



Sun Java System Message Queue 4.1 릴리스 노트



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

부품 번호: 820-3192
2007년 9월

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 모든 권리는 저작권자의 소유입니다.

Sun Microsystems, Inc.는 이 문서에 설명된 제품의 기술 관련 지적 재산권을 소유합니다. 특히 이 지적 재산권에는 하나 이상의 미국 특허권이 포함될 수 있으며, 미국 및 다른 국가에서 하나 이상의 출원 중인 특허권이 제한 없이 포함될 수 있습니다.

미국 정부 권한 및 상용 소프트웨어. 정부 사용자는 Sun Microsystems, Inc.의 표준 사용권 계약과 해당 FAR 규정 및 보충 규정을 준수해야 합니다.

이 배포판에는 타사에서 개발한 자료가 포함되어 있을 수 있습니다.

제품 중에는 캘리포니아 대학에서 허가한 Berkeley BSD 시스템에서 파생된 부분이 포함되어 있을 수 있습니다. UNIX는 미국 및 다른 국가에서 X/Open Company, Ltd.를 통해 독점적으로 사용권이 부여되는 등록 상표입니다.

Sun, Sun Microsystems, Sun 로고, Solaris 로고, Java Coffee Cup 로고, docs.sun.com, Java 및 Solaris는 미국 및 다른 국가에서 Sun Microsystems, Inc.의 상표 또는 등록 상표입니다. 모든 SPARC 상표는 사용 허가를 받았으며 미국 및 다른 국가에서 SPARC International, Inc.의 상표 또는 등록 상표입니다. SPARC 상표를 사용하는 제품은 Sun Microsystems, Inc.에서 개발한 구조에 기반을 두고 있습니다.

OPEN LOOK 및 SunTM 그래픽 사용자 인터페이스(GUI)는 Sun Microsystems, Inc.가 자사의 사용자 및 정식 사용자로 개발했습니다. Sun은 컴퓨터 업계를 위한 시각적 또는 그래픽 사용자 인터페이스(GUI)의 개념을 연구 개발한 Xerox사의 선구적인 노력을 높이 평가하고 있습니다. Sun은 Xerox와 Xerox 그래픽 사용자 인터페이스(GUI)에 대한 비독점적 사용권을 보유하고 있습니다. 이 사용권은 OPEN LOOK GUI를 구현하는 Sun의 정식 사용자에게도 적용되며 그렇지 않은 경우에는 Sun의 서면 사용권 계약을 준수해야 합니다.

이 설명서에 포함된 정보 및 이 설명서에서 다루는 제품은 미국 수출 관리법이 적용되며 다른 국가의 수출 또는 수입 관리법의 적용을 받을 수 있습니다. 이 제품과 정보를 직간접적으로 핵무기, 미사일 또는 생화학 무기에 사용하거나 핵과 관련하여 해상에서 사용하는 것은 엄격하게 금지합니다. 미국 수출 금지 국가 또는 금지된 개인과 특별히 지정된 국민 목록을 포함하여 미국 수출 금지 목록에 지정된 대상으로의 수출이나 재수출은 엄격하게 금지됩니다.

설명서는 "있는 그대로" 제공되며, 법률을 위반하지 않는 범위 내에서 상품성, 특정 목적에 대한 적합성 또는 비침해에 대한 묵시적인 보증을 포함하여 모든 명시적 또는 묵시적 조건, 표현 및 보증을 배제합니다.

목차

1 Sun Java System Message Queue 4.1 릴리스 노트	5
릴리스 노트 개정 내역	6
Message Queue 4.1 정보	6
4.1 릴리스의 새로운 기능	7
하드웨어 및 소프트웨어 요구 사항	15
Message Queue 4.0 정보	16
4.0 릴리스의 새로운 기능	16
하드웨어 및 소프트웨어 요구 사항	31
이 릴리스에서 해결된 버그	31
중요 정보	33
설치 정보	33
호환성 문제	33
Message Queue 4.1 설명서 업데이트	34
알려진 문제점 및 제한 사항	35
설치 문제	35
더 이상 사용되지 않는 비밀번호 옵션	40
일반적인 문제	41
관리/구성 문제	41
브로커 문제	42
브로커 클러스터	42
JMX 문제	44
SOAP 지원	45
재배포 가능 파일	45
내게 필요한 옵션 기능	45
문제 보고 및 사용자 의견 제공 방법	45
Sun Java System 소프트웨어 포럼	46
Java 기술 포럼	46
사용자 의견 환영	46

Sun의 추가 자원	47
------------------	----

Sun Java System Message Queue 4.1 릴리스 노트

버전 4.1

부품 번호 820-3192

이 릴리스 노트에는 Sun Java™ System Message Queue 4.1의 릴리스 시점에 알려진 중요한 정보가 들어 있습니다. 이 문서에서는 새로 추가된 기능과 향상된 기능, 알려진 제한 사항과 문제점 및 기타 정보를 제공합니다. Message Queue를 사용하기 전에 먼저 이 문서를 읽어 보십시오. 이 릴리스 노트에는 Message Queue 4.0 릴리스에 대한 정보도 포함되어 있습니다. 이 릴리스에서 소개된 기능에 대한 정보는 [16 페이지 “Message Queue 4.0 정보”](#)를 참조하십시오.

이 릴리스 노트의 최신 버전은 Sun Java System Message Queue 설명서 웹 사이트에서 찾아볼 수 있습니다. 소프트웨어를 설치 및 설정하기 전과 후에도 웹 사이트를 검토하여 최신 릴리스 노트와 제품 설명서를 정기적으로 확인하시기 바랍니다.

이 릴리스 노트는 다음 내용으로 구성되어 있습니다.

- 6 페이지 “릴리스 노트 개정 내역”
- 6 페이지 “Message Queue 4.1 정보”
- 16 페이지 “Message Queue 4.0 정보”
- 31 페이지 “이 릴리스에서 해결된 버그”
- 33 페이지 “중요 정보”
- 35 페이지 “알려진 문제점 및 제한 사항”
- 45 페이지 “재배포 가능 파일”
- 45 페이지 “내게 필요한 옵션 기능”
- 45 페이지 “문제 보고 및 사용자 의견 제공 방법”
- 46 페이지 “사용자 의견 환영”
- 47 페이지 “Sun의 추가 자원”

이 문서에는 관련된 정보를 추가로 얻을 수 있는 타사 URL도 나와 있습니다.

Sun은 이 문서에 언급된 타사 웹 사이트의 이용 가능성에 대해 책임을 지지 않습니다. Sun은 그러한 사이트 또는 자원에 있거나 사용 가능한 내용, 광고, 제품 또는 기타 자료에

대하여 보증하지 않으며 책임 또는 의무를 지지 않습니다. Sun은 해당 사이트나 자원을 통해 사용 가능한 내용, 상품 또는 서비스의 사용과 관련하여 발생했거나 발생했다고 간주되는 손해나 손실에 대해 책임 또는 의무를 지지 않습니다.

릴리스 노트 개정 내역

다음 표에서는 Message Queue 제품의 모든 4.x 릴리스 날짜를 나열하고 각 릴리스와 관련된 주요 변경 사항에 대해 설명합니다.

표 1-1 개정 내역

날짜	변경 사항 설명
2006년 5월	Message Queue의 버전 4.0에 대한 이 문서의 초기 릴리스입니다.
2007년 1월	Message Queue의 베타 버전 4.1에 대한 이 문서의 초기 릴리스입니다. JAAS 지원에 대한 설명이 추가되었습니다.
2007년 4월	Message Queue의 베타 버전 4.1에 대한 이 문서의 두 번째 릴리스입니다. 고가용성 기능이 추가되었습니다.
2007년 9월	공식 출시를 위한 이 문서의 세 번째 릴리스입니다. Java Enterprise System Monitoring Framework, 수정된 C 포트, 버그 수정 사항 및 기타 기능의 지원에 대한 설명이 추가되었습니다.

Message Queue 4.1 정보

Sun Java System Message Queue는 완벽한 기능의 메시지 서비스로 Java Messaging Specification(JMS) 1.1을 따르는 안정적인 비동기식 메시징을 제공합니다. 또한 Message Queue는 JMS 사양을 능가하는 다양한 기능을 제공하여 대규모 엔터프라이즈 배포 요구를 충족시킵니다.

Message Queue 버전 4.1에는 고가용성, JAAS(Java Authentication and Authorization Service), 수정된 C 포트 사용 및 Java Enterprise System Monitoring Framework에 대한 지원과 몇 가지 향상된 기능 및 버그 수정 사항이 추가되었습니다. 이 절은 다음 내용으로 구성되어 있습니다.

- 7 페이지 “4.1 릴리스의 새로운 기능”
- 15 페이지 “하드웨어 및 소프트웨어 요구 사항”

Message Queue 4.0에 소개된 기능에 대한 내용은 16 페이지 “Message Queue 4.0 정보”를 참조하십시오.

4.1 릴리스의 새로운 기능

Message Queue 4.1에는 고가용성(데이터 및 서비스 가용성) 브로커 클러스터, JAAS 지원 및 기타 여러 가지 사소한 기능이 도입되었습니다. 이 절에서는 이러한 기능과 추가 참조 정보에 대해 설명합니다.

- 7 페이지 “고가용성”
- 8 페이지 “JAAS 지원”
- 13 페이지 “영구 저장소 형식 변경 사항”
- 14 페이지 “브로커 구성”
- 14 페이지 “JES Monitoring Framework 지원”
- 15 페이지 “트랜잭션 관리”
- 15 페이지 “C 클라이언트 연결을 위한 고정 포트”

고가용성

Message Queue 4.1에 서비스 가용성 및 데이터 가용성을 제공하는 고가용성 클러스터가 도입되었습니다. 클라이언트와 고가용성 브로커의 연결이 끊어지면 클라이언트가 클러스터 내의 다른 브로커와 자동으로 다시 연결됩니다. 새로 연결될 브로커는 실패한 브로커의 영구 데이터 및 상태를 인계 받으며 실패한 브로커의 클라이언트에 서비스를 중단 없이 계속 제공합니다. 보안 연결을 통해 고가용성 브로커를 실행할 수 있습니다.

고가용성 브로커에는 고가용성 데이터베이스(HADB) 사용이 필수입니다. 이러한 유형의 데이터베이스가 없거나 데이터 가용성이 중요하지 않은 경우에는 자동 다시 연결과 서비스 가용성을 제공하는 기존 클러스터를 계속 사용할 수 있습니다.

고가용성 브로커를 구성하는 방법은 간단합니다. 클러스터의 각 브로커에 다음과 같은 종류의 브로커 등록 정보를 지정합니다.

- **클러스터 구성원 등록 정보** - 브로커가 고가용성 클러스터의 일부인지 여부, 클러스터 아이디 및 브로커 아이디를 지정합니다.
- **고가용성 데이터베이스(HADB) 등록 정보** - 영구 메시지 모델(JDBC), HADB 공급업체 이름 및 데이터베이스에 대한 공급업체 특정 구성 등록 정보를 지정합니다.
- **오류 감지 및 인계 등록 정보** - 브로커 오류를 감지하고 처리하는 방법을 지정합니다.

이 기능을 사용하려면 다음 작업을 수행해야 합니다.

1. 고가용성 데이터베이스를 설치합니다.
2. JDBC 드라이버의 .jar 파일을 설치합니다.
3. 고가용성 영구 저장소에 데이터베이스 스키마를 만듭니다.
4. 클러스터의 각 브로커에 고가용성과 관련된 등록 정보를 설정합니다.
5. 클러스터의 각 브로커를 시작합니다.

고가용성에 대한 개념 설명과 일반 클러스터와의 비교에 대해서는 **Sun Java System Message Queue 4.1 Technical Overview**의 4 장, “Broker Clusters”를 참조하십시오.

고가용성에 대한 절차 및 참조 정보에 대해서는 **Sun Java System Message Queue 4.1**

Administration Guide의 8 장, “Broker Clusters” 및 **Sun Java System Message Queue 4.1 Administration Guide**의 “Cluster Configuration Properties”를 참조하십시오.

Message Queue 버전 4.0에서 HADB 데이터베이스를 사용하고 있는 경우 고가용성 클러스터를 사용하려면 dbmgr 유틸리티를 사용하여 공유 HADB 저장소로 업그레이드할 수 있습니다. 자세한 내용은 [42 페이지](#) “브로커 클러스터”를 참조하십시오.

JAAS 지원

Message Queue는 기본으로 제공된 파일 기반 및 LDAP 기반 인증 메커니즘 이외에도 다양한 서비스를 브로커로 플러그인하여 Message Queue 클라이언트를 인증할 수 있도록 해주는 JAAS(Java Authentication and Authorization Service)도 지원합니다. 이 절에서는 브로커를 통해 JAAS 호환 인증 서비스에 사용할 수 있는 정보와 브로커를 구성하여 이러한 서비스를 사용하는 방법에 대해 설명합니다.

JAAS API에 대한 설명은 이 문서에서 다루지 않습니다. 자세한 내용을 보려면 다음 소스를 참조하십시오.

- JAAS API에 대한 전체 내용은 **Java Authentication and Authorization Service (JAAS) Reference Guide**를 참조하십시오.
<http://java.sun.com/j2se/1.5.0/docs/guide/security/jaas/JAASRefGuide.html>
- LoginModule 쓰기에 대한 내용은 **Java Authentication and Authorization Service (JAAS) LoginModule Developer's Guide**를 참조하십시오.
<http://java.sun.com/j2se/1.5.0/docs/guide/security/jaas/JAASLMDevGuide.html>

JAAS API는 J2SE에서 코어 API이므로 이는 Message Queue 런타임 환경을 구성하는 일부분입니다. JAAS는 응용 프로그램과 인증 메커니즘 간의 추상화 계층을 정의하며 원하는 메커니즘을 응용 프로그램 코드 변경 없이 플러그인할 수 있도록 해줍니다. Message Queue 서비스의 경우 추상화 계층은 브로커(응용 프로그램)와 인증 공급자 사이에 놓입니다. 몇 가지 브로커 등록 정보를 설정하여 JAAS 호환 인증 서비스를 플러그인할 수 있으며 브로커 코드의 중단 또는 변경 없이 이 서비스를 업그레이드할 수 있습니다.

