリケーション内部からプログラムによってブローカ関数を設定および監視することができます。以前のバージョンのメッセージキューでは、これらの関数にはコマンド行管理ユーティリティまたは管理コンソールからしかアクセスできませんでした。

詳細は、『[Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients](#)』を参照してください。

クライアントランタイムログ

メッセージキュー 4.0 では、接続およびセッション関連のイベントに関するクライアントランタイムログのサポートが導入されました。

クライアントランタイムログおよびその設定方法については、『Java Dev Guide』の 137 ページを参照してください。

接続イベント通知 API

メッセージキュー 4.0 にはイベント通知 API が導入され、クライアントランタイムが接続状態の変更をアプリケーションに通知できるようになりました。接続イベント通知を使用すると、Message Queue クライアントは接続のクローズおよび再接続イベントを待機して、通知タイプと接続状態に基づいて適切なアクションを起こすことができます。たとえば、フェイルオーバーが発生してクライアントが別のブローカに再接続された場合、アプリケーションはそのトランザクションの状態をクリーンアップしてから新しいトランザクションに進む必要があるかもしれません。

接続イベント、およびイベントリスナの作成方法については、『Java Dev Guide』の 96 ページを参照してください。

ブローカ管理の拡張機能

メッセージキュー 4.0 では、コマンドユーティリティー (`imqcmd`) に、サブコマンドといくつかのコマンドオプションが追加されました。管理者はこれらを使用して、ブローカを休止したり、指定した間隔の後でブローカをシャットダウンしたり、接続を破棄したり、Java システムプロパティー (たとえば、コネクション関連のプロパティー) を設定したりできます。

- ブローカを休止すると休止状態になり、ブローカをシャットダウンまたは再起動する前に、メッセージを排出してしまふことができます。休止状態にあるブローカに新しく接続が作成されることはありません。ブローカを休止するには、次のようなコマンドを入力します。

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

- 指定した間隔の後でブローカをシャットダウンするには、次のようなコマンドを入力します。(時間間隔は、ブローカをシャットダウンするまでの待機を秒数で指定します。)

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

時間間隔を指定した場合、ブローカはシャットダウンが発生するタイミングを示すメッセージを記録します。次に例を示します。

```
Shutting down the broker in 29 seconds (29996 milliseconds)
```

ブローカがシャットダウンを待っている間、ブローカの動作は次のような影響を受けます。

- 管理 JMS 接続は引き続き受け付けられます。

- 新しい JMS 接続は受け付けられません。
- 既存の JMS 接続は引き続き機能します。
- ブローカが拡張ブローカクラスタ内のほかのブローカを継承することはできません。
- `imqcmd` ユーティリティーはブロックはせず、シャットダウンの要求をブローカに送信してすぐに返します。
- 接続を破棄するには、次のようなコマンドを入力します。

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

コマンド `imqcmd list cxn` または `imqcmd query cxn` を使用してコネクション ID を取得します。

- `imqcmd` を使用してシステムプロパティを設定するには、新しい `-D` オプションを使用します。これは、JMS コネクションファクトリのプロパティ、またはコネクション関連の Java システムのプロパティの設定または上書きに便利です。次に例を示します。

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

`imqcmd` コマンドの構文については、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の第 16 章「[Command Line Reference](#)」を参照してください。

JDBC ベースのデータストアに関する情報の表示

メッセージキュー 4.0 では、データベースマネージャユーティリティー `imqdbmgr` に、新しい `query` サブコマンドが追加されました。このサブコマンドは、JDBC ベースのデータストアに関する情報(データベースバージョン、データベースユーザー、データベーステーブルが作成されたかどうかなど)を表示するために使用します。

次に、このコマンドによって表示される情報の例を示します。

```
imqdbmgr query
```

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

JDBC プロバイダのサポート

メッセージキュー 4.0 では、Apache Derby Version 10.1.1 が、JDBC ベースのデータストアプロバイダとしてサポートされるようになりました。

持続データストアの形式の変更

メッセージキュー 4.0 では、最適化のため、および今後の拡張機能をサポートするために、JDBC ベースのデータストアに変更が加えられました。このため、JDBC ベースのデータストアの形式はバージョン 400 に更新されました。Message Queue 4.0 では、ファイルベースのデータストアのバージョンは、何も変更されていないので 370 のままです。

追加のメッセージプロパティ

メッセージキュー 4.0 では、デッドメッセージキューに配置されたすべてのメッセージで設定される 2 つの新しいプロパティが追加されました。

- `JMS_SUN_DMQ_PRODUCING_BROKER` は、メッセージが生成されたブローカを指定します。
- `JMS_SUN_DMQ_DEAD_BROKER` は、メッセージをデッドとして指定したブローカを指定します。

SSL サポート

メッセージキュー 4.0 から、クライアントコネクションファクトリのプロパティ `imqSSLIsHostTrusted` のデフォルト値が `false` になりました。使用しているアプリケーションが以前のデフォルト値の `true` に依存している場合は、再設定を行い、プロパティを明示的に `true` に設定する必要があります。

ブローカが自己署名付き証明書を使用するように設定されているときは、ホストを信頼することもできます。この場合、接続で SSL ベースの接続サービスを使用するように指定する (`imqConnectionType` プロパティを使用する) ことに加え、`imqSSLIsHostTrusted` プロパティを `true` に設定することをお勧めします。

たとえば、ブローカが自己署名付き証明書を使用するときに安全にクライアントアプリケーションを実行するには、次のようなコマンドを使用します。

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

ブローカが自己署名付き証明書を使用するときにコマンドユーティリティー (`imqcmd`) を安全に使用するには、次のようなコマンドを使用します (コネクタサービスのリストを表示する場合)。

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

今後のリリースで使用されなくなる機能

次の機能は、今後のリリースで使用されなくなります。

- メッセージベースの監視

メッセージベースの監視では、ブローカの設定可能なメトリックスメッセージプロデューサを利用して、メトリックデータを JMS メッセージに書き込みます。この JMS メッセージは、メッセージに含まれているメトリック情報の種類に応じてメトリックトピック送信先に送信されます。このメトリック情報にアクセスするには、適切なメトリックトピック送信先へのサブスクライブ、そのメッセージの消費、および必要に応じたデータ処理を行うクライアントアプリケーションを記述します。

メッセージベースの監視機能の代わりに、MQ 4.0 で導入された JMX 管理 API が使用されるようになっていきます (27 ページの「JMX 管理 API のサポート」を参照)。この JMX API は、より包括的 (トピック送信先に書き込まれるよりも多くのメトリックデータが含まれる) であり、JMX 業界標準に基づいています。

メッセージキューで JMX API がサポートされている現在、あえてメッセージベースの監視を使用する理由はありません。メッセージベースの監視機能に関する情報は、この機能が正式に使用されなくなるまでメッセージキュードキュメントに残されます。

- テキストベースのインストーラ

メッセージキュー インストーラのテキストモード (`installer -t`) は、すべてのオペレーティングシステムプラットフォームで使用できなくなります。テキストモードでは、端末ウィンドウにグラフィカルユーザーインターフェース (GUI) モードの表示をシミュレートするプレーンテキストが表示されます。GUI モードおよびサイレントモードのサポートは継続されます。

- プラットフォームサポート

今後のリリースでは、Windows 2000 と Red Hat Linux 3 はサポートされなくなります。

- JMSRA リソースアダプタ

メッセージキューのリソースアダプタ `imqjmsra.rar` (JMSRA) は、今後のメッセージキューのリリースで新しいリソースアダプタに置き換えられます。JMSRA は、メッセージキューと Sun Java System Application Server を統合するために使用されます。

新しいリソースアダプタは、JMSRA の既存の機能とほかの Sun JMS リソースアダプタの機能を組み合わせて、Java EE 5 Application Server 環境でメッセージキューやほかのプロバイダに特別なサポートを提供します。したがって、メッセージキューを Sun GlassFish Enterprise Server および Sun Java Composite Application Platform Suite (Java CAPS) に統合するために使用される予定です。

