



Sun Java™ System  
Message Queue 3  
관리 설명서

---

2005Q1

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

부품 번호: 819-2218

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 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 <http://www.sun.com/patents>에 나열된 하나 이상의 미국 특허권이 포함될 수 있으며, 미국 및 다른 국가에서 하나 이상의 추가 특허권 또는 출원 중인 특허권이 제한 없이 포함될 수 있습니다.

미국 정부의 권리 - 상용 소프트웨어. 정부 사용자는 Sun Microsystems, Inc. 표준 사용권 계약과 해당 FAR 규정 및 보충 규정을 준수해야 합니다. 본 제품의 사용은 사용권 조항의 적용을 받습니다. 이 배포에는 타사에서 개발한 자료가 포함되어 있을 수 있습니다.

Sun, Sun Microsystems, Sun 로고, Java, Solaris, Sun[tm] ONE, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp 및 Javadoc는 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.가 개발한 구조를 기반으로 하고 있습니다.

UNIX는 미국 및 다른 국가에서 X/Open Company, Ltd.를 통해 독점적으로 사용권이 부여되는 등록 상표입니다.

이 제품은 미국 수출 관리법에 의해 규제되며 다른 국가의 수출 또는 수입 관리법의 적용을 받을 수도 있습니다. 이 제품과 정보를 직간접적으로 핵 무기, 미사일 또는 생화학 무기에 사용하거나 핵과 관련하여 해상에서 사용하는 것은 엄격하게 금지됩니다. 미국 수출 금지 국가 또는 금지된 개인과 특별히 지정된 국민 목록을 포함하여 미국 수출 금지 목록에 지정된 대상으로의 수출이나 재수출은 엄격하게 금지됩니다.

# 목차

<b>그림 목차</b> .....	<b>13</b>
<b>표 목차</b> .....	<b>15</b>
<b>절차 목록</b> .....	<b>19</b>
<b>머리말</b> .....	<b>21</b>
대상 .....	22
설명서를 읽기 전에 .....	22
구성 .....	22
본 설명서에 사용된 규약 .....	24
텍스트 표기 규칙 .....	24
디렉토리 변수 규칙 .....	25
관련 문서 .....	27
Message Queue 설명서 집합 .....	27
온라인 도움말 .....	28
JavaDoc .....	28
클라이언트 응용 프로그램의 예 .....	28
JMS(Java Message Service) 사양 .....	29
관련 타사 웹 사이트 참조 .....	29
사용자 의견 환영 .....	29

**I부 Message Queue 관리 소개 ..... 31**

**1장 관리 작업 및 도구 ..... 33**

개발 환경의 관리 작업 ..... 34  
작업 환경의 관리 작업 ..... 34  
    설정 작업 ..... 35  
    유지 보수 작업 ..... 36  
관리 도구 ..... 37  
    명령줄 유틸리티 ..... 37  
    관리 콘솔 ..... 39

**2장 관리 빠른 시작 ..... 41**

준비 ..... 42  
관리 콘솔 시작 ..... 42  
    도움말 정보 ..... 44  
브로커 시작 ..... 45  
브로커 추가 ..... 46  
브로커에 연결 ..... 48  
    연결 서비스 보기 ..... 48  
    브로커에 물리적 대상 추가 ..... 50  
    물리적 대상 관리 ..... 51  
    항목 정보 얻기 ..... 53  
객체 저장소 작업 ..... 54  
    객체 저장소 추가 ..... 54  
    객체 저장소 등록 정보 확인 ..... 57  
    객체 저장소에 연결 ..... 57  
    연결 팩토리 관리 대상 객체 추가 ..... 57  
대상 객체 추가 ..... 59  
관리 대상 객체 등록 정보 보기 ..... 61  
콘솔 정보 업데이트 ..... 62  
샘플 응용 프로그램 실행 ..... 62

**II부 관리 작업 ..... 65**

**3장 브로커 및 클라이언트 시작 ..... 65**

시스템 자원 준비 ..... 66  
    시스템 클럭 동기화 ..... 66  
    파일 설명자 제한 설정(Solaris 또는 Linux) ..... 66  
대화식으로 브로커 시작 ..... 67

자동으로 브로커 시작 .....	68
자동 시작(Solaris 및 Linux) .....	68
자동 시작(Windows) .....	69
Message Queue 클라이언트 시작 .....	71
브로커 인스턴스 제거 .....	72
<b>4장 브로커 구성 .....</b>	<b>73</b>
구성 가능한 브로커 구성 요소 정보 .....	74
연결 서비스 .....	75
메시지 라우터 .....	79
지속성 관리자 .....	83
보안 관리자 .....	88
모니터링 서비스 .....	91
구성 파일 정보 .....	96
인스턴스 구성 파일 .....	96
등록 정보 값 병합 .....	97
등록 정보 이름 지정 구문 .....	98
인스턴스 구성 파일 편집 .....	98
명령줄에 구성 옵션 입력 .....	99
영구 저장소 설정 .....	99
파일 시스템 저장소 구성 .....	100
JDBC 저장소 구성 .....	100
지속성 데이터 보안 .....	104
기본 제공(파일 기반) 영구 저장소 .....	105
플러그 인(JDBC) 영구 저장소 .....	105
<b>5장 브로커 관리 .....</b>	<b>107</b>
전제 조건 .....	108
imqcmd 명령 유틸리티 사용 .....	108
아이디 및 비밀번호 지정 .....	109
브로커 이름 및 포트 지정 .....	109
예 .....	110
도움말 표시 .....	110
제품 버전 표시 .....	111
브로커 정보 표시 .....	111
브로커 등록 정보 업데이트 .....	112
브로커 일시 중지 및 다시 시작 .....	113
브로커 일시 중지 .....	113
브로커 다시 시작 .....	114
브로커 종료 및 다시 시작 .....	114
브로커 메트릭 표시 .....	115

연결 서비스 관리 .....	116
연결 서비스 나열 .....	117
연결 서비스 정보 표시 .....	118
연결 서비스 등록 정보 업데이트 .....	118
연결 서비스 매트릭 표시 .....	119
연결 서비스 일시 중지 및 다시 시작 .....	120
연결 정보 얻기 .....	121
영구 가입 관리 .....	122
트랜잭션 관리 .....	123

## **6장 물리적 대상 관리 .....** **127**

imqcmd 명령 유틸리티 사용 .....	128
하위 명령 .....	128
물리적 대상 만들기 .....	129
물리적 대상 나열 .....	131
물리적 대상 정보 표시 .....	131
물리적 대상 등록 정보 업데이트 .....	133
물리적 대상 일시 중지 및 다시 시작 .....	133
물리적 대상 제거 .....	134
물리적 대상 완전 삭제 .....	135
물리적 대상 압축 .....	136
사용 불능 메시지 대기열 사용 구성 .....	138
사용 불능 메시지 대기열 사용 구성 .....	138
사용 불능 메시지 대기열 구성 및 관리 .....	139
사용 불능 메시지 로깅 사용 .....	140

## **7장 보안 관리 .....** **141**

사용자 인증 .....	142
플랫 파일 사용자 저장소 사용 .....	142
사용자 저장소에 LDAP 서버 사용 .....	149
사용자 권한 부여: 액세스 제어 등록 정보 파일 .....	152
액세스 제어 등록 정보 파일 작성 .....	153
액세스 규칙 구문 .....	154
사용 권한 계산 방법 .....	155
연결 서비스에 대한 액세스 제어 .....	156
물리적 대상에 대한 액세스 제어 .....	157
자동 작성 물리적 대상에 대한 액세스 제어 .....	158
SSL 기반 서비스 작업 .....	159
TCP/IP에 대한 보안 연결 서비스 .....	160
자체 서명된 인증서 사용 구성 .....	160
서명된 인증서 사용 구성 .....	166

Passfile 사용 .....	169
보안 문제 .....	170
Passfile 내용 .....	170
감사 로그 만들기 .....	171
<b>8장 관리 대상 객체 관리 .....</b>	<b>173</b>
객체 저장소 정보 .....	174
LDAP 서버 객체 저장소 .....	174
파일 시스템 객체 저장소 .....	175
관리 대상 객체 속성 정보 .....	176
연결 팩토리 속성 .....	177
클라이언트 아이디 .....	180
대상 관리 대상 객체 속성 .....	185
객체 관리자 유틸리티(imqobjmgr) 사용 .....	185
필요한 정보 .....	185
명령 파일 사용 .....	186
관리 대상 객체 추가 및 삭제 .....	189
연결 팩토리 추가 .....	189
주제 또는 대기열 추가 .....	190
관리 대상 객체 삭제 .....	192
관리 대상 객체 나열 .....	193
단일 객체 관련 정보 얻기 .....	193
관리 대상 객체 업데이트 .....	194
<b>9장 브로커 클러스터 작업 .....</b>	<b>195</b>
클러스터 구성 등록 정보 .....	196
개별 브로커에 대한 클러스터 등록 정보 설정 .....	197
클러스터 구성 파일 사용 .....	197
클러스터 관리 .....	198
브로커 연결 .....	198
클러스터에 브로커 추가 .....	199
클러스터에서 브로커 제거 .....	200
마스터 브로커 .....	201
구성 변경 기록 관리 .....	201
마스터 브로커를 사용할 수 없는 경우 .....	202
<b>10장 메시지 서버 모니터링 .....</b>	<b>203</b>
모니터링 도구 소개 .....	203
브로커 로깅 구성 및 사용 .....	205
기본 로깅 구성 .....	205
로그 메시지 형식 .....	206
로거 구성 변경 .....	206

대화식 메트릭 표시 .....	210
imqcmd 메트릭 .....	211
메트릭 하위 명령을 사용하여 메트릭 데이터 표시 .....	212
메트릭 출력: imqcmd metrics .....	213
imqcmd query .....	214
브로커를 모니터링하는 응용 프로그램 작성 .....	215
메시지 기반 모니터링 설정 .....	216
보안 및 액세스 고려 사항 .....	217
메트릭 출력: 메트릭 메시지 .....	218

## **11장 메시지 서비스 분석 및 조정 .....** **219**

성능 정보 .....	219
성능 조정 프로세스 .....	219
성능 요소 .....	220
벤치마크 .....	221
기본 사용 패턴 .....	222
성능에 영향을 미치는 요소 .....	223
성능에 영향을 미치는 응용 프로그램 설계 요소 .....	224
성능에 영향을 미치는 메시지 서비스 요소 .....	232
성능 향상을 위한 구성 조정 .....	237
시스템 조정 .....	237
브로커 조정 .....	242
클라이언트 런타임 메시지 흐름 조정 .....	244

## **12장 문제 해결 .....** **247**

클라이언트가 연결을 설정할 수 없음 .....	248
연결 처리 속도가 너무 느림 .....	253
클라이언트가 메시지 생성자를 만들 수 없음 .....	255
메시지 생성이 지연되거나 느림 .....	256
메시지가 백로그됨 .....	259
메시지 서버 처리 능력이 일정하지 않음 .....	264
메시지가 사용자에게 도달하지 않음 .....	265
사용 불능 메시지 대기열에 메시지가 포함되어 있음 .....	269

## **III부 참조 .....** **277**

### **13장 명령 참조 .....** **279**

명령줄 구문 .....	280
명령 입력 규칙 .....	280
명령줄 예 .....	280
공통 명령 옵션 .....	281



imqbrokerd	282
구문	282
명령 옵션	282
참조	286
imqcmd	287
구문	287
하위 명령	287
명령 옵션	294
참조	296
imqobjmgr	297
구문	297
하위 명령	297
명령 옵션	298
참조	299
imqdbmgr	300
구문	300
하위 명령	300
명령 옵션	301
참조	301
imqusermgr	302
구문	302
하위 명령	302
명령 옵션	303
참조	303
imqsvcadm	304
구문	304
하위 명령	304
명령 옵션	304
참조	305
imqkeytool	306
구문	306
참조	306
<b>14장 브로커 등록 정보 참조</b>	<b>307</b>
알파벳 순 등록 정보 목록	307
연결 서비스 등록 정보	311
메시지 라우터 등록 정보	313
지속성 관리자 등록 정보	316
파일 기반 지속성	316
JDBC 기반 지속성	317
보안 관리자 등록 정보	320
모니터링 및 로깅 등록 정보	324
클러스터 구성 등록 정보	326

<b>15장 물리적 대상 등록 정보 참조</b> .....	<b>329</b>
<b>16장 관리 대상 객체 속성 참조</b> .....	<b>333</b>
대상 등록 정보 .....	333
연결 팩토리 속성 .....	334
연결 처리 .....	334
클라이언트 아이디 .....	338
메시지 헤더 무시 .....	338
안정성 및 흐름 제어 .....	339
대기열 브라우저 동작 및 서버 세션 .....	340
JMS 정의 등록 정보 지원 .....	341
SOAP 종점 속성 .....	342
<b>17장 JMS 자원 어댑터 속성 참조</b> .....	<b>343</b>
ResourceAdapter JavaBean .....	344
ManagedConnectionFactory JavaBean .....	345
ActivationSpec JavaBean .....	346
<b>18장 메트릭 참조</b> .....	<b>349</b>
JVM 메트릭 .....	349
브로커 전체 메트릭 .....	350
연결 서비스 메트릭 .....	352
대상 메트릭 .....	354
<b>IV부 부록</b> .....	<b>357</b>
<b>부록 A 운영 체제별 Message Queue 데이터의 위치</b> .....	<b>359</b>
Solaris .....	359
Linux .....	361
Windows .....	362
<b>부록 B Message Queue 인터페이스의 안정성</b> .....	<b>365</b>

<b>부록 C HTTP/HTTPS 지원</b> .....	<b>369</b>
HTTP/HTTPS 지원 구조 .....	370
HTTP 지원 활성화 .....	371
1단계. 웹 서버에 HTTP 터널 서블릿 배포 .....	372
2단계. httpjms 연결 서비스 구성 .....	373
3단계. HTTP 연결 구성 .....	374
예 1: Sun Java System Web Server에 HTTP 터널 서블릿 배포 .....	376
예 2: Sun Java System Application Server 7.0에 HTTP 터널 서블릿 배포 .....	380
HTTPS 지원 활성화 .....	382
1단계. HTTPS 터널 서블릿에 대해 자체 서명된 인증서 생성 .....	382
2단계. 웹 서버에 HTTPS 터널 서블릿 배포 .....	383
3단계. httpsjms 연결 서비스 구성 .....	385
4단계. HTTPS 연결 구성 .....	386
예 3: Sun Java System Web Server에 HTTPS 터널 서블릿 배포 .....	389
예 4: Sun Java System Application Server 7.0에 HTTPS 터널 서블릿 배포 .....	394
문제 해결 .....	396
서버 또는 브로커 오류 .....	396
터널 서블릿을 통한 클라이언트 연결 오류 .....	396
<b>용어집</b> .....	<b>397</b>
<b>색인</b> .....	<b>399</b>



# 그림 목차

그림 1-1	로컬 및 원격 관리 유틸리티 .....	38
그림 4-1	브로커 서비스 구성 요소 .....	74
그림 4-2	연결 서비스 지원 .....	76
그림 4-3	지속성 관리자 지원 .....	84
그림 4-4	보안 관리자 지원 .....	89
그림 4-5	모니터링 서비스 지원 .....	92
그림 4-6	브로커 구성 파일 .....	97
그림 11-1	Message Queue 서비스를 통한 메시지 전달 .....	223
그림 11-2	전달 모드의 성능 영향 .....	227
그림 11-3	가입 유형의 성능 영향 .....	229
그림 11-4	메시지 크기의 성능 영향 .....	231
그림 11-5	전송 프로토콜 속도 .....	234
그림 11-6	전송 프로토콜의 성능 영향 .....	235
그림 11-7	1k(1024바이트) 패킷에 대한 inbufsz 변경의 결과 .....	240
그림 11-8	1k(1024바이트) 패킷에 대한 outbufsz 변경의 결과 .....	241
그림 12-1	QBrowser 창 .....	267
그림 12-2	QBrowser 메시지 세부 정보 .....	268
그림 C-1	HTTP/HTTPS 지원 구조 .....	370



# 표 목차

표 1	책 내용 .....	22
표 2	문서 표기 규칙 .....	24
표 3	Message Queue 디렉토리 변수 .....	25
표 4	Message Queue 설명서 집합 .....	27
표 4-1	주요 브로커 구성 요소 및 기능 .....	75
표 4-2	브로커가 지원하는 연결 서비스 .....	76
표 4-3	메트릭 주제 대상 .....	93
표 5-1	브로커가 지원하는 연결 서비스 .....	116
표 5-2	imqcmd가 업데이트하는 연결 서비스 등록 정보 .....	118
표 6-1	imqcmd 명령 유틸리티의 물리적 대상 하위 명령 .....	128
표 6-2	물리적 대상 디스크 사용률 메트릭 .....	137
표 6-3	사용 불가능 메시지 대기열에서 처리하는 표준 물리적 대상 등록 정보 .....	139
표 7-1	사용자 저장소 초기 항목 .....	143
표 7-2	imqusermgr 옵션 .....	145
표 7-3	액세스 규칙의 구문 요소 .....	154
표 7-4	물리적 대상 액세스 제어 규칙의 요소 .....	157
표 7-5	자체 서명된 인증서에 필요한 고유 이름 정보 .....	161
표 7-6	비밀번호를 사용하는 명령 .....	169
표 7-7	Passfile의 비밀번호 .....	170
표 8-1	LDAP 객체 저장소 속성 .....	174
표 8-2	파일 시스템 객체 저장소 속성 .....	176
표 8-3	이름 지정 규약 예 .....	190
표 10-1	메트릭 모니터링 도구의 장점 및 제한 .....	204
표 10-2	로깅 수준 .....	206
표 10-3	imqbrokerd 로거 옵션 및 해당 등록 정보 .....	207
표 10-4	imqcmd metrics 하위 명령 구문 .....	211
표 10-5	imqcmd metrics 하위 명령 옵션 .....	212