JAAS 기반 인증을 사용하고 있는 경우 JMX 클라이언트를 사용하여 브로커를 관리할 수 있지만 이 경우 브로커를 시작하기 전에 JAAS 지원을 수동으로 설정(JAAS 관련 브로커 등록 정보 설정)해야 합니다. JMX API를 사용하여 해당 등록 정보를 변경할 수 없습니다.

JAAS 요소

[그림 1-1](#)에서는 JAAS의 기본 요소를 보여 줍니다. JAAS의 기본 요소(JAAS 클라이언트, JAAS 호환 인증 서비스 및 JAAS 구성 파일)를 나타냅니다.

- JAAS 클라이언트는 JAAS 호환 인증 서비스를 사용하여 인증을 수행하려는 응용 프로그램입니다. 이는 LoginModule을 사용하여 본 서비스와 통신하며 LoginModule이 사용자 이름, 비밀번호 및 기타 관련 정보를 가져오기 위해 호출할 수 있는 콜백 처리기의 제공을 담당합니다.

- JAAS 호환 인증 서비스는 하나 이상의 LoginModule 및 필요 인증을 수행하는 논리로 구성됩니다. LoginModule은 인증 논리를 포함하거나 전용 프로토콜 또는 API를 사용하여 해당 논리를 제공하는 모듈과 통신할 수 있습니다.
- JAAS 구성 파일은 JAAS 클라이언트가 JAAS 호환 서비스와의 통신에 필요한 LoginModule을 찾는 데 사용하는 텍스트 파일입니다.

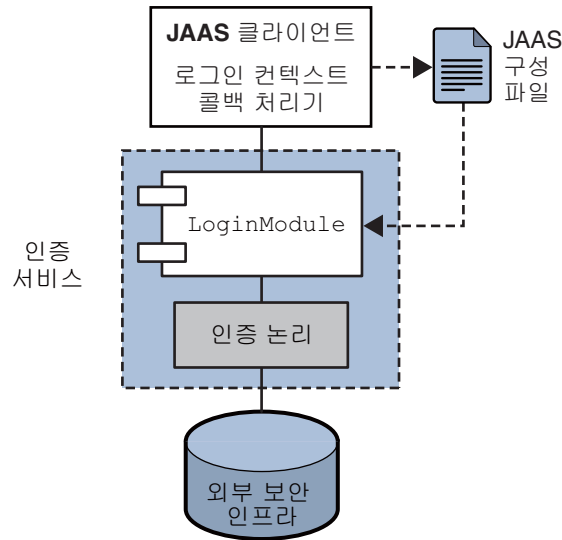


그림 1-1 JAAS 요소

다음 절에서는 Message Queue 서비스에서 이 요소를 사용해 JAAS 호환 인증을 제공하는 방법에 대해 설명합니다.

JAAS 및 Message Queue

다음 그림은 Message Queue 브로커가 JAAS를 사용하는 방법을 나타냅니다. 이 그림은 이전 그림의 JAAS 모델에 비해 좀 더 복잡한 구현을 나타냅니다.

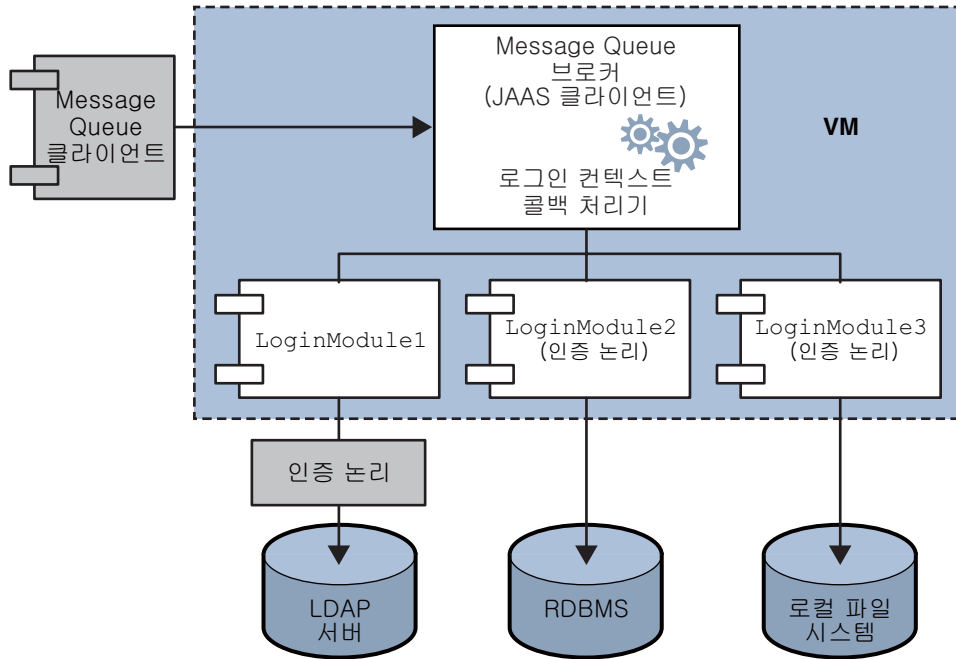


그림 1-2 Message Queue가 JAAS를 사용하는 방법

이전 그림의 단순한 경우에서 보았듯이 인증 서비스 계층은 브로커와 구분되어 있습니다. 인증 서비스는 하나 이상의 로그인 모듈(LoginModule) 및 추가 인증 모듈(필요 시)로 구성됩니다. 로그인 모듈은 브로커와 동일한 Java 가상 머신에서 실행됩니다. Message Queue 브로커는 로그인 모듈에 LogInContext로 표현되며 브로커 런타임 코드의 일부인 CallBackHandler를 통해 로그인 모듈과 통신합니다.

또한 인증 서비스는 로그인 모듈에 대한 항목이 포함된 JAAS 구성 파일을 제공합니다. 구성 파일은 모듈의 사용 순서와 일부 사용 조건을 지정합니다. 브로커가 시작되면 JAAS는 Java 시스템 등록 정보 java.security.auth.login.config 또는 Java 보안 등록 정보 파일로 구성 파일을 찾습니다. 그런 다음 JAAS 구성 파일에서 브로커 등록 정보 imq.user_repository.jaas.name 값에 따른 항목을 선택합니다. 이 항목에는 인증에 사용할 로그인 모듈이 지정되어 있습니다. 그림에서와 같이 브로커는 로그인 모듈을 두 개 이상 사용할 수 있습니다. 구성 파일, 로그인 모듈 및 브로커 간의 관계는 그림 1-3에서 볼 수 있습니다.

브로커에서 JAAS 플러그인 인증 서비스를 사용하면 이 내용이 Message Queue 클라이언트에도 완전히 공개됩니다. 클라이언트는 사용자 이름과 비밀번호를 전달하여 이전과 같이 브로커에 계속 연결됩니다. 한편 브로커는 콜백 처리기를 사용하여 이 정보를 인증 서비스에 전달하며 서비스는 이 정보를 사용하여 사용자를 인증하고 결과를 반환합니다. 인증에 성공하면 브로커에서 연결을 부여하며, 인증에 실패하면 클라이언트 런타임에서 클라이언트가 처리해야 할 JMS 보안 예외를 반환합니다.

Message Queue 클라이언트가 인증된 후 추가 권한 부여가 필요할 경우에는 브로커에서 액세스 제어 파일을 참조하여 인증된 클라이언트가 대상 액세스, 메시지 사용, 대기열 찾아보기 등의 작업을 수행할 수 있는 권한이 부여되어 있는지 여부를 결정합니다(브로커가 정상일 경우).

JAAS 호환 인증 설정

JAAS 호환 인증 설정에는 이 유형의 인증을 선택하고 구성 파일의 위치를 지정하며 사용할 로그인 모듈에 대한 항목을 지정할 수 있는 브로커 및 시스템 등록 정보의 설정이 포함됩니다.

이 절에서는 JAAS 클라이언트, 로그인 모듈 및 JAAS 구성 파일 간의 관계와 JAAS 호환 인증 설정에 필요한 프로세스에 대해 설명합니다. 다음 그림은 구성 파일, 로그인 모듈 및 브로커 간의 관계를 나타냅니다.

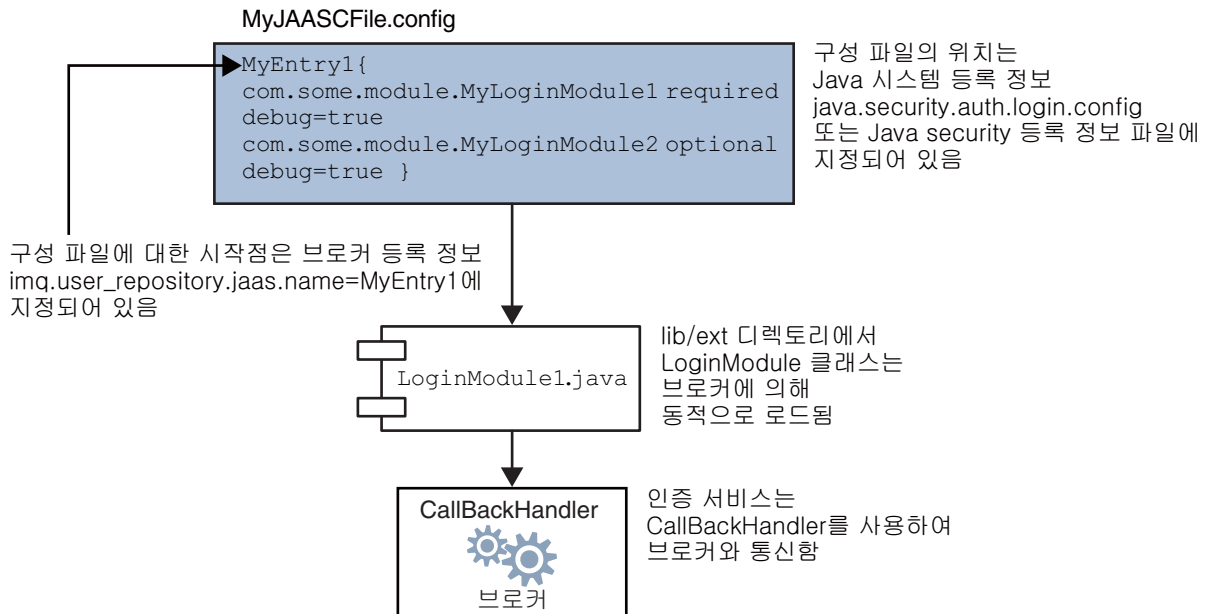


그림 1-3 JAAS 지원 설정

그림에서와 같이 JAAS 구성 파일 MyJAASFile.config에는 시작점에서 그룹화된 여러 로그인 모듈에 대한 참조 항목이 포함됩니다. 브로커에서는 Java 시스템 등록 정보 java.security.auth.login.config 또는 Java 보안 등록 정보 파일을 참조하여 구성 파일을 찾습니다. 사용할 로그인 모듈은 구성 파일에 원하는 항목을 지정할 수 있는 브로커 등록 정보 imq.user_repository.jaas.name을 참조하여 결정합니다. 이 모듈에 대한 클래스는 lib/ext 디렉토리에 있습니다.

Message Queue에 JAAS 지원을 설정하려면 다음 단계를 완료해야 합니다.(개발 환경에서는 개발자가 다음의 모든 단계를 수행할 수 있습니다. 프로덕션 환경에서는 관리자가 이 작업의 일부를 인계 받습니다.)

1. 인증 서비스를 구현하는 로그인 모듈 클래스를 하나 이상 만듭니다. 브로커에서 지원되는 JAAS 콜백 유형은 다음과 같습니다.

`javax.security.auth.callback.LanguageCallback`

브로커에서 이 콜백을 사용하여 브로커가 실행될 로케를 인증 서비스에 전달합니다. 이 값은 현지화에 사용할 수 있습니다.

`javax.security.auth.callback.NameCallback`

브로커에서 이 콜백을 사용하여 연결 요청 시 Message Queue 클라이언트가 지정한 사용자 이름을 인증 서비스에 전달합니다.

`javax.security.auth.callback.TextInputCallback`

`TextInputCallback.getPrompt()`가 `imq.authentication.type`일 때 브로커에서 이 콜백을 사용하여 `imq.authentication.type` 값을 인증 서비스에 지정합니다. 이때 이 필드에 지정할 수 있는 유일한 값은 `basic`입니다. 이는 Base 64 비밀번호 인코딩을 나타냅니다.

`javax.security.auth.callback.PasswordCallback`

브로커에서 이 콜백을 사용하여 연결 요청 시 Message Queue 클라이언트가 지정한 비밀번호를 인증 서비스에 전달합니다.

`javax.security.auth.callback.TextOutputCallback`

브로커에서 이 콜백을 사용하여 텍스트 출력을 브로커 로그 파일에 기록함으로써 인증 서비스에 로깅 서비스를 제공합니다. 콜백의 `MessageType` `ERROR`, `INFORMATION`, `WARNING`은 각각 브로커 로그 수준 `ERROR`, `INFO` 및 `WARNING`에 매핑됩니다.

2. 로그인 모듈 클래스를 참조하는 항목을 사용하여 JAAS 구성 파일을 만들고 이 파일의 위치를 Message Queue 관리자에 지정합니다. (이 파일은 원격으로 찾을 수 있으며 해당 위치를 URL로 지정할 수 있습니다.)
3. 로그인 구현 클래스를 참조하는 항목 이름을 JAAS 구성 파일에 기록합니다.
4. 로그인 모듈을 구현하는 클래스를 `jar` 파일로 아카이브하고 이 `jar` 파일을 Message Queue `lib/ext` 디렉토리에 둡니다.
5. JAAS 지원과 관련된 브로커 등록 정보를 구성합니다. 이에 대해서는 표 1-2에서 설명합니다.
6. 다음 시스템 등록 정보를 설정하여 JAAS 구성 파일의 위치를 지정합니다.

`java.security.auth.login.config=JAAS_Config_File_Location`

예를 들어, 브로커를 시작할 때 다음과 같이 구성 파일을 지정할 수 있습니다.

`imqbrokerd -Djava.security.auth.login.config=JAAS_Config_File_Location`

다른 방법을 사용하여 JAAS 구성 파일의 위치를 지정할 수도 있습니다. 추가 정보는 다음 사이트를 참조하십시오.

<http://java.sun.com/j2se/1.5.0/docs/guide/security/jaas/tutorials/LoginConfigFile.html>

다음 표는 JAAS 지원 설정에 필요한 브로커 등록 정보에 대해 설명합니다.