メッセージキュー 4.4 Update 1 で修正されたバグおよび最近のリリース

メッセージキュー 4.4 Update 1 には、新しいバグの修正と、メッセージキュー 4.x ファミリのこれまでのリリースで解決されたバグの修正が含まれます。

次の各節では、それぞれのリリースで修正されたバグの一覧を示します。

- 32 ページの「メッセージキュー 4.4 Update 1 で修正されたバグ」
- 35 ページの「メッセージキュー 4.4 で修正されたバグ」
- 37 ページの「メッセージキュー 4.3 で修正されたバグ」
- 38 ページの「メッセージキュー 4.2 で修正されたバグ」
- 39 ページの「メッセージキュー 4.1 で修正されたバグ」
- 40 ページの「メッセージキュー 4.0 で修正されたバグ」

メッセージキュー 4.4 Update 1 で修正されたバグ

次の表に、メッセージキュー 4.4 Update 1 で修正されたバグを示します。これらの問題の一部には、「(OpenMQ)」のマークが付いています。これは、Sun GlassFish メッセージキューの基になっているオープンソースプロジェクト [Open Message Queue](#) で問題が解決されたことを示しています。

表 1-7 メッセージキュー 4.4 Update 1 で修正されたバグ

バグ	説明
6590909	アドレスリストが上書きされる時に、DIRECT モードの MDB がリモートブローカーに接続しません。
6616704	1つのセッション内に多数のコンシューマが作成されたときに、ブローカーのメモリーが増大します。
6745761	2つの接続が同じ XA TX で使用されている場合 (JMSJCA を使用)、XAResource.isSameRM() は true を返すべきです。
6745763	2つの接続が同じ XA TX で使用されている場合 (JMSRA DIRECT モード)、XAResource.isSameRM() は true を返すべきです。
6745768	2つの接続が同じ XA TX で使用されている場合 (JMSRA LOCAL/REMOTE)、XAResource.isSameRM() は true を返すべきです。
6760450	MQ (GF) インスタンスを停止せずにマシンを再起動すると、メッセージストアが破損します。
6766241	UMS: SendMsg.html の AJAX の例が、デフォルトのコンテキストルートとして /ums を使用します。/imqums を使用するべきです。

表 1-7 メッセージキュー 4.4 Update 1 で修正されたバグ (続き)

バグ	説明
6766852	DirectXAResource が、ブローカの CONFLICT 状態を「TxID はすでに使用されています」に変換します。
6799428	DMQ の非持続メッセージまたは送信された非永続メッセージは、消費できませんが参照可能です。
6809353	postgresql (8.1) では openmq 4.3 HA が動作しません (imqbrokerd を起動できません)。
6809750	クライアント ID 接続の (JMSRA からの) 接続プールが動作しません。
6812198	MQ トピックのメトリックを使用して監視しているときに、Classcast 例外がスローされます。
6832000	MQ reapExcessConnection JDBC 接続で、CPU の負荷が高くなります。
6833109	MQClusterMonitor の JMX サンプルアプリケーションが、JDK 6 を使用する AIX で例外をスローします。
6835420	NoGCDefault のデフォルト値の計算が間違っています。メモリーが少ないときに GC が大きくなりすぎます。
6852018	エラーメッセージの「永続コンシューマ {0} を追加できません。接続に対するクライアント ID が設定されていません。」は誤解を招きます
6856991	ブローカを再起動して、永続コンシューマの PREPARED トランザクションをロールバックしたあとに、NullPointerException が発生します。
6874125	警告: MQJMSRA_DC2001: connectionId=555670328604044289: _destroy(): ... 接続で呼び出されました。
6878945	RFE: JMSBridge: ユーザー名とパスワードを指定して接続ファクトリから接続を作成することの許可。
6881493	管理一時送信先は、HA ブローカに保存されるべきではありません。
6881753	RFE JMSBridge: ターゲットに転送する前に各メッセージに jmsbridge 名のタグを付けることの許可。
6884673	MQ 4.4 Broker が MQ 3.7 または 3.6 Broker とクラスタ接続を確立できませんでした。
6886390	発行された持続またはトランザクションのメッセージが DMQ に送られると、これらを DMQ から消費するときに、mq.sys.dmqs が見つからないというエラーが発生する場合があります。
6886515	JMX を使用して組み込みブローカ内の送信先を削除するときに、AccessControlException が発生します。
6890628	ブローカのプロパティ imq.autocreate.destination.isLocalOnly=true を設定しても効果がありません。
6891615	ブローカ 4.3 を GlassFish で実行する場合、常にセレクトが動作するわけではありません。

表 1-7 メッセージキュー 4.4 Update 1 で修正されたバグ (続き)

バグ	説明
6891624	「imqcmd list dst」を実行したときに、メッセージの「Remote」が「Count」よりも大きくなる場合があります。
6891629	セレクトで算術例外が発生した場合は、ユーザーにわかりやすいメッセージが必要です。
6891717	ifmq.transaction.autorollback=true、autorollbackのPREPAREDの確認がクリアされていないと、TransactionAckExistExが発生します。
6891802	リモートトランザクションの確認をテイクオーバーしたあと、ブローカを再起動すると、「[B4061]: トランザクション ID を使用できません。現在使用されています」と表示されます。
6892512	メモリーリーク: tempDest.delete() を呼び出したときに、一時送信先が接続から削除されません。
6895040	マスターブローカに一時送信先がある場合、uidprefix ロックのタイムアウトの経過後、スレーブブローカは起動時に uidprefix を取得できません。
6896230	マスターブローカをスレーブと同期したあと、マスターブローカの再起動中にマスターブローカで作成された新しいコンシューマが伝達されない場合があります。
6896764	TransactionAcknowledgement の equals メソッドが正しくありません。
6898355	ブローカ再起動時のクラスタマネージャーの初期化中に、テイクオーバーの完了を待たずに、テイクオーバーのロックがリセットされます。
6901405	RFE: 指定した場合に、JDBC ベンダー情報とベンダーのプロパティをログに記録する。
16 (OpenMQ)	ブローカ 4.3 を GlassFish で実行する場合、常にセレクトが動作するわけではありません。
17 (OpenMQ)	postgresql (8.1) では openmq 4.3 HA が動作しません (imqbrokerd を起動できません)。
22 (OpenMQ)	存在していないバイナリをインストーラが参照して失敗します。
25 (OpenMQ)	TemporaryTopic の作成時にメモリーがリークします。
29 (OpenMQ)	ブローカの分離。
30 (OpenMQ)	「imqcmd list dst」を実行したときに、メッセージの「Remote」が「Count」よりも大きくなる場合があります。
31 (OpenMQ)	セレクトで算術例外が発生したときに、ユーザーにわかりやすいメッセージが必要です。
32 (OpenMQ)	int->long のオーバーフローの修正。
33 (OpenMQ)	OpenMQ インストーラ: 日本語のロケールで実行したときに、「無効な SwiXML 記述子」のエラーが発生します。

メッセージキュー 4.4 で修正されたバグ

次の表に、メッセージキュー 4.4 で修正されたバグを示します。

表 1-8 メッセージキュー 4.4 で修正されたバグ

バグ	説明
6242247	2つのブローカが同じマシン上にあり、同じ名前である場合、マスターブローカを持つ MQ クラスタが起動後にハングアップします。
6760937	再起動されたブローカが DB に再接続しません。
6763252	期限切れまたは削除されたメッセージを確認するときに、ブローカは NPE よりも意味のあるメッセージをログに記録するべきです。
6765410	マスターブローカがローカルの配信対象に 2 回送信を行うと、スレーブで「永続サブスクリプションがすでにアクティブである」という例外が発生します。
6796506	リモートの PREPARE 応答の受信中にタイムアウトが発生した場合、リモートの PREPARED メッセージがロールバック後に再配信されません。
6807708	マスターブローカが動作していない場合、TemporaryDestination.delete が失敗します。
6812037	RFE: MQMessageListenerFunc がエラーを返す場合は、MQ_CALLBACK_RUNTIME_ERROR を afterMessageDelivery に渡す。
6812755	before/afterMessageDelivery のコールバックがエラーを返す場合、FINE レベルのログメッセージは WARNING レベルであるべきです。
6816023	Message.setStringProperty() の例外が、無効な文字の例外でプロパティ名を表示しません。
6819095	RFE: クラスタは、入力と出力ストリームのバッファサイズの設定と TcpNoDelay をサポートすべきです。
6820585	「imqcmd list txn」が、リモートブローカの完了を待機している COMMITTED クラスタランザクションを表示しません。
6820588	ローカルとリモートの両方のメッセージを消費するクラスタランザクションが、待機中の状態で COMMITTED のままになります。
6821639	MQRA-DIRECT モードで AS 回復中に、rollback/commitTransaction で NPE が発生します。
6823364	RFE: C-API コンパイラの Solaris 上の Sun Studio 12 へのアップグレード。
6829113	Tuxedo TM が負荷の高い状況でタイムアウトしたトランザクションをロールバックすると、ConcurrentModificationException が発生します。
6832197	クライアントが確認の確認を要求しなかった場合に、非トランザクションのリモート確認はリモートの応答を待つべきではありません。