표 10-6	imqcmd query 하위 명령 구분	215
표 10-7	메트릭 주제 대상	216
표 11-1	높은 안정성 및 높은 성능 시나리오 비교	225
표 13-1	공통 Message Queue 명령줄 옵션	281
표 13-2	imqbrokerd 옵션	282
표 13-3	imqcmd 하위 명령	287
표 13-4	브로커 관리에 사용되는 imqcmd 하위 명령	289
표 13-5	대상 관리에 사용되는 imqcmd 하위 명령	290
표 13-6	연결 서비스 관리에 사용되는 imqcmd 하위 명령	292
표 13-7	연결 서비스 관리에 사용되는 imqcmd 하위 명령	293
표 13-8	영구 가입 관리에 사용되는 imqcmd 하위 명령	293
표 13-9	트랜잭션 관리에 사용되는 imqcmd 하위 명령	294
표 13-10	imqcmd 옵션	294
표 13-11	imqobjmgr 하위 명령	297
표 13-12	imqobjmgr 옵션	298
표 13-13	imqdbmgr 하위 명령	300
표 13-14	imqdbmgr 옵션	301
표 13-15	imqusermgr 하위 명령	302
표 13-16	imqusermgr 옵션	303
표 13-17	imqsvcadm 하위 명령	304
표 13-18	imqsvcadm 옵션	304
표 14-1	브로커 인스턴스 구성 등록 정보	308
표 14-2	연결 서비스 등록 정보	311
표 14-3	메시지 라우터 등록 정보	313
표 14-4	자동 작성 구성 등록 정보	314
표 14-5	필수 지속성 관리자 등록 정보	316
표 14-6	파일 기반 지속성 등록 정보	317
표 14-7	JDBC 기반 지속성 등록 정보	317
표 14-8	보안 관리자 등록 정보	320
표 14-9	키 저장소 등록 정보	323
표 14-10	모니터링 서비스 등록 정보	324
표 14-11	클러스터 구성 등록 정보	326
표 15-1	물리적 대상 등록 정보	329
표 16-1	대상 관리 대상 객체 속성	333
표 16-2	연결 팩토리 속성: 연결 처리	334
표 16-3	imqAddressList 속성의 주소 지정 체계	336
표 16-4	메시지 서버 주소 예	337



표 16-5	연결 팩토리 속성: 클라이언트 아이디 .....	338
표 16-6	연결 팩토리 속성: 메시지 헤더 무시 .....	338
표 16-7	연결 팩토리 속성: 안정성 및 흐름 제어 .....	339
표 16-8	연결 팩토리 속성: 대기열 브라우저 동작 .....	340
표 16-9	연결 팩토리 속성: JMS 정의 등록 정보 지원 .....	341
표 16-10	SOAP 중점 속성 .....	342
표 17-1	자원 어댑터 속성 .....	344
표 17-2	관리 대상 연결 팩토리 속성 .....	345
표 17-3	활성화 사양 속성 .....	347
표 18-1	JVM 메트릭 .....	349
표 18-2	브로커 전체 메트릭 .....	350
표 18-3	연결 서비스 메트릭 .....	352
표 18-4	대상 메트릭 .....	354
표 A-1	Solaris에서 Message Queue 데이터의 위치 .....	360
표 A-2	Linux에서 Message Queue 데이터의 위치 .....	361
표 A-3	Windows에서 Message Queue 데이터의 위치 .....	362
표 B-1	인터페이스 안정성 분류 체계 .....	365
표 B-2	Message Queue 인터페이스의 안정성 .....	366
표 C-1	httpjms 연결 서비스 등록 정보 .....	373
표 C-2	HTTP 터널 서블릿 Jar 파일 배포에 사용되는 서블릿 인수 .....	377
표 C-3	httpsjms 연결 서비스 등록 정보 .....	385
표 C-4	HTTPS 터널 서블릿 Jar 파일 배포에 사용되는 서블릿 인수 .....	390



# 절차 목록

관리 콘솔 도움말 정보를 표시하는 방법	44
관리 콘솔에 브로커를 추가하는 방법	46
브로커에 연결하는 방법	48
사용 가능한 연결 서비스를 보는 방법	48
브로커에 대기열 대상을 추가하는 방법	50
물리적 대상의 등록 정보를 보는 방법	52
물리적 대상에서 메시지를 제거하는 방법	53
대상을 삭제하는 방법	53
파일 시스템 객체 저장소를 추가하는 방법	54
객체 저장소 등록 정보를 표시하는 방법	57
객체 저장소에 연결하는 방법	57
객체 저장소에 연결 팩토리를 추가하는 방법	58
객체 저장소에 대상을 추가하는 방법	60
대상 객체의 등록 정보를 확인 또는 업데이트하는 방법	61
HelloWorldMessageJNDI 응용 프로그램을 실행하는 방법	62
기록된 서비스 오류 이벤트를 보는 방법	70
기본적인 전달 메커니즘	79
JDBC로 액세스할 수 있는 데이터 저장소를 플러그 인하는 방법	101
물리적 대상 생성 방법	130
사용되지 않는 물리적 대상 디스크 공간을 확보하는 방법	138
LDAP 서버를 사용하도록 구성 파일을 편집하는 방법	149
관리 사용자 설정 방법	151
SSL 기반 연결 서비스를 설정하는 방법	160
키 쌍을 다시 생성하는 방법	163
브로커에서 SSL 기반 서비스를 활성화하는 방법	163
서명된 인증서를 가져오는 방법	166
서명된 인증서 설치 방법	167

Java 클라이언트 런타임 구성 방법 .....	167
클러스터 구성 파일을 사용하여 클러스터에 새 브로커를 추가하는 방법 .....	199
클러스터 구성 파일을 사용하지 않고 클러스터에 새 브로커를 추가하는 방법 .....	199
명령줄을 사용하여 클러스터에서 브로커를 제거하는 방법 .....	200
클러스터 구성 파일을 사용하여 클러스터에서 브로커를 제거하는 방법 .....	200
구성 변경 기록을 백업하는 방법 .....	201
구성 변경 기록 복원 방법 .....	202
브로커의 로거 구성을 변경하는 방법 .....	206
로그 파일을 사용하여 메트릭 정보를 보고하는 방법 .....	209
메트릭 하위 명령을 사용하는 방법 .....	212
메시지 기반 모니터링을 설정하는 방법 .....	216
HTTP 지원을 활성화하는 방법 .....	371
httpjms 연결 서비스를 활성화하는 방법 .....	373
터널 서블릿을 추가하는 방법 .....	376
터널 서블릿에 대한 가상 경로(서블릿 URL)를 구성하는 방법 .....	377
웹 서버 시작 시 터널 서블릿을 로드하는 방법 .....	378
서버 액세스 로그를 비활성화하는 방법 .....	378
http 터널 서블릿을 WAR 파일로 배포하는 방법 .....	378
Application Server 7.0 환경에 HTTP 터널 서블릿을 배포하는 방법 .....	380
응용 프로그램 서버의 server.policy 파일을 수정하는 방법 .....	381
HTTPS 지원을 활성화하는 방법 .....	382
httpsjms 연결 서비스를 활성화하는 방법 .....	385
JSSE를 구성하는 방법 .....	386
터널 서블릿을 추가하는 방법 .....	389
터널 서블릿에 대한 가상 경로(서블릿 URL)를 구성하는 방법 .....	391
웹 서버 시작 시 터널 서블릿을 로드하는 방법 .....	391
서버 액세스 로그를 비활성화하는 방법 .....	391
HTTPS 터널 서블릿 WAR 파일을 수정하는 방법 .....	392
https 터널 서블릿을 WAR 파일로 배포하는 방법 .....	393
Application Server 7.0 환경에 HTTPS 터널 서블릿을 배포하는 방법 .....	394
응용 프로그램 서버의 server.policy 파일을 수정하는 방법 .....	395

# 머리말

Sun Java™ System Message Queue *관리 설명서*에서는 Message Queue 메시징 시스템을 관리하는 데 필요한 정보를 제공합니다.

이 설명서에서는 Sun Java™ System Message Queue 3 2005Q1(Message Queue 3.6)에 대해 설명합니다.

머리말은 다음과 같은 절로 구성되어 있습니다.

- 22페이지의 "대상"
- 22페이지의 "설명서를 읽기 전에"
- 22페이지의 "구성"
- 24페이지의 "본 설명서에 사용된 규약"
- 27페이지의 "관련 문서"
- 29페이지의 "관련 타사 웹 사이트 참조"
- 29페이지의 "사용자 의견 환영"

# 대상

이 설명서는 Message Queue 관리 작업을 수행해야 하는 관리자와 응용 프로그램 개발자를 대상으로 합니다.

Message Queue 관리자는 Message Queue 메시징 시스템, 특히 이 시스템의 핵심인 Message Queue 메시지 서버를 설정하고 관리합니다.

## 설명서를 읽기 전에

*Message Queue 기술 개요*를 읽어 Java Message Specification의 Message Queue 구현, Message Queue 서비스 구성 요소 및 Message Queue 응용 프로그램 개발, 배포, 관리의 기본 프로세스를 잘 알고 있어야 합니다.

## 구성

다음 표에서는 설명서의 내용을 간단히 설명합니다.

**표 1** 설명서 내용

부/장	설명
<b>I부, "Message Queue 관리 소개"</b>	
1장, "관리 작업 및 도구"	Message Queue 관리 작업 및 도구에 대해 소개합니다.
2장, "관리 빠른 시작"	관리 콘솔을 익힐 수 있는 실습 자습서를 제공합니다.
<b>II부, "관리 작업"</b>	
3장, "브로커 및 클라이언트 시작"	Message Queue 브로커 및 클라이언트를 시작하는 방법을 설명합니다.
4장, "브로커 구성"	구성 등록 정보를 설정하고 읽는 방법을 설명하고 브로커의 구성 가능 측면에 대해 소개합니다. 또한, 지속성 기능을 수행하도록 파일 또는 데이터베이스를 설정하는 방법을 설명합니다.
5장, "브로커 관리"	브로커 관리 작업을 설명합니다.

표 1 설명서 내용(계속)

부/장	설명
6장, "물리적 대상 관리"	주제 및 대기열 관련 관리 작업에 대해 설명합니다.
7장, "보안 관리"	비밀번호 파일 관리, 인증, 권한 부여, 암호화 등과 같은 보안 관련 작업에 대해 설명합니다.
8장, "관리 대상 객체 관리"	대상 관리 대상 객체 및 연결 팩토리 관리 대상 객체 관련 작업을 수행하는 방법과 객체 저장소에 대해 설명합니다.
9장, "브로커 클러스터 작업"	Message Queue 브로커 클러스터 설정 및 관리 방법에 대해 설명합니다.
10장, "메시지 서버 모니터링"	Message Queue 모니터링 기능 설정 및 사용 방법에 대해 설명합니다.
11장, "메시지 서비스 분석 및 조정"	메시지 서버 성능을 분석하는 기술과 메시지 서버를 조정하여 성능을 최적화하는 방법을 설명합니다.
12장, "문제 해결"	일반적인 Message Queue 문제의 원인을 확인하는 방법과 문제 해결을 위해 수행 가능한 작업에 대해 제안합니다.
<b>III부, "참조"</b>	
13장, "명령 참조"	Message Queue 명령 유틸리티에 대한 구문과 설명을 제공합니다.
14장, "브로커 등록 정보 참조"	브로커를 구성할 때 사용할 수 있는 등록 정보를 나열하고 설명합니다.
15장, "물리적 대상 등록 정보 참조"	주제와 대기열을 구성할 때 사용할 수 있는 등록 정보를 나열하고 설명합니다.
16장, "관리 대상 객체 속성 참조"	대상 관리 대상 객체 및 연결 팩토리 관리 대상 객체를 구성할 때 사용할 수 있는 등록 정보를 나열하고 설명합니다.
17장, "JMS 자원 어댑터 속성 참조"	Application Server에서 사용할 수 있도록 Message Queue 자원 어댑터를 구성할 때 사용 가능한 등록 정보를 나열하고 설명합니다.
18장, "메트릭 참조"	Message Queue 브로커에서 생성되는 메트릭을 나열하고 설명합니다.
<b>IV부, "부록"</b>	
부록 A, "운영 체제별 Message Queue 데이터의 위치"	지원되는 각 플랫폼에서 Message Queue 파일의 위치를 나열합니다.
부록 B, "Message Queue 인터페이스의 안정성"	다양한 Message Queue 인터페이스의 안정성에 대해 설명합니다.
부록 C, "HTTP/HTTPS 지원"	Message Queue 통신을 위한 HTTP 사용 설정 방법을 설명합니다.

# 본 설명서에 사용된 규약

이 절에서는 본 설명서에 사용되는 표기 규칙에 관한 정보를 제공합니다.

## 활자체 규약

**표 2** 문서 표기 규칙

형식	설명
<i>기울임꼴</i>	기울임꼴 텍스트는 위치 표시자를 나타냅니다. 기울임꼴 텍스트로 표시된 부분은 적절한 절 또는 값으로 대체합니다. 기울임꼴 텍스트는 문서 제목, 강조 또는 소개할 단어나 구를 지정할 때도 사용됩니다.
고정 폭	고정 폭 텍스트는 코드의 예, 명령줄에 입력하는 명령, 디렉토리/파일/경로 이름, 오류 메시지 텍스트, 클래스 이름, 메소드 이름(서명의 모든 요소 포함), 패키지 이름, 예약어, URL을 나타냅니다.
[]	대괄호는 명령줄 구문에 선택적으로 사용되는 값을 나타냅니다.
모두 대문자	모두 대문자로 표시된 텍스트는 파일 시스템 유형(GIF, TXT, HTML 등), 환경 변수(IMQ_HOME) 또는 머릿글자(Message Queue, JSP)를 나타냅니다.
키+키	동시 키 입력은 더하기 기호와 함께 표시됩니다. Ctrl+A는 두 키를 동시에 누르라는 의미입니다.
키-키	연속적인 키 입력은 하이픈과 함께 표시됩니다. Esc-S는 Esc를 누르고 이를 놓은 다음 S 키를 누르라는 의미입니다.



## 디렉토리 변수 규칙

Message Queue에서는 세 가지 디렉토리 변수를 사용하며 설정 방법은 플랫폼에 따라 다릅니다. 표 3에서는 이러한 변수를 설명하고 이들이 Solaris™, Windows, Linux 플랫폼에 사용되는 방법을 개괄적으로 설명합니다.

**표 3** Message Queue 디렉토리 변수

변수	설명
<b>IMQ_HOME</b>	<p>보통은 다음과 같이 Message Queue 설명서에서 Message Queue 기본 디렉토리(루트 설치 디렉토리)를 참조할 때 사용됩니다.</p> <ul style="list-style-type: none"> <li>Solaris 및 Linux에는 루트 Message Queue 설치 디렉토리가 없습니다. 따라서 Message Queue 설명서에서 Solaris의 파일 위치를 참조하는 경우에는 IMQ_HOME이 사용되지 않습니다.</li> <li>Solaris와 Windows에서 Sun Java System Application Server의 루트 Message Queue 설치 디렉토리는 Application Server 기본 디렉토리 아래에 있는 /imq입니다.</li> <li>Windows의 경우, 루트 Message Queue 설치 디렉토리는 Message Queue 설치 프로그램에서 설정합니다(기본값은 C:\Program Files\Sun\MessageQueue3).</li> </ul>
<b>IMQ_VARHOME</b>	<p>Message Queue 임시 또는 동적으로 작성된 구성 및 데이터 파일이 저장되는 /var 디렉토리입니다. 모든 디렉토리를 가리키는 환경 변수로 설정할 수 있습니다.</p> <ul style="list-style-type: none"> <li>Solaris에서 IMQ_VARHOME의 기본값은 /var/imq 디렉토리입니다.</li> <li>Solaris에서 Sun Java System Application Server 평가판의 IMQ_VARHOME 기본값은 IMQ_HOME/var 디렉토리입니다.</li> <li>Windows에서 IMQ_VARHOME의 기본값은 IMQ_HOME\var 디렉토리입니다.</li> <li>Windows에서 Sun Java System Application Server의 IMQ_VARHOME 기본값은 IMQ_HOME\var 디렉토리입니다.</li> <li>Linux에서 IMQ_VARHOME의 기본값은 /var/opt/sun/mq 디렉토리입니다.</li> </ul>

**표 3** Message Queue 디렉토리 변수(계속)

변수	설명
<b>IMQ_JAVAHOME</b>	<p>Message Queue 실행 파일에 필요한 Java™ runtime(JRE)의 위치를 가리키는 환경 변수입니다.</p> <ul style="list-style-type: none"> <li> <p>Solaris에서 IMQ_JAVAHOME은 다음과 같은 순서로 Java 런타임을 찾지만, 선택적으로 사용자는 필요한 JRE가 있는 적절한 위치로 값을 설정할 수 있습니다.</p> <p>Solaris 8 또는 9:</p> <pre> /usr/jdk/entsys-j2se /usr/jdk/jdk1.5.* /usr/jdk/j2sdk1.5.* /usr/j2se </pre> <p>Solaris 10:</p> <pre> /usr/jdk/entsys-j2se /usr/java /usr/j2se </pre> </li> <li> <p>Linux에서 Message Queue가 먼저 다음 순서로 Java 런타임을 찾지만, 선택적으로 사용자는 IMQ_JAVAHOME 값을 필요한 JRE가 있는 적절한 위치로 설정할 수 있습니다.</p> <pre> /usr/jdk/entsys-j2se /usr/java/jre1.5.* /usr/java/jdk1.5.* /usr/java/jre1.4.2* /usr/java/j2sdk1.4.2* </pre> </li> <li> <p>Windows에서 IMQ_JAVAHOME의 기본값은 IMQ_HOME\jre이지만, 선택적으로 사용자는 필요한 JRE가 있는 적절한 위치로 값을 설정할 수 있습니다.</p> </li> </ul>

이 설명서에서 IMQ\_HOME, IMQ\_VARHOME 및 IMQ\_JAVAHOME은 플랫폼별 환경 변수 표시나 구문(예: UNIX®의 \$IMQ\_HOME) 없이 표시됩니다. 경로 이름에는 일반적으로 UNIX 디렉토리 구분자 표시(/)를 사용합니다.

## 관련 문서

Message Queue에는 이 설명서 외에도 추가 설명서 자원이 제공됩니다.

### Message Queue 설명서 집합

Message Queue 설명서 집합을 구성하는 문서는 표 4에서 일반적으로 사용되는 순서에 따라 나열되어 있습니다.