표 1-2 JAAS 지원을 위한 브로커 등록 정보

등록 정보	설명
<code>imq.authentication.type</code>	Base 64 비밀번호 인코딩을 지정하려면 <code>basic</code> 으로 설정합니다. 이는 JAAS 인증에 지정할 수 있는 유일한 값입니다.
<code>imq.authentication.basic.user_repository</code>	JAAS 인증을 지정하려면 <code>jaas</code> 로 설정합니다.
<code>imq.accesscontrol.type</code>	<code>file</code> 로 설정합니다.
<code>imq.user_repository.jaas.name</code>	JAAS 구성 파일에서 인증 메커니즘으로 사용할 로그인 모듈이 참조된 항목 중 원하는 항목 이름으로 설정합니다. 이 이름은 단계 3에서 기록한 이름입니다.
<code>imq.user_repository.jaas.userPrincipalClass</code>	Message Queue 액세스 제어에서 사용하는 이 등록 정보는 Message Queue 액세스 제어 파일에서 사용자 엔티티를 나타내는 <code>Principal</code> 이름을 추출하기 위해 브로커가 사용하는 로그인 모듈에 <code>java.security.Principal</code> 구현 클래스를 지정합니다. 이 등록 정보를 지정하지 않으면 연결 요청 시 Message Queue 클라이언트로부터 전달된 사용자 이름이 대신 사용됩니다.
<code>imq.user_repository.jaas.groupPrincipalClass</code>	Message Queue 액세스 제어에서 사용하는 이 등록 정보는 Message Queue 액세스 제어 파일에서 그룹 엔티티를 나타내는 <code>Principal</code> 이름을 추출하기 위해 브로커가 사용하는 로그인 모듈에 <code>java.security.Principal</code> 구현 클래스를 지정합니다. 이 등록 정보를 지정하지 않으면 Message Queue 액세스 제어 파일의 그룹 규칙(있는 경우)이 무시됩니다.

영구 저장소 형식 변경 사항

Message Queue 버전 4.1에서는 고가용성을 지원할 수 있도록 JDBC 저장소를 변경합니다. 이러한 이유로 JDBC 저장소 버전이 410으로 업그레이드됩니다. JDBC 저장소 버전 350, 370 및 400이 410 버전 형식으로 자동으로 마이그레이션됩니다.

파일 기반의 영구 저장소 버전은 변경되지 않고 370으로 유지됩니다.

브로커 구성

imqenv.conf 파일에 IMQ_DEFAULT_EXT_JARS 등록 정보가 추가되었습니다. 이 등록 정보를 설정하여 브로커가 시작될 때 CLASSPATH에 포함될 수 있도록 외부 .jar 파일의 경로 이름을 지정할 수 있습니다. 이 등록 정보를 사용하여 외부 .jar 파일의 위치를 지정하면 이 파일을 더 이상 lib/ext 디렉토리에 복사할 필요가 없습니다. 외부 jar 파일은 JDBC 드라이버 또는 JAAS 로그인 모듈과 관련된 것일 수 있습니다. 다음 예제 명령은 JDBC 드라이버의 위치를 지정합니다.

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

JES Monitoring Framework 지원

Message Queue는 공통 그래픽 인터페이스를 사용하여 Java Enterprise System 구성 요소를 모니터링할 수 있도록 해주는 Sun Java Enterprise System(JES) Monitoring Framework를 지원합니다. 이 인터페이스는 Sun Java System Monitoring Console이라고 하는 웹 기반의 콘솔로 구현됩니다. Message Queue를 다른 JES 구성 요소와 함께 실행 중인 경우에는 단일 인터페이스를 사용하여 이러한 모든 구성 요소를 관리하는 것이 편리할 수 있습니다.

JES Monitoring Framework는 모든 JES 구성 요소 제품에서 사용할 공통 모니터링 모델(CMM)을 정의합니다. 이 모델을 사용하면 모든 JES 구성 요소의 중앙 집중식 동일 보기가 가능해집니다. Message Queue는 JES Monitoring Framework에 다음 객체를 제공합니다.

- 설치된 제품
- 브로커 인스턴스 이름
- 브로커 포트 매핑
- 각 연결 서비스
- 각 물리적 대상
- 영구 저장소
- 사용자 저장소

각 객체는 JES Monitoring Console을 사용하여 속성을 모니터링할 수 있는 CMM 객체에 매핑됩니다. 런타임 시 관리자는 콘솔을 사용하여 성능 통계를 보고, 자동으로 모니터링할 수 있는 규칙을 만들고, 경보를 확인할 수 있습니다. CMM 객체에 Message Queue 객체를 매핑하는 방법에 대한 자세한 내용은 **Sun Java Enterprise System Monitoring Guide**를 참조하십시오.

JES 모니터링을 활성화하려면 다음 작업을 수행해야 합니다.

1. **Sun Java Enterprise System Installation Guide**의 지침에 따라 해당 배포(Message Queue 및 기타 구성 요소)에 모든 구성 요소를 설치 및 구성합니다.
2. **Sun Java Enterprise System Monitoring Guide**의 설명에 따라 모니터링되는 모든 구성 요소에 대해 모니터링 프레임워크를 활성화하고 구성합니다.
3. **Sun Java Enterprise System Monitoring Guide**의 설명에 따라 별도의 호스트에 모니터링 콘솔을 설치하고 마스터 에이전트를 시작한 다음 웹 서버를 시작합니다.

모든 메트릭 수집 작업은 브로커의 기존 모니터링 데이터 인프라로부터 데이터를 가져오는 모니터링 프레임워크에서 수행하기 때문에 JES Monitoring Framework를 사용해도 브로커 성능에 영향을 주지 않습니다.

트랜잭션 관리

이전에는 PREPARED 상태의 트랜잭션만 관리 목적으로 롤백할 수 있었습니다. 즉, 분산 트랜잭션의 일부인 세션이 정상적으로 종료되지 않은 경우 트랜잭션이 브로커 관리자에 의해 정리되지 않은 상태로 유지되었습니다. Message Queue 4.1에서는 `imqcmd` 유틸리티를 사용하여 STARTED, FAILED, INCOMPLETE, COMPLETE 및 PREPARED와 같은 상태의 트랜잭션을 정리(롤백)할 수 있습니다.

특정 트랜잭션을 롤백할 수 있는지 여부(특히 트랜잭션이 PREPARED 상태가 아닐 경우)를 결정할 수 있도록 `imqcmd` 유틸리티가 `imqcmd query txn` 출력의 일부로 추가 데이터를 제공합니다. 이 유틸리티는 트랜잭션을 시작한 연결의 연결 아이디를 제공하며 트랜잭션이 만들어진 시간을 지정합니다. 관리자는 이 정보를 사용하여 트랜잭션을 롤백해야 하는지 여부를 결정할 수 있습니다. 일반적으로 관리자는 트랜잭션을 조기에 롤백해서는 안 됩니다.

C클라이언트 연결을 위한 고정 포트

C 클라이언트는 `MQ_SERVICE_PORT_PROPERTY` 연결 등록 정보를 사용하여 연결할 고정 포트를 지정할 수 있습니다. 이는 포트를 동적으로 할당하는 브로커의 포트 매핑 서비스를 우회해야 하거나 방화벽을 통과하려고 시도할 경우에 유용할 수 있습니다.

이 포트를 지정할 경우 브로커측에 JMS 서비스 포트도 구성해야 함에 유의하십시오. 예를 들어, 클라이언트를 `ssljms`를 통해 포트 1756에 연결할 경우 다음 작업을 수행합니다.

- 클라이언트측: 1756에는 `MQ_SERVICE_PORT_PROPERTY`를, SSL에는 `MQ_CONNECTION_TYPE_PROPERTY`를 설정합니다.
- 브로커측: 다음과 같이 1756에 `imq.serviceNameType.protocol.port` 등록 정보를 설정합니다.

```
imq.ssljms.ssl.port=1756
```

주 - `MQ_SERVICE_PORT_PROPERTY` 연결 등록 정보는 Message Queue 버전 3.7 Update 2에서 도입되었습니다.

하드웨어 및 소프트웨어 요구 사항

버전 4.1에 대한 하드웨어 및 소프트웨어 요구 사항에 대해서는 **Sun Java System Message Queue 4.1 Installation Guide**를 참조하십시오.

Message Queue 4.0 정보

Message Queue 4.0은 Application Server 9 PE를 지원하기 위해 제한되었으며 몇 가지 새로운 기능과 소수 향상된 기능 및 버그 수정 사항이 포함된 부 릴리스입니다. 이 절은 다음 내용으로 구성되어 있습니다.

- 16 페이지 “4.0 릴리스의 새로운 기능”
- 31 페이지 “하드웨어 및 소프트웨어 요구 사항”

4.0 릴리스의 새로운 기능

Message Queue 4.0에는 다음과 같은 새로운 기능이 추가되었습니다.

- 16 페이지 “C API 및 C 클라이언트 런타임의 인터페이스 변경 사항”
- 16 페이지 “Java API 및 Java 클라이언트 런타임의 인터페이스 변경 사항”
- 17 페이지 “영구 저장소에 대한 정보 표시”
- 17 페이지 “영구 저장소 형식 변경 사항”
- 18 페이지 “브로커 관리”
- 19 페이지 “JDBC 지속성 지원”
- 19 페이지 “SSL 지원”
- 19 페이지 “JMX 지원”
- 24 페이지 “클라이언트 런타임 로깅”
- 28 페이지 “연결 이벤트 알림”

이러한 기능에 대해서는 이어지는 절에서 설명합니다.



주의 - 버전 4.0에 도입된, 작지만 잠재적으로 혁신적인 변경 사항 중 하나는 비밀번호를 지정하는 명령줄 옵션이 폐지되었다는 점입니다. 이제부터는 **40 페이지 “더 이상 사용되지 않는 비밀번호 옵션”**의 설명대로 한 파일에 모든 비밀번호를 저장해야 합니다.

C API 및 C 클라이언트 런타임의 인터페이스 변경 사항

Message Queue 버전 4.0에는 사용 불능 메시지 대기열에 있던 모든 메시지에 설정할 수 있는 등록 정보 두 가지가 새롭게 추가되었습니다.

- JMS_SUN_DMQ_PRODUCING_BROKER는 메시지를 생성한 브로커를 나타냅니다.
- JMS_SUN_DMQ_DEAD_BROKER는 메시지를 사용 불능으로 표시한 브로커를 나타냅니다.

Java API 및 Java 클라이언트 런타임의 인터페이스 변경 사항

Message Queue 버전 4.0에는 사용 불능 메시지 대기열에 있던 모든 메시지에 설정할 수 있는 등록 정보 두 가지가 새롭게 추가되었습니다.

- JMS_SUN_DMQ_PRODUCING_BROKER는 메시지를 생성한 브로커를 나타냅니다.
- JMS_SUN_DMQ_DEAD_BROKER는 메시지를 사용 불능으로 표시한 브로커를 나타냅니다.

영구 저장소에 대한 정보 표시

`imqdbmgr` 명령에 `query` 하위 명령이 추가되었습니다. 저장소 버전, 데이터베이스 사용자 및 데이터베이스 테이블이 만들어졌는지의 여부를 포함한 영구 저장소에 대한 정보를 표시하려면 이 하위 명령을 사용합니다.

다음은 이 명령을 사용하여 표시한 정보의 예를 나타냅니다.

```
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.
```

영구 저장소 형식 변경 사항

Message Queue 버전 3.7 UR1에는 성능을 향상시킬 수 있는, 영구 저장소 형식에 대한 두 가지 변경 사항이 도입되었습니다. 하나는 파일 저장소에 대한 변경 사항이며 다른 하나는 JDBC 저장소에 대한 변경 사항입니다.

- 파일 저장소에 저장되는 트랜잭션 데이터의 형식

Message Queue의 파일 기반 영구 저장소에 저장되는 트랜잭션 상태 정보 형식이 변경되어 디스크 입출력이 줄었으며 JMS 트랜잭션의 성능이 향상되었습니다.

- Oracle JDBC 저장소

이전 버전의 Message Queue에서 Oracle용 저장소 스키마는 **LONG RAW** 데이터 유형을 사용하여 메시지 데이터를 저장했습니다. Oracle 8에서는 **BLOB** 데이터 유형을 도입했으며 **LONG RAW** 유형은 더 이상 사용하지 않습니다. Message Queue 3.7 UR1에서는 성능 및 지원 가능성 향상을 위해 **BLOB** 데이터 유형으로 전환했습니다.

이러한 변경 사항은 저장소 호환성에 영향을 주기 때문에 Message Queue 버전 3.7 UR1에서 파일 저장소 및 JDBC 저장소의 버전이 모두 350에서 370으로 변경되었습니다.

Message Queue 버전 4.0에서는 JDBC 저장소를 최적화하고 향후 향상된 기능을 지원할 수 있는 변경 내용이 도입되었습니다. 이러한 이유로 JDBC 저장소 버전이 400으로 업그레이드되었습니다. 버전 4.0에서는 파일 기반의 영구 저장소 버전이 변경되지 않고 370으로 유지됩니다.

Message Queue 4.0은 파일 기반 영구 저장소 및 JDBC 영구 저장소를 최신 버전으로 자동으로 변환할 수 있도록 지원합니다. 처음 `imqbrokerd`가 시작될 때 유틸리티에서 이전 저장소를 감지하면 이전 저장소를 그대로 두고 새 형식의 저장소로 마이그레이션합니다.

- 파일 기반의 저장소 버전 200 및 350은 370 버전 형식으로 마이그레이션됩니다.

- JDBC 저장소 버전 350 및 370은 400 버전 형식으로 마이그레이션됩니다. (200 저장소를 업그레이드해야 하는 경우에는 먼저 중간에 있는 3.5 또는 3.6 릴리스 단계를 밟아야 합니다.)

이 업그레이드를 롤백해야 하는 경우 Message Queue 4.0 설치를 제거한 다음 이전에 실행 중이던 버전을 다시 설치할 수 있습니다. 이전 저장소의 복사본이 그대로 유지되므로 브로커는 이전 저장소 복사본을 실행할 수 있습니다.

브로커 관리

관리자가 브로커를 정지하거나, 지정된 간격 후 브로커를 종료하거나, 연결을 삭제하거나, java 시스템 등록 정보(예: 연결 관련 등록 정보)를 설정할 수 있도록 해주는 하위 명령 및 여러 옵션이 명령 유틸리티(imqcmd)에 받습니다.)