表 1-8 メッセージキュー 4.4 で修正されたバグ (続き)

バグ	説明
6834735	Tuxedo TM が START 状態のトランザクションをタイムアウトするときに、混乱を招く「予期しないブローカ内部エラー」のログメッセージが記録されます。
6836364	トピックがサブスクライバより前に作成されている場合、ワイルドカードサブスクライバがリモートメッセージを受信しません。
6836691	HA(JCAPS): XA の受信側がメッセージをロールバックしてコミットしたあと、受信時に「メッセージがすでに削除されている」という例外が発生します。
6836749	HA(JCAPS): 永続サブスクリプションの 1 つでメッセージをロールバックしてコミットしたあと、受信時に「ストアに確認応答が存在する」という例外が発生します。
6837671	HA(JCAPS): コミットが成功したあと、XAResourceImpl.rollback の実行時にコミットしたメッセージが繰り返し配信されます。
6839193	RFE: C++ コンパイラの Visual Studio 2008 SP1 へのアップグレード。
6845625	リモートコンシューマの作成と終了を繰り返すと、ブローカが低メモリー状態になります。
6852207	リモートブローカにメッセージを送信したときの NPE により、メッセージパケットの読み取り時にリモートブローカがメッセージを処理できない状態になります。
6853822	FAILED トランザクションを終了するときの、「操作 END_TRANSACTION を実行できません」という例外のメッセージは混乱を招きます。
6854142	「クラスタ接続を待っています」、「クラスタ接続を終了しました」のメッセージが 3 分ごとにリモートブローカに送られます。
6858121	リモートトランザクションが存在する場合に、「imqcmd list txn」でブローカのログに記録される「不明なトランザクション」の警告は混乱を招きます。
6858488	リモート参加のブローカが COMMITTED トランザクションを削除した場合、その COMMITTED トランザクションはトランザクションのホームブローカから削除されません。
6858905	Consumer.destroyConsumer で ConcurrentModificationException が発生します。
6861362	RFE: JMSBridge: ターゲット送信先とソース Message.getJMSDestination の自動マップのサポート。
6861528	RFE: JMSBridge: MessageTransformer.transform() で、ターゲットにある別の送信先へのメッセージの分岐を許可する。
6861653	トランザクションの負荷が高い状態で、過度のクラスタトランザクション情報が COMMIT が完了していない停止中のリモートブローカに送信されます。
6862413	「mq://xxx.xxx.xx.xx:pppp/ ... に 60 秒以内に到達できます」のメッセージは混乱を招きます。

表 1-8 メッセージキュー 4.4 で修正されたバグ (続き)

バグ	説明
6863867	停止したリモートブローカからの保留中の COMMITTED がある場合、HA ブローカの再起動時に MissingResourceException が発生します。
6867596	ブローカの再起動後に回復した PREPARED トランザクションが、ブローカをもう一度再起動すると、再び PREPARED 状態に戻ります。
6868525	リンク確立時に一時送信先をリモートブローカに転送すると、NullPointerException が発生します。
6868578	ブロードキャストまたはユニキャストの一部は、リンクが確立されたことを確認しないため、リンクのハンドシェイクに干渉してリンクが停止します。
6871612	HA:保留中のブローカが停止している場合、リモートメッセージを消費するときに、「トランザクションの完了を通知できません..」のメッセージが記録されません。
6886391	メッセージがすでに削除されている場合、メッセージの確認時に NullPointerException が発生します。

メッセージキュー 4.3 で修正されたバグ

次の表に、メッセージキュー 4.3 で修正されたバグを示します。

表 1-9 Message Queue 4 で修正されたバグ3

バグ	説明
6634033	クライアントが作成されたときに、クラスタプロトコルが <code>imqConsumerFlowLimit</code> の値をリモートブローカに伝達しません。
6713012	リモートブローカの再起動と同時に、クラスタ内のブローカでコンシューマを破棄すると、一部のメッセージが配信されない場合があります。
6727555	ブローカのログメッセージ「Max bytes per msg exceeded」が実際のメッセージサイズを表し、 <code>max bytes per message</code> の値と入れ替わっています。
6737404	JMX メトリックは、送信先(トピックおよびキュー)からディスパッチされているが、まだコンシューマに配信されていないメッセージの数を示す必要があります。
6740568	単一のトランザクションで消費されたメッセージが多すぎると、ブローカが例外をスローします。
6758524	永続サブスクリプションを一覧表示するコマンド (<code>imqcmd list dur -d "foo.*"</code>) で、送信先名にワイルドカード文字を使用できません。
6758952	<code>imq.portmapper.hostname=localhost</code> を設定すると、ブローカがクラスタ内に接続できなくなります。

表 1-9 Message Queue 4 で修正されたバグ3 (続き)

バグ	説明
6758817	imq.cluster.hostname=localhost (非推奨) を設定すると、別のマシンのブローカがクラスタ内に接続できなくなります。

メッセージキュー 4.2 で修正されたバグ

次の表に、メッセージキュー 4.2 で修正されたバグを示します。

表 1-10 Message Queue 4.2 で修正されたバグ

バグ	説明
6581592	インストーラまたはアンインストーラがテキストモード (<code>installer -t</code>) で実行されているときは、「概要」画面にログファイルまたは概要ファイルを含むディレクトリが表示されますが、これらのファイルの名前のリストは表示されません。
6585911	インストーラにバンドルされている、インストーラの実行に使用する JRE が、インストーラの「JDK の選択」画面に正しく表示されません。
6587112	複数バイトロケールで、インストーラの「概要」画面に無意味な文字が表示されます。
6587127	回答ファイルを参照して (<code>installer -a filename -s</code>) インストーラを実行するときに、回答ファイルが存在しない場合、矛盾した不明瞭なエラーメッセージが表示されます。
6590969	クライアント接続認証で、DN ユーザー名形式が許可されます。
6594381	Message Queue 4.1 ローカリゼーション RPM のインストール(「多言語パッケージ」画面で「Message Queue 多言語パッケージのインストール」チェックボックスにチェックマークを付けた場合に実行される)は、古いバージョンの Message Queue ローカリゼーション RPM がシステムに存在すると、異常終了します。
6599144	Message Queue 4.2 のアンインストール時に、Java SE 6 ではスプラッシュ画面とアンインストーラがハングアップして空のグレー画面になりますが、Java SE 5 では正しく動作します。
6615741	ロールバックの前に元のコンシューマが閉じられると、ロールバックされたトランザクションコンシューマセッションで配信されたメッセージは再配信されません。
6629922	分散トランザクションハンドラは、非アクティブなコンシューマに正しい順序でメッセージを再配信しません。
6635130	送信先のメモリー制限またはメッセージ制限に達したために一時停止された後、ブローカは、非持続メッセージのプロデューサに生成を再開するように通知できません。

表 1-10 Message Queue 4.2 で修正されたバグ (続き)

バグ	説明
6641117	ロールバックの後に元のコンシューマが閉じられると、ロールバックされたトランザクションコンシューマセッションで配信されたメッセージは再配信されません。
6683897	設定が正常に完了したように見える場合でも、Message Queue インストーラの概要画面に設定エラーが報告されます。インストーラは、一部のコンピュータの /dev/sterr に書き込むことができません。
6684069	コンシューマトランザクションで多数のメッセージがリモートクライアントに配信されるブローカクラスターで、トランザクションのコミットに障害が発生します。
6688935	Portmapper の読み取りのタイムアウトが短すぎます。
6695238	C クライアントアプリケーションは、パスにスペースが含まれている場所にインストールされたブローカに接続できません。
6710168	送信先が再開されないまま 2 回続けて一時停止されると、コンシューマはメッセージを消費しなくなります。
6710169	JMX 操作 ConsumerManagerMonitor.getConsumerInfo は、通知モードでは常に SESSION_TRANSACTED を返します。