**표 4** Message Queue 설명서 집합

문서	대상	설명
<i>Message Queue 설치 설명서</i>	개발자와 관리자	Solaris, Linux, Windows 플랫폼에서 Message Queue 소프트웨어를 설치하는 방법을 설명합니다.
<i>Message Queue 릴리스 노트</i>	개발자와 관리자	새로운 기능, 제한, 알려진 버그 및 기술 노트에 관한 설명이 포함되어 있습니다.
<i>Message Queue 기술 개요</i>	개발자와 관리자	Message Queue 개념, 기능 및 구성 요소에 대해 설명합니다.
<i>Message Queue 관리 설명서</i>	관리자 및 개발자	Message Queue 관리 도구를 사용한 관리 작업 수행 시 필요한 배경 및 정보를 제공합니다.
<i>Message Queue Developer's Guide for Java Clients</i>	개발자	JMS 및 SOAP/JAXM 사양의 Message Queue 구현을 사용하는 Java 클라이언트 프로그램을 개발하는 방법에 대한 정보를 제공합니다.
<i>Message Queue Developer's Guide for C Clients</i>	개발자	Message Queue 메시지 서비스에 대해 C 인터페이스(C-API)를 사용하는 C 클라이언트 프로그램을 개발하는 방법에 대한 정보를 제공합니다.

## 온라인 도움말

Message Queue는 Message Queue 메시지 서비스 관리 작업을 수행하는 명령줄 유틸리티를 포함합니다. 이 유틸리티에 대한 온라인 도움말을 보려면 [13장, "명령 참조"](#)를 참조하십시오.

또한 Message Queue에는 그래픽 사용자 인터페이스(GUI) 관리 도구인 관리 콘솔(imqadmin)이 포함되어 있습니다. 컨텍스트 관련 온라인 도움말은 관리 콘솔에 포함되어 있습니다.

## JavaDoc

JavaDoc 형식의 JMS 및 Message Queue API 설명서는 다음 위치에 있습니다.

플랫폼	위치
Solaris	/usr/share/javadoc/imq/index.html
Linux	/opt/sun/mq/javadoc/index.html/
Windows	IMQ_HOME/javadoc/index.html

이 설명서는 Netscape, Internet Explorer와 같은 모든 HTML 브라우저로 볼 수 있습니다. 표준 JMS API 설명서 및 Message Queue 관리 대상 객체에 대한 Message Queue별 API를 포함하며(*Message Queue Developer's Guide for Java Clients*의 3장 참조), 이는 메시지 응용 프로그램 개발자에게 도움이 됩니다.

## 클라이언트 응용 프로그램의 예

샘플 클라이언트 응용 프로그램 코드를 제공하는 여러 응용 프로그램의 예가 플랫폼에 따라 다른 디렉토리에 포함되어 있습니다([부록 A, "운영 체제별 Message Queue 데이터의 위치"](#) 참조).

해당 디렉토리 및 각 하위 디렉토리에 있는 README 파일을 참조하십시오.

## JMS(Java Message Service) 사양

다음 위치에서 JMS 사양을 확인할 수 있습니다.

<http://java.sun.com/products/jms/docs.html>

이 사양에는 클라이언트 코드 샘플이 포함되어 있습니다.

## 관련 타사 웹 사이트 참조

이 문서에 있는 타사 URL에서는 관련 추가 정보를 제공합니다.

---

<b>주</b>	<p>Sun은 이 문서에 언급된 타사 웹 사이트의 사용 가능성에 대해 책임지지 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 관련해 발생했거나 발생했다고 간주되는 손해나 손실에 대해 책임이나 의무를 지지 않습니다.</p>
----------	---

---

## 사용자 의견 환영

Sun은 본 설명서의 개선을 위해 지속적으로 노력하고 있으며 고객의 의견과 제안을 환영합니다.

사용자 의견을 나누시려면 <http://docs.sun.com>으로 이동하여 Send Comments(의견 보내기)를 누르십시오. 온라인 양식에서 문서 제목과 부품 번호를 입력하십시오. 부품 번호는 해당 설명서의 제목 페이지나 문서 맨 위에 있는 7자리 또는 9자리 숫자입니다.

사용자 의견을 제출할 때 해당 양식에 영문 설명서의 제목과 부품 번호를 입력해야 할 수도 있습니다. 본 설명서의 영문 제목과 부품 번호는 Sun Java™ System Message Queue 3 Administration Guide 2005Q1(819-0066)입니다.

사용자 의견 환영

# Message Queue 관리 소개

1장, "관리 작업 및 도구"

2장, "관리 빠른 시작"





# 관리 작업 및 도구

Sun Java™ System Message Queue 관리는 다양한 작업과 해당 작업을 수행하기 위한 다양한 도구로 구성됩니다.

이 장에서는 먼저 관리 작업을 요약 소개하고 명령줄 관리 유틸리티의 공통 기능에 중점을 두어 관리 도구를 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 34페이지의 "개발 환경의 관리 작업"
- 34페이지의 "작업 환경의 관리 작업"
- 37페이지의 "관리 도구"

## 개발 환경의 관리 작업

개발 환경에서의 작업은 Message Queue 클라이언트 응용 프로그램을 프로그래밍하는데 중점을 두며 프로그래머가 시스템을 관리하는 경우도 있습니다. Message Queue 메시지 서버는 주로 테스트 용도로 필요합니다. 개발 환경에서는 유연성이 강조되며 관리하는 일반적으로 다음과 같은 방법을 포함합니다.

- 관리는 주로 개발자가 테스트에 사용할 브로커를 시작하는 정도로 최소화합니다.
- 기본 제공 파일 기반 지속성, 파일 기반 사용자 저장소 및 파일 시스템 저장소를 사용합니다. 이러한 단순 구성은 일반적으로 개발 테스트에 적합합니다.
- 멀티 브로커 테스트에서는 마스터 브로커를 사용하지 않습니다.
- 관리자가 작성한 대상 대신 자동 작성된 대상을 사용합니다.
- 관리자 대신 클라이언트 코드에서 관리 대상 객체를 인스턴스화합니다.

## 작업 환경의 관리 작업

응용 프로그램을 안정적으로 배포하고 실행해야 하는 작업 환경에서는 관리가 훨씬 더 중요합니다. 수행할 관리 작업은 메시징 시스템 및 이 시스템이 지원해야 하는 응용 프로그램의 복잡성에 따라 달라집니다. 일반적으로 설정 작업과 유지 보수 작업으로 구분할 수 있습니다.

## 설정 작업

보통 다음 설정 작업 중 전부는 아니더라도 최소한 일부는 수행해야 합니다.

- **관리자 보안**(관리 도구 사용 보호)
  - 권한 부여: 특정 개인 또는 그룹이 관리 연결 서비스에 액세스하여 사용 불능 메시지 대기열의 메시지를 사용할 수 있도록 허용합니다(156페이지의 "[연결 서비스에 대한 액세스 제어](#)" 및 157페이지의 "[물리적 대상에 대한 액세스 제어](#)" 참조).
  - 기본 관리 사용자(admin)와 파일 기반 사용자 저장소를 사용하는 경우 사용자 비밀번호를 변경합니다(148페이지의 "[기본 관리자 비밀번호 변경](#)" 참조).
  - 그룹에 권한을 부여하는 경우 각 관리자가 해당 그룹에 속해야 합니다.
    - 파일 기반 사용자 저장소
 

파일 기반 사용자 저장소는 단일 관리자 그룹(admin)을 갖습니다. 새 관리 사용자를 만들 경우 새 사용자가 admin 그룹에 있어야 합니다.
    - LDAP 사용자 저장소
 

LDAP 서버에서 그룹을 만들거나 기존 그룹을 사용합니다. 관리 권한을 부여할 사용자가 해당 그룹의 구성원인지 확인한 다음 해당 그룹 구성원에 대한 관리 연결 권한을 부여합니다.

자세한 내용은 149페이지의 "[사용자 저장소에 LDAP 서버 사용](#)"을 참조하십시오.
- **일반 보안**(7장, "[보안 관리](#)" 참조)
  - 인증: 파일 기반 사용자 저장소에 항목을 만들거나 브로커가 기존 LDAP 사용자 저장소를 사용하도록 구성합니다.
 

(최소한 관리 기능은 비밀번호로 보호합니다.)
  - 권한 부여: 액세스 제어 등록 정보 파일에서 액세스 설정을 수정합니다.
  - 암호화: SSL 기반 연결 서비스를 설정합니다(159페이지의 "[SSL 기반 서비스 작업](#)" 참조).

- **관리 대상 객체**(8장, "관리 대상 객체 관리" 참조)
  - LDAP 객체 저장소를 구성하거나 설정합니다.
  - ConnectionFactory 및 대상 관리 대상 객체를 만듭니다.
- **브로커 클러스터**(9장, "브로커 클러스터 작업" 참조)
  - 중앙 구성 파일을 만듭니다.
  - 마스터 브로커를 사용합니다.
- **지속성**: 브로커가 플러그인 지속성을 사용하는지 기본 제공 지속성을 사용하는지를 결정하고 원하는 저장소를 설정합니다(99페이지의 "영구 저장소 설정" 참조).
- **메모리 관리**: 메시지에 할당된 메시지 수와 메모리 양이 사용 가능한 브로커 메모리 자원에 적합하도록 대상 속성을 설정합니다(329페이지의 표 15-1 참조).

## 유지 보수 작업

작업 환경에서는 Message Queue 메시지 서버 자원을 엄격하게 모니터링하고 제어해야 합니다. 응용 프로그램 성능, 신뢰성 및 보안이 매우 중요하며 Message Queue 관리 도구를 사용하여 다음과 같은 다양한 작업을 지속적으로 수행해야 합니다.

- **응용 프로그램 관리**
  - `imq.autocreate.queue` 및 `imq.autocreate.topic` 등록 정보의 값을 설정하여 브로커의 자동 작성 기능을 사용 불가능하게 합니다(313페이지의 "메시지 라우터 등록 정보" 참조).
  - 응용 프로그램을 대신하여 물리적 대상을 작성합니다(127페이지의 6장, "물리적 대상 관리" 참조).
  - 대상에 대한 사용자 액세스를 설정합니다(152페이지의 "사용자 권한 부여: 액세스 제어 등록 정보 파일" 참조).
  - 대상을 모니터링하고 관리합니다(122페이지의 "영구 가입 관리" 참조).
  - 영구 가입을 모니터링하고 관리합니다(122페이지의 "영구 가입 관리" 참조).
  - 트랜잭션을 모니터링하고 관리합니다(123페이지의 "트랜잭션 관리" 참조).

- **브로커 관리 및 조정**
  - 브로커 메트릭을 사용하여 브로커를 조정하고 재구성합니다(219페이지의 11장, "메시지 서비스 분석 및 조정" 참조).
  - 브로커 메모리 자원을 관리합니다(219페이지의 11장, "메시지 서비스 분석 및 조정" 참조).
  - 클러스터에 브로커를 추가하여 로드 균형을 조정합니다(9장, "브로커 클러스터 작업" 참조).
  - 오류가 발생한 브로커를 복구합니다(67페이지의 "대화식으로 브로커 시작" 참조).
- **응용 프로그램 관리**
  - 필요에 따라 `ConnectionFactory` 및 대상 관리 대상 객체를 추가로 작성합니다(189페이지의 "관리 대상 객체 추가 및 삭제" 참조).
  - Java 클라이언트 응용 프로그램이 올바르게 동작하도록 `ConnectionFactory` 속성 값을 조정합니다(8장, "관리 대상 객체 관리" 참조).

## 관리 도구

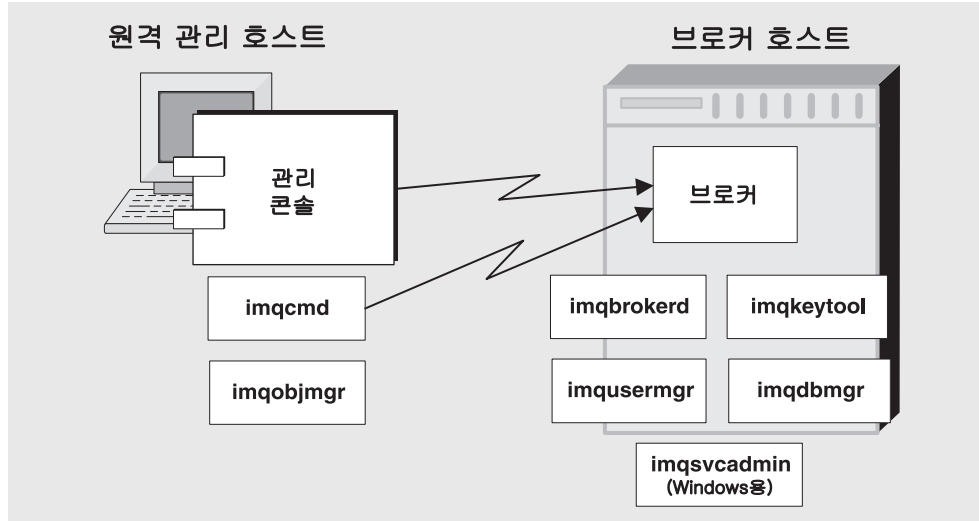
Message Queue 관리 도구는 다음 두 범주로 구분됩니다.

- 명령줄 유틸리티
- 그래픽 관리 콘솔(imqadmin)

### 명령줄 유틸리티

이 절에서는 Message Queue 관리 작업 수행 시 사용하는 명령줄 유틸리티를 소개합니다. Message Queue 유틸리티는 브로커를 시작 및 관리하고 보다 특별한 다른 관리 작업을 수행할 때 사용합니다.

그림 1-1 로컬 및 원격 관리 유틸리티



모든 Message Queue 유틸리티는 명령줄 인터페이스(CLI)에서 액세스할 수 있습니다. 이 장의 뒷부분에서 설명하겠지만 유틸리티 명령은 공통의 형식, 구문 규칙 및 옵션을 공유합니다. 명령줄 유틸리티 사용에 관한 자세한 내용은 13장, "명령 참조"를 참조하십시오.

**브로커(imqbrokerd)** 브로커 유틸리티는 브로커를 시작할 때 사용합니다. imqbrokerd 명령에 옵션을 사용하여 브로커와 클러스터의 연결 여부를 지정하고 시작할 때 브로커가 사용하는 추가 구성 정보를 지정합니다.

**명령(imqcmd)** 브로커를 시작한 후 명령 유틸리티를 사용하여 물리적 대상을 작성, 업데이트 및 삭제하고, 브로커와 해당 연결 서비스를 제어하고, 브로커의 자원을 관리합니다.

**객체 관리자(imqobjmgr)** 객체 관리자 유틸리티는 JNDI를 통해 액세스 가능한 객체 저장소에서 관리 대상 객체를 추가, 나열, 업데이트 및 삭제할 때 사용합니다. 관리 대상 객체는 JMS 클라이언트를 공급자별 이름 지정 및 구성 형식과 분리하여 공급자 독립성을 부여할 수 있습니다.

**사용자 관리자(imqusermgr)** 사용자 관리자 유틸리티는 사용자 인증 및 권한 부여에 사용하는 파일 기반 사용자 저장소를 채울 때 사용합니다.

**키 도구(imqkeytool)** 키 도구 유틸리티는 SSL 인증에 사용하는 자체 서명된 인증서를 생성할 때 사용합니다.

**데이터베이스 관리자(imqdbmgr)** 데이터베이스 관리자 유틸리티는 영구 저장소에 사용하는 JDBC 호환 데이터베이스를 작성하고 관리할 때 사용합니다.

**서비스 관리자(imqsvcaadmin)** 서비스 관리자 유틸리티는 브로커를 Windows 서비스로서 설치, 쿼리 및 제거할 때 사용합니다.

## 관리 콘솔

관리 콘솔은 명령 유틸리티(imqcmd)와 객체 관리자 유틸리티(imqobjmgr)의 두 가지 명령줄 유틸리티 기능을 결합합니다.

관리 콘솔과 이 두 가지 명령줄 유틸리티를 사용하여 브로커를 원격 관리하고 Message Queue 관리 대상 객체를 관리할 수 있습니다. 다른 명령줄 유틸리티(imqusermgr, imqdbmgr 및 imqkeytool)는 [38페이지의 그림 1-1](#)에 표시된 대로, 연결된 브로커와 동일한 호스트에서 실행되어야 합니다.

관리 콘솔에 대한 자세한 내용은 온라인 도움말을 참조하십시오. 일반적으로 특별한 작업을 수행하는 데 사용하는 명령줄 유틸리티는 "[명령줄 유틸리티](#)"에서 설명합니다.

관리 콘솔을 사용하여 다음 작업을 수행할 수 있습니다.

- 브로커에 연결하여 관리
- 브로커에 물리적 대상 만들기 및 관리
- 객체 저장소에 연결
- 객체 저장소에 관리 대상 객체 추가 및 관리

관리 콘솔로 수행할 수 없는 작업이 있으며, 그 대표적인 예로 브로커 시작, 브로커 클러스터 작성, 보다 특별한 브로커 및 물리적 대상 등록 정보 구성, 사용자 데이터베이스 관리 등이 있습니다.

[2장, "관리 빠른 시작"](#)에서 제공하는 간단한 실습 자습서를 통해 관리 콘솔 기능을 익히고 이 기능을 사용하여 기본적인 작업을 수행하는 방법을 학습할 수 있습니다.





## 관리 빠른 시작

이 빠른 시작은 Message Queue 브로커 및 객체 저장소 관리를 위해 그래픽 인터페이스인 관리 콘솔을 사용하는 기본 관리 작업에 대해 중점적으로 다룹니다. 이 장의 지침에 따라 다음 작업 방법을 학습할 수 있습니다.

- 브로커 시작
- 브로커 연결 및 관리 콘솔을 사용하여 브로커 관리
- 브로커에 물리적 대상 만들기
- 객체 저장소를 작성하고 관리 콘솔을 사용하여 연결
- 객체 저장소에 물리적 대상을 추가하고 해당 등록 정보 보기

빠른 시작은 간단한 JMS 호환 응용 프로그램인 HelloWorldMessageJNDI를 실행하는 데 필요한 물리적 대상과 관리 대상 객체를 설정합니다. 응용 프로그램은 예제 응용 프로그램 디렉토리(Solaris 및 Windows 플랫폼의 경우, demo 또는 Linux의 경우 examples), [부록 A, "운영 체제별 Message Queue 데이터의 위치"](#) 참조)의 helloworld 하위 디렉토리에 사용할 수 있습니다. 빠른 시작의 마지막 부분에서 이 응용 프로그램을 실행합니다.

이 빠른 시작은 관리 콘솔을 사용하여 기본 관리 작업을 수행하는 방법을 안내하기 위해 작성된 것입니다. 이 빠른 시작은 보조 자료이기 때문에 본 설명서를 읽고 참조해야 합니다.