- 브로커를 정지하면 브로커가 자동 상태가 되며 브로커가 종료되거나 다시 시작되기 전에 메시지가 드레인됩니다. 정지된 브로커에는 새 연결을 만들 수 없습니다. 브로커를 정지하려면 다음 명령을 입력하십시오.

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

- 지정된 간격 후 브로커를 종료하려면 다음 명령을 입력하십시오. (시간 간격은 브로커가 종료되기 전에 대기하는 시간(초)을 지정합니다.)

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

시간 간격을 지정하면 브로커에서 종료 시간을 나타내는 메시지를 기록합니다. 예를 들면 다음과 같습니다.

Shutting down the broker in 29 seconds (29996 milliseconds)

브로커 종료 대기 중 해당 동작은 다음과 같은 방식으로 영향을 받습니다.

- 관리 jms 연결은 계속 허용됩니다.
- 새 jms 연결은 허용되지 않습니다.
- 기존 jms 연결은 계속 작동합니다.
- 브로커는 고가용성 클러스터 내의 다른 브로커를 인계 받을 수 없습니다.
- imqcmd 유틸리티는 차단되지 않고 종료 요청을 브로커에 전송하며 즉시 반환됩니다.
- 연결을 삭제하려면 다음 명령을 입력하십시오.

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

연결 아이디를 가져오려면 imqcmd list cxn 또는 imqcmd query cxn 명령을 사용합니다.

- imqcmd를 사용하여 시스템 등록 정보를 설정하려면 새로운 -D 옵션을 사용하십시오. 이 명령은 JMS 연결 팩토리 등록 정보 또는 연결 관련 java 시스템 등록 정보를 설정하거나 대체할 때 유용합니다. 예를 들면 다음과 같습니다.

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

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

`imqcmd` 명령의 구문에 대한 자세한 내용은 **Sun Java System Message Queue 4.1 Administration Guide**의 13 장, “Command Line Reference”를 참조하십시오.

JDBC 지속성 지원

이제 Apache Derby 버전 10.1.1이 JDBC 호환 영구 저장소 공급자로 지원됩니다.

SSL 지원

릴리스 4.0 시작 시 클라이언트 연결 팩토리 등록 정보 `imqSSLIsHostTrusted`의 기본값은 `false`입니다. 응용 프로그램이 이전 기본값 `true`를 따르는 경우에는 등록 정보를 재구성하여 명시적으로 `true`로 설정해야 합니다.

자체 서명된 인증서를 사용하도록 브로커를 구성한 경우 호스트를 신뢰하도록 선택할 수 있습니다. 이 경우 `imqConnectionType` 등록 정보를 사용하여 연결이 SSL 기반 연결 서비스를 사용하도록 지정해야 할 뿐 아니라 `imqSSLIsHostTrusted` 등록 정보를 `true`로 설정해야 합니다.

예를 들어, 브로커가 자체 서명된 인증서를 사용할 때 클라이언트 응용 프로그램을 안전하게 실행하려면 다음 명령을 사용합니다.

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

브로커가 자체 서명된 인증서를 사용할 때 관리 도구 `imqcmd`를 안전하게 실행하려면 다음 명령을 사용합니다.

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

JMX 지원

JMX(Java Management Extensions) 사양에 따라 Message Queue 브로커 구성 및 모니터링을 위한 새로운 API가 추가되었습니다. 이 API를 사용하여 Message Queue 클라이언트 응용 프로그램 내에서 브로커 기능을 프로그래밍 방식으로 구성하고 모니터링할 수 있습니다. 이전 버전의 Message Queue에서는 명령줄이나 관리 콘솔을 통해서만 이 기능에 액세스할 수 있었습니다.

API는 다음 Message Queue 관련 자원을 관리하기 위한 *JMX Managed Beans(MBeans)* 집합으로 구성됩니다.

- 메시지 브로커
- 연결 서비스
- 연결
- 대상
- 메시지 생성자
- 메시지 사용자
- 트랜잭션

- 브로커 클러스터
- 로깅
- JVM(Java Virtual Machine)

이 MBean은 상태 변경 사항이 발생할 때 클라이언트 응용 프로그램이 이 변경 사항을 수신하고 이에 대해 비동기식으로 응답할 수 있도록 해주는 **알림**뿐 아니라 기본 자원의 상태를 동기식으로 폴링하고 조작하는 데 필요한 **속성** 및 **작업**을 제공합니다. 클라이언트 응용 프로그램은 JMX API를 사용하여 다음과 같은 구성 및 모니터링 작업을 수행할 수 있습니다.

- 브로커의 포트 번호 설정
- 브로커의 최대 메시지 크기 설정
- 연결 서비스 일시 중지
- 연결 서비스에 대한 최대 스레드 수 설정
- 서비스에서 현재 연결 수 가져오기
- 연결 삭제
- 대상 만들기
- 대상 삭제
- 대상 자동 만들기 활성화 또는 비활성화
- 대상에서 모든 메시지 제거
- 브로커가 시작된 후 대상에서 수신한 누적 메시지 수 가져오기
- 대기열의 현재 상태(실행 중 또는 일시 중지됨) 가져오기
- 주제에 대한 현재 메시지 생성자 수 가져오기
- 영구 가입자의 모든 메시지 제거
- 현재 JVM 힙 크기 가져오기

JMX API에 대한 소개와 전체 참조 정보에 대해서는 **Sun Java System Message Queue 4.1 Developer's Guide for JMX Clients**를 참조하십시오.

브로커 지원: JMX 관련 등록 정보

JMX API를 지원하기 위한 몇 개의 새로운 브로커 등록 정보가 추가되었습니다([표 1-3](#) 참조). 이 등록 정보는 명령줄에서 Message Queue 명령 유틸리티(imqcmd)를 사용하여 설정할 수 없습니다. 대신, 브로커 유틸리티(imqbrokerd)의 -D 옵션을 사용하여 설정하거나 브로커의 인스턴스 구성 파일(config.properties)에서 직접 편집할 수 있습니다. 또한 이러한 등록 정보의 일부(imq.jmx.rmiregistry.start, imq.jmx.rmiregistry.use, imq.jmx.rmiregistry.port)는 [표 1-4](#)에서 설명하는 새로운 브로커 유틸리티 옵션을 사용하여 설정할 수 있습니다. 다음 표에서는 각 옵션을 나열하고 해당 유형 및 용도를 지정 및 설명합니다.

표 1-3 JMX 지원을 위한 새 브로커 등록 정보

등록 정보	유형	설명
<code>imq.jmx.rmiregistry.start</code>	Boolean	<p>브로커 시작 시 RMI 레지스트리를 시작할지 여부를 지정합니다.</p> <p><code>true</code>로 설정하면 브로커가 <code>imq.jmx.rmiregistry.port</code>에 지정한 포트에서 RMI 레지스트리를 시작하며 이를 사용하여 JMX 커넥터에 RMI 스탬을 저장합니다. 이 경우 <code>imq.jmx.rmiregistry.use</code> 값은 무시됩니다.</p> <p>기본값: <code>false</code></p>
<code>imq.jmx.rmiregistry.use</code>	Boolean	<p>외부 RMI 레지스트리를 사용할지 여부를 지정합니다.</p> <p>이 값은 <code>imq.jmx.rmiregistry.start</code>가 <code>false</code>일 경우에만 적용됩니다.</p> <p><code>true</code>로 설정하면 브로커가 <code>imq.jmx.rmiregistry.port</code>에 지정한 포트에서 외부 RMI 레지스트리를 사용하여 JMX 커넥터에 RMI 스탬을 저장합니다. 브로커 시작 시 외부 RMI 레지스트리가 이미 실행 중이어야 합니다.</p> <p>기본값: <code>false</code></p>
<code>imq.jmx.rmiregistry.port</code>	Integer	<p>RMI 레지스트리의 포트 번호입니다.</p> <p>이 값은 <code>imq.jmx.rmiregistry.start</code> 또는 <code>imq.jmx.rmiregistry.use</code>가 <code>true</code>인 경우에만 적용됩니다. 이렇게 위의 값이 <code>true</code>로 설정되면 이 포트 번호를 JMX 서비스 URL의 URL 경로에 포함시켜 RMI 레지스트리를 사용할 수 있도록 JMX 커넥터를 구성할 수 있습니다.</p> <p>기본값: <code>1099</code></p>
<code>imq.jmx.connector.list</code>	String	<p>사전 구성된 JMX 커넥터 이름입니다(쉼표로 구분).</p> <p>기본값: <code>jmxrmi,ssljmxrmi</code></p>
<code>imq.jmx.connector.activelist</code>	String	<p>브로커 시작 시 활성화할 JMX 커넥터 이름입니다(쉼표로 구분).</p> <p>기본값: <code>jmxrmi</code></p>

표 1-3 JMX 지원을 위한 새 브로커 등록 정보 (계속)

등록 정보	유형	설명
<code>imq.jmx.connector.connectorName.urlpath</code>	String	<p><code>connectorName</code> 커넥터에 대한 JMX 서비스 URL의 <code>urlPath</code> 구성 요소입니다.</p> <p>JMX 서비스 URL 경로를 명시적으로 설정해야 할 경우(예: 공유된 외부 RMI 레지스트리를 사용하는 경우)에 유용합니다.</p> <p>기본값: JMX 커넥터에 RMI 스텝을 저장하는 데 RMI 레지스트리를 사용하는 경우(즉, <code>imq.jmx.registry.start</code> 또는 <code>imq.jmx.registry.use</code>가 <code>true</code>인 경우)에는 다음과 같습니다.</p> <pre>/jndi/rmi://brokerHost:rmiPort /brokerHost/brokerPort/connectorName</pre> <p>RMI 레지스트리를 사용하지 않는 경우(기본적으로 <code>imq.jmx.registry.start</code> 및 <code>imq.jmx.registry.use</code>가 모두 <code>false</code>)에는 다음과 같습니다.</p> <pre>/stub/rmiStub</pre> <p>여기서 <code>rmiStub</code>은 RMI 스텝의 인코딩되고 일련화된 표현입니다.</p>
<code>imq.jmx.connector.connectorName.useSSL</code>	Boolean	<p><code>connectorName</code> 커넥터에 SSL(Secure Socket Layer)을 사용할지 여부를 지정합니다.</p> <p>기본값: <code>false</code></p>
<code>imq.jmx.connector.connectorName.brokerHostTrusted</code>	Boolean	<p><code>connectorName</code> 커넥터에 대해 브로커에서 제공한 인증서를 신뢰할지 여부를 지정합니다.</p> <p>이 값은 <code>imq.jmx.connector.connectorName.useSSL</code>이 <code>true</code>인 경우에만 적용됩니다.</p> <p>이 값이 <code>false</code>인 경우에는 Message Queue 클라이언트 런타임이 제공되는 모든 인증서를 검증합니다. 인증서 서명자가 클라이언트의 트러스트 저장소에 없는 경우 검증이 실패합니다.</p> <p>이 값이 <code>true</code>인 경우에는 인증서 검증이 생략됩니다. 이 설정은, 예를 들어 자체 서명된 인증서를 사용하는 소프트웨어를 테스트할 때 유용할 수 있습니다.</p> <p>기본값: <code>false</code></p>

`imq.jmx.connector.list` 등록 정보는 브로커 시작 시 명명된 JMX 커넥터 집합을 만들도록 정의합니다. `imq.jmx.connector.activelist`는 만들어진 커넥터 집합에서 활성화할 커넥터를 지정합니다. 명명된 각 커넥터에는 다음과 같은 고유 등록 정보 집합이 있습니다.

```
imq.jmx.connector.connectorName .urlpath
imq.jmx.connector.connectorName .useSSL
imq.jmx.connector.connectorName .brokerHostTrusted
```

기본적으로 `jmxrmi`와 `ssljmxrmi`라고 하는 두 가지 JMX 커넥터가 만들어집니다. 첫 번째 커넥터는 SSL 암호화를 사용하지 않도록 구성되며(`imq.jmx.connector.jmxrmi.useSSL = false`), 두 번째 커넥터는 이 암호화를 사용하도록 구성됩니다(`imq.jmx.connector.ssljmxrmi.useSSL = true`). 기본적으로 브로커 시작 시 `jmxrmi` 커넥터만 활성화됩니다. 보안 통신을 위해 `ssljmxrmi` 커넥터를 활성화하는 방법에 대한 내용은 [24 페이지 “JMX 클라이언트에 대한 SSL 지원”](#)을 참조하십시오.

편의상 새 옵션([표 1-4](#))도 RMI 레지스트리에 대한 사용, 시작 및 포트를 제어하기 위해 명령줄 브로커 유틸리티(`imqbrokerd`)에 추가되었습니다. 이러한 옵션의 사용과 효과는 [표 1-3](#)에서 설명하는 것처럼 해당 브로커 등록 정보와 동일합니다. 다음 표에서는 각 옵션을 나열하고 해당 브로커 등록 정보 및 용도를 지정 및 설명합니다.

표 1-4 JMX 지원을 위한 새 브로커 유틸리티 옵션

옵션	해당 브로커 등록 정보	설명
<code>-startRmiRegistry</code>	<code>imq.jmx.rmiregistry.start</code>	브로커 시작 시 RMI 레지스트리를 시작할지 여부를 지정합니다.
<code>-useRmiRegistry</code>	<code>imq.jmx.rmiregistry.use</code>	외부 RMI 레지스트리를 사용할지 여부를 지정합니다.
<code>-rmiRegistryPort</code>	<code>imq.jmx.rmiregistry.port</code>	RMI 레지스트리의 포트 번호입니다.

브로커 시작 시 작성 및 시작되는 JMX 커넥터의 JMX 서비스 URL을 나열하기 위한 새로운 하위 명령([표 1-5](#))이 명령줄 명령 유틸리티(`imqcmd`)에 추가되었습니다. 이 정보는 JMX 커넥터를 가져오는 데 Message Queue 편의 클래스 `AdminConnectionFactory`를 사용하지 않는 JMX 클라이언트에 필요하며, Java 모니터링 및 관리 콘솔(`jconsole`)과 같은 일반 JMX 브라우저를 통해 Message Queue를 관리하거나 모니터링하는 데 사용할 수도 있습니다.