メッセージキュー 4.1 で修正されたバグ

次の表に、メッセージキュー 4.1 で修正されたバグを示します。

表 1-11 Message Queue 4.1 で修正されたバグ

バグ	説明
6381703	処理されたりリモートメッセージが、そのメッセージの発信元のブローカが再起動された場合に、2 回コミットされることがあります。
6388049	未完了の分散トランザクションをクリーンアップできません。
6401169	imqcmd のコミットおよびロールバックオプションで、確認のプロンプトが表示されません。
6473052	自動作成されたキューのデフォルトはラウンドロビンであるべきです。(MaxNumberConsumers = -1)。
6474990	ブローカのログに、imqcmd list dst コマンドに対して ConcurrentModificationException が示されます。
6487413	LimitBehavior が REMOVE_OLDEST または REMOVE_LOWER_PRIORITY のときに、メモリーがリークします。
6488340	ブローカがスピンし、クライアントは確認に対する応答を待機します。

表 1-11 Message Queue 4.1 で修正されたバグ (続き)

バグ	説明
6502744	ブローカが、デッドメッセージキューのデフォルト制限(1000 のメッセージ)を尊重しません。
6517341	クライアントが拡張ブローカクラスタへ接続している際の、再接続ロジックを、 <i>imqReconnectEnabled</i> プロパティの値に関係なく再接続を可能にするように改善すべきです。
6528736	起動時に Windows 自動起動サービス (<i>imqbrokersvc</i>) に障害が発生します。
6561494	両者が1つのセッションを共有していると、メッセージが間違ったコンシューマに配信されます。
6567439	PREPARED トランザクションの生成メッセージがブローカの再起動後にコミットされた場合、それらのメッセージは順序どおりに配信されません。

メッセージキュー 4.0 で修正されたバグ

次の表に、メッセージキュー 4.0 で修正されたバグを示します。

表 1-12 メッセージキュー 4.0 で修正されたバグ

バグ番号	説明
4986481	メッセージキュー 3.5 では、自動再接続モードでの <i>Session.recover</i> の呼び出しがハングすることがありました。
4987325	<i>Session.recover</i> の呼び出し後に、再配信されたフラグは再配信されたメッセージに対して <i>false</i> に設定されました。
6157073	新しく接続メッセージが変更され、接続の合計数のほかにサービス上の接続数も表示されるようになりました。
6193884	メッセージキュー は、メッセージに非アスキー文字を使用するロケールで文字化けメッセージを <i>syslog</i> に出力します。
6196233	<i>JMSMessageID</i> を使用したメッセージ選択が機能しません。
6251450	クラスタのシャットダウン時、 <i>connectList</i> に <i>ConcurrentModificationException</i> が発生します。
6252763	<i>java.nio.HeapByteBuffer.putLong/Int</i> に <i>java.nio.BufferOverflowException</i> が発生します。
6260076	Oracle ストレージを使用すると、起動後に発行される最初のメッセージが遅くなります。
6260814	<i>JMSXUserID</i> 上のセレクト処理が、常に <i>false</i> と評価されてしまいます。
6264003	キューブラウザにコミットされていないメッセージが表示されます。

表 1-12 メッセージキュー 4.0 で修正されたバグ (続き)

バグ番号	説明
6271876	消費されていないメッセージを含むコンシューマを閉じようとする、接続フロー制御が正しく動作しなくなります。
6279833	メッセージキューでは、2つのブローカが同じ JDBC テーブルを使用することはできません。
6293053	システム IP アドレスが変更された場合、 <code>-reset store</code> を使用してストアをクリアしていないと、マスターブローカが正しく起動しません。
6294767	メッセージキューブローカがネットワークソケットを開く場合、そのネットワークソケット上に <code>SO_REUSEADDR</code> を設定する必要があります。
6304949	<code>TopicConnectionFactory</code> に <code>ClientID</code>
6307056	<code>txn</code> ログがパフォーマンス上のボトルネックになっています。
6320138	メッセージキュー C API では <code>reply-to</code> ヘッダーからキューの名前を決定することができません。
6320325	Solaris 上に JDK 1.4 と JDK 1.5 の両方がインストールされている場合、ブローカが JDK 1.5 より前に JDK 1.4 を検出してしまうことがあります。
6321117	マルチブローカクラスタの初期化で <code>java.lang.NullPointerException</code> が発生します。
6330053	サブスクリバからトランザクションをコミットしている場合、JMS クライアントで <code>java.lang.NoClassDefFoundError</code> が発生します。
6340250	C-API で MESSAGE タイプをサポート。
6351293	Apache Derby データベースへのサポートを追加。

メッセージキュー 4.4 Update 1 のドキュメントの更新

この節では、メッセージキュー 4.4 Update 1 のドキュメントの更新について説明します。

- [41 ページの「互換性の問題」](#)
- [42 ページの「メッセージキュー 4.4 Update 1 ドキュメントセットの変更点」](#)

互換性の問題

この節では、メッセージキュー 4.4 Update 1 に関する互換性の問題を説明していません。

インタフェースの安定性

Sun GlassFish メッセージキュー で使用される多くのインタフェースは、時間の経過につれて変更される可能性があります。『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の付録 B 「[Stability of Message Queue Interfaces](#)」では、インタフェースを安定性に従って分類しています。安定性に優れるインタフェースほど、製品の後継バージョンで変更される可能性は低くなります。

メッセージキューの次のメジャーリリースに関する問題

メッセージキューの次のメジャーリリースでは、現在のメッセージキュークライアントアプリケーションがそのリリースと互換性がなくなるような変更が導入される可能性があります。この情報は、完全な情報開示の目的で提供しています。

- Sun GlassFish メッセージキューの一部としてインストールされる各ファイルの場所が変更される可能性があります。これによって、特定のメッセージキューファイルの現在の場所に依存する既存のアプリケーションの動作が中断する可能性があります。
- メッセージキュー 3.5 以前のブローカは、これより新しいブローカのクラスタ内では動作できなくなる可能性があります。
- 今後のリリースでは、メッセージキュークライアントは 1.5 より前のバージョンの JDK を使用できなくなる可能性があります。
- 今後のリリースでは、メッセージキュークライアントは 1.6 より前のバージョンの JDK を使用できなくなる可能性があります。

メッセージキュー 4.4 Update 1 ドキュメントセットの変更点

メッセージキュー 4.4 Update 1 ドキュメントセットは、次に説明するように、Message Queue 4.4 ドキュメントセットから更新されています。

『技術の概要』

『[Sun GlassFish Message Queue 4.4 Technical Overview](#)』は、メッセージキュー 4.4 の新しい機能を反映するように更新されました。

管理ガイド

『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』は、小規模なバグ修正を行い、メッセージキュー 4.4 の新しい機能を反映するように更新されました。

『Developer's Guide for Java Clients』

『[Sun GlassFish Message Queue 4.4 Developer's Guide for Java Clients](#)』は、小規模なバグ修正を行いました。

『Developer's Guide for C Clients』

『Sun GlassFish Message Queue 4.4 Developer's Guide for C Clients』は、小規模なバグ修正を行いました。

『Developer's Guide for JMX Clients』

『Sun GlassFish Message Queue 4.4 Developer's Guide for JMX Clients』は、メッセージキュー製品の名前とバージョン番号を反映するように更新されました。

既知の問題点と制限事項

この節には、メッセージキュー 4.4 Update 1 の既知の問題についてのリストが含まれています。次の内容について説明します。

- 43 ページの「インストールに関する情報」
- 50 ページの「使用されなくなったパスワードオプション」
- 51 ページの「管理および設定上の問題」
- 52 ページの「ブローカの問題」
- 54 ページの「ブローカクラスタ」
- 57 ページの「JMX の問題」
- 57 ページの「SOAP サポート」

現時点のバグ、その状態、および回避策の一覧については、Java Developer Connection™ メンバーは、Java Developer Connection Web サイトの「Bug Parade」ページを参照してください。新しいバグを報告する前に、このページをチェックしてください。すべてのメッセージキューバグがリストされているわけではありませんが、このページはある問題が報告済みかどうかを知りたい場合に活用できます。