일부 Message Queue 관리 작업은 관리 콘솔을 사용하여 수행할 수 없습니다. 그런 작업은 다음과 같이 명령줄 유틸리티를 사용하여 수행해야 합니다.

- 특정 물리적 대상 등록 정보 구성
- 브로커 클러스터 만들기
- 사용자 데이터베이스 관리

이러한 작업을 수행하는 방법에 대한 자세한 내용은 [6장, "물리적 대상 관리"](#), [9장, "브로커 클러스터 작업"](#) 및 [7장, "보안 관리"](#)를 참조하십시오.

## 준비

시작하기 전에 Message Queue 제품을 설치해야 합니다. 자세한 내용은 *Message Queue 설치 설명서*를 참조하십시오. 이 장은 Windows를 중심으로 구성되어 있고 UNIX® 사용자를 위한 참고 내용이 추가되어 있습니다.

이 장에서 항목1 > 항목2 > 항목3을 선택하는 것은 항목1이라는 폴다운 메뉴에서 항목2를 선택하고 항목2가 제시하는 선택 사항 중 항목3을 선택한다는 의미입니다.

## 관리 콘솔 시작

관리 콘솔을 시작하려면 다음 방법 중 하나를 사용합니다.

- Windows의 경우 시작 > 프로그램 > Sun Microsystems > Sun Java System Message Queue 3.6 > Administration을 선택합니다.

- Solaris의 경우 다음 명령을 입력합니다.

```
/usr/bin/imqadmin
```

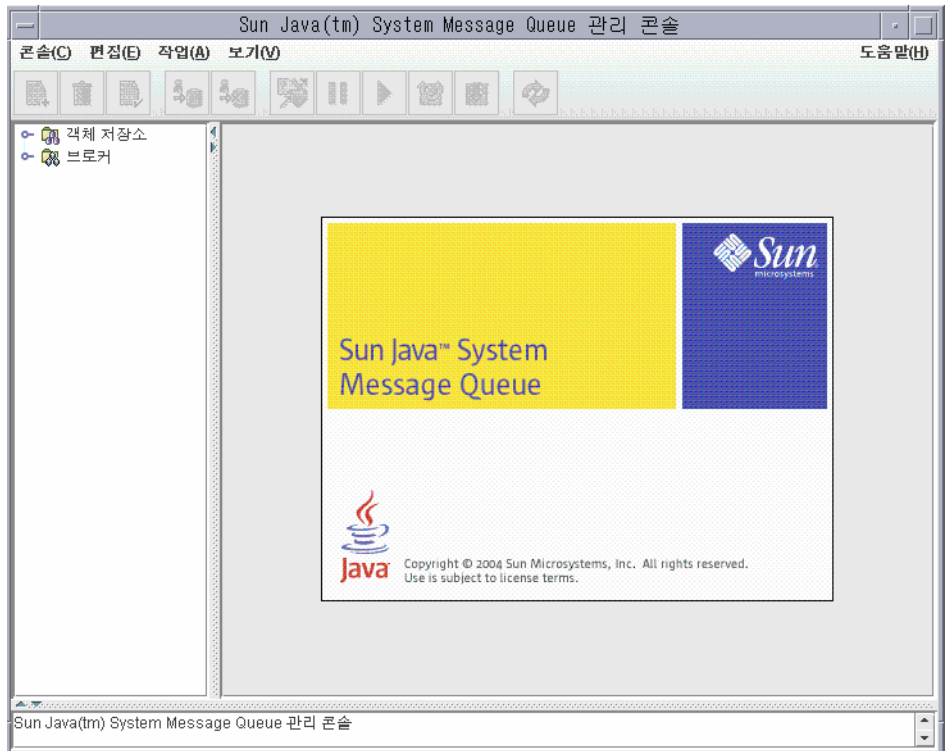
- Linux의 경우 다음 명령을 입력합니다.

```
/opt/sun/mq/bin/imqadmin
```

콘솔 창이 표시되려면 몇 초 정도 기다려야 합니다.

몇 초간 콘솔 창을 검사합니다.

콘솔에서 맨 위에 메뉴 모음, 메뉴 모음 바로 아래에 도구 모음, 왼쪽에는 탐색 창, 오른쪽에는 결과 창(현재 Sun Java System Message Queue 제품을 나타내는 그래픽 표시) 그리고 맨 아래에 상태 창이 표시됩니다.



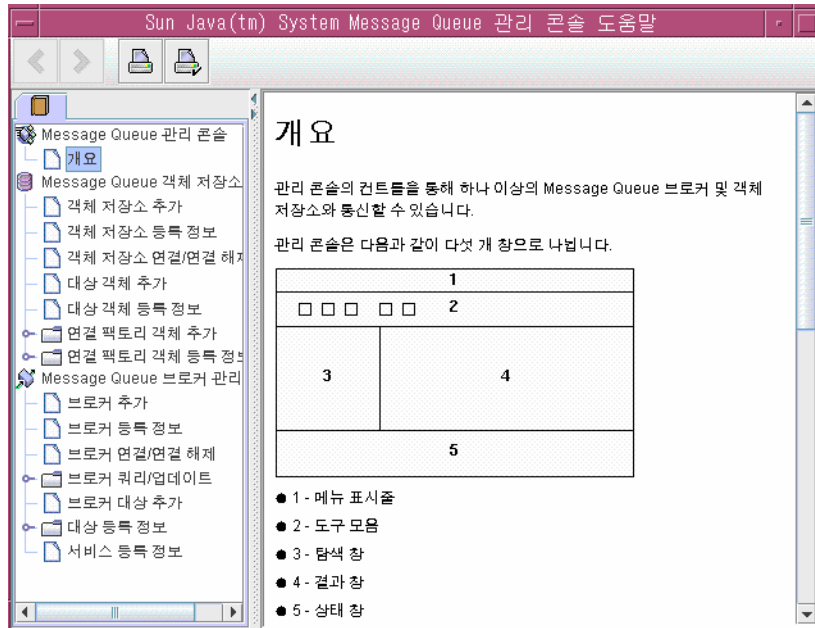
이 장만으로는 완전한 정보를 제공할 수 없으므로, 우선 관리 콘솔에 대한 도움말 정보를 얻는 방법을 알아 봅니다.

## 도움말 정보

메뉴 표시줄 맨 오른쪽에서 도움말 메뉴를 찾습니다.

▶ **관리 콘솔 도움말 정보를 표시하는 방법**

1. 도움말 풀다운 메뉴에서 개요(Overview)를 선택합니다. 도움말 창이 표시됩니다.



도움말 정보의 구성 방식에 유의하십시오. 왼쪽 탐색 창은 목차를 표시하고 오른쪽 결과 창은 탐색 창에서 선택한 항목의 내용을 표시합니다.

도움말 창의 결과 창을 확인합니다. 이 창에는 관리 콘솔의 구조 보기를 표시하고 각 콘솔 창의 사용을 보여줍니다.

2. 도움말 창의 탐색 창을 확인합니다. 이 창은 개요, 객체 저장소 관리 및 브로커 관리의 3가지 영역의 주제로 구성되어 있습니다. 각 영역은 파일과 폴더로 구성됩니다. 각 폴더는 여러 탭으로 구성된 대화 상자에 대한 도움말을 제공하고, 각 파일은 단순한 대화 상자나 탭에 대한 도움말을 제공합니다.

첫 번째 콘솔 관리 작업인 46페이지의 "브로커 추가"에서는 콘솔을 통해 관리하는 브로커에 대한 참조를 작성합니다. 그러나 시작하기 전에 온라인 도움말에서 정보를 확인하십시오.

3. 도움말 창의 탐색 창에서 브로커 추가(Add Broker) 항목을 누릅니다.  
결과 창이 변경됩니다. 이제 브로커 추가의 의미 및 브로커 추가 대화 상자의 각 필드 사용법을 설명하는 텍스트가 표시됩니다. 필드 이름은 굵은 체로 표시됩니다.
4. 도움말 텍스트를 읽어 봅니다.
5. 도움말 창을 닫습니다.

## 브로커 시작

관리 콘솔을 사용하여 브로커를 시작할 수 없습니다. 대신 다음 방법 중 하나를 사용합니다.

- Windows의 경우 시작 > 프로그램 > Sun Microsystems > Sun Java System Message Queue 3.6 > 메시지 브로커를 선택합니다.
- Solaris의 경우 다음 명령을 입력합니다.

```
/usr/bin/imbrokerd
```

- Linux의 경우 다음 명령을 입력합니다.

```
/opt/sun/mq/bin/imbrokerd
```

Windows 시작 메뉴를 사용한 경우 명령 창이 나타납니다. 명령 응답이 나타나고 다음과 같은 행을 표시하여 브로커가 준비 중임을 나타냅니다.

```
Loading persistent data...
Broker "imbroker@stan:7676 ready.
```

다시 관리 콘솔 창으로 돌아옵니다. 이제 콘솔에 브로커를 추가하여 연결할 준비가 되었습니다.

관리 콘솔에서 브로커에 대한 참조를 추가하기 전에 브로커를 시작할 필요는 없지만, 브로커에 연결하려면 먼저 브로커를 시작해야 합니다.

# 브로커 추가

브로커를 추가하면 관리 콘솔에서 해당 브로커에 대한 참조가 생성됩니다. 브로커를 추가한 후 연결할 수 있습니다.

➤ **관리 콘솔에 브로커를 추가하는 방법**

1. 탐색 창에서 브로커를 마우스 오른쪽 버튼으로 누른 다음 브로커 추가(Add Broker)를 선택합니다.
2. 브로커 레이블(Broker Label) 필드에 MyBroker라고 입력합니다.  
관리 콘솔에서 이 브로커를 식별하는 레이블을 제공합니다.

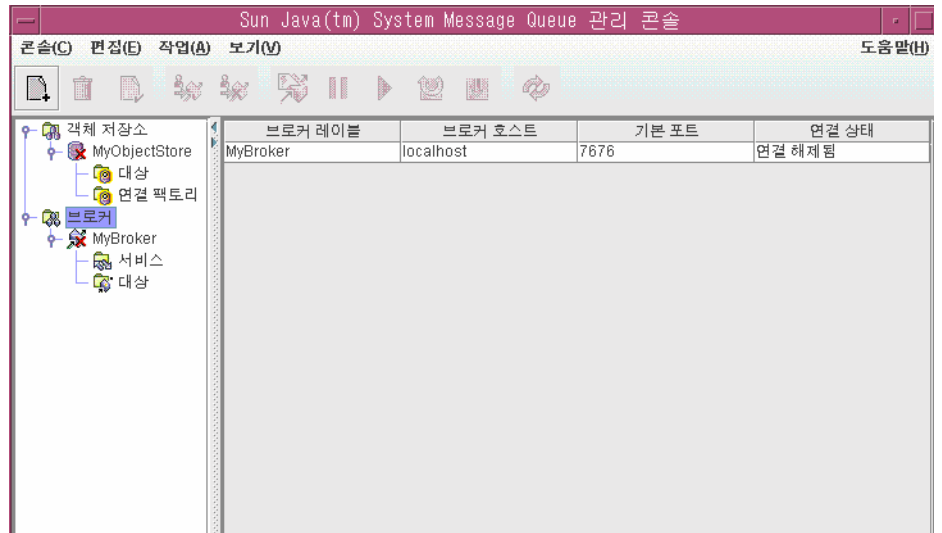


대화 상자에 지정된 기본 호스트 이름(localhost) 및 기본 포트(7676)에 유의합니다. 나중에 클라이언트가 이 브로커에 대해 연결을 설정하기 위해 사용할 연결 팩토리를 구성할 때 지정해야 하는 값입니다.

비밀번호>Password) 필드는 비워 둡니다. 연결할 때 비밀번호를 지정하면 더 안전합니다.

3. 확인을 눌러 브로커를 추가합니다.

탐색 창을 확인합니다. 방금 추가한 브로커가 브로커 아래에 표시됩니다. 브로커 아이콘 위의 빨간색 X 표시는 이 브로커가 현재 콘솔에 연결되어 있지 않다는 의미입니다.



4. MyBroker를 마우스 오른쪽 버튼으로 누르고 팝업 메뉴에서 등록 정보(Properties)를 선택합니다.

브로커 등록 정보 대화 상자가 표시됩니다. 이 대화 상자에서는 브로커를 추가할 때 지정했던 등록 정보를 업데이트할 수 있습니다.

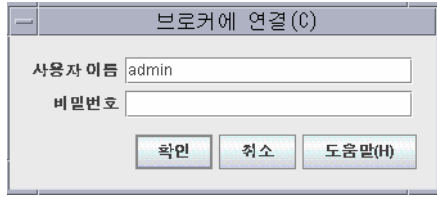
5. 취소(Cancel)를 눌러 대화 상자를 닫습니다.

# 브로커에 연결

## ▶ 브로커에 연결하는 방법

1. MyBroker를 마우스 오른쪽 버튼으로 누르고 브로커에 연결(Connect to Broker)을 선택합니다.

대화 상자가 표시되고 아이디(Username)와 비밀번호(Password)를 묻습니다.



기본적으로 관리 콘솔은 admin 아이디와 admin 비밀번호를 사용하여 브로커에 연결할 수 있습니다. 이 연습에서는 기본값을 사용합니다. 실제 환경에서는 가능한 안전한 아이디와 비밀번호를 설정해야 합니다. 자세한 내용은 [142페이지의 "사용자 인증"](#)을 참조하십시오.

2. 비밀번호(Password) 필드에 admin을 입력합니다.

admin 아이디와 정확한 비밀번호를 제시하면 관리자 권한으로 브로커에 연결됩니다.

3. 확인(OK)을 눌러 브로커에 연결합니다.

브로커에 연결한 후, 작업(Actions) 메뉴에서 브로커에 대한 정보 얻기, 브로커 중지 및 다시 시작, 브로커 종료 및 다시 시작 그리고 브로커와의 연결 해제를 선택할 수 있습니다.

## 연결 서비스 보기

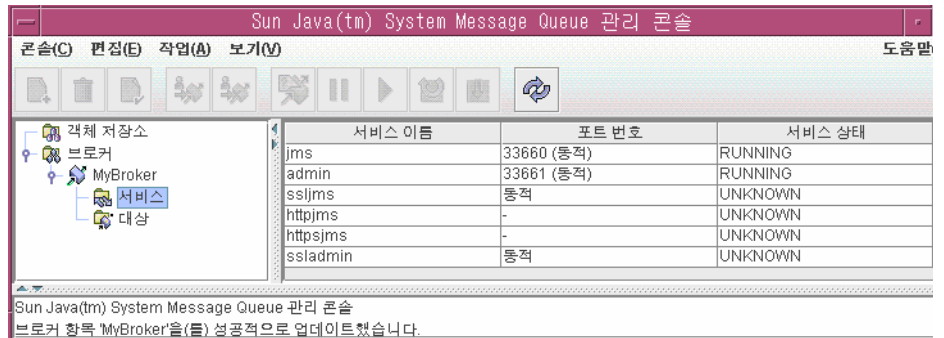
브로커는 해당 브로커에서 제공하는 연결 서비스와 지원하는 물리적 대상으로 구별됩니다.

## ▶ 사용 가능한 연결 서비스를 보는 방법

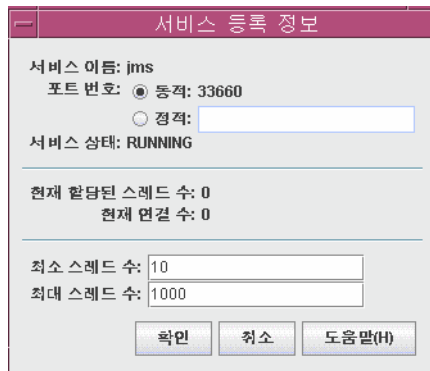
1. 탐색 창에서 서비스(Services)를 선택합니다.

사용 가능한 서비스가 결과 창에 표시됩니다. 각 서비스마다 이름, 포트 번호 및 상태가 제시됩니다.





2. 결과 창에서 jms 서비스를 눌러 선택합니다.
3. 작업(Actions) 풀다운 메뉴를 누르고 강조 표시된 항목을 확인합니다.  
jms 서비스를 중지하거나 그 등록 정보를 확인하고 업데이트할 수 있습니다.
4. 작업(Actions) 메뉴에서 등록 정보(Properties)를 선택합니다.  
서비스 등록 정보 대화 상자를 사용하여 서비스에 정적 포트 번호를 할당하고 이 서비스에 할당된 최소 및 최대 스레드 수를 변경할 수 있습니다.



5. 확인(OK)이나 취소(Cancel)를 눌러 등록 정보(Properties) 대화 상자를 닫습니다.
6. 결과 창에서 관리 서비스를 선택합니다.
7. 작업(Actions) 풀다운 메뉴를 누릅니다.

이 서비스는 일시 중지할 수 없습니다(일시 중지 항목 사용 불가). 관리 서비스는 관리자만 브로커에 연결하는 링크입니다. 이 링크를 일시 중지하면 관리자는 더 이상 브로커에 액세스할 수 없습니다.

8. 작업(Actions) > 등록 정보(Properties)를 선택하여 관리 서비스의 등록 정보를 확인합니다.
9. 작업을 마쳤으면 확인(OK)이나 취소(Cancel)를 누릅니다.

## 브로커에 물리적 대상 추가

기본적으로 브로커에 대한 물리적 대상 자동 생성이 활성화됩니다. 브로커는 자동 생성을 사용하여 물리적 대상을 동적으로 만들 수 있습니다.

따라서 개발 환경에서는 클라이언트 코드를 테스트하기 위해 물리적 대상을 명시적으로 만들 필요가 없습니다.

그러나 작업 설정에서는 물리적 대상을 명시적으로 만드는 것이 좋습니다. 그렇게 하면 관리자가 브로커에서 사용 중인 물리적 대상을 완전히 파악할 수 있습니다.

이제 브로커에 물리적 대상을 추가합니다. 대상에 지정한 이름을 적어 두십시오. 나중에 이 물리적 대상에 해당되는 관리 대상 객체를 작성할 때 이 이름이 필요합니다.

### ▶ 브로커에 대기열 대상을 추가하는 방법

1. MyBroker의 대상(Destinations) 노드를 마우스 오른쪽 버튼으로 누르고 브로커 대상 추가(Add Broker Destination)를 선택합니다.

다음 대화 상자가 표시됩니다.