표 1-5 새 명령 유틸리티 하위 명령

하위 명령	설명
<code>list jmx</code>	JMX 커넥터의 JMX 서비스 URL을 나열합니다.

JMX 클라이언트에 대한 SSL 지원

위에 언급한 바와 같이 Message Queue 메시지 브로커는 기본적으로 사전 구성된 JMX 커넥터 `jmxrmi`를 사용하므로 보안 통신이 설정되지 않은 상태로 구성됩니다. 응용 프로그램에 보안 통신을 위한 SSL(Secure Socket Layer)을 사용하려면 대신 사용할 수 있는 보안 JMX 커넥터 `ssljmxrmi`를 활성화해야 합니다. 이 커넥터를 활성화하려면 다음 단계를 수행해야 합니다.

1. **Message Queue 관리 설명서**에 설명된 대로 `ssljms`, `ssladmin` 또는 `cluster` 연결 서비스와 동일한 방법으로 서명된 인증서를 가져오고 설치합니다.
2. 필요 시 트러스트 저장소에 루트 인증 기관 인증서를 설치합니다.
3. 브로커 시작 시 활성화될 수 있도록 다음과 같이 JMX 커넥터 목록에 `ssljmxrmi` 커넥터를 추가합니다.

```
imq.jmx.connector.activelist=jmxrmi,ssljmxrmi
```

4. Message Queue 브로커 유틸리티(`imqbrokerd`)에 비밀번호 파일의 키 저장소 비밀번호를 전달하거나 프롬프트 창의 명령줄에서 이 유틸리티를 입력하여 브로커를 시작합니다.
5. 기본적으로 `ssljmxrmi` 커넥터(또는 다른 SSL 기반 커넥터)는 제공되는 모든 브로커 SSL 인증서를 검증하도록 구성됩니다. 이 검증을 수행하지 않으려면(예를 들어, 소프트웨어 테스트 중 자체 서명된 인증서를 사용할 경우) 브로커 등록 정보 `imq.jmx.connector.ssljmxrmi.brokerHostTrusted`를 `true`로 설정합니다.

클라이언트 측에서는 다음과 같이 `ssljmxrmi`를 기본 커넥터로 지정하는 URL로 관리자 연결 팩토리(`AdminConnectionFactory`)를 구성해야 합니다.

```
AdminConnectionFactory acf = new AdminConnectionFactory();
acf.setProperty(AdminConnectionConfiguration.imqAddress, "mq://myhost:7676/ssljmxrmi");
```

필요 시 시스템 등록 정보 `javax.net.ssl.trustStore` 및 `javax.net.ssl.trustStorePassword`를 사용하여 JMX 클라이언트가 트러스트 저장소를 가리키도록 설정합니다.

클라이언트 런타임 로깅

이 절에서는 Message Queue 4.0의 연결 및 세션 관련 이벤트에 대한 클라이언트 런타임 로깅 지원에 대해 설명합니다.

JDK 1.4 이상 버전에는 `java.util.logging` 라이브러리가 포함되어 있습니다. 이 라이브러리는 응용 프로그램 특정 로깅에 사용할 수 있는 표준 로거 인터페이스를 구현합니다.

Message Queue 클라이언트 런타임은 Java 로깅 API를 사용하여 로깅 기능을 구현합니다. 모든 J2SE 1.4 로깅 기능을 사용하여 로깅 작업을 구성할 수 있습니다. 예를 들어, 응용 프로그램은 다음과 같은 Java 로깅 기능을 사용하여 Message Queue 클라이언트 런타임이 해당 로깅 정보를 출력하는 방법을 구성할 수 있습니다.

- 로깅 처리기
- 로깅 필터
- 로깅 포맷터
- 로깅 수준

Java 로깅 API에 대한 자세한 내용은

<http://java.sun.com/j2se/1.4.2/docs/guide/util/logging/overview.html>의 Java Logging Overview를 참조하십시오.

로깅 이름 공간, 수준 및 작업

Message Queue 공급자는 로깅 구성을 적절하게 설정한 경우 Message Queue 클라이언트가 연결 및 세션 이벤트를 기록할 수 있도록 해주는 로깅 수준 및 로깅 작업과 연관된 로깅 이름 공간 집합을 정의합니다.

Message Queue 클라이언트 런타임의 루트 로깅 이름 공간은 `javax.jms`로 정의됩니다. Message Queue 클라이언트 런타임의 모든 로거에서 이 이름을 부모 이름 공간으로 사용합니다.

Message Queue 클라이언트 런타임에 사용되는 로깅 수준은 `java.util.logging.Level` 클래스에 정의된 로깅 수준과 동일합니다. 이 클래스는 7개의 표준 로그 수준과 로깅 기능을 활성화 및 비활성화하는 데 사용할 수 있는 2가지 추가 설정을 정의합니다.

OFF 로깅을 비활성화합니다.

SEVERE 최고 우선 순위로 가장 높은 값입니다. 응용 프로그램에서 정의됩니다.

WARNING 응용 프로그램에서 정의됩니다.

INFO 응용 프로그램에서 정의됩니다.

CONFIG 응용 프로그램에서 정의됩니다.

FINE 응용 프로그램에서 정의됩니다.

FINER 응용 프로그램에서 정의됩니다.

FINEST 최저 우선 순위로 가장 낮은 값입니다. 응용 프로그램에서 정의됩니다.

ALL 모든 메시지 로깅을 활성화합니다.

일반적으로 Message Queue 클라이언트 런타임에서 발생한 예외 및 오류는 이름 공간이 `javax.jms`인 로거에서 기록합니다.

- JVM에서 발생하여 클라이언트 런타임이 수신하는 예외(예: `IOException`)는 로깅 이름 공간이 `javax.jms`인 로거가 **WARNING** 수준에서 기록합니다.
- 클라이언트 런타임에서 발생하는 JMS 예외(예: `IllegalStateException`)는 로깅 이름 공간이 `javax.jms`인 로거가 **FINER** 수준에서 기록합니다.
- JVM에서 발생하여 클라이언트 런타임이 수신하는 오류(예: `OutOfMemoryError`)는 로깅 이름 공간이 `javax.jms`인 로거가 **SEVERE** 수준에서 기록합니다.

다음에 나오는 표에는 JMS 연결 및 세션에 대한 이벤트를 기록하기 위해 설정해야 하는 로그 수준과 기록할 수 있는 이벤트가 나열되어 있습니다.

다음 표에서는 연결에 대한 로그 수준과 이벤트에 대해 설명합니다.

표 1-6 javax.jms.connection 이름 공간에 대한 로그 수준 및 이벤트

로그 수준	이벤트
FINE	연결 생성
FINE	연결 시작
FINE	연결 닫힘
FINE	연결 끊어짐
FINE	다시 연결됨
FINER	setClientID와 같은 기타 연결 작업
FINEST	메시지, 확인, Message Queue 작업 및 제어 메시지(예: 트랜잭션 완결)

세션의 경우에는 다음 정보가 로그 레코드에 기록됩니다.

- 사용자에게 전송된 메시지에 대한 각 로그 레코드에는 ConnectionID, SessionID 및 ConsumerID가 포함됩니다.
- 생성자에서 전송한 메시지에 대한 각 로그 레코드에는 ConnectionID, SessionID, ProducerID 및 대상 이름이 포함됩니다.

다음 표에서는 세션에 대한 로그 수준과 이벤트에 대해 설명합니다.

표 1-7 javax.jms.session 이름 공간에 대한 로그 수준 및 이벤트

로그 수준	이벤트
FINE	세션 생성
FINE	세션 닫힘
FINE	생성자 생성
FINE	사용자 생성
FINE	대상 생성
FINER	세션 완결과 같은 기타 세션 작업
FINEST	메시지 생성 및 사용(메시지 등록 정보 및 본문은 로그 레코드에 기록되지 않습니다.)

기본적으로 출력 로그 수준은 응용 프로그램이 실행 중인 JRE에서 상속됩니다. 해당 수준을 확인하려면 `JRE_DIRECTORY/lib/logging.properties` 파일을 확인하십시오.

프로그래밍 방식이나 구성 파일을 사용하여 로깅을 구성할 수 있고 로깅의 범위를 제어할 수 있습니다. 다음 절에서는 이러한 작업에 대해 설명합니다.

JRE 로깅 구성 파일 사용

다음 예는 Java 런타임 환경의 로그 수준을 설정하는 데 사용되는 `JRE_DIRECTORY/lib/logging.properties` 파일에 로깅 이름 공간 및 수준을 설정하는 방법을 나타냅니다. 이 JRE를 사용하는 모든 응용 프로그램에는 동일한 로깅 구성이 있습니다. 다음의 예제 구성은 `javax.jms.connection` 이름 공간에 대해 로깅 수준을 `INFO`로 설정하며 출력을 `java.util.logging.ConsoleHandler`에 기록할 것을 지정합니다.

```
#logging.properties file.
# "handlers" specifies a comma separated list of log Handler
# classes. These handlers will be installed during VM startup.
# Note that these classes must be on the system classpath.
# By default we only configure a ConsoleHandler, which will only
# show messages at the INFO and above levels.

handlers= java.util.logging.ConsoleHandler

# Default global logging level.
# This specifies which kinds of events are logged across
# all loggers. For any given facility this global level
# can be overridden by a facility-specific level.
# Note that the ConsoleHandler also has a separate level
# setting to limit messages printed to the console.

.level= INFO

# Limit the messages that are printed on the console to INFO and above.

java.util.logging.ConsoleHandler.level = INFO
java.util.logging.ConsoleHandler.formatter =
    java.util.logging.SimpleFormatter

# The logger with javax.jms.connection name space will write
# Level.INFO messages to its output handler(s). In this configuration
# the output handler is set to java.util.logging.ConsoleHandler.

javax.jms.connection.level = INFO
```

특정 응용 프로그램에 대한 로깅 구성 파일 사용

응용 프로그램을 실행하는 데 사용하는 java 명령줄에서 로깅 구성 파일을 정의할 수도 있습니다. 응용 프로그램은 지정된 로깅 파일에 정의된 구성을 사용합니다. 다음 예에서 configFile은 JRE_DIRECTORY/lib/logging.properties 파일에 정의된 형식과 동일한 형식을 사용합니다.

```
java -Djava.util.logging.config.file=configFile MQApplication
```

프로그래밍 방식으로 로깅 구성 설정

java.util.logging API를 사용하는 다음 코드는 javax.jms.connection 이름 공간 로그 수준을 FINE으로 변경하여 연결 이벤트를 기록합니다. 이러한 코드를 응용 프로그램에 포함시킴으로써 프로그래밍 방식으로 로깅 구성을 설정할 수 있습니다.

```
import java.util.logging.*;
//construct a file handler and output to the mq.log file
//in the system's temp directory.

    Handler fh = new FileHandler("%t/mq.log");
    fh.setLevel (Level.FINE);

//Get Logger for "javax.jms.connection" domain.

    Logger logger = Logger.getLogger("javax.jms.connection");
    logger.addHandler (fh);

//javax.jms.connection logger would log activities
//with level FINE and above.

    logger.setLevel (Level.FINE);
```

연결 이벤트 알림

Message Queue 클라이언트는 연결 이벤트 알림을 통해 닫힘 및 다시 연결 이벤트를 수신하고 알림 유형 및 연결 상태를 기반으로 적절한 작업을 수행할 수 있습니다. 예를 들어, 페일오버가 발생하고 클라이언트가 다른 브로커에 다시 연결된 경우 응용 프로그램은 해당 트랜잭션 상태를 정리하고 새 트랜잭션으로 작업을 계속하려 할 수 있습니다.

Message Queue 공급자가 연결에서 심각한 문제를 감지한 경우에는 연결 객체의 등록된 예외 수신기를 호출한 후 이 수신기의 onException 메소드를 호출하고, 문제를 설명하는 JMSEException 인수를 이 메소드에 전달합니다. 또한 Message Queue 공급자는 클라이언트 런타임이 연결 상태 변경 사항을 응용 프로그램에 알릴 수 있도록 해주는 이벤트 알림 API를 제공하기도 합니다. 알림 API는 다음 요소로 정의됩니다.

- 이벤트 수신기 및 알림 이벤트 객체를 정의하는 com.sun.messaging.jms.notification 패키지

- `javax.jms.Connection` 인터페이스에 대한 확장을 정의하는 `com.sun.messaging.jms.Connection` 인터페이스

다음 절에서는 알림을 트리거할 수 있는 이벤트에 대한 정보와 이벤트 수신기를 만들 수 있는 방법에 대해 설명합니다.

연결 이벤트

다음 표에서는 이벤트 수신기에서 반환할 수 있는 이벤트를 나열하고 설명합니다.

연결 이벤트가 발생한 경우에는 JMS 예외 수신기가 호출되지 않습니다. 이 예외 수신기는 클라이언트 런타임이 다시 연결 시도를 모두 소진한 경우에만 호출됩니다. 클라이언트 런타임은 항상 예외 수신기보다 이벤트 수신기를 먼저 호출합니다.

표 1-8 알림 이벤트

이벤트 유형	의미
<code>ConnectionClosingEvent</code>	<code>Message Queue</code> 클라이언트 런타임이 관리자의 종료 요청으로 인해 연결이 닫히려 하고 하는 브로커에서 알림을 수신한 경우 이 이벤트를 생성합니다.
<code>ConnectionClosedEvent</code>	브로커 오류나 관리자의 종료 또는 다시 시작 요청으로 인해 연결이 닫힌 경우 <code>Message Queue</code> 클라이언트 런타임이 이 이벤트를 생성합니다. 이벤트 수신기가 <code>ConnectionClosedEvent</code> 를 수신한 경우 응용 프로그램은 수신된 이벤트의 <code>getEventCode()</code> 메소드를 사용하여 연결이 닫힌 원인을 지정하는 이벤트 코드를 가져올 수 있습니다.
<code>ConnectionReconnectedEvent</code>	<code>Message Queue</code> 클라이언트 런타임이 브로커에 다시 연결되었음을 나타냅니다. 이는 이전에 클라이언트에 연결되었던 동일한 브로커이거나 다른 브로커일 수 있습니다. 응용 프로그램은 수신된 이벤트의 <code>getBrokerAddress</code> 메소드를 사용하여 다시 연결된 브로커의 주소를 가져올 수 있습니다.