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

注 - Java Developer Connection のメンバーになるのは無料ですが、登録が必要です。Java Developer Connection のメンバーになる方法についての詳細は、Sun の「For Developers」 Web ページを参照してください。

新しいバグの報告や機能に関する要求を行うには、imq-feedback@sun.com 宛てにメールを送信してください。

インストールに関する情報

この節では、Message Queue version 4.4 Update 1 のインストールに関連した問題について説明します。

製品レジストリと Java ES

メッセージキュー 4.4 は、メッセージキュー 4.2 および 4.1 と同様に、新しいインストーラでインストールされます。このインストーラでは、JDK、NSS、JavaHelp など、Message Queue に必要な Java Enterprise System (Java ES) 共有コンポーネントもインストールされます。

新しい Message Queue インストーラと、以前のバージョンの Message Queue で使用されていた古い Java ES インストーラは、同じ製品レジストリを共有しません。Java ES でインストールされた Message Queue を削除したあと、Message Queue インストーラでメッセージキュー 4.4 をインストールすると、Java ES 製品レジストリが矛盾した状態になる場合があります。その結果、Java ES アンインストーラを実行すると、Java ES でインストールしなかったとしても、メッセージキュー 4.4 とそれが依存する共有コンポーネントが意図せずに削除されることがあります。

Java ES インストーラでインストールした Message Queue ソフトウェアをアップグレードする最善の方法は、次のとおりです。

1. Java ES アンインストーラを使用して、メッセージキューとその共有コンポーネントを削除します。
2. Message Queue インストーラを使用して、メッセージキュー 4.4 をインストールします。

すべてのプラットフォームでのインストール

次に示す問題は、すべてのプラットフォームでのインストールに影響します。

- 「インストールの準備完了」画面で、製品名が「Sun Java System Message Queue 4.3」ではなく「mq」と表示されます。(バグ 6650841)
- インストーラが Message Queue 4.3 のインストールを処理中で、「進行状況」画面が表示されているときは、「取消し」ボタンがアクティブになります。ここで「取消し」ボタンを選択すると、インストールが完了しないか、破棄されず。(バグ 6595578)
- インストーラの「概要」画面には、クリックするとログまたは概要ページビューが開くリンクがいくつか含まれています。「閉じる」というラベルの付いたボタンの代わりに、ウィンドウの閉じるボタン「X」を使用してこのビューアウィンドウを閉じた場合、このビューアウィンドウを再度表示することができません。(バグ 6587138)

回避方法: 「閉じる」というラベルの付いたボタンを使用してウィンドウを閉じます。

- 登録が省略されるサイレントインストールを実行したあとに、インストーラを登録のみのモード (`installer -r`) で実行すると、「ファイルの途中終了」エラーにより登録が失敗します。(バグ 6767988)

- JDKがインストールされていないコンピュータで Message Queue インストーラを実行すると、「Invalid root in registry key HKLM\SOFTWARE\JavaSoft\Java Runtime Environment\CurrentVersion」というエラーメッセージが表示されます。(バグ 6764358)
回避方法: インストーラを実行する前に、JDK をインストールします。
- 「インストールの準備完了」画面の「インストール」ボタンをクリックする前に、Message Queue インストーラによって mqInstallHome ディレクトリが作成されます。(Bug 6595590)

Windows でのインストール

Message Queue を Windows にインストールするときは、次の制限事項に注意してください。

- Windows プラットフォームにインストールされた Message Queue 4.3 のディレクトリ構造は、これまでのリリースのディレクトリ構造と異なります。『『Sun Java System Message Queue 4.3 Installation Guide』』の「「Installed Directory Structure」」を参照してください。
- インストーラは、Message Queue 用のエントリを「スタート」>「すべてのプログラム」メニューに追加しません。(バグ 6567258)
回避方法: 管理コンソールを起動するには、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「Starting the Administration Console」で説明されているようにコマンド行を使用します。
- インストーラは、IMQ_HOME\mq\bin ディレクトリを PATH 環境変数に追加しません。(バグ 6567197)
回避方法: ユーザーは Message Queue ユーティリティ (IMQ_HOME\mq\bin\command) を起動するときに、このエントリを PATH 環境変数に追加するか、フルパス名を指定する必要があります。
- インストーラは、Message Queue がインストールされたことを示すために Windows レジストリにエントリを追加しません。(バグ 6586389)
- インストーラは、メッセージキューブローカを Windows サービスとして追加しません。
回避方法: imqsvcadm コマンドを使用して、メッセージキューブローカを Windows サービスとして手動で追加します。
- JDK がインストールされていない場合、インストーラが「Invalid root in registry key HKLM\SOFTWARE\JavaSoft\Java Runtime Environment\CurrentVersion」というエラーをスローします。(バグ 6764358)
回避方法: このエラーが表示された場合は、JDK をインストールしてから操作を続行します。

- 回答ファイルを使用してサイレントモードで実行すると、インストーラはすぐに返されます。インストールは行われますが、ユーザーにはサイレントインストールがいつ実際に実行されたかを知る方法がありません。(バグ 6586560)
- オペレーティングシステムが別のドライブにインストールされている場合でも、インストーラは Message Queue を C:\ にインストールします。(バグ 6673511)
- Windows でのインストールおよびアンインストールでは、ユーザーが実行できる bat ファイルはありません。また、ユーザーが Windows コントロールパネルの「プログラムの追加と削除」を使用してアンインストールすることもできません。(バグ 6673417)
- Windows Vista では、コマンドプロンプトから管理者としてインストールを実行しなければ、Message Queue を C:\Program Files にインストールできません。(バグ 6701661)

回避方法: 管理者としてコマンドプロンプトからインストールするには、次の手順に従います。

1. 「スタート」 → 「すべてのプログラム」 → 「アクセサリ」の順にクリックし、「コマンドプロンプト」をポイントします。
2. 「コマンドプロンプト」を右クリックします。
3. 「管理者として実行」をクリックします。
4. ディレクトリを Message Queue 4.2 のインストールイメージに変更します。
5. `installer.vbs` を実行します。

- アンインストーラを予行モード (`uninstaller -n`) で実行すると、アンインストールが誤って実行されます。(バグ 6719051)

回避方法: 次のコマンドを使用して、サイレントインストールを実行します。

```
uninstaller -s
```

- インストーラのホームページで「Install Home」の文字列がローカライズされません。(バグ 6592491)
- Message Queue の ZIP ベースのアンインストーラが、Windows 2003 でハングアップします。(バグ 6764370)

回避方法: `mqInstallHome` ディレクトリを手動で削除します。

Solaris でのインストール

- インストーラを予行モード (`installer -n`) で実行すると、「概要」画面にいくつかのエラーメッセージが表示され、「未完了」のインストール状態も表示されません。これは誤りで、誤解を招きます。予行モードでは、何もシステムにインストールされません。後でサイレントインストールの実行に使用できる回答ファイルが作成されるだけです。(バグ 6594351)
- 回答ファイルを使用してサイレントモード (`installer -a filename -s`) でインストーラを実行すると、Sun Connection への登録が実行されません。(バグ 6710268)

Linuxでのインストール

次に示す問題は、Linux プラットフォームでのインストールに影響します。

- Red Hat Linux 5 では、C クライアントアプリケーションの実行に必要な `compat-libstdc++` ライブラリが、メッセージキューの配布に含まれていません。したがって、メッセージキューインストーラではこのライブラリをインストールできません。C クライアントを開発および実行している場合は、このライブラリを手動でインストールする必要があります。

通常、`compat-libstdc++ RPM` は使用している Linux のバージョンのインストールメディアにあります。次のコマンドを使用してインストールできます。

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

x はバージョン番号を表します。

ライブラリが正常にインストールされたかどうかを確認するには、次のコマンドを使用します。

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

- Red Hat Linux 5 では、C クライアントが `PR_LOAD_LIBRARY_ERROR` エラーで失敗する場合があります。(バグ 6885978)