2. 대상 이름(Destination Name) 필드에 MyQueueDest를 입력합니다.
3. 대기열(Queue) 라디오 버튼이 선택되어 있지 않으면 선택합니다.
4. 확인(OK)을 눌러 물리적 대상을 추가합니다.  
이제 결과 창에 물리적 대상이 표시됩니다.

## 물리적 대상 관리

브로커에 물리적 대상을 추가했다면 아래 절차에 설명된 대로 다음 작업을 수행할 수 있습니다.

- 물리적 대상의 등록 정보 확인 및 업데이트
- 물리적 대상의 메시지 제거
- 물리적 대상 삭제

▶ 물리적 대상의 등록 정보를 보는 방법

1. MyBroker의 대상 노드를 선택합니다.

결과 패널에 MyQueueDest와 mq.sys.dmq라는 두 개의 물리적 대상이 표시됩니다. mq.sys.dmq 대상은 시스템에서 생성된 대기열이며 브로커에 대해 만료되었거나 거부된 메시지를 저장합니다. 지금은 이 사용 불가능 메시지 대기열을 무시합니다.

2. 결과 창에서 MyQueueDest를 선택합니다.
3. 작업(Actions) > 등록 정보(Properties)를 선택합니다.

다음 대화 상자가 표시됩니다.

The screenshot shows a window titled "브로커 대상 등록 정보" (Broker Target Registration Information). It has two tabs: "기본" (Basic) and "영구 가입" (Permanent Registration). The "기본" tab is active and displays the following information:

- 대상 이름: MyQueueDest
- 대상 유형: 대기열
- 대상 상태: RUNNING

---

Current statistics:

- 현재 메시지 수: 0
- 현재 메시지 전체 크기: 0 바이트
- 현재 생성자 수: 0
- 현재 활성 사용자 수: 0
- 현재 백업 사용자 수: 0

---

Configuration options:

- 최대 메시지 수:  제한 없음
- 최대 메시지 전체 크기:  제한 없음
- 메시지당 최대 크기:  제한 없음
- 최대 생성자 수:  제한 없음,  100
- 최대 활성 사용자 수:  제한 없음,  1
- 최대 백업 사용자 수:  제한 없음,  0

---

Reject Newest:  ▼

사용 불가능 메시지 대기열 사용:

Buttons at the bottom: 확인 (OK), 취소 (Cancel), 도움말(H) (Help).

대화 상자에는 대기열에 대해 현재 상태 정보와 변경 가능한 일부 등록 정보가 표시됩니다.

4. 취소(Cancel)를 눌러 대화 상자를 닫습니다.

▶ 물리적 대상에서 메시지를 제거하는 방법

1. 결과 창에서 물리적 대상을 선택합니다.
2. 작업(Actions) > 메시지 제거(Purge Messages)를 선택합니다.

확인 대화 상자가 표시됩니다.

메시지 제거 기능은 메시지를 삭제하고 빈 대상을 남겨 둡니다.

▶ 대상을 삭제하는 방법

1. 결과 창에서 물리적 대상을 선택합니다.
2. 편집(Edit) > 삭제(Delete)를 선택합니다.

확인 대화 상자가 표시됩니다.

---

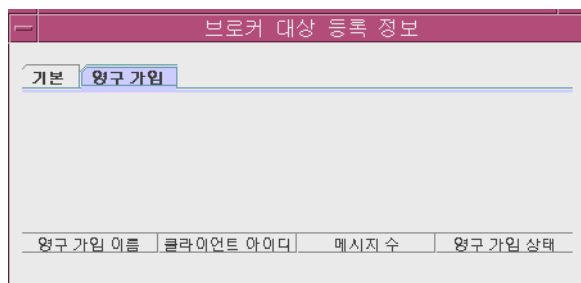
**주** MyQueueDest 대기열 대상은 삭제하지 마십시오.

---

물리적 대상 삭제 기능은 대상에서 메시지를 제거하고 대상도 제거합니다.

## 항목 정보 얻기

브로커 주제 대상 등록 정보 대화 상자에는 영구 가입 정보를 나열하는 추가 탭이 있습니다. 대기열의 경우 이 탭이 비활성화됩니다.



이 대화 상자에서는 다음 작업을 할 수 있습니다.

- 영구 가입 제거, 영구 가입과 관련된 모든 메시지 제거
- 영구 가입 삭제, 영구 가입과 관련된 모든 메시지 및 해당 영구 가입 제거

## 객체 저장소 작업

객체 저장소는 Message Queue 관리 대상 객체를 저장하는 데 사용됩니다. 이러한 관리 대상 객체는 클라이언트 응용 프로그램에 사용되는 객체에 대한 Message Queue 관련 구현 및 구성 정보를 캡슐화합니다. 객체 저장소는 LDAP 디렉토리 서버이거나 파일 시스템 저장소(파일 시스템의 디렉토리)일 수 있습니다.

관리 대상 객체는 클라이언트 코드 내에서 인스턴스화하고 구성할 수 있습니다. 하지만 클라이언트 응용 프로그램이 JNDI를 사용하여 액세스할 수 있는 객체 저장소에서 관리자가 이러한 객체를 생성, 구성 및 저장하는 것이 좋습니다. 이 경우 클라이언트 코드는 공급자에 영향을 받지 않을 수 있습니다.

관리 콘솔을 사용하여 객체 저장소를 작성할 수 없습니다. 다음 절에 설명된 대로 미리 작성해두어야 합니다.

## 객체 저장소 추가

객체 저장소를 추가하면 관리 콘솔의 기존 객체 저장소에 대해 참조가 생성됩니다. 이 참조는 콘솔을 종료하고 다시 시작하는 경우에도 유지됩니다.

### ▶ 파일 시스템 객체 저장소를 추가하는 방법

1. 아직 C 드라이브에 Temp라는 이름의 폴더가 없다면 지금 만듭니다.

이 장에서 사용하는 샘플 응용 프로그램에서는 객체 저장소가 C 드라이브의 Temp 폴더라고 가정합니다. 일반적으로 파일 시스템 객체 저장소는 어떤 드라이브의 어떤 디렉토리도 가능합니다.

**비 Windows:** 이미 존재하는 /tmp 디렉토리를 사용할 수 있습니다.

2. 객체 저장소(Object Stores)를 마우스 오른쪽 버튼으로 눌러 객체 저장소 추가(Add Object Store)를 선택합니다.

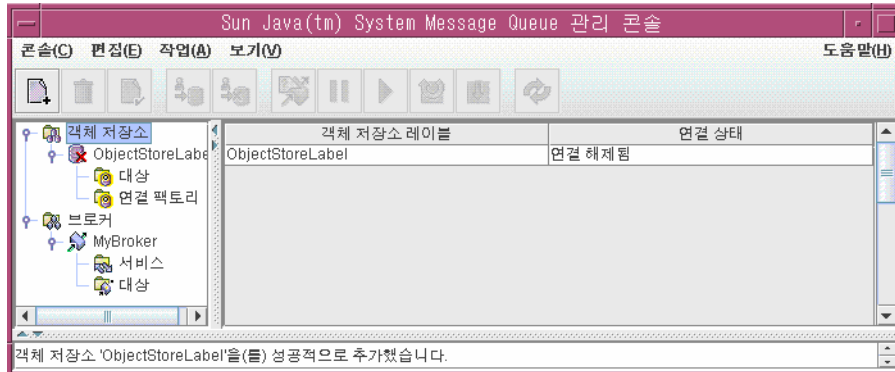
다음 대화 상자가 표시됩니다.

3. 객체 저장소 레이블(Object Store Label)이라는 이름의 필드에 MyObjectStore라고 입력합니다.  
 이는 관리 콘솔에서 객체 저장소를 표시할 레이블을 제공합니다.  
 다음 단계에서는 JNDI 이름/값 쌍을 입력해야 합니다. 이 쌍은 JMS 호환 응용 프로그램에서 관리 대상 객체를 조회할 때 사용됩니다.
4. 이름(Name) 드롭다운 목록에서 `java.naming.factory.initial`을 선택합니다.  
 이 등록 정보에서는 어떤 JNDI 서비스 공급자를 사용할 것인지 지정할 수 있습니다. 파일 시스템 서비스 제공자나 LDAP 서비스 제공자를 예로 들 수 있습니다.
5. 값(Value) 필드에 다음을 입력합니다.  

```
com.sun.jndi.fscontext.RefFSContextFactory
```

 이는 파일 시스템 저장소를 사용한다는 의미입니다 (LDAP 저장소의 경우 `com.sun.jndi.ldap.LdapCtxFactory`를 지정).  
 작업 환경에서는 LDAP 디렉토리 서버를 객체 저장소로 사용할 것입니다. 서버 설정 및 JNDI 조회 수행에 대한 정보는 [174페이지](#)의 "LDAP 서버 객체 저장소"를 참조하십시오.
6. 추가(Add) 버튼을 누릅니다.  
 등록 정보 및 그 값은 이제 등록 정보 요약 창에 표시됩니다.

7. 이름(Name) 드롭다운 목록에서 `java.naming.provider.url`을 선택합니다.  
이 등록 정보에서 객체 저장소의 정확한 위치를 지정할 수 있습니다. 파일 시스템 유형의 객체 저장소의 경우 이것이 기존 디렉토리의 이름이 됩니다.
8. 값(Value) 필드에 다음을 입력합니다.  
`file:///C:/Temp`  
(Solaris 및 Linux의 경우 `file:///tmp`)
9. 추가(Add) 버튼을 누릅니다.  
두 등록 정보와 그 값이 이제 등록 정보 요약 창에 표시됩니다. LDAP 서버를 사용하는 경우, 인증 정보를 지정해야 하는 경우가 있습니다. 파일 시스템 저장소에 대해서는 지정할 필요가 없습니다.
10. 확인(OK)을 눌러 객체 저장소를 추가합니다.
11. 탐색 창에서 `MyObjectStore` 노드가 선택되지 않았다면 지금 선택합니다.  
이제 관리 콘솔(Administration Console)은 다음과 같이 표시됩니다.



객체 저장소는 탐색 창에, 그 내용, 대상(Destinations) 및 연결 팩토리(Connection Factories)는 결과 창에 표시됩니다. 아직 객체 저장소에 관리 대상 객체를 추가하지 않았으며, 이는 결과 창의 수 열에서 확인할 수 있습니다.

탐색 창의 객체 저장소 아이콘에 빨간색 X가 표시됩니다. 이 표시가 나타나면 연결이 끊겼다는 의미입니다. 객체 저장소를 사용하려면 먼저 연결해야 합니다.



## 객체 저장소 등록 정보 확인

관리 콘솔(Administration Console)이 객체 저장소에 연결되지 않은 상태에서 객체 저장소의 일부 등록 정보를 확인하고 변경할 수 있습니다.

### ▶ 객체 저장소 등록 정보를 표시하는 방법

1. 탐색 창에서 MyObjectStore를 마우스 오른쪽 버튼으로 누릅니다.
2. 팝업 메뉴에서 등록 정보(Properties)를 선택합니다.  
객체 저장소를 추가할 때 지정한 모든 등록 정보를 보여주는 대화 상자가 표시됩니다. 이 등록 정보를 변경하고 확인(OK)을 누르면 기존 정보를 업데이트합니다.
3. 확인(OK) 또는 취소(Cancel)를 눌러 대화 상자를 닫습니다.

## 객체 저장소에 연결

객체 저장소에 객체를 추가하려면 먼저 객체 저장소에 연결해야 합니다.

### ▶ 객체 저장소에 연결하는 방법

1. 탐색 창에서 MyObjectStore를 마우스 오른쪽 버튼으로 누릅니다.
2. 팝업 메뉴에서 객체 저장소에 연결(Connect to Object Store)을 선택합니다.  
객체 저장소 아이콘에 더 이상 X 표시가 없습니다. 이제 객체 저장소에 객체, 연결 팩토리 및 대상을 추가할 수 있습니다.

## 연결 팩토리 관리 대상 객체 추가

관리 콘솔을 사용하여 연결 팩토리를 작성하고 구성할 수 있습니다. 연결 팩토리는 클라이언트 코드가 브로커에 연결할 때 사용합니다. 연결 팩토리 구성을 통해 이 연결 팩토리를 사용하여 작성되는 연결의 동작을 제어할 수 있습니다.

연결 팩토리 구성에 대한 정보는 온라인 도움말 및 *Message Queue Developer's Guide for Java Clients*를 참조하십시오.

---

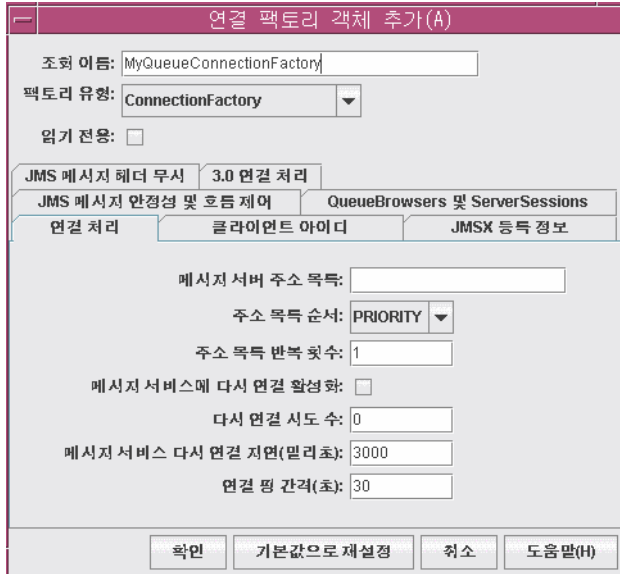
**주** 관리 콘솔은 Message Queue 관리 대상 객체만 나열하고 표시합니다. 추가할 관리 대상 객체와 동일한 조회 이름을 가진 비 Message Queue 객체를 객체 저장소가 포함하는 경우 이 객체를 추가하려고 하면 오류 메시지가 표시됩니다.

---

▶ 객체 저장소에 연결 팩토리를 추가하는 방법

1. 아직 MyObjectStore에 연결되지 않았다면 지금 연결합니다(57페이지의 "객체 저장소에 연결" 참조).
2. 연결 팩토리(Connection Factories) 노드를 마우스 오른쪽 버튼으로 누르고 연결 팩토리 객체 추가(Add Connection Factory Object)를 선택합니다.

연결 팩토리 객체 추가(Add Connection Factory Object) 대화 상자가 표시됩니다.



3. 조회 이름(Lookup Name) 필드에 "MyQueueConnectionFactory"라는 이름을 입력합니다.

이 이름은 HelloWorldMessageJNDI.java의 다음 행에서 알 수 있듯이 클라이언트 코드가 연결 팩토리를 조회할 때 사용하는 이름입니다.

```
qcf= (javax.jms.QueueConnectionFactory)
    ctx.lookup ("MyQueueConnectionFactory")
```

4. 풀다운 메뉴에서 QueueConnectionFactory를 선택하여 연결 팩토리 유형을 지정합니다.
5. 연결 처리(Connection Handling) 탭을 누릅니다.

6. 메시지 서버 주소 목록 필드에는 일반적으로 클라이언트가 연결할 브로커의 주소를 입력합니다. 다음은 이 필드의 예입니다.  
mq://localhost:7676/jms  
기본적으로 연결 팩토리가 로컬 호스트의 포트 7676에서 실행 중인 브로커에 연결하도록 구성되므로(이 빠른 시작 예에서 사용하는 구성) 값을 입력할 필요가 없습니다.
7. 이 대화 상자에 있는 탭을 차례로 눌러 연결 팩토리에 대해 구성 가능한 정보 종류를 확인합니다. 연결 팩토리 객체 추가(Add Connection Factory Object) 대화 상자의 오른쪽 아래에 있는 도움말(Help) 버튼을 사용하여 각 탭에 대한 정보를 얻을 수 있습니다. 지금은 기본값을 바꾸지 마십시오.
8. 확인(OK)을 눌러 대기열 연결 팩토리를 만듭니다.
9. 결과 창을 확인합니다. 새로 만든 연결 팩토리의 조회 이름 및 유형이 표시됩니다.

## 대상 객체 추가

대상 관리 대상 객체는 브로커의 물리적 대상에 연결되어 해당 대상을 가리킵니다. 대상 관리 대상 객체를 사용하면 클라이언트가 공급자별 대상 이름과 구성에 관계없이 물리적 대상을 조회하여 찾을 수 있습니다.

클라이언트는 메시지를 보낼 때 대상 관리 대상 객체를 조회 또는 인스턴스화하고 이를 JMS API의 `send()` 메소드에서 참조합니다. 그런 다음 브로커는 다음과 같이 관리 대상 객체에 연결되는 물리적 대상에게 메시지를 전달합니다.

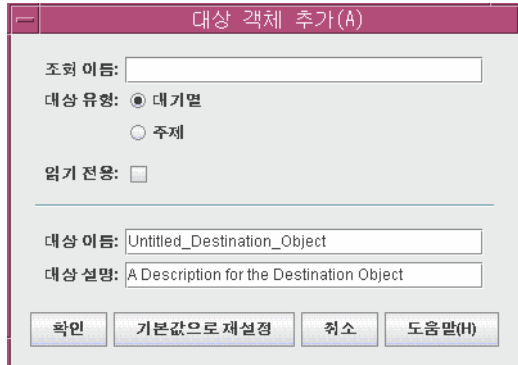
- 관리 대상 객체에 해당되는 물리적 대상이 작성되어 있다면, 브로커는 이 물리적 대상에게 메시지를 전달합니다.
- 물리적 대상을 만들지 않았고 물리적 대상의 자동 생성 기능이 사용 가능한 상태라면 브로커 스스로 물리적 대상을 작성하고 그 대상에게 메시지를 전달합니다.
- 물리적 대상을 만들지 않았고 물리적 대상의 자동 생성 기능도 사용 불가 상태라면 브로커는 물리적 대상을 생성할 수도, 메시지를 전달할 수도 없습니다.

빠른 시작의 다음 부분에서는 이미 추가해 놓은 물리적 대상에 해당되는 관리 대상 객체를 추가합니다.