표 1-8 알림 이벤트 (계속)

이벤트 유형	의미
ConnectionReconnectFailedEvent	<p>Message Queue 클라이언트 런타임을 브로커에 다시 연결하지 못했음을 나타냅니다. 다시 연결 시도에 실패할 때마다 런타임은 새 이벤트를 생성하고 이를 이벤트 수신기에 전송합니다.</p> <p>연결 이벤트가 발생한 경우에는 JMS 예외 수신기가 호출되지 않습니다. 이 예외 수신기는 클라이언트 런타임이 다시 연결 시도를 모두 소진한 경우에만 호출됩니다. 클라이언트 런타임은 항상 예외 수신기보다 이벤트 수신기를 먼저 호출합니다.</p>

이벤트 수신기 만들기

다음 코드 예는 연결 이벤트 수신기를 설정하는 방법을 나타냅니다. 연결 이벤트가 발생할 때마다 클라이언트 런타임에 의해 이벤트 수신기의 `onEvent` 메소드가 호출됩니다.

```
//create an MQ connection factory.

com.sun.messaging.ConnectionFactory factory =
    new com.sun.messaging.ConnectionFactory();

//create an MQ connection.

com.sun.messaging.jms.Connection connection =
    (com.sun.messaging.jms.Connection )factory.createConnection();

//construct an MQ event listener. The listener implements
//com.sun.messaging.jms.notification.EventListener interface.

com.sun.messaging.jms.notification.EventListener eListener =
    new ApplicationEventListener();

//set event listener to the MQ connection.

connection.setEventListener ( eListener );
```

이벤트 수신기 예

이 예에서 응용 프로그램은 해당 이벤트 수신기가 연결 이벤트를 응용 프로그램의 로깅 시스템에 기록하도록 선택합니다.

```
public class ApplicationEventListener implements
    com.sun.messaging.jms.notification.EventListener {
```

```

public void onEvent ( com.sun.messaging.jms.notification.Event connEvent ) {
    log (connEvent);
}
private void log ( com.sun.messaging.jms.notification.Event connEvent ) {
    String eventCode = connEvent.getEventCode();
    String eventMessage = connEvent.getEventMessage();
    //write event information to the output stream.
}
}

```

하드웨어 및 소프트웨어 요구 사항

버전 4.0에 대한 하드웨어 및 소프트웨어 요구 사항은 Sun Java System Application Server Platform Edition 9의 릴리스 노트를 참조하십시오.

이 릴리스에서 해결된 버그

다음 표는 Message Queue 4.1 버전에서 해결된 버그를 설명합니다.

표 1-9 Message Queue 4.1에서 해결된 버그

버그	설명
6381703	메시지를 만든 브로커가 다시 시작되면 트랜잭션된 원격 메시지가 두 번 완결될 수 있음
6388049	완료되지 않은 분산 트랜잭션을 정리할 수 없음
6401169	imqcmd에 대한 완결 및 롤백 옵션이 확인을 요청하는 메시지를 표시하지 않음
6473052	자동 생성된 대기열의 기본값은 라운드 로빈되어야 함(MaxNumberConsumers = -1)
6474990	imqcmd list dst 명령에 대해 브로커 로그가 ConcurrentModificationException을 표시함
6487413	제한 동작이 REMOVE_OLDEST 또는 REMOVE_LOWER_PRIORITY일 경우 메모리 누출이 발생함
6488340	브로커가 회전하고 클라이언트가 응답 확인을 기다림
6502744	브로커가 사용 불능 메시지 대기열의 기본 제한인 1000개 메시지를 따르지 않음
6517341	imqReconnectEnabled 등록 정보 값에 관계 없이 클라이언트가 다시 연결될 수 있도록 하여 클라이언트를고가용성 클러스터에 연결할 경우 클라이언트 런타임의 다시 연결 논리가 향상되어야 함
6528736	시작 도중 Windows 자동 시작 서비스(imqbrokersvc)에 충돌이 발생함
6561494	메시지와 사용자가 한 세션을 공유한 경우 메시지가 잘못된 사용자에게 전송됨

표 1-9 Message Queue 4.1에서 해결된 버그 (계속)

버그	설명
6567439	PREPARED 트랜잭션에서 생성된 메시지가 브로커 다시 시작 후에 완결되면 메시지가 잘못된 순서로 전송됨

다음 표는 Message Queue 4.0에서 해결된 버그를 설명합니다.

표 1-10 Message Queue 4.0에서 해결된 버그

버그 번호	설명
4986481	Message Queue 3.5의 경우 자동 다시 연결 모드에서 <code>Session.recover</code> 호출이 정지될 수 있었음
4987325	<code>Session.recover</code> 호출 후 재전송된 플래그가 재전송된 메시지에 대해 <code>false</code> 로 설정되었음
6157073	연결 총 수뿐 아니라 서비스의 연결 수를 포함시키기 위해 새 연결 메시지를 변경함
6193884	메시지에 ASCII가 아닌 문자를 사용하는 로케일의 경우 Message Queue는 알아 볼 수 없는 메시지를 syslog로 출력함
6196233	JMSMessageID를 사용하여 메시지를 선택할 수 없음
6251450	클러스터 종료 도중 <code>connectList</code> 의 <code>ConcurrentModificationException</code>
6252763	<code>java.nio.HeapByteBuffer.putLong/Int</code> 의 <code>java.nio.BufferOverflowException</code>
6260076	Oracle 저장소에서 시작 후 첫 번째로 게시되는 메시지가 느림
6260814	JMSXUserID에서의 선택기 처리가 항상 <code>false</code> 로 처리됨
6264003	대기열 브라우저가 완결되지 않은 트랜잭션의 일부인 메시지를 표시함
6271876	사용되지 않은 메시지를 가진 사용자를 종료할 때 연결 흐름 제어가 제대로 작동하지 않음
6279833	Message Queue에서 두 개의 브로커가 동일한 jdbc 테이블을 사용해서는 안 됨
6293053	마스터 브로커는 시스템의 IOP 주소가 변경된 경우 저장소를 지우기 전에는(<code>-reset store</code> 를 사용하여) 올바르게 시작되지 않습니다.반습니다.)
6294767	Message Queue 브로커가 여는 네트워크 소켓에 <code>SO_REUSEADDR</code> 을 설정해야 함
6304949	ClientID 등록 정보를 TopicConnectionFactory에 설정할 수 없음
6307056	txn 로그에 성능 병목 현상 발생
6320138	Message Queue C API에 회신 헤더로부터 대기열의 이름을 확인하는 기능이 없음
6320325	JDK 1.4와 JDK 1.5가 모두 설치된 경우 브로커가 JDK 1.5보다 JDK 1.4를 먼저 선택하는 경우가 있음

표 1-10 Message Queue 4.0에서 해결된 버그 (계속)

버그 번호	설명
6321117	멀티 브로커 클러스터 초기화 시 <code>java.lang.NullPointerException</code> 발생
6330053	가입자의 트랜잭션을 완결할 때 <code>jms</code> 클라이언트에서 <code>java.lang.NoClassDefFoundError</code> 가 발생함
6340250	C-API에서 MESSAGE 유형 지원
6351293	Apache Derby 데이터베이스 지원 추가

중요 정보

핵심 제품 설명서에 들어 있지 않은 최신 정보에 대한 내용은 다음 절을 참조하십시오.

- [33 페이지 “설치 정보”](#)
- [33 페이지 “호환성 문제”](#)
- [34 페이지 “Message Queue 4.1 설명서 업데이트”](#)

설치 정보

사전 설치 지침, 업그레이드 절차에 대한 정보와 Solaris, Linux, Windows 플랫폼에 Message Queue, Platform Edition을 설치하는 것과 관련된 기타 모든 정보는 **Sun Java System Message Queue 4.1 Installation Guide**를 참조하십시오.

Solaris, Linux 및 HP/UX 플랫폼에 Message Queue Enterprise Edition을 설치하는 작업과 관련된 사전 설치 지침 및 기타 모든 정보에 대한 자세한 내용은 **Sun Java Enterprise System Installation Guide**를 참조하십시오.

Solaris, Linux, HP/UX 및 Windows 플랫폼에서 Message Queue Enterprise Edition으로 업그레이드하는 작업과 관련된 업그레이드 및 마이그레이션 지침에 대한 자세한 내용은 **Sun Java Enterprise System Upgrade and Migration Guide**를 참조하십시오.

호환성 문제

이 절에서는 Message Queue 4.1의 호환성 문제를 설명합니다.

인터페이스 안정성

Sun Java System Message Queue는 시간이 흐름에 따라 변경될 수 있는 많은 인터페이스를 사용합니다. **Sun Java System Message Queue 4.1 Administration Guide**의 부록 B, “Stability of Message Queue Interfaces”에서는 인터페이스를 안정성에 따라 분류합니다. 인터페이스가 안정적일수록 제품의 향후 버전에서 변경될 가능성이 적어집니다.

Message Queue의 다음 주요 릴리스와 관련된 문제

Message Queue의 다음 주요 릴리스에서는 클라이언트가 해당 릴리스와 호환되지 않을 수 있습니다. 이러한 변경 사항에 대비할 수 있도록 이 정보를 미리 제공합니다.

- Sun Java System Message Queue의 일부로 설치된 개별 파일의 위치가 변경될 수 있습니다. 따라서 특정 Message Queue 파일의 현재 위치에 의존하는 기존 응용 프로그램이 작동하지 않을 수 있습니다.
- 최신 브로커가 있는 클러스터에서 3.5 이전 브로커가 더 이상 작동하지 않을 수 있습니다.
- Message Queue의 향후 릴리스에서 클라이언트가 1.5 이전 버전의 JDK를 사용하지 못할 수 있습니다.

Message Queue 4.1 설명서 업데이트

Message Queue 4.1에서는 본 릴리스 노트 문서 외에 **Sun Java System Message Queue 4.1 Developer's Guide for JMX Clients** 문서만 **Sun Java System Message Queue 4.1 Developer's Guide for JMX Clients**. 이 문서는 Message Queue 4.0 릴리스에서 소개되었습니다. 4.1 버전에서는 JMX 모델을 소개하는 개념 정보가 추가되었습니다.

Message Queue 3.6 SP3, 2005Q4에서 제공된 Message Queue 설명서는 Application Server 9 PE 클라이언트의 필요에 따라 최신 상태로 유지됩니다. 이 설명서 모음은 다음 위치에서 사용할 수 있습니다.

<http://docs.sun.com/app/docs/coll/1307.1>

설치 및 업그레이드 정보

Sun Java System Message Queue 4.1 Installation Guide가 플랫폼별 정보를 반영하기 위해 업데이트되었습니다. 이 문서에 Message Queue 4.1과 관련된 설치 및 업그레이드 정보가 추가되었습니다.

관리 설명서

고가용성 클러스터, JAAS 지원 및 JMX 지원에 대한 정보를 제공할 수 있도록 **Administration Guide**가 업데이트되었습니다.

Java 클라이언트용 개발 안내서

클라이언트 런타임 로깅 지원 및 연결 이벤트 알림과 관련된 추가 사항을 반영할 수 있도록 **Developer's Guide for Java Clients**가 업데이트되었습니다.

C 클라이언트용 개발 안내서

MQGetDestinationName 기능, MQ_Message 메시지 유형 및 고정 포트와 관련된 추가 사항을 반영할 수 있도록 **Developer's Guide for C Clients**가 업데이트되었습니다.

알려진 문제점 및 제한 사항

이 절에는 Message Queue 4.1의 알려진 문제점이 나열되어 있습니다. 적용되는 제품 영역은 다음과 같습니다.

- 35 페이지 “설치 문제”
- 40 페이지 “더 이상 사용되지 않는 비밀번호 옵션”
- 41 페이지 “일반적인 문제”
- 41 페이지 “관리/구성 문제”
- 42 페이지 “브로커 문제”
- 42 페이지 “브로커 클러스터”
- 44 페이지 “JMX 문제”
- 45 페이지 “SOAP 지원”

현재 버그와 상태, 해결 방법에 대한 목록을 보려면 Java Developer Connection™ 회원은 Java Developer Connection 웹 사이트의 Bug Parade 페이지를 확인해야 합니다. 새로운 버그를 보고하기 전에 이 페이지를 확인하십시오. 여기에 모든 Message Queue 버그가 나열되어 있지는 않지만 문제가 보고된 적이 있는지 여부를 확인할 수 있습니다.

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

주 - Java Developer Connection 구성원의 경우 무료이지만 등록해야 합니다. Java Developer Connection 구성원이 되는 방법에 대한 자세한 내용은 Sun의 "For Developers" 웹 페이지를 참조하십시오.

새 버그를 보고하거나 기능 요청서를 제출하려면 imq-feedback@sun.com으로 메일을 보내십시오.

설치 문제

이 절에서는 Message Queue 버전 4.1 설치와 관련된 문제에 대해 설명합니다.

제품 레지스트리 및 JES

Message Queue 버전 4.1은 Message Queue에 필요한 공유 구성 요소(예: JDK, NSS 라이브러리, JavaHelp 등)를 설치 및 업그레이드할 때에도 사용할 수 있는 새 설치 프로그램으로 설치합니다. 이 설치 프로그램은 Java Enterprise System(JES) 설치 프로그램과 제품 레지스트리를 공유하지 않습니다. JES에 설치된 Message Queue 버전이 제거되고 Message Queue 설치 프로그램을 통해 Message Queue 4.1로 업그레이드된 경우에는 JES 제품 레지스트리가 불일치 상태가 될 수 있습니다. 따라서, JES 제거 프로그램이 실행되면 Message Queue 4.1 및 여기에서 사용하는 공유 구성 요소(JES가 설치하지 않은 구성 요소)가 부주의로 제거될 수 있습니다.

JES 설치 프로그램에서 설치한 소프트웨어를 업그레이드하는 가장 좋은 방법은 다음과 같습니다.