Red Hat Linux 5 では、C クライアントが失敗して、次のようなメッセージが表示される場合があります。

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

producer(): Error: PR_LOAD_LIBRARY_ERROR

このエラーは、NSS または NSPR ライブラリにアクセスできないために発生します。

この問題を解決するには、`LD_LIBRARY_PATH` 環境変数を、NSS または NSPR ライブラリのパス (`IMQ_HOME/nss/lib`) を含むように設定します。

- 「JDK 選択」パネルで、スクロールリストに項目が1つしか表示されません。このため、リスト内のほかのJDKを選択することが難しくなります。(バグ 6584735)
- JDK が現在のもので、ユーザーが「JDK 選択」画面で「Java(TM) SDK のデフォルトバージョンをインストールして使用します。」を選択した場合、インストーラはそのJDKをインストールしようとし、パッケージをインストールできないというレポートを返します。この問題があっても、インストールは正常に完了します。(バグ 6581310)
- 現在インストールされているJDKがJDK 1.5.0_15(メッセージキューインストーラによって標準でインストールされるバージョン)より新しいバージョンの場合、メッセージキューインストーラはデフォルトの `IMQ_JAVAHOME` ディレクトリを見つけることができず、エラーを返します。(バグ 6673415)

回避方法: メッセージキュー インストーラを実行する前に、次のコマンドを使用して JDK 1.5 を手動でインストールします。

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

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

arch は *i586* か *amd64* のいずれかです。

- インストーラを予行モード (`installer -n`) で実行すると、「概要」画面にいくつかのエラーメッセージが表示され、「未完了」のインストール状態も表示されません。これは誤りで、誤解を招きます。予行モードでは、何もシステムにインストールされません。後でサイレントインストールの実行に使用できる回答ファイルが作成されるだけです。(バグ 6594351)

インストーラのバージョン表示の問題

インストーラには、メッセージキューのバージョン情報が不明瞭な形式で表示されます。(バグ 6586507)

Solaris プラットフォーム

Solaris プラットフォームの場合は、次の表を参照して、インストーラに表示されるメッセージキューのバージョンを判別してください。

表 1-13 バージョン文字列の読み換え

Solaris OS でインストーラに表示されるバージョン	対応するメッセージキューのリリース
4.4.1.0	4.4 Update 1
4.4.0.0	4.4
4.3.0.0	4.3
4.2.0.0	4.2
4.1.0.2	4.1 Patch 2
4.1.0.1	4.1 Patch 1
4.1.0.0	4.1
3.7.2.1	3.7 UR2 Patch 1
3.7.0.2	3.7 UR2
3.7.0.1	3.7 UR1
3.6.0.0	3.6
3.6.0.4	3.6 SP4

表 1-13 バージョン文字列の読み換え (続き)

Solaris OS でインストーラに 表示されるバージョン	対応するメッセージキューのリリース
3.6.0.3	3.6 SP3
3.6.0.2	3.6 SP2
3.6.0.1	3.6 SP1

注-3.6 SP4 のパッチリリースの場合 (たとえば、3.6 SP4 Patch 1)、インストーラで表示されるリリース文字列は同じままです。厳密なバージョンを判別するには、コマンド `imqbrokerd -version` を実行する必要があります。

Linux プラットフォーム

Linux のプラットフォームの場合、インストーラで表示されるバージョン番号は次の形式になります。

majorReleaseNumber.minorReleaseNumber-someNumber

たとえば、3.7-22 のようになります。これにより、3.7 リリースの 1 つであることだけはわかりますが、どの特定のバージョンかはわかりません。インストールされているメッセージキューのバージョンを判別するには、次のコマンドを実行します。

`imqbrokerd -version.`

ローカリゼーションの問題

次に示す問題は、ローカリゼーションの問題に関係しています。

- インストーラを英語以外のロケールでテキストモード (`installer -t`) で実行すると、マルチバイト文字が文字化けします。(バグ 6586923)
- インストーラの「進行状況」画面で、進行状況のバーに見慣れない文字が表示されます。ツールヒントは、英語以外のロケールではハードコードされていません。(バグ 6591632)
- インストーラの「ライセンス」画面には、インストーラが実行されるロケールに関係なく、英語のライセンステキストが表示されます。(バグ 6592399)
回避方法: ローカライズされたライセンスファイルにアクセスするには、`LICENSE_MULTILANGUAGE.pdf` ファイルを検索してください。
- インストーラの使用方法に関するヘルプテキストがローカライズされていません。(バグ 6592493)
- インストーラの概要 HTML ページに表示される文字列「None」は、英語でハードコードされています。(バグ 6593089)

- インストーラをドイツ語のロケールで実行すると、ほかのロケールでは表示される開始画面のテキストが正しく表示されません。(バグ 6592666)
- インストーラの「Install Home」画面に表示される文字列「Install Home」がローカライズされていません。その文字列は、英語以外のロケールでインストーラを実行しても、英語で表示されます。(バグ 6592491)
- テキストモード (`installer -t`) でインストーラを実行すると、どのロケールでインストーラを実行するかに関係なく、英語の応答選択肢「Yes」および「No」が使用されます。(バグ 6593230)
- インストーラの「JDK 選択」画面の参照ボタンのツールヒントは、英語でハードコードされています。(バグ 6593085)

使用されなくなったパスワードオプション

以前のバージョンのメッセージキューでは、`-p` または `-password` オプションを使用して、`imqcmd`、`imqbrokerd`、および `imdbmgr` コマンドのパスワードを対話形式で指定することができました。version 4.0 から、これらのオプションは使用できなくなりました。

代わりに、関連するパスワードを指定したパスワードファイルを作成し、`-passfile` コマンドオプションを使用してそのパスワードファイルを参照できます。または、コマンドで要求されたときにパスワードを入力することもできます。

パスワードファイルには、次に示すパスワードを1つ以上格納することができます。

- SSL キーストアを開くために使用するキーストアパスワード。このパスワードを指定するには `imq.keystore.password` プロパティを使用します。
- 接続が匿名でない場合に LDAP ディレクトリとの接続のセキュリティを確保するために使用する LDAP リポジトリパスワード。このパスワードを指定するには `imq.user_repository.ldap.password` プロパティを使用します。
- JDBC 準拠のデータベースとの接続に使用する JDBC データベースパスワード。このパスワードを指定するには `imq.persist.jdbc.vendorName.password` プロパティを使用します。プロパティ名の `vendorName` コンポーネントは、データベースベンダーを指定する変数です。 `hadb`、`derby`、`pointbase`、`oracle`、`mysql` から選択できます。
- `imqcmd` コマンド (ブローカ管理タスクを実行する) に対するパスワード。このパスワードを指定するには `imq.imqcmd.password` プロパティを使用します。

次の例では、JDBC データベースに対するパスワード `abracadabra` をパスワードファイルに設定しています。

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

パスワードファイルは、次のいずれかの方法で使用できます。

- ブローカがパスワードファイルを使用するように、ブローカの `config.properties` ファイルに次のプロパティを設定する。

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

- 関連するコマンドの `-passfile` オプションを使用する。次に例を示します。

```
imqbrokerd -passfile passwordFileName
```

管理および設定上の問題

次に示す問題はメッセージキューの管理および設定に関係するものです。

- Windows プラットフォームでは、`imqsvcadm` コマンドを使用して、メッセージキューブローカを Windows サービスとして手動で追加する必要があります。インストーラはこの処理を実行しません。
- Windows プラットフォームの場合、デフォルトで有効になっている組み込みの Windows ファイアウォールで、ブローカがクライアントから受信接続を受け付けるようにするためのファイアウォール規則を手動で設定する必要があります。(バグ 6675595)

1. コントロールパネルの「Windows ファイアウォール」をダブルクリックします。

Windows ファイアウォールの設定ダイアログを開くには、「ユーザーアカウント制御」ダイアログで「続行」をクリックします。

2. Windows ファイアウォールの設定ダイアログで、「例外」タブをクリックします。
3. 「プログラムの追加」をクリックします。
4. 「プログラムの追加」ダイアログで、「`java.exe`」を選択し、「参照」をクリックします。

Windows はブローカプロセスを Java Platform SE バイナリとして識別します。従って、ブローカが使用する `java.exe` (通常は `jdk1.5.0_15\jre\bin\java.exe`) を探します。