▶ 객체 저장소에 대상을 추가하는 방법

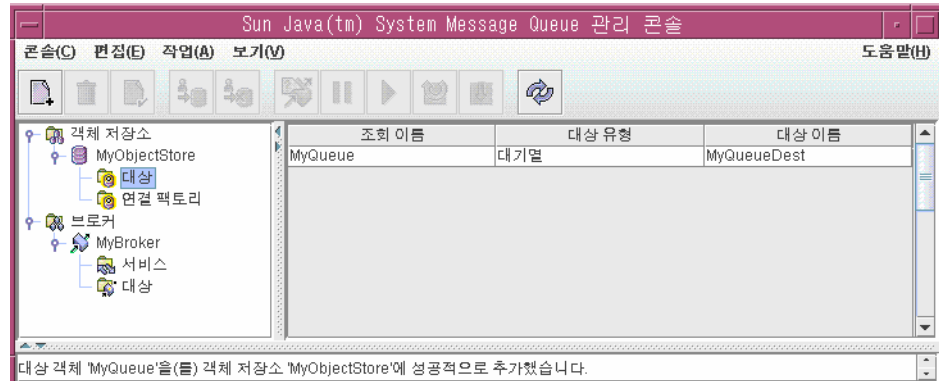
1. 탐색 창의 대상(Destinations) 노드(MyObjectStore 노드 아래)를 마우스 오른쪽 버튼으로 누릅니다.
2. 대상 객체 추가(Add Destination Object)를 선택합니다.

관리 콘솔(Administration Console)은 객체에 대한 정보를 지정할 때 사용하는 대상 객체 추가(Add Destination Object) 대화 상자를 표시합니다.



3. 조회 이름(Lookup Name) 필드에 "MyQueue"라고 입력합니다.  
이 조회 이름은 JNDI 조회 호출을 사용하여 객체를 찾을 때 사용됩니다. 이 샘플 응용 프로그램에서는 다음과 같이 호출합니다.  

```
queue=(javax.jms.Queue) ctx.lookup("MyQueue?");
```
4. 대상 유형(Destination Type)의 대기열(Queue) 라디오 버튼을 선택합니다.
5. 대상 이름(Destination Name) 필드에 MyQueueDest를 입력합니다.  
이것은 브로커에 물리적 대상을 추가했을 때 지정한 이름입니다(50페이지의 "브로커에 물리적 대상 추가" 참조).
6. 확인(OK)을 누릅니다.
7. 탐색 창에서 대상을 선택하고 방금 추가한 대기열 대상 관리 대상 객체에 대한 정보가 결과 창에 어떻게 표시되는지 확인합니다.



## 관리 대상 객체 등록 정보 보기

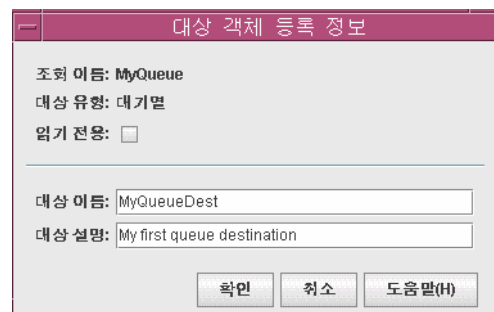
관리 대상 객체의 등록 정보를 보거나 업데이트하려면 탐색 창에서 대상(Destinations)이나 연결 팩토리(Connection Factories)를 선택하고 결과 창에서 특정 객체를 선택한 다음 작업(Actions) > 등록 정보(Properties)를 선택합니다.

### ▶ 대상 객체의 등록 정보를 확인 또는 업데이트하는 방법

1. 탐색 창에서 MyObjectStore의 대상(Destinations) 노드를 선택합니다.
2. 결과 창에서 MyQueue를 선택합니다.
3. 작업(Actions) > 등록 정보(Properties)를 선택하여 대상 객체 등록 정보(Destination Object Properties) 대화 상자를 확인합니다.

변경 가능한 값은 대상 이름과 설명뿐입니다. 조회 이름을 변경하려면 객체를 삭제한 다음 원하는 조회 이름으로 새 대기열 관리 대상 객체를 추가해야 합니다.

4. 취소(Cancel)를 눌러 대화 상자를 닫습니다.



## 콘솔 정보 업데이트

객체 저장소나 브라우저로 작업하는 모든 경우에 보기 > 새로 고침을 선택하면 어떤 요소나 요소 그룹의 시각적 표시를 업데이트할 수 있습니다.

## 샘플 응용 프로그램 실행

이 빠른 시작에 사용할 샘플 응용 프로그램 HelloWorldMessageJNDI가 제공됩니다. 이 응용 프로그램은 사용자가 만든 물리적 대상과 관리 대상 객체를 사용합니다.

- MyQueueDest라는 이름의 대기열 물리적 대상
- JNDI 조회 이름이 각각 MyQueueConnectionFactory 및 MyQueue인 대기열 연결 팩토리 관리 대상 객체와 대기열 관리 대상 객체

이 코드는 간단한 대기열 발신기 및 수신기를 만들고 "Hello World" 메시지를 보내고 받습니다.

### ▶ HelloWorldMessageJNDI 응용 프로그램을 실행하는 방법

1. HelloWorldMessageJNDI 응용 프로그램을 포함하는 디렉토리가 현재 디렉토리가 되게 합니다. 예를 들면 다음과 같습니다.

```
cd %IMQ_HOME%\demo\helloworld\helloworldmessagejndi(Windows)
```

```
cd /usr/demo/imq/helloworld/helloworldmessagejndi(Solaris)
```

```
cd /opt/sun/mq/examples/helloworld/helloworldmessagejndi(Linux)
```

HelloWorldMessageJNDI.class 파일이 있는지 확인합니다 (응용 프로그램을 변경할 경우, *Message Queue Developer's Guide for C Clients*의 Quick Start Tutorial에서 소개하는 클라이언트 응용 프로그램 컴파일 지침을 사용하여 다시 컴파일해야 함).

CLASSPATH 변수가 HelloWorldMessageJNDI.class 파일을 비롯하여 Message Queue 제품에 포함된 jms.jar, imq.jar 및 fscontext.jar 파일이 들어 있는 현재 디렉토리를 포함하도록 설정합니다. CLASSPATH 설정 지침은 *Message Queue Developer's Guide for Java Clients*를 참조하십시오.

JNDI jar 파일(jndi.jar)은 JDK 1.4와 함께 제공됩니다. 이 JDK를 사용하는 경우 jndi.jar 파일을 CLASSPATH 설정에 추가할 필요가 없습니다. 이전 버전 JDK를 사용하는 경우 jndi.jar을 CLASSPATH에 포함시켜야 합니다. 자세한 내용은 *Message Queue Developer's Guide for Java Clients*를 참조하십시오.

2. 응용 프로그램을 실행하기 전에 HelloWorldMessageJNDI.java 소스 파일을 열고 확인합니다. 소스는 간단하지만 확실하게 기록되어 있으며 이 자습서에서 만든 관리 대상 객체 및 대상을 어떻게 사용하는지에 대해 명확하게 설명합니다.

3. 아래 명령 중 하나를 실행하여 HelloWorldMessageJNDI 응용 프로그램을 실행합니다.

```
java HelloWorldMessageJNDI (Windows)
```

```
% java HelloWorldMessageJNDI file:///tmp (Solaris 및 Linux)
```

응용 프로그램이 성공적으로 실행되면 다음과 같이 출력됩니다.

```
java HelloWorldMessageJNDI
Using file:///C:/Temp for Context.PROVIDER_URL

Looking up Queue Connection Factory object with lookup name:
MyQueueConnectionFactory
Queue Connection Factory object found.
Looking up Queue object with lookup name: MyQueue
Queue object found.

Creating connection to broker.
Connection to broker created.

Publishing a message to Queue: MyQueueDest
Received the following message: Hello World
```

샘플 응용 프로그램 실행



## 관리 작업

- 3장, "브로커 및 클라이언트 시작"
- 4장, "브로커 구성"
- 5장, "브로커 관리"
- 6장, "물리적 대상 관리"
- 7장, "보안 관리"
- 8장, "관리 대상 객체 관리"
- 9장, "브로커 클러스터 작업"
- 10장, "메시지 서버 모니터링"
- 11장, "메시지 서비스 분석 및 조정"
- 12장, "문제 해결"



# 브로커 및 클라이언트 시작

Sun Java™ System Message Queue를 설치하고 몇 가지 준비 단계를 수행한 후 브로커와 클라이언트를 시작할 수 있습니다.

이 장은 다음 내용으로 구성되어 있습니다.

- 66페이지의 "시스템 자원 준비"
- 67페이지의 "대화식으로 브로커 시작"
- 68페이지의 "자동으로 브로커 시작"
- 71페이지의 "Message Queue 클라이언트 시작"
- 72페이지의 "브로커 인스턴스 제거"

브로커 인스턴스의 구성은 일련의 구성 파일 및 `imqbrokerd` 명령과 함께 전달되는 옵션에 따라 결정되며, 옵션은 구성 파일의 해당 등록 정보를 대체합니다. 브로커 구성에 대한 자세한 내용은 73페이지의 4장, "브로커 구성"을 참조하십시오.

## 시스템 자원 준비

브로커를 시작하기 전에 시스템 클럭 동기화 및 파일 설명자 제한 설정(Solaris 또는 Linux)의 두 가지 시스템 수준 작업을 수행해야 합니다. 다음 절에서는 이러한 작업에 대해서 설명합니다.

### 시스템 클럭 동기화

브로커 또는 클라이언트를 시작하기 전에 Message Queue 시스템과 상호 작용하는 모든 호스트에서 클럭을 동기화해야 합니다. 동기화는 메시지 만료(TimeToLive)를 사용할 경우에 특히 중요합니다. 동기화되지 않은 클럭의 타임스탬프는 TimeToLive 기능이 예상한 대로 작동하지 않게 하여 메시지 전달을 막을 수 있습니다. 브로커 클러스터에도 동기화가 중요합니다.

SNTP(Simple Network Time Protocol)와 같은 시간 동기화 프로토콜을 실행하도록 시스템을 구성합니다. 시간 동기화는 일반적으로 xntpd 데몬(Solaris 및 Linux)과 W32Time 시간 서비스(Windows)에 의해 지원됩니다. 이러한 서비스 구성에 대한 자세한 내용은 운영 체제 설명서를 참조하십시오.

브로커가 실행된 이후에는 시스템 클럭을 뒤로 설정하지 마십시오.

### 파일 설명자 제한 설정(Solaris 또는 Linux)

Solaris과 Linux 플랫폼에서는 클라이언트 또는 브로커가 실행 중인 셸에 프로세스가 사용할 수 있는 파일 설명자 수에 대한 소프트 한계가 있습니다. Message Queue 시스템에서는 각 클라이언트가 만드는 연결 또는 각 브로커가 받아들이는 연결에 이 파일 설명자 중 하나를 사용합니다. 지속성 메시지를 갖는 각 물리적 대상 역시 파일 설명자를 사용합니다.

결과적으로 이러한 요소가 연결 수를 제한합니다. 따라서 파일 설명자 제한을 변경하지 않으면 Solaris에서는 256개, Linux에서는 1024개를 초과하는 연결에서 브로커나 클라이언트를 실행할 수 없습니다(지속성을 위해 파일 설명자를 사용하기 때문에 연결 제한은 실제로 더 낮음).

파일 설명자 제한을 변경하려면 ulimit 설명서 페이지를 참조하십시오. 클라이언트 또는 브로커가 실행될 각 셸에서 한계를 변경해야 합니다.

## 대화식으로 브로커 시작

명령줄에서 `imqbrokerd` 명령을 사용하여 브로커를 대화식으로 시작할 수 있습니다. Windows에서는 시작 메뉴에서 브로커를 시작할 수도 있습니다. 관리 콘솔(`imqadmin`) 또는 명령 유틸리티(`imqcmd`)를 사용하여 브로커를 시작할 수는 없습니다. 이러한 도구를 사용하려면 브로커가 이미 실행되고 있어야 합니다.

Solaris 및 Linux 플랫폼의 경우 브로커 인스턴스는 처음에 시작한 사용자가 항상 시작해야 합니다. 브로커 인스턴스가 처음 시작되면 Message Queue는 해당 사용자의 `umask`를 사용하여 구성 정보와 지속성 데이터가 들어 있는 브로커 인스턴스 디렉토리에 대한 사용 권한을 설정합니다. 각 브로커 인스턴스는 자체 구성 등록 정보 및 파일 기반 메시지 저장소를 가지고 있습니다.

브로커 인스턴스는 기본적으로 인스턴스 이름이 `imqbroker`입니다. 이 이름과 기본 구성을 사용하여 명령줄에서 브로커를 시작하려면 다음 명령을 사용하기만 하면 됩니다.

```
imqbrokerd
```

그러면 기본 포트 7676에 포트 매핑이 있는 로컬 시스템에서 `imqbroker`라는 브로커 인스턴스를 시작합니다.

기본값이 아닌 인스턴스 이름을 지정하려면 `-name` 옵션을 `imqbrokerd` 명령에 사용합니다. 다음 명령은 인스턴스 이름이 `myBroker`인 브로커를 시작합니다.

```
imqbrokerd -name myBroker
```

`imqbrokerd` 명령줄에서 다른 옵션을 사용하여 다양한 브로커 작업을 제어할 수 있습니다. 다음 예에서는 `-tty` 옵션을 사용하여 명령 창에 오류 및 경고를 보냅니다(표준 출력).

```
imqbrokerd -name myBroker -tty
```

명령줄에서 `-D` 옵션을 사용하여 브로커의 인스턴스 구성 파일(`config.properties`)에 지정된 등록 정보 값을 무시할 수도 있습니다. 이 예에서는 `imq.jms.max_threads` 등록 정보를 사용하여 `jms` 연결 서비스에 사용할 수 있는 최대 스레드 수를 2000으로 늘립니다.

```
imqbrokerd -name myBroker -Dimq.jms.max_threads=2000
```

`imqbrokerd` 명령의 구문, 하위 명령 및 옵션에 대한 자세한 내용은 [13장, "명령 참조"](#)를 참조하십시오. 이 정보에 대한 빠른 요약은 보려면 다음을 입력합니다.

```
imqbrokerd -help
```

---

**주** Sun Java System Message Queue 플랫폼판 사용권이 있는 경우 `imqbrokerd` 명령의 `-license` 옵션을 사용하여 시험 플랫폼판 사용권을 활성화하여 플랫폼판 기능을 90일 동안 사용할 수 있습니다. `try`를 라이선스 이름으로 지정합니다.

```
imqbrokerd -license try
```

이 옵션은 브로커를 시작할 때마다 사용해야 하며 그렇지 않으면 브로커의 기본값이 표준 플랫폼판 사용권으로 돌아갑니다.

---

## 자동으로 브로커 시작

브로커를 명령줄에서 명시적으로 시작하는 대신 시스템을 시작할 때 자동으로 시작하도록 설정할 수 있습니다. 이 방법은 브로커를 실행 중인 플랫폼(Solaris, Linux 또는 Windows)에 따라 다릅니다.

### 자동 시작(Solaris 및 Linux)

Solaris 및 Linux 시스템에서는 자동 시작을 활성화하는 스크립트가 Message Queue 설치 중에 `/etc/rc*` 디렉토리 트리에 저장됩니다. 이러한 스크립트를 사용하려면 구성 파일 `/etc/imq/imqbrokerd.conf`(Solaris) 또는 `/etc/opt/sun/mq/imqbrokerd.conf`(Linux)를 다음과 같이 편집해야 합니다.

- 시스템을 시작할 때 브로커가 자동으로 시작되게 하려면 `AUTOSTART` 등록 정보를 `YES`로 설정합니다.
- 비정상적인 종료 이후에 브로커가 자동으로 다시 시작되게 하려면 `RESTART` 등록 정보를 `YES`로 설정합니다.
- 브로커에 대한 시작 명령줄 인수를 설정하려면 `ARGS` 등록 정보에 대해 하나 이상의 값을 지정합니다.

## 자동 시작(Windows)

Windows 시스템을 시작할 때 브로커가 자동으로 시작되게 하려면 브로커를 Windows 서비스로 정의해야 합니다. Message Queue를 Windows 시스템에 설치할 때 브로커를 서비스로 설치할 수 있습니다. 설치 이후에 서비스 관리자 유틸리티 `imqsvcadmin`을 사용하여 다음 작업을 수행할 수 있습니다.

- 브로커를 Windows 서비스로 추가
- 브로커 서비스에 대한 시작 옵션을 결정
- Windows 서비스로 실행 중인 브로커 제거

`imqsvcadmin` 명령의 구문, 하위 명령 및 옵션에 대한 자세한 내용은 [13장, "명령 참조"](#)를 참조하십시오.

브로커를 Windows 서비스로 설치한다는 것은 시스템을 시작할 때 시작하여 종료할 때 까지 백그라운드로 실행하는 것을 의미합니다. 따라서 추가 인스턴스를 시작하는 경우가 아니면 `imqbrokerd` 명령을 사용해서 브로커를 시작하지 마십시오.

브로커에 시작 옵션을 전달하려면 `-args` 인수를 `imqsvcadmin` 명령에 사용합니다. [67페이지](#)의 "[대화식으로 브로커 시작](#)"에서 설명한 대로 이 옵션은 `imqbrokerd` 명령의 `-D` 옵션과 동일하게 작동합니다. `imqcmd` 명령을 사용해서 보통의 경우와 같이 브로커를 제어합니다.

브로커가 Windows 서비스로 실행 중이면 작업 관리자에 해당 브로커가 두 개의 실행 가능 프로세스로 나열됩니다.

- Windows 고유의 서비스 래퍼 `imqbrokersvc.exe`
- 브로커를 실행 중인 Java 런타임

Windows 서비스로 실행 중인 브로커는 하나의 시스템에 하나만 있을 수 있습니다.

### 브로커 서비스 재구성

Windows 서비스를 재구성하는 순서는 다음과 같습니다.

1. 서비스를 정지합니다.
2. 서비스를 제거합니다.
3. `-args` 옵션으로 다른 브로커 시작 옵션을 지정하거나 `-vmargs` 옵션으로 다른 Java 버전 인수를 지정하여 서비스를 추가합니다.

## 대체 Java 런타임 사용

-javahome 또는 -jrehome 옵션을 사용하여 대체 Java 런타임의 위치를 지정할 수 있습니다. Windows 서비스 제어판 시작 매개 변수 필드에서 이러한 옵션을 지정할 수도 있습니다.