1. JES 제거 프로그램을 사용하여 Message Queue와 해당 공유 구성 요소를 제거합니다.
2. Message Queue 설치 프로그램을 사용하여 Message Queue 4.1을 설치합니다.

적합한 JRE 선택

Message Queue 4.1 설치 프로그램 JDK 선택 화면을 사용하여 Message Queue에서 사용할 용도로 시스템에 있는 기존 JDK/JRE를 선택할 수 있습니다. 하지만, 표시되는 목록에는 설치 응용 프로그램을 실행할 때 사용되는 JRE도 포함되어 있습니다. 이 JRE는 설치 프로그램 번들의 일부이며 실제로 시스템에 설치되지 않습니다. (버그 6585911)

설치 프로그램에서 사용하는 JRE는 경로로 인식되며, 이 경로는 압축 해제된 설치 프로그램 디렉토리 안에 있어야 하며 하위 디렉토리 mq4_1-installer가 포함되어 있어야 합니다. 예를 들면 다음과 같습니다.

```
some_directory/mq4_1-installer/usr/jdk/instances/jdk1.5.0/jre
```

Message Queue에서 사용할 용도로 이 JRE를 선택하지 마십시오. 대신 시스템상의 다른 JDK를 선택하십시오. 시스템상에 JDK가 존재하지 않으면 다음과 같이 해당 플랫폼에서 적절한 조치를 취하십시오.

- Solaris 또는 Linux: "기본 JDK 설치 및 사용"을 선택합니다.
- Windows: Message Queue 4.1 설치 프로그램을 실행하기 전에 JDK를 다운로드하여 설치합니다.

Windows 설치

Message Queue를 Windows에 설치할 경우에는 다음 제한 사항에 유의하십시오.

- 설치 프로그램을 실행해도 Message Queue에 대한 항목이 시작 > 프로그램 메뉴에 추가되지 않습니다(버그 6567258). 관리 콘솔을 시작하려면 **Sun Java System Message Queue 4.1 Administration Guide**의 "Starting the Administration Console"에 보이는 명령줄을 사용하십시오.
- 설치 프로그램을 실행해도 PATH 환경 변수에 IMQ_HOME\mq\bin 디렉토리가 추가되지 않습니다(버그 6567197). PATH 환경 변수에 이 항목을 직접 추가하거나 Message Queue 유틸리티를 호출할 때 전체 경로 이름(IMQ_HOME\mq\bin\command)을 입력해야 합니다.
- 설치 프로그램을 실행해도 Message Queue가 설치되었음을 나타내는 항목이 Windows 레지스트리에 추가되지 않습니다.
- 설치 프로그램을 자동 모드로 실행하면 프로그램이 즉시 반환됩니다. 설치가 수행되지만 사용자는 자동 설치가 실제로 완료된 시점을 알 수 없습니다.(버그 6586560)
- 텍스트 모드(installer -t)가 Windows에서 지원되지 않습니다. Windows에서 설치 프로그램을 텍스트 모드로 실행하면 오류 메시지가 표시됩니다. 설치 프로그램을 영어 이외의 로케일로 실행한 경우에도 이 메시지는 영어로 표시됩니다. (버그 6594142)

- 설치 프로그램을 영어 이외의 로캘로 실행한 경우에도 설치 프로그램 설치 홈 화면에 표시되어야 할 “설치 홈” 문자열이 영어로 표시됩니다. (버그 6592491)

Solaris 설치

사용자가 `installer-n` 명령을 사용하여 설치를 시도하면 오류 메시지와 "불완전" 요약 상태가 잘못 표시됩니다. 실제로는 명령에 성공합니다. (버그 6594351)

Linux 설치

Linux 플랫폼에서는 다음 문제가 설치 작업에 영향을 줍니다.

- JDK 선택 패널에서 스크롤 목록에 한 가지 항목만 표시됩니다. 이로 인해 목록에서 다른 JDK를 선택하기 어렵습니다. (버그 6584735)
- JDK가 이미 설치되어 있는 경우 사용자가 JDK 선택 화면에서 "기본 JDK 설치"를 선택하면 설치 프로그램이 JDK 설치를 계속 시도하며 패키지를 설치할 수 없다고 보고합니다. 이 문제에도 불구하고 설치는 성공적으로 완료됩니다. (버그 6581310)
- 설치 프로그램이 드라이 런 모드에서 실행되는 경우(`installer-n`), 요약 화면에 오류 메시지가 표시되고 설치 상태가 "불완전"으로 표시됩니다. 이는 잘못된 정보입니다. 드라이 런에서는 시스템에 어떠한 것도 설치되지 않습니다. 다음 설치에 사용할 수 있는 응답 파일만 만들어집니다. (버그 6594351)
- 시스템에 이전 버전의 Message Queue 현지화 RPM이 있으면 Message Queue 4.1 현지화 RPM 설치(다국어 패키지 화면에서 "Message Queue 다국어 패키지 설치" 확인란 선택)에 실패합니다. 이전 3.7 UR1 설치의 II8 패키지와의 충돌로 인해 설치에 실패합니다. (버그 6594381)

해결 방법 4.1 설치 프로그램을 실행하기 전에 `rpm -e` 명령을 사용하여 현지화 RPM을 제거하십시오. 관련된 RPM이 무엇인지 확인하려면 **Sun Java System Message Queue 4.1 Installation Guide**의 “Message Queue Packages (RPMs)”를 참조하십시오.

모든 플랫폼 설치

여기서 설명하는 문제는 모든 플랫폼 설치에 적용됩니다.

- 설치 프로그램이 Message Queue 4.1을 설치하는 도중 진행률 화면이 표시될 때 취소 버튼이 활성화됩니다. 이때 취소 버튼을 선택하면 설치가 완료되지 않거나 손상됩니다. (버그 6595578)
- 설치 프로그램 요약 화면에 클릭 시 로그 또는 요약 페이지 뷰어를 실행하는 많은 링크가 포함됩니다. "닫기" 레이블로 표시된 버튼 대신 창 닫기 버튼 "X"를 사용하여 이 뷰어 창을 닫으면 이 뷰어 창의 백업을 가져올 수 없게 됩니다. (버그 6587138)

해결 방법: 닫기 레이블로 표시된 버튼을 사용하여 창을 닫습니다.

- 시스템에 이전 버전의 Message Queue와 NSS/NSPR이 있는 경우 설치 프로그램의 업그레이드에 업그레이드가 필요한 대상으로 Message Queue만 표시되며 NSS/NSPR의 업그레이드 필요성은 언급되지 않습니다. 올바른 정보를 제공하는 설치

준비 완료 화면에 나타나는 바대로 관련 소프트웨어가 설치 프로세스의 일부로 모두 업그레이드되므로 이 문제는 업데이트 화면에만 국한되는 문제입니다. (버그 6580696)

해결 방법: NSS/NSPR 파일이 설치되어 있지 않다면 이 파일이 설치될 것이고 이전 버전은 제거될 것이므로 해결 방법이 필요 없습니다.

- 설치 프로그램 또는 제거 프로그램이 텍스트 모드에서 실행될 때(`installer-t`), 요약 화면에 로그/요약 파일이 포함된 디렉토리가 표시되지만 이 파일의 이름은 나열되지 않습니다. (버그 6581592)
- 존재하지 않는 파일 이름을 지정하면 일관성 없고 모호한 오류 메시지가 생성됩니다. (버그 6587127)

버전 정보

설치 프로그램에서 불분명한 형식으로 Message Queue 버전 정보를 표시합니다. (버그 6586507)

Solaris 플랫폼에서 설치된 버전을 확인하려면 아래 표를 참조하십시오.

표 1-11 버전 형식

설치 프로그램에 표시되는 버전	Message Queue 릴리스
4.1.0.0	4.1
3.7.0.1	3.7 UR1
3.7.0.2	3.7 UR2
3.7.0.3	3.7 UR3
3.6.0.0	3.6
3.6.0.1	3.6 SP1
3.6.0.2	3.6 SP2
3.6.0.3	3.6 SP3
3.6.0.4	3.6 SP4

주 - 3.6 SP4에 대한 패치 릴리스(예: 3.6 SP4 패치 1)의 경우 설치 프로그램에 표시되는 릴리스 문자열이 동일합니다. 정확한 버전을 확인하려면 `imqbrokerd -version`을 실행해야 합니다.

Linux 플랫폼에서는 단순 형식 변환을 제공할 수 없습니다. Linux에서 설치 프로그램에 표시되는 버전 번호의 형식은 다음과 같습니다.

<majorReleaseNumber>.<minorReleaseNumber>-<someNumber>

예: 3.7-22. 이는 3.7 릴리스 중 하나를 나타내지만 특정 릴리스를 가리키지는 않습니다. 이를 확인하려면 `imqbrokerd -version` 명령을 실행하십시오.

현지화 문제

다음 문제는 현지화 문제와 관련된 내용입니다.

- 설치 프로그램이 텍스트 모드(`installer-t`), 비영어 로케에서 실행될 때 다중 바이트 문자가 알아볼 수 없게 표시됩니다.(버그 6586923)
- 설치 프로그램 요약 화면을 통해 요약 보고서를 볼 수 있습니다. 하지만, 설치 프로그램을 멀티바이트 로케로 실행하면 이 보고서(HTML 페이지)에 알 수 없는 문자가 표시됩니다.(버그 6587112)

해결 방법: HTML 파일을 편집하여 이 파일에 지정된 문자 집합을 수정합니다. HTML 파일에는 다음과 같은 내용이 포함되어야 합니다.

```
meta http-equiv= Content-Type content= text/html; charset=UTF-8
```

“UTF-8”을 `locale_name.UTF-8`로 바꿉니다. 예를 들어, Solaris의 경우 `ja_JA.UTF-8` 또는 `ko.UTF-8`로, Linux의 경우 `ja_JA.utf8` 또는 `ko_KO.utf8`로 바꿉니다.

- 설치 프로그램 진행률 화면에서 진행률 표시줄에 알 수 없는 문자가 표시됩니다. 도구 설명이 영어 이외의 로케로 하드 코드되어 있습니다.(버그 6591632)
- 텍스트 모드(`installer-t`)가 Windows에서 지원되지 않습니다. Windows에서 설치 프로그램을 텍스트 모드로 실행하면 오류 메시지가 표시됩니다. 설치 프로그램을 영어 이외의 로케로 실행한 경우에도 이 메시지는 영어로 표시됩니다.(버그 6594142)
- 설치 프로그램이 실행되는 로케에 관계 없이 설치 프로그램의 사용권 화면에 영문 사용권 텍스트가 표시됩니다.(버그 6592399)

해결 방법: 현지화된 사용권 파일에 액세스하려면 `LICENSE_MULTILANGUAGE.pdf` 파일을 사용하십시오.

- 설치 프로그램 사용 도움말 텍스트가 현지화되지 않았습니다.(버그 6592493)
- 설치 프로그램 요약 HTML 페이지에 표시되는 문자열 "None"이 영어로 하드 코드되어 있습니다.(버그 6593089)
- 프랑스어를 제외한 나머지 로케에서 저작권 페이지가 현지화되지 않았습니다.(버그 6590992)
- 설치 프로그램을 독일어 로케로 실행하면 시작 화면에 다른 로케에서는 표시되는 전체 텍스트가 표시되지 않습니다.(버그 6592666)
- 설치 프로그램 설치 홈 화면에 표시되어야 할 "설치 홈" 문자열이 현지화되지 않았습니다. 설치 프로그램을 영어 이외의 로케로 실행한 경우에도 이 문자열은 영어로 표시됩니다.(버그 6592491)
- 설치 프로그램이 텍스트 모드에서 실행될 때는(`installer-t`), 설치 프로그램이 실행되는 로케에 관계 없이 영어 응답 선택 항목 "Yes"와 "No"가 사용됩니다.(버그 6593230)

- 설치 프로그램 JDK 선택 화면의 찾아보기 버튼에 대한 도구 설명이 영문으로 하드 코드되어 있습니다. (버그 6593085)

더 이상 사용되지 않는 비밀번호 옵션

이전 버전의 Message Queue에서는 `-p` 또는 `-password` 옵션을 사용하여 다음 명령에 대한 비밀번호를 대화식으로 지정할 수 있었습니다. 비밀번호를 대화식으로 지정할 수 있었습니다. 버전 4.0부터는 이러한 옵션이 더 이상 사용되지 않았습니다. 비밀번호는 다음과 같이 지정해야 합니다.

1. 비밀번호만을 저장하는 데 사용되는 파일에서 비밀번호 등록 정보를 원하는 값으로 설정합니다.

다음 구문을 사용하여 비밀번호 파일에 비밀번호를 지정합니다.

```
PasswordPropertyName= MyPassword
```

2. `-passfile` 옵션을 사용하여 비밀번호 파일의 이름을 전달합니다.

비밀번호 파일에는 아래 나열된 비밀번호가 하나 이상 포함될 수 있습니다.

- SSL 키 저장소를 여는 데 사용되는 키 저장소 비밀번호. 이 비밀번호를 지정하려면 `imq.keystore.password` 등록 정보를 사용합니다.
- 익명 연결이 아닌 경우 LDAP 디렉토리에 안전하게 연결하는 데 사용되는 LDAP 저장소 비밀번호. 이 비밀번호를 지정하려면 `imq.user_repository.ldap.password` 등록 정보를 사용합니다.
- JDBC 호환 데이터베이스에 연결하는 데 사용되는 JDBC 데이터베이스 비밀번호. 이 비밀번호를 지정하려면 `imq.persist.jdbc.vendorName.password` 등록 정보를 사용합니다. 등록 정보 이름의 `vendorName` 구성 요소는 데이터베이스 공급업체를 지정하는 변수입니다. 이 항목은 `hadb`, `derby`, `pointbase`, `oracle` 또는 `mysql`일 수 있습니다.
- 브로커 관리 작업을 수행하는 데 사용되는 `imqcmd` 명령에 대한 비밀번호. 이 비밀번호를 지정하려면 `imq.imqcmd.password` 등록 정보를 사용합니다.

다음 예에서는 JDBC 데이터베이스에 대한 비밀번호가 `abracadabra`로 설정되어 있습니다.

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

다음 방법 중 하나를 사용하여 사용자가 만든 비밀번호 파일을 사용하도록 브로커를 구성할 수 있습니다.

- 브로커의 `config.properties` 파일에서 다음 등록 정보를 설정합니다.

```
imq.passfile.enabled=true
imq.passfile.dirpath=MyFileDirectory
```



```
imq.passfile.name=MyPassfileName
```

- imqbrokerd 명령의 -passfile 옵션을 사용합니다.

```
imqbrokerd -passfile MyPassfileName
```

일반적인 문제

이 절에서는 Message Queue 4.1의 일반적인 문제를 설명합니다. 이 중 일부는 이전 Message Queue 버전에서 소개되었습니다.