5. 「スコープの変更」をクリックします。
6. 「スコープの変更」ダイアログで、「任意のコンピュータ (インターネット上のコンピュータを含む)」を選択します。
7. 「了解」をクリックします。
8. 「プログラムの追加」ダイアログで「OK」をクリックします。
9. Windows ファイアウォールの設定ダイアログで「OK」をクリックします。

- Windows プラットフォームの場合、CLASSPATH に二重引用符が含まれていると、mqadmin および mqobjmgr コマンドの実行時にエラーがスローされます。(バグ 5060769)

回避方法: コマンドプロンプトウィンドウを開き、次のように入力して CLASSPATH を設定解除します。

```
set classpath=
```

その後、同じコマンドプロンプトウィンドウで、必要なコマンドを実行します。次に例を示します。

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

- すべての Solaris および Windows スクリプトで、-javahome オプションの値に空白文字が含まれると動作しません。(バグ 4683029)

javahome オプションはメッセージキュー コマンドおよびユーティリティで使用し、使用する代替の Java 2 互換のランタイムを指定します。ただし、代替の Java 2 互換のランタイムへのパスには、空白文字を含めることはできません。空白文字を含むパスの例は次のとおりです。

Windows: C:\jdk 1.4

Solaris: /work/java 1.4

回避方法: Java ランタイムを、空白文字を含まない場所またはパスにインストールします。

- mqQueueBrowserMaxMessagesPerRetrieve 属性は、クライアントランタイムがキューの内容を検索するとき一度に取得することのできるメッセージの最大数を指定します。この属性は、キューに入れられたメッセージがバッチ処理され、クライアントランタイムに配信される方法には影響しますが、参照されるメッセージの総数には影響しません。この属性は参照機構のみに影響し、キューのメッセージの配信には影響しません。(バグ 6387631)
- SELinux を実行している Linux プラットフォームでは、Update Center の pkg コマンドが失敗します。(バグ 6892062)

回避方法: この問題は Update Center の既知の問題により発生します (https://updatecenter2.dev.java.net/issues/show_bug.cgi?id=1211)。次のコマンドを使用して、Enforcement が有効な SELinux で pkg コマンドが機能するようにします。

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

ブローカの問題

次に示す問題はメッセージキューブローカに影響します。

- メッセージキュー 4.4 クライアントは、メッセージキュー 3.7 ブローカに接続するときに不明瞭な警告を受信します。(バグ 6899886)

メッセージキュー 4.4 クライアントがメッセージキュー 3.7 ブローカに接続するときに、クライアントは次の形式の警告を受信します。

```
WARNING [I500]: Caught JVM exception: ...
[4036]: A broker error occurred. :[505] bad version ...
```

この「正しくないバージョン」の警告は、プロトコルレベルを下げてもクライアントをブローカに再接続すべきであることを示しています。

- JDBC データストアを使用する場合、データベースのパスワードが平文で保存されます。(バグ 6691717)

回避方法: 『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「[Password Files](#)」で説明しているように、データベースのパスワードを含むパスワードファイルをセキュリティー保護します。

- 持続データストアがあまりにも多くの送信先を開く場合、ブローカがアクセス不可能になります。(バグ 4953354)

回避方法: この状態はブローカがシステムのオープンファイル記述子の制限に達したことが原因です。Solaris や Linux では、ulimit コマンドを使って、ファイル記述子の制限を増やします。

- 送信先が破棄された場合、コンシューマが孤立します。(バグ 5060787)

送信先が破棄された場合、アクティブコンシューマが孤立します。いったんコンシューマが孤立すると、送信先が再作成された場合でもメッセージを受信しなくなります。

- HTTP 接続サービスを使用している JMS クライアントが、Ctrl-C の使用などにより突然終了した場合、ブローカがクライアント接続や関連するすべてのリソースを解放するまでに、およそ 1 分かかります。

この 1 分間にクライアントのほかのインスタンスが起動し、同じ ClientID、永続サブスクリプション、またはキューを使おうとした場合、そのインスタンスは「クライアント ID はすでに使用されています」の例外を受け取ります。このことは実際の問題ではなく、上記の終了処理の結果にすぎません。およそ 1 分経過後にクライアントが起動すると、すべて問題なく動作します。

- データストアに MySQL データベースを使用している場合、1M バイトを超えるメッセージを格納すると、「クエリーのバケットが大きすぎます...」という SQLException がスローされます。(バグ 6682815)

回避方法: `--max_allowed_packet` オプションでデフォルトの 1M バイトより大きい値を設定して MySQL サーバーを起動します。たとえば、次の値を使用します。

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

- 高可用性共有データストアに MySQL データベースを使用している場合、MySQL ストレージエンジンを NDBCLUSTER として設定する機構が必要です。(バグ 6691394)

回避方法: ブローカの `config.properties` ファイルに、次のプロパティ値を追加します (『Sun GlassFish Message Queue 4.4 Administration Guide』の「Enhanced Clusters: JDBC Configuration Properties」を参照)。

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

- Oracle の 9i (JDBC 9.2.0.x) ドライバを使用する場合、ブローカが「Failed to persist property...」という例外をスローします。(バグ 6626825)

回避方法: Oracle の 10g (JDBC 10.2.0.x) ドライバを使用します。ブローカはこのドライバに対して最適化されます。

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

- データストアに Java DB データベースを使用している場合、メッセージを格納すると、「要求された時間内に例外を取得できませんでした」という `SQLException` がスローされます。(バグ 6691394)

回避方法: ブローカの `config.properties` ファイルに次のプロパティ値を設定します。

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

- AIX で IBM JVM を使用しているときに、明白な理由なく、ブローカが低メモリまたはメモリ不足の状態に陥ります。(バグ 6899526)

回避方法: IBM JVM の最新のバージョン (Java Runtime 1.6.0 IBM Corporation 以降) を使用して、`imqbrokerd` に次の IBM JVM GC オプションを渡します。

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

ブローカクラスタ

次に示す問題は、ブローカクラスタに影響します。

- MySQL Cluster データストアを使用する高可用性ブローカは、異常終了した場合、再起動に失敗します。(バグ 6896877)

回避方法: この問題は MySQL Cluster の既知の問題により発生します (<http://bugs.mysql.com/bug.php?id=47955>)。この問題に対する修正は、MySQL バージョン 5.1.39-ndb-6.3.28、5.1.39-ndb-7.0.9、および 5.1.39-ndb-7.1.0 に組み込まれています。

- このリリースでは、フル接続のブローカクラスタのみサポートされています。つまり、クラスタ内のすべてのブローカは、そのクラスタ内のほかのブローカと相互に直接やり取りする必要があります。`imqbrokerd -cluster` コマンド行引数を使用してブローカを従来のクラスタに接続する場合は、そのクラスタ内のすべてのブローカが含まれていることを確認してください。

- クライアントが拡張ブローカクラスタ内のブローカに接続されている場合、クライアントランタイムは、`imqAddressListIterations` 接続ファクトリ属性の値を無視して、成功するまで再接続を試みます。
- クライアントが検索できるのは、ホームブローカにあるキューの内容のみです。この場合でも、クライアントは、クラスタ内の任意のブローカに対してキューにメッセージを送信したりキューからのメッセージを消費したりできません。制約を受けるのはキューの検索のみです。
- Version 4.3 のブローカが含まれる従来のクラスタでは、すべてのクラスタが Version 3.5 以降でなければなりません。
- デフォルトでは、メッセージキュー 4.3、4.2、および 4.1 のブローカをメッセージキュー 3.7 または 3.6 のブローカと同時に使用できません。これは、これらのバージョン間で `imq.autocreate.queue.maxNumActiveConsumers` のデフォルト値が変更されているためです。(バグ 6716400)

回避方法: すべてのブローカで `imq.autocreate.queue.maxNumActiveConsumers` の値を同じにします。通常は、メッセージキュー 4.3、4.2、および 4.1 の設定を変更して、3.7 または 3.6 のブローカで使用されている設定に一致させます(デフォルトでは、-1 の値を以前のバージョンのデフォルト値の 1 に変更します)。