시작 매개 변수 필드에서는 백슬래시(\)가 제어 문자로 간주되기 때문에 백슬래시를 경로 구분자로 사용할 경우 두 번 입력해야 합니다. 예를 들면 다음과 같습니다.

```
-javahome d:\\jdk1.3.
```

## 브로커 서비스 시작 옵션 표시

브로커 서비스의 시작 옵션을 결정하려면 `imqsvcadm query` 명령에 `query` 옵션을 사용합니다.

```
imqsvcadm query

Service iMQ_Broker is installed.
Display Name: iMQ_Broker
Start Type: Manual
Binary location: C:Program FilesSun Microsystems
                    Message Queue 3.5\bin\imqbrokersvc
JavaHome: c:\j2sdk1.4.0
Broker Args: -passfile d:\imqpassfile
```

## 서비스 시작 문제 해결

서비스를 시작할 때 오류가 발생하는 경우 기록된 오류 이벤트를 볼 수 있습니다.

### ▶ 기록된 서비스 오류 이벤트를 보는 방법

1. 이벤트 뷰어를 시작합니다.
2. 로그 > 응용 프로그램을 확인합니다.
3. 보기 > 새로 고침을 선택하여 오류 이벤트를 봅니다.

## Windows 서비스로 실행 중인 브로커 제거

서비스로 실행 중인 브로커를 제거하려면 다음 중 하나를 수행합니다.

- 명령을 사용합니다. `imqcmd shutdown bkr` 명령을 사용하여 브로커를 종료한 다음 `imqsvcadm remove` 명령을 사용하여 서비스를 제거합니다.
- Windows 서비스의 경우 제어판의 관리 도구를 사용합니다. 이 기능을 사용할 수 있는 위치는 Windows 버전마다 다릅니다.

작업이 완료되면 컴퓨터를 다시 시작합니다.



# Message Queue 클라이언트 시작

클라이언트 응용 프로그램을 시작하기 전에 응용 프로그램 개발자로부터 시스템 설정 방법에 대한 정보를 얻습니다. Java 클라이언트 응용 프로그램을 시작할 경우 CLASSPATH 변수를 설정하고 올바른 jar 파일이 설치되었는지 확인해야 합니다. *Message Queue Developer's Guide for Java Clients*에 시스템 설정에 대한 일반 단계와 관한 자세한 내용이 있지만 개발자에게 추가 정보가 있을 수도 있습니다.

Java 클라이언트 응용 프로그램을 시작하려면 다음 명령줄 형식을 사용합니다.

```
java clientAppName
```

C 클라이언트 응용 프로그램을 시작하려면 응용 프로그램 개발자가 제공한 형식을 사용합니다.

응용 프로그램에서 설정하는 속성 값에 대한 자세한 내용은 응용 프로그램 개발자에게 문의하거나 응용 프로그램 설명서를 참조하십시오. 응용 프로그램에서 설정한 일부 속성을 대체하려는 경우 명령줄에서 해당 속성을 지정하면 됩니다.

Java 클라이언트의 명령줄에서 JNDI 조회를 사용하여 연결 팩토리를 찾는 속성을 지정할 수도 있습니다. 조회에서 응용 프로그램보다 더 오래된 연결 팩토리를 반환할 경우 해당 연결 팩토리는 최신 속성을 지원하지 못할 수 있습니다. 그럴 경우 Message Queue는 해당 속성을 기본값으로 설정합니다. 명령줄에서 속성을 지정하여 기본값이 아닌 값으로 설정할 수 있습니다.

명령줄에서 속성 값을 제공하려면 Java 응용 프로그램에 대해 다음 명령줄 구문을 사용합니다.

```
java [-Dattribute=value ]... clientAppName
```

16장, "관리 대상 객체 속성 참조"에 표시된 대로 *attribute* 값은 연결 팩토리 관리 대상 객체 속성이어야 합니다. 값에 공백이 있는 경우 명령줄의 *attribute=value* 부분을 따옴표로 묶어야 합니다.

다음 예에서는 클라이언트 응용 프로그램 MyMQClient를 시작합니다. 응용 프로그램은 응용 프로그램에 설정된 호스트 이름과 포트를 무시하고 호스트 OtherHost의 7677 포트에 있는 브로커에 연결합니다.

```
java -DmqAddressList=mq://OtherHost:7677/jms MyMQClient
```

명령줄을 사용하여 속성 값을 지정할 수 없는 경우도 있습니다. 관리자가 읽기 전용 액세스만 허용하도록 관리 대상 객체를 설정하거나 응용 프로그램 개발자가 읽기 전용 액세스만 허용하도록 클라이언트를 코딩할 수 있습니다. 클라이언트 프로그램을 시작하는 가장 좋은 방법을 알아보려면 응용 프로그램 개발자와 대화해야 합니다.

## 브로커 인스턴스 제거

이 절에서는 Solaris 또는 Linux에서 브로커 인스턴스를 제거하는 방법에 관한 정보를 제공합니다. Windows 서비스 제거에 대한 자세한 내용은 [70페이지의 "Windows 서비스로 실행 중인 브로커 제거"](#)를 참조하십시오.

브로커 인스턴스를 제거하려면 `imqbrokerd` 명령에서 `-remove` 옵션을 사용합니다. 브로커 인스턴스 제거를 위한 명령 형식은 다음과 같습니다.

```
imqbrokerd [options..] -remove instance
```

예를 들어, 브로커 이름이 `myBroker`인 경우 명령은 다음과 같습니다.

```
imqbrokerd -name myBroker -remove instance
```

이 명령은 지정된 브로커에 대해 전체 인스턴스 디렉토리를 삭제합니다.

브로커를 제거할 때 사용할 수 있는 옵션 목록은 [279페이지의 "명령 참조"](#)의 `imqbrokerd` 참조 정보를 참조하십시오.

Solaris 또는 Linux에서 시스템을 시작할 때 브로커를 자동으로 시작하도록 설정하려면 구성 파일 `/etc/imq/imqbrokerd.conf`(Solaris) 또는 `/etc/opt/sun/mq/imqbrokerd.conf`(Linux)를 편집하고 `AUTOSTART` 등록 정보를 `NO`로 설정합니다.

# 브로커 구성

브로커 인스턴스가 시작될 때 브로커 인스턴스의 구성은 일련의 구성 파일 및 `imgbrokerd` 명령에 전달되는 옵션에 따라 결정됩니다. 이 장에서는 구성 파일과 명령줄 옵션이 브로커 인스턴스를 구성하기 위해 상호 작용하는 방법을 설명하고 각 브로커 구성 요소의 기능을 설명하며 해당 구성 등록 정보를 나열한 다음 구성 설정 방법을 설명합니다.

이 장은 다음 내용으로 구성되어 있습니다.

- 74페이지의 "구성 가능한 브로커 구성 요소 정보"
- 96페이지의 "구성 파일 정보"
- 98페이지의 "인스턴스 구성 파일 편집"
- 99페이지의 "명령줄에 구성 옵션 입력"
- 99페이지의 "영구 저장소 설정"
- 104페이지의 "지속성 데이터 보안"

구성 등록 정보에 대한 자세한 내용은 14장, "브로커 등록 정보 참조"를 참조하십시오.

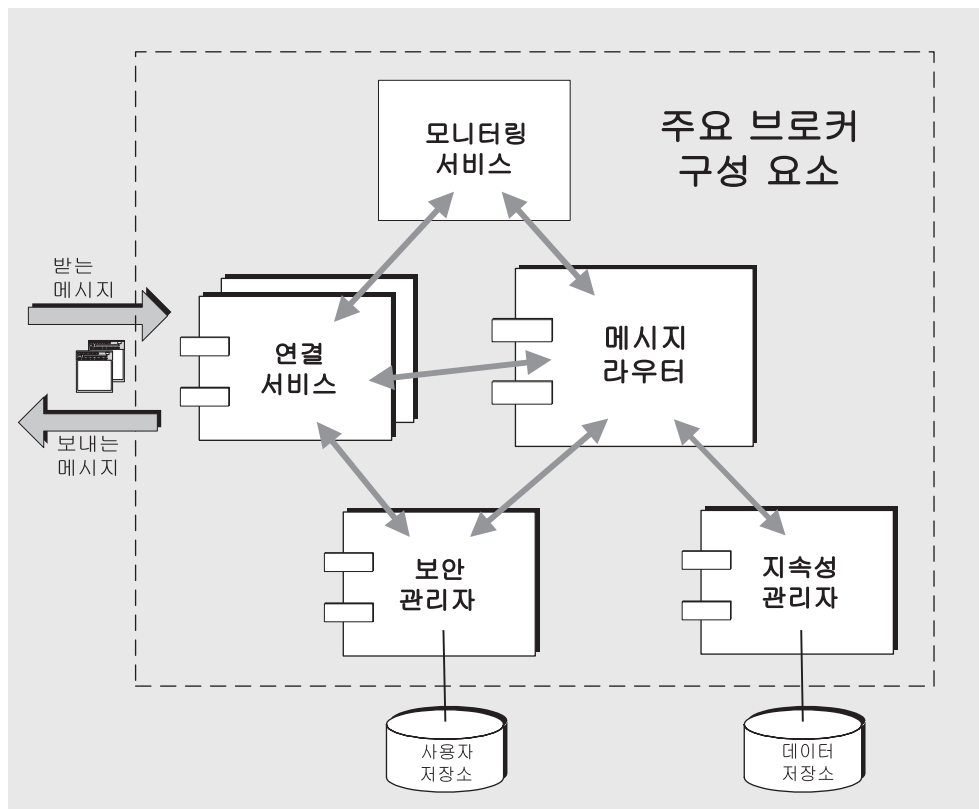
## 구성 가능한 브로커 구성 요소 정보

Message Queue 메시징 시스템의 메시지 전달은 브로커 또는 탠덤으로 작동하는 브로커 인스턴스로 구성된 클러스터가 수행합니다. 메시지는 생성자 클라이언트에서 대상으로 전달된 다음 대상에서 하나 이상의 사용자 클라이언트로 전달됩니다.

메시지 전달을 수행하려면 브로커가 클라이언트와 통신 채널을 설정하고, 인증 및 권한 부여를 수행하며, 메시지 경로를 올바르게 지정하고, 안정적인 전달을 보장하며, 시스템 성능을 모니터링할 데이터를 제공해야 합니다.

이러한 기능들을 수행하기 위해 브로커는 전달 과정에서 특별한 역할을 맡는 다양한 내부 구성 요소를 사용합니다. 이러한 브로커 구성 요소는 [그림 4-1](#)에 표시되어 있습니다.

그림 4-1 브로커 서비스 구성 요소



메시지 라우터 구성 요소가 주요 메시지 경로 지정 및 전달 서비스를 수행하고 다른 구성 요소는 중요한 지원 서비스를 제공합니다. 표 4-1에서는 각 구성 요소에 대해 간단히 설명합니다.

표 4-1 주요 브로커 구성 요소 및 기능

구성 요소	설명/기능	등록 정보 설명...
연결 서비스	브로커와 클라이언트 사이의 물리적 연결을 관리하면서 받고 보내는 메시지 전송을 담당합니다.	311페이지의 "연결 서비스 등록 정보"
메시지 라우터	메시지 라우팅 및 전달을 관리합니다. 여기에는 JMS 메시지를 비롯하여 Message Queue 메시징 시스템이 JMS 메시지 전달을 지원하기 위해 사용하는 제어 메시지도 포함됩니다.	313페이지의 "메시지 라우터 등록 정보"
지속성 관리자	영구 저장소에 데이터 쓰기 및 영구 저장소로부터의 데이터 검색을 관리합니다.	316페이지의 "지속성 관리자 등록 정보"
보안 관리자	브로커와의 연결을 요청하는 사용자에게 인증 서비스를 제공하고 인증된 사용자에게 권한 부여 서비스 (액세스 제어)를 제공합니다.	320페이지의 "보안 관리자 등록 정보"
모니터링 서비스	관리자가 브로커를 모니터링 및 관리할 때 사용할 수 있는 다양한 출력 채널에 기록 가능한 메트릭 및 진단 정보를 생성합니다.	324페이지의 "모니터링 및 로깅 등록 정보"

로드 상태, 응용 프로그램 복잡성 등에 따라 브로커 성능을 최적화하도록 이 구성 요소들을 구성할 수 있습니다. 다음 절에서는 각 구성 요소가 수행하는 기능과 해당 동작에 영향을 미치도록 설정할 수 있는 등록 정보에 대해 살펴봅니다.

## 연결 서비스

Message Queue 브로커는 Message Queue 응용 프로그램 클라이언트 및 Message Queue 관리 클라이언트 모두와의 통신을 지원합니다. 각 연결 서비스는 해당 서비스 유형 및 프로토콜 유형을 통해 다음과 같이 지정됩니다.

- *service type*은 서비스가 JMS 메시지 전달(NORMAL)을 제공하는지 또는 Message Queue 관리(ADMIN) 서비스를 제공하는지 지정합니다.
- *protocol type*은 서비스를 지원하는 기본 전송 프로토콜 계층을 지정합니다.

표 4-2에 Message Queue 브로커에서 사용 가능한 연결 서비스가 나열되어 있습니다.

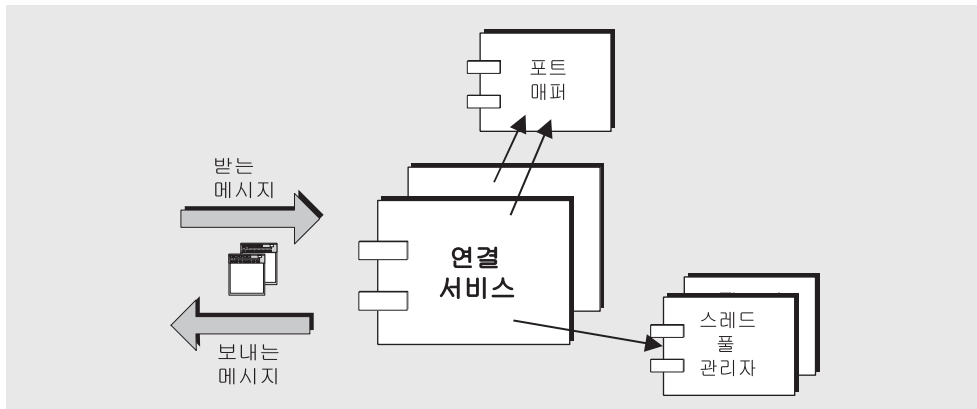
**표 4-2**      브로커가 지원하는 연결 서비스

서비스 이름	서비스 유형	프로토콜 유형
jms	NORMAL	tcp
ssljms(엔터프라이즈판)	NORMAL	tls(SSL 기반 보안)
httpjms(엔터프라이즈판)	NORMAL	http
httpsjms(엔터프라이즈판)	NORMAL	https(SSL 기반 보안)
admin	ADMIN	tcp
ssladmin	ADMIN	tls(SSL 기반 보안)

이 연결 서비스 중 어느 것이라도 또는 전부 실행하도록 브로커를 구성할 수 있습니다. 각 연결 서비스는 특정 포트에서 사용 가능하며, 브로커의 호스트 이름과 포트 번호로 지정됩니다. jms 및 admin 서비스가 기본적으로 실행됩니다.

Message Queue가 연결 서비스를 포트 번호에 동적으로 매핑하거나 사용자가 포트를 명시적으로 할당할 수 있습니다. 그림 4-2에 표시된 것처럼 각 서비스는 공통 포트 매핑에 등록되지만 자체 스레드 풀 관리자도 있습니다.

**그림 4-2**      연결 서비스 지원



다음 절에서는 연결 서비스와 포트 매핑 및 스레드 풀 관리자 사이의 관계에 대해 설명합니다.

## 포트 매핑

Message Queue는 포트를 연결 서비스에 할당하는 *포트 매핑*을 제공합니다. 포트 매핑은 표준 포트 번호 7676에 위치합니다. 클라이언트가 브로커와 연결을 설정할 때 이 클라이언트는 먼저 포트 매핑에 접속하여 지정한 연결 서비스의 포트 번호를 요청합니다.

jms, ssljms, admin 및 ssladmin 연결 서비스의 포트 번호는 동적이거나 정적일 수 있습니다. 기본적으로 연결 서비스는 시작될 때 해당 포트를 동적으로 구성합니다. 사용자가 서비스에 대해 정적 포트를 지정할 수도 있지만 정적 포트 번호는 사용하지 않는 것이 좋습니다. 일반적으로 정적 포트 번호는 방화벽을 통한 연결과 같은 특수한 상황에서만 사용됩니다.

httpjms 및 httpsjms 서비스는 각각 **부록 C, "HTTP/HTTPS 지원"**에서 **373페이지의 표 C-1**과 **385페이지의 표 C-3**에 설명되어 있는 등록 정보를 사용하여 구성됩니다.

## 스레드 풀 관리자

각 연결 서비스는 다중 스레드 방식으로서, 다중 연결을 지원합니다. 이 연결에 필요한 스레드는 *스레드 풀 관리자* 구성 요소가 관리하는 스레드 풀에서 유지 관리됩니다.

스레드 풀 관리자를 구성하여 스레드 풀에서 유지 관리되는 최소 스레드 수와 최대 스레드 수를 설정할 수 있습니다. 연결 시 스레드가 필요하면 스레드 풀에 해당 스레드가 추가됩니다. 최소 스레드 수를 초과할 경우, 시스템은 최소 수 임계값에 도달할 때까지 스레드를 종료시켜 여유 스레드를 확보하는 방법으로 메모리 자원을 절약합니다. 새 스레드가 계속 작성될 필요가 없도록 이 수를 충분히 크게 설정해야 합니다. 연결 로드가 많은 경우, 스레드 풀의 최대 수에 도달할 때까지 스레드 수가 증가하기도 합니다. 그 후 스레드가 사용 가능해질 때까지 연결은 대기합니다.

스레드 풀의 스레드는 단일 연결(*전용 모델*) 전용으로 사용되거나 필요에 따라 여러 연결(*공유 모델*)에 지정될 수 있습니다.

**전용 모델** 브로커와의 연결마다 연결에서 받는 메시지와 보내는 메시지를 각각 처리하는 2개의 전용 스레드가 필요합니다. 따라서 연결 수가 스레드 풀에 있는 최대 스레드 수의 절반으로 제한되지만, 이 방법은 우수한 성능을 발휘합니다.