- HTTP 전송을 사용하는 JMS 클라이언트가 갑자기 종료되면(예: Ctrl-C 사용) 브로커는 약 1분이 경과한 후 클라이언트 연결 및 연결된 모든 자원을 해제합니다. 클라이언트의 다른 인스턴스가 1분 이내에 시작되어 동일한 클라이언트 아이디, 영구 가입 또는 대기열을 사용하려고 하면 "클라이언트 아이디가 이미 사용 중"이라는 예외가 발생할 수 있습니다. 이것은 실제 문제가 아니라 위에서 설명한 종료 프로세스의 부작용일 뿐입니다. 약 1분 간의 지연 후에 클라이언트가 시작되면 모든 문제가 해결됩니다.
- SOAP 클라이언트 문제. 이전에는 mail.jar 및 mail.jar을 참조하는 데 사용되는 SAAJ 1.2 구현 jar이 CLASSPATH에 있을 필요가 없었습니다. SAAJ 1.3에서는 이 참조가 제거되었으므로 Message Queue 클라이언트에서 mail.jar을 명시적으로 CLASSPATH에 두어야 합니다.

관리/구성 문제

다음 문제는 Message Queue 관리 및 구성에 관련된 내용입니다.

- Windows 시스템에서 CLASSPATH에 큰따옴표가 포함되어 있으면 imqadmin 및 imqobjmgr 유틸리티에서 오류가 발생합니다(**버그 아이디 5060769**).
해결 방법: 이 오류 메시지는 무시해도 됩니다. 브로커가 모든 오류를 사용자에게 알리는 작업을 올바르게 수행합니다. 이 오류는 시스템의 안정성에 영향을 주지 않습니다.
- Solaris 및 Windows 스크립트의 -javahome 옵션에 입력한 값에 공백이 있는 경우 해당 옵션이 작동되지 않습니다(**버그 아이디 4683029**).
javahome 옵션은 Message Queue 명령 및 유틸리티에서 사용할 대체 Java 2 호환 런타임을 지정하는 데 사용됩니다. 그러나 대체 Java 런타임의 경로 이름에는 공백이 포함되지 않아야 합니다. 다음은 공백을 포함하는 경로의 예입니다.

Windows: C:/jdk 1.4

Solaris: /work/java 1.4

해결 방법: 공백이 없는 위치 또는 경로에 Java 런타임을 설치합니다.

- `imqQueueBrowserMaxMessagesPerRetrieve` 속성은 클라이언트 런타임에서 대기열의 내용을 찾아볼 때 한 번에 검색하는 메시지의 최대 수를 지정합니다. 클라이언트 응용 프로그램에는 항상 대기열의 모든 메시지가 표시됩니다. 따라서 `imqQueueBrowserMaxMessagesPerRetrieve` 속성은 클라이언트 런타임에 전달될 대기 메시지가 체크되는 방식(큰 메모리 체크 또는 작은 메모리 체크)에 영향을 미치지 않지만 검색되는 전체 메시지에 영향을 미치지는 않습니다. 이 속성 값을 변경하면 성능에 영향을 미칠 수 있지만 클라이언트 응용 프로그램에서 더 많은 데이터 또는 더 적은 데이터를 얻도록 하지는 않습니다(**버그 아이디 6387631**).

브로커 문제

다음 문제는 Message Queue 브로커에 영향을 줍니다.

- 라운드 로빈 전달에 대한 브로커 구성 방법에 약간의 혼동이 있었습니다. 해결책을 간단하게 구성할 수 있습니다.
 1. 대상 속성 `maxNumActiveConsumers`를 -1로 설정하여 라운드 로빈 전달을 활성화합니다.
 2. 대상 속성 `consumerFlowLimit`를 1로 설정하여 다음 사용자에게 전달하기 전에 단일 사용자에게 전달할 메시지의 수를 지정합니다. 다른 체크에서는 이 속성을 원하는 값으로 설정합니다. 기본적으로 100개 메시지가 각 사용자에게 전달됩니다.
- 영구 저장소가 너무 많은 대상을 열면 브로커에 액세스할 수 없게 됩니다(**버그 아이디 4953354**).

해결 방법: 이 상황은 브로커가 시스템 파일 열기 설명자 제한에 도달하면 발생합니다. Solaris와 Linux에서 `ulimit` 명령을 사용하여 파일 설명자 제한을 늘립니다.
- 대상이 삭제되면 사용자와의 연결이 끊어집니다(**버그 아이디 5060787**).

대상이 삭제되면 활성 사용자와의 연결이 끊어집니다. 사용자와의 연결이 끊어진 후에는 대상이 다시 생성된 후에도 더 이상 메시지를 받을 수 없습니다.

해결 방법: 이 문제는 해결 방법이 없습니다.

브로커 클러스터

다음 문제는 클러스터된 브로커에 영향을 줍니다.

- 이 릴리스에서는 완전히 연결된 브로커 클러스터만 지원됩니다. 즉, 클러스터에 있는 모든 브로커가 해당 클러스터 내의 다른 모든 브로커와 직접 통신해야 합니다. `imqbrokerd -cluster` 명령줄 인수를 사용하여 브로커를 연결하는 경우 해당 클러스터의 모든 브로커가 포함되어 있는지 확인해야 합니다.
- HADB를 사용하는 브로커는 10MB를 초과하는 메시지를 처리할 수 없습니다. (**버그 6531734**)

- 클라이언트가 고가용성 브로커에 연결되면 클라이언트 런타임이 `imqAddressListIterations` 설정 값에 관계 없이 성공할 때까지 재연결을 받습니다.)
- 클러스터의 일부인 브로커에 연결된 클라이언트는 현재 해당 클러스터에서 원격 브로커에 있는 대기열을 찾아보기 위해 `QueueBrowser`를 사용할 수 없습니다. 클라이언트는 직접 연결된 브로커에 있는 대기열의 내용만 찾아볼 수 있습니다. 그렇지만 클라이언트는 여전히 모든 대기열로 메시지를 보내거나 클러스터에 있는 모든 브로커의 대기열에서 메시지를 사용할 수 있습니다. 찾아보기만 제한을 받습니다.
- 기존 클러스터에서 3.x 브로커와 4.1 브로커를 클러스터링하려면 4.1 브로커에 대해 등록 정보 `imq.autocreate.queue.maxNumActiveConsumers=1`을 설정해야 합니다. 그렇지 않으면 브로커는 클러스터 연결을 설정할 수 없습니다.
- 고가용성 클러스터로 변환한 경우 Message Queue 관리자 유틸리티(`imqdbmgr`)를 사용하여 기존의 독립 실행형 HADB 영구 데이터 저장소를 공유 HADB 저장소로 변환할 수 있습니다. 명령은 다음과 같습니다.

```
imqdbmgr upgrade hystore
```

이 유틸리티는 다음과 같은 경우에 사용할 수 있습니다.

- 4.0 독립 실행형 HADB 저장소를 4.1 공유 HADB 저장소로 바꿀 경우. 이 경우 브로커는 저장소를 자동으로 업그레이드합니다. 이때 `imqdbmgr` 명령을 실행하여 업그레이드된 데이터 저장소가 공유될 수 있도록 변환할 수 있습니다.
- 4.1 독립 실행형 HADB 저장소를 공유 HADB 저장소로 바꿀 경우. 이 경우에는 위에 표시된 `imqdbmgr` 명령을 실행하여 데이터 저장소가 공유될 수 있도록 변환하기만 하면 됩니다.

이 명령은 HADB 저장소 변환만을 지원하므로 이를 사용하여 파일 기반의 저장소 또는 다른 JDBC 저장소를 공유 HADB 저장소로 변환할 수 없습니다. 이전에 Message Queue 3.x 버전을 실행 중인 경우 고가용성 기능을 사용하기 위해서는 HADB 저장소를 만든 다음 데이터를 해당 저장소로 수동으로 마이그레이션해야 합니다.

- `imqdbmgr upgrade hystore` 명령을 사용하여 HADB 저장소로 변환한 경우 저장소에 메시지가 10,000개 이상 저장되면 "too many locks are set" 메시지가 발생하면서 작업에 실패할 수 있습니다(**버그 아이디** 6588856).

해결 방법: 다음 명령을 사용하여 잠금 수를 늘립니다.

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

추가 정보에 대해서는 **Sun Java System Application Server 9.1 Enterprise Edition Troubleshooting Guide**의 "HADB Problems"를 참조하십시오.

- 한 트랜잭션에서 원격 메시지가 500개 이상 완결되면 브로커에서 "HADB-E-12815: Table memory space exhausted" 오류가 발생할 수 있습니다(**버그 아이디** 6550483)

추가 정보에 대해서는 **Sun Java System Application Server 9.1 Enterprise Edition Troubleshooting Guide**의 "HADB Problems"를 참조하십시오.

- 브로커 클러스터에서 브로커는 시작되지 않은 원격 연결에 메시지 대기열을 만듭니다(**버그 아이디** 4951010).

해결 방법: 일단 연결이 시작되면 사용자가 메시지를 수신합니다. 사용자의 연결이 종료된 경우 메시지는 다른 사용자에게 전달됩니다.

- 한 트랜잭션 내에서 원격 브로커의 메시지를 두 개 이상 사용할 경우 브로커에 다음 오류 메시지가 기록될 수 있습니다. 이 메시지는 문제를 일으키지 않으며 무시할 수 있습니다.

```
[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
```

이 메시지는 `imq.txn.reapLimit` 등록 정보가 한 트랜잭션 내의 원격 메시지 수에 비해 낮을 경우 트랜잭션 내의 최근 메시지에 대한 완결을 메시지 홈 브로커에 알릴 때 기록됩니다. (**버그 6585449**)

해결 방법: 이 메시지를 방지하려면 `imq.txn.reapLimit` 등록 정보 값을 늘립니다.

JMX 문제

Windows 플랫폼에서 트랜잭션 관리자 모니터 MBean의 `getTransactionInfo` 메소드가 잘못된 트랜잭션 작성 시간이 포함된 트랜잭션 정보를 반환합니다(**버그 아이디 6393359**).

해결 방법: 트랜잭션 관리자 모니터 MBean의 `getTransactionInfoByID` 메소드를 대신 사용합니다.

SOAP 지원

SOAP 지원과 관련된 다음의 두 가지 문제에 대해 잘 알고 있어야 합니다.

- Message Queue 버전 4.0 릴리스부터는 SOAP 관리 대상 객체가 더 이상 지원되지 않습니다.
- SOAP 개발은 SUNwjaf, SUNwjmail, SUNwxprt 및 SUNwjapx의 몇 가지 파일에 따라 달라집니다. Message Queue 버전 4.1에서는 JDK 버전 1.6.0 이상에서 Message Queue를 실행 중인 경우에만 사용자가 이 파일을 사용할 수 있습니다.

재배포 가능 파일

Sun Java System Message Queue 4.1에는 이진 형식으로 사용하고 자유롭게 배포할 수 있는 다음과 같은 파일 집합이 포함되어 있습니다.

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

LICENSE 및 COPYRIGHT 파일도 재배포할 수 있습니다.

내게 필요한 옵션 기능

이 매체를 발행한 이후 릴리스된 내게 필요한 옵션 기능을 사용하려면 Sun에 요청하여 구할 수 있는 섹션 508 제품 평가를 참조하여 관련 솔루션을 배포하는 데 가장 적합한 버전을 확인하십시오. 업데이트된 응용 프로그램 버전은

<http://sun.com/software/javaenterprisesystem/get.html>에서 볼 수 있습니다.

내게 필요한 옵션 기능 구현을 위한 Sun의 방침에 대해 자세히 알아보려면

<http://sun.com/access>를 방문하십시오.

문제 보고 및 사용자 의견 제공 방법

Sun Java System Message Queue에 문제가 있으면 다음 방법 중 하나를 사용하여 Sun 고객 지원부에 문의하십시오.

- Sun 소프트웨어 지원 온라인 서비스(<http://www.sun.com/service/sunone/software>)
이 사이트에는 유지 보수 프로그램과 지원 담당자 연락처뿐만 아니라 기술 자료, 온라인 지원 센터 및 ProductTracker에 대한 링크가 있습니다.
- 유지 보수 계약과 관련된 직통 전화번호

고객 지원부에 문의할 때에는 문제를 해결하는 데 최선의 도움을 제공할 수 있도록 다음 정보를 미리 준비해 두십시오.

- 문제가 발생한 상황과 해당 문제가 작업에 미친 영향 등을 포함한 문제에 대한 자세한 설명
- 문제에 영향을 줄 수 있는 모든 패치 및 기타 소프트웨어를 포함한 컴퓨터 유형, 운영 체제 버전 및 제품 버전 등의 정보
- 문제를 재현하기 위해 사용한 방법에 대한 자세한 단계
- 오류 로그 또는 코어 덤프

Sun Java System 소프트웨어 포럼

다음 웹 사이트에서는 Sun Java System Message Queue 포럼을 제공합니다.

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

여러분의 참여를 환영합니다.

Java 기술 포럼

Java 기술 포럼에는 관련 JMS 포럼이 있습니다.

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

사용자 의견 환영

Sun은 설명서의 내용 개선에 노력을 기울이고 있으며, 여러분의 의견과 제안을 환영합니다.

사용자 의견을 보내시려면 <http://docs.sun.com>에서 의견 보내기를 누르십시오. 온라인 양식에 설명서 제목과 부품 번호를 입력합니다. 부품 번호는 해당 설명서의 제목 페이지나 문서 맨 위에 있는 7자리 또는 9자리 숫자입니다. 예를 들어, 이 설명서의 제목은 Sun Java System Message Queue 4.1 릴리스 노트이며 부품 번호는 820-3192입니다.

Sun의 추가 자원

다음 인터넷 사이트에서 유용한 Sun Java System 관련 정보를 찾을 수 있습니다.

- 설명서
<http://docs.sun.com/prod/java.sys>
- 전문가 서비스
<http://www.sun.com/service/sunps/sunone>
- 소프트웨어 제품 및 서비스
<http://www.sun.com/software>
- 소프트웨어 지원 서비스
<http://www.sun.com/service/sunone/software>
- 지원 및 기술 자료
<http://www.sun.com/service/support/software>
- Sun 지원 및 교육 서비스
<http://training.sun.com>
- 컨설팅 및 전문가 서비스
<http://www.sun.com/service/sunps/sunone>
- 개발자 정보
<http://developers.sun.com>
- Sun 개발자 지원 서비스
<http://www.sun.com/developers/support>
- 소프트웨어 교육
<http://www.sun.com/software/training>