- Message Queue 4.3 (または 4.x) ブローカを Message Queue 3.x ブローカクラスタに追加するには、マスターブローカが動作している必要があります。(バグ 6763796)
- 従来のクラスタから拡張クラスタに変換するときに、メッセージキューデータベースマネージャユーティリティ (`imqdbmgr`) を使用して、既存のスタンドアロン JDBC ベースのデータストアから共有 JDBC データストアに変換できます。方法については、『[Sun GlassFish Message Queue 4.4 Administration Guide](#)』の「[Cluster Conversion: JDBC-Based Data Store](#)」を参照してください。
- HADB を使用するブローカは、10M バイトより大きなメッセージを処理できません。(バグ 6531734)
- コマンド `imqdbmgr upgrade hastore` を使用した HADB ストアへの変換は、ストアに 10,000 を超えるメッセージが保持されている場合、「設定されているロックの数が多すぎます」というメッセージが表示されて失敗することがあります。(バグ 6588856)

回避方法: 次のコマンドを使用して、ロックの数を増やします。

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

詳細は、『[Sun Java System Application Server 9.1 Enterprise Edition トラブルシューティングガイド](#)』の「HADB の問題」を参照してください。

- 1 つのトランザクションに 500 を超えるリモートメッセージがコミットされている場合、ブローカは「HADB-E-12815: Table memory space exhausted」というエラーを返します。(バグ 6550483)

詳細は、『[Sun Java System Application Server 9.1 Enterprise Edition トラブルシューティングガイド](#)』の「HADB の問題」を参照してください。

- ブローカクラスタで、ブローカが開かれていないリモート接続へのメッセージをキューに入れます。(バグ 4951010)

回避方法: いったんその接続が開かれると、メッセージはコンシューマによって受信されます。コンシューマの接続が閉じたままの場合、メッセージは別のコンシューマへ再配信されます。

- 1つのトランザクションでリモートブローカから複数のメッセージを消費しているときは、次のエラーメッセージがブローカに記録される可能性があります。このメッセージは無視しても問題ありません。

```
[26/Jul/2007:13:18:27 PDT] WARNING [B2117]:
Message acknowledgement failed from
mq://129.145.130.95:7677/?instName=a&brokerSessionUID=3209681167602264320:
  ackStatus = NOT_FOUND(404)\
  Reason = Update remote transaction state to COMMITTED(6):
transaction 3534784765719091968 not found, the transaction
may have already been committed.
  AckType = MSG_CONSUMED
  MessageBrokerSession = 3209681167602264320
  TransactionID = 3534784765719091968
  SysMessageID = 8-129.145.130.95(95:fd:93:91:ec:a0)-33220-1185481094690
  ConsumerUID = 3534784765719133952\par
```

```
[26/Jul/2007:13:18:27 PDT] WARNING Notify commit transaction
[8-129.145.130.95(95:fd:93:91:ec:a0)-33220-1185481094690,
[consumer:3534784765719133952, type=NONE]]
TUID=3534784765719091968 got response:
com.sun.messaging.jmq.jmsserver.util.BrokerException:
Update remote transaction state to COMMITTED(6):
transaction 3534784765719091968 not found, the transaction may have already
been committed.:
com.sun.messaging.jmq.jmsserver.util.BrokerException: Update remote transaction
state to COMMITTED(6): transaction 3534784765719091968 not found, the transaction
may have already been committed.r
```

このメッセージは、`imq.txn.reapLimit` プロパティが1つのトランザクション内のリモートメッセージの数と比較して少ない場合に、トランザクション内の後のメッセージのためにコミットをメッセージホームブローカに通知するときに記録されます。(バグ 6585449)

回避方法: このメッセージを回避するには、`imq.txn.reapLimit` プロパティの値を増やします。

JMXの問題

Windows プラットフォームでは、トランザクションマネージャーの監視 MBean の `getTransactionInfo` メソッドが不正なトランザクションの作成時間を含むトランザクション情報を返します。(バグ 6393359)

回避方法: 代わりに、トランザクションマネージャーの監視 MBean の `getTransactionInfoByID` メソッドを使用します。

SOAP サポート

SOAP サポートに関連した 2 つの問題に注意する必要があります。

- Message Queue の version 4.0 のリリースから、SOAP 管理によるオブジェクトのサポートは中止されています。
- SOAP 開発は、`SUNWjaf`、`SUNWjmail`、`SUNWxsrt`、および `SUNWjaxp` ファイルに依存します。Message Queue の version 4.1 では、これらのファイルを利用できるのは JDK version 1.6.0 以降で Message Queue を実行している場合のみです。
- 以前は、`SAAJ 1.2` 実装 `.jar` が `mail.jar` を直接参照していました。`SAAJ 1.3` では、この参照が削除されたため、Message Queue クライアントが `mail.jar` を明示的に `CLASSPATH` 内に配置する必要があります。

再配布可能なファイル

Sun GlassFish メッセージキュー 4.4 Update 1 には、バイナリ形式で使用でき自由に配布可能な次のファイルのセットが含まれています。

<code>fscontext.jar</code>	<code>jaxm-api.jar</code>
<code>imq.jar</code>	<code>jms.jar</code>
<code>imqjmx.jar</code>	<code>libmqcrt.sl (HP-UX)</code>
<code>imqxm.jar</code>	<code>libmqcrt.so (UNIX)</code>
<code>imqums.war</code>	<code>mqcrt1.dll (Windows)</code>

さらに、`LICENSE` ファイルおよび `COPYRIGHT` ファイルも再配布できます。

障害を持つユーザー向けのアクセシビリティ機能

このメディアの出版後にリリースされたアクセシビリティを入手する場合は、請求に応じて Sun から提供される 508 条に関する製品評価資料を参照し、使いやすいソリューションの配備にもっとも適したバージョンを調べてください。アプリケーションの最新バージョンは、<http://sun.com/software/javaenterprisesystem/get.html> で入手できます。

アクセシビリティ機能に対する Sun のコミットメントについては、<http://sun.com/access> を参照してください。

問題の報告とフィードバックの方法

Sun GlassFish メッセージキューに問題が発生した場合は、次のいずれかの方法で Sun のカスタマサポートにお問い合わせください。

- Sun ソフトウェアサポートサービスオンライン (<http://www.sun.com/service/sunone/software>)。

このサイトには、ナレッジベース、オンラインサポートセンター、ProductTracker へのリンクと保守プログラムおよびサポートの連絡先電話番号へのリンクがあります。

- 保守契約を結んでいるお客様の場合は、専用ダイヤルをご利用ください。

最善の問題解決のため、サポートに連絡する際には次の情報をご用意ください。

- 問題が発生した箇所や動作への影響など、問題の具体的な説明
- マシンのタイプ、OS のバージョン、および製品のバージョン (問題に関連している可能性のあるパッチやその他のソフトウェアを含む)
- 問題を再現するために用いた詳細な手順。
- すべてのエラーログまたはコアダンプ。

Sun GlassFish Software Forum

次のサイトでは、Sun GlassFish メッセージキューフォーラムが利用できます。

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

ご参加を歓迎いたします。

Java Technology Forum

Java Technology Forums には、関連する JMS のフォーラムがあります。

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

このドキュメントに関するコメント

弊社では、ドキュメントの改善に努めており、お客様からのコメントおよびご忠告をお受けしております。

<http://docs.sun.com> にアクセスして「コメントの送信」をクリックしてください。このオンラインフォームでは、ドキュメントのタイトルと Part No. もご記入ください。Part No. は、7桁か9桁の番号で、ドキュメントのタイトルページまたは最初のページに記載されています。このドキュメントのタイトルは『リリースノート』で、Part No. は 821-1514-11 です。

その他の情報

次のインターネットのサイトで、Sun GlassFish の情報を参照できます。

- ドキュメント
<http://docs.sun.com/prod/java.sys>
- プロフェッショナルサービス
<http://www.sun.com/service/sunps/sunone>
- ソフトウェア製品およびサービス
<http://www.sun.com/software>
- ソフトウェアサポートサービス
<http://www.sun.com/service/sunone/software>
- サポートおよびナレッジベース
<http://www.sun.com/service/support/software>
- Sun サポートおよびトレーニングサービス
<http://training.sun.com>
- コンサルティングおよびプロフェッショナルサービス
<http://www.sun.com/service/sunps/sunone>
- 開発者向けの情報
<http://developers.sun.com>
- Sun 開発者サポートサービス

<http://www.sun.com/developers/support>

- ソフトウェアトレーニング

<http://www.sun.com/software/training>