**공유 모델(엔터프라이즈판)** 메시지를 보내거나 받을 때마다 공유 스레드에서 연결을 처리합니다. 이 모델에서는 각 연결에 전용 스레드가 필요하지 않기 때문에 연결 서비스(및 브로커)가 지원할 수 있는 연결 수가 증가합니다. 그러나 스레드 공유와 관련된 어느 정도의 성능 오버헤드가 있습니다. 스레드 풀 관리자는 연결 활동을 모니터링하고 필요에 따라 스레드에 연결을 지정하는 분산자 스레드 집합을 사용합니다. 이러한 각 분산자 스레드가 모니터링하는 연결 수를 제한하여 이 활동과 관련된 성능 오버헤드를 최소화할 수 있습니다.

## 보안

각 연결 서비스는 특정 인증 및 권한 부여(액세스 제어) 기능을 지원합니다([88페이지의 "보안 관리자"](#) 참조).

## 연결 서비스 등록 정보

연결 서비스와 관련된 구성 가능한 등록 정보는 다음과 같습니다.

- `img.service.activelist`. 브로커 시작 시 시작될 연결 서비스 목록입니다.
- `img.hostname`. 사용 가능한 호스트가 두 개 이상 있는 경우(예: 컴퓨터에 네트워크 인터페이스 카드가 두 개 이상인 경우) 모든 연결 서비스를 바인드할 호스트를 지정합니다.
- `img.portmapper.port`. 브로커의 기본 포트(포트 매핑이 위치한 포트)를 지정합니다.
- `img.portmapper.hostname`. 사용 가능한 호스트가 두 개 이상인 경우 포트 매핑을 바인드할 호스트를 지정합니다.
- `img.portmapper.backlog`. 포트 매핑이 요청 거부 전까지 처리할 수 있는 최대 동시 요청 수를 지정합니다. 이 등록 정보는 운영 체제 백로그에 저장되어 포트 매핑의 처리를 대기할 수 있는 요청 수를 설정합니다.
- `img.service_name.protocol_type.port`. `jms`, `ssljms`, `admin` 및 `ssladmin` 서비스에 한해 명명된 연결 서비스의 포트 번호를 지정합니다.
- `img.service_name.protocol_type.hostname`. `jms`, `ssljms`, `admin` 및 `ssladmin` 서비스에 한해 사용 가능한 호스트가 두 개 이상인 경우에 명명된 연결 서비스를 바인드할 호스트를 지정합니다.
- `img.service_name.min_threads`. 이 스레드 수에 도달하면 명명된 연결 서비스가 사용할 수 있도록 스레드 풀에서 스레드가 유지 관리됩니다.



- `imq.service_name.max_threads`. 이 스레드 수를 초과하면 명명된 연결 서비스가 사용할 수 있는 신규 스레드가 더 이상 스레드 풀에 추가되지 않습니다.
- `imq.service_name.threadpool_model`. 명명된 연결 서비스에 대해 스레드가 연결 전용인지 또는 필요에 따라 여러 연결에 의해 공유되는지를 지정합니다.
- `imq.shared.connectionMonitor_limit`. 공유 스레드 풀 모델에 한해 분산자 스레드가 모니터링할 수 있는 최대 연결 수를 지정합니다

이러한 등록 정보에 대한 자세한 내용은 [311페이지의 표 14-2](#)를 참조하십시오.

## 메시지 라우터

지원되는 연결 서비스를 사용하여 클라이언트와 브로커 사이에 연결이 설정되면 메시지 경로 지정 및 전달을 계속할 수 있습니다.

### 기본적인 전달 메커니즘

포괄적으로 말하자면, 브로커가 처리하는 메시지는 다음과 같은 2가지 범주로 구분됩니다.

- 생성자 클라이언트가 사용자 클라이언트를 대상으로 보내는 **JMS 페이로드** 메시지
- **JMS** 메시지 전달을 지원하려고 클라이언트를 오가는 제어 메시지

받는 메시지가 **JMS** 메시지인 경우 다음과 같이 브로커는 대상이 대기열인지 주제인지에 따라 사용자 클라이언트로 경로를 지정합니다.

- 대상이 주제라면 **JMS** 메시지는 해당 주제의 모든 활성 가입자로 향하는 경로를 즉시 지정합니다. 영구 가입자가 비활성 상태인 경우 메시지 라우터는 메시지를 보관했다가 해당 가입자가 활성 상태가 되면 전달합니다.
- 대상이 대기열이라면 **JMS** 메시지는 해당 대기열로 들어가고, 대기열의 맨 앞에 도달하는 시점에 해당 사용자에게 전달됩니다. 메시지가 대기열의 맨 앞에 도달하는 순서는 도착 순서 및 우선 순위에 따라 결정됩니다.

메시지 라우터가 목표한 모든 사용자에게 메시지를 전달했다면 해당 메시지를 메모리에서 삭제합니다. 지속성 메시지의 경우 메시지 라우터는 해당 메시지를 브로커의 영구 데이터 저장소에서 제거합니다.

## 안정적인 전달: 확인 및 트랜잭션

지금까지 설명한 전달 메커니즘은 *안정적인* 전달을 위한 요구 사항이 추가될 경우 더 복잡해집니다. 안정적인 전달에는 다음과 같은 두 가지 측면이 있습니다.

- 브로커를 오가는 메시지가 성공적으로 전달되게 함
- 메시지를 실제로 전달하기 이전에 브로커가 메시지나 전달 정보를 잃어버리지 않게 함

브로커를 오가는 메시지가 성공적으로 전달되도록 하기 위해 **Message Queue**는 다양한 응답 제어 메시지를 사용합니다.

예를 들어, 생성자가 어떤 대상으로 **JMS** 메시지(페이로드 메시지)를 대상에 보낼 때 브로커는 **JMS** 메시지를 수신했음을 확인하는 응답을 보냅니다(기본적으로 생성자가 **JMS** 메시지를 지속성 메시지로 지정한 경우에만 **Message Queue**가 이 기능을 수행함). 생성자 클라이언트는 브로커 응답을 통해 대상에게 확실히 전달되도록 합니다.

그와 비슷하게 브로커가 사용자에게 **JMS** 메시지를 전달할 경우, 사용자 클라이언트는 해당 메시지를 수신 및 처리했음을 알리는 확인을 회신합니다. 클라이언트는 세션 객체 작성 시 이 확인을 얼마나 자동으로 또는 얼마나 자주 보낼 것인지 지정하지만, 메시지 라우터는 자신이 메시지를 전달했던 각 사용자로부터 확인을 받을 때까지(예: 여러 가입자 각각으로부터 한 주제로) 해당 **JMS** 메시지를 메모리에서 삭제하지 않습니다.

어떤 주제에 대한 영구 가입의 경우, 메시지 라우터는 각 **JMS** 메시지를 대상에 보존하고 각 영구 가입자가 활성 사용자가 되면 메시지를 전달합니다.

메시지 라우터는 클라이언트 확인을 수신하여 이를 기록하고, 그 전까지 **JMS** 메시지만 완료되지 않는 한 모든 확인을 수신한 후에야 **JMS** 메시지를 삭제합니다.

또한 메시지 라우터는 클라이언트에게 브로커 응답을 회신하여 클라이언트 확인을 받았음을 알립니다. 사용자 클라이언트는 브로커 응답을 사용하여 브로커가 **JMS** 메시지를 2회 이상 전달하지 않도록 합니다. 이는 브로커가 클라이언트 확인을 받지 못한 경우에 발생할 수 있습니다.

브로커가 클라이언트 확인을 받지 않고 **JMS** 메시지를 두 번째 전달하는 경우, 메시지에 재전송 플래그가 표시됩니다. 브로커는 일반적으로 다음과 같은 경우에 **JMS** 메시지를 다시 전달합니다.

- 브로커가 클라이언트 확인을 받기 전에 클라이언트 연결이 닫히고 새 연결이 이어져 열리는 경우
- 클라이언트 응용 프로그램이 세션을 복구하는 경우
- 클라이언트 응용 프로그램이 롤백된 트랜잭션을 복구하는 경우

예를 들어 메시지 확인을 받기 전에 대기열의 메시지 사용자가 오프라인되고 이어서 다른 사용자가 해당 대기열에 등록하면 브로커는 미확인 메시지를 새 사용자에게 재전송합니다.

위에서 설명한 클라이언트 확인과 브로커 응답 과정은 JMS 메시지 전달이 트랜잭션으로 그룹화된 경우에도 적용됩니다. 이 경우 클라이언트 확인과 브로커 응답 과정은 개별 JMS 메시지 송수신 수준 뿐 아니라 트랜잭션 수준에서도 수행됩니다. 트랜잭션이 완료되면 브로커 응답이 자동으로 보내집니다.

브로커는 트랜잭션을 추적하면서 트랜잭션을 완료하거나 실패 시 롤백이 가능하게 합니다. 또한 이 트랜잭션 관리는 더 큰 규모의 분산 트랜잭션에 포함되는 로컬 트랜잭션을 지원합니다. 브로커는 트랜잭션이 완료될 때까지 그 상태를 추적합니다. 브로커가 시작되면 이 브로커는 완료되지 않은 모든 트랜잭션을 검사하며, PREPARED 상태의 트랜잭션을 제외하고 모든 트랜잭션을 롤백하도록 기본 설정되어 있습니다.

`img.transaction.autorollback` 등록 정보를 설정한 경우에도 브로커는 PREPARED 상태인 트랜잭션을 롤백합니다.

## 안정적인 전달: 지속성

안정적인 전달의 또 다른 측면은 실제로 메시지가 전달될 때까지 브로커가 메시지나 전달 정보를 잃어버리지 않게 하는 것입니다. 일반적으로 메시지는 전달되거나 만료될 때까지 메모리에 남아 있습니다. 하지만 브로커 실패 시 메시지가 손실됩니다.

생성자 클라이언트가 메시지를 지속성 메시지로 지정한 경우 메시지 라우터는 해당 메시지를 *지속성 관리자에게 전달합니다*. 지속성 관리자는 브로커가 실패할 경우에 메시지를 복구할 수 있도록 메시지를 데이터베이스나 파일 시스템에 저장합니다([83페이지의 "지속성 관리자"](#) 참조).

## 메모리 자원 및 메시지 흐름 관리

브로커의 성능과 안정성은 사용 가능한 시스템 자원 그리고 메모리와 같은 자원이 얼마나 효율적으로 활용되는가에 따라 달라집니다. 특히 메시지의 생성이 사용보다 훨씬 빠를 경우 메시지 라우터가 넘치게 되어 모든 메모리 자원을 소진할 수 있습니다. 이런 현상을 방지하기 위해 메시지 라우터는 자원이 부족해질 경우 다음과 같은 세 가지 메모리 보호 수준을 사용하여 시스템을 운영합니다.

**개별 대상에 대한 메시지 제한** 메시지 수와 메시지가 사용하는 총 메모리를 제한하는 물리적 대상 등록 정보를 설정할 수 있습니다(15장, "물리적 대상 등록 정보 참조" 참조). 제한 값에 도달할 경우의 메시지 라우터 동작을 지정할 수도 있습니다. 네 가지 제한 동작은 다음과 같습니다.

- 메시지 생성자 표시(FLOW\_CONTROL)
- 메모리에서 가장 오래된 메시지 삭제(REMOVE\_OLDEST)
- 메시지 보존 기간을 기준으로 메모리에서 우선 순위가 가장 낮은 메시지 삭제(REMOVE\_LOW\_PRIORITY)
- 최신 메시지 거부(REJECT\_NEWEST)

**시스템 전체 메시지 제한** 시스템 전체 메시지 제한은 두 번째 보호 집합을 구성합니다. 총 메시지 수, 모든 메시지가 사용하는 메모리 등과 같이 시스템의 모든 대상에 한꺼번에 적용되는 시스템 전체 제한을 지정할 수 있습니다(313페이지의 표 14-3 참조). 시스템 전체 메시지 제한에 도달하면 메시지 라우터는 새 메시지를 거부합니다.

**시스템 메모리 임계값** 시스템 메모리 임계값은 세 번째 보호 집합입니다. 브로커가 메모리 과부하 방지를 위한 조치의 수위를 점점 더 높이게 되는 사용 가능한 시스템 메모리의 임계값을 지정할 수 있습니다. 조치는 다음과 같이 메모리 자원 상태에 따라 달라집니다.

- 초록(사용 가능한 메모리 충분)
- 노랑(브로커 메모리 감소 중)
- 주황(브로커 메모리 부족)
- 빨강(브로커가 사용 가능한 메모리 없음)

브로커의 메모리 상태가 초록에서 노랑 및 주황을 거쳐 빨강으로 변하면 브로커는 다음과 같이 점점 더 높은 수준의 조치를 수행합니다.

- 활성 메모리의 메시지를 영구 저장소에 스왑(83페이지의 "지속성 관리자" 참조)
- 비지속성 메시지 생성자를 억제한 뒤 결국 브로커로 향하는 메시지 흐름을 중지시킵니다. 지속성 메시지 흐름은 브로커가 각 메시지를 확인해야 하기 때문에 자동으로 제한됩니다.

두 조치 모두 성능을 떨어뜨립니다.

시스템 메모리 임계값에 도달하는 경우는 대상 메시지 제한과 시스템 전체 메시지 제한을 너무 작게 설정했기 때문입니다. 임계값만으론 잠재적 메모리 과부하를 제때에 잡을 수 없는 경우도 있습니다. 따라서 이 기능에만 의존하여 메모리 자원을 제어해서는 안 되고 대상을 개별적 및 전체적으로 구성하여 메모리 자원을 최적화하는 것이 좋습니다.

## 메시지 라우터 등록 정보

다음은 메모리 자원 관리를 위한 시스템 전체 제한 및 시스템 메모리 임계값입니다.

- `imq.destination.DMQ.truncateBody`. 사용 불능 메시지 대기열에 메시지 헤더와 등록 정보 데이터만 포함되도록 지정합니다. 메시지 본문 내용은 삭제됩니다.
- `imq.message.expiration.interval`. 만료된 메시지 재생 이용이 발생하는 빈도를 초 단위로 지정합니다.
- `imq.system.max_count`. 브로커가 보관하는 최대 메시지 수를 지정합니다.
- `imq.system.max_size`. 브로커가 보관하는 메시지의 최대 전체 크기를 지정합니다.
- `imq.message.max_size`. 메시지 본문의 최대 크기를 지정합니다.
- `imq.resource_state.threshold`. 각 메모리 자원 상태가 트리거되는 메모리 사용률을 지정합니다.
- `imq.resource_state.count`. 일괄 처리에서 허용되는 받는 메시지의 최대 수를 지정합니다. 이 제한에 이르면 각 메모리 자원 상태가 트리거됩니다.
- `imq.transaction.autorollback`. PREPARED 상태로 남아 있는 분산 트랜잭션이 브로커 시작 시 자동으로 롤백되는지 여부를 지정합니다.

이러한 등록 정보에 대한 자세한 내용은 [313페이지의 표 14-3](#)을 참조하십시오.

## 지속성 관리자

오류 발생 시 브로커를 복구하려면 메시지 전달 작업 상태를 다시 작성해야 합니다. 그러기 위해서는 모든 지속성 메시지와 기본적인 라우팅 및 전달 정보를 데이터 저장소에 저장해야 합니다. *지속성 관리자* 구성 요소는 이 정보의 작성 및 검색을 관리합니다.

오류가 발생한 브로커를 복구하려면 전달되지 못한 메시지를 복원하는 것 이상의 작업이 필요합니다. 또한 브로커는 다음을 수행할 수 있어야 합니다.

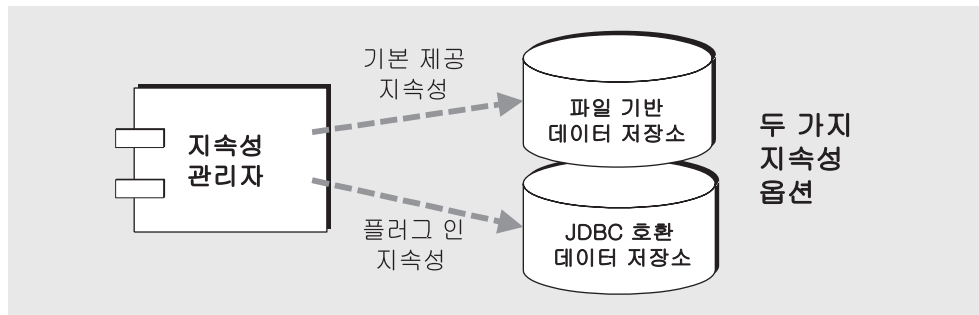
- 대상 다시 작성
- 각 주제의 영구 가입 목록 복원
- 각 메시지의 확인 목록 복원
- 완결된 모든 트랜잭션 상태 복제

지속성 관리자는 이 모든 상태 정보의 저장 및 복원을 관리합니다.

브로커가 다시 시작되면 대상 및 영구 가입을 다시 작성하고 지속성 메시지를 복구하며 모든 트랜잭션의 상태를 복원하고 전달되지 못한 메시지의 라우팅 테이블을 다시 작성합니다. 그런 다음 메시지 전달을 다시 시작할 수 있습니다.

Message Queue는 기본 제공 및 플러그인 지속성 모듈을 모두 지원합니다(그림 4-3 참조). 기본 제공 지속성은 파일 기반 데이터 저장소입니다. 플러그인 지속성은 JDBC™(Java Database Connectivity) 인터페이스를 사용하며 JDBC 데이터 저장소가 필요합니다. 일반적으로 기본 제공 지속성이 플러그인 지속성보다 더 빠르지만, JDBC 호환 데이터베이스 시스템의 중복 및 관리 기능을 선호하는 사용자도 있습니다.

그림 4-3 지속성 관리자 지원



## 기본 제공 지속성

기본 Message Queue 영구 저장소 솔루션은 파일 기반 데이터 저장소입니다. 이 방법에서는 개별 파일을 사용하여 메시지, 대상, 영구 가입, 트랜잭션과 같은 지속성 데이터를 저장합니다.

파일 기반 데이터 저장소는 데이터 저장소와 연관된 브로커 인스턴스의 이름 (*instanceName*)으로 식별되는 디렉토리에 있습니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

```
.../instances/instanceName/fs350/
```

메시지가 있는 대상에 따라 지속성 메시지를 디렉토리에 저장하도록 파일 기반 데이터 저장소를 구성합니다. 대부분의 메시지는 가변 크기 레코드로 구성되는 단일 파일에 저장됩니다.

메시지를 추가 및 제거할 때 단편화를 줄이기 위해 가변 크기 레코드 파일을 압축할 수 있습니다(136페이지의 "물리적 대상 압축" 참조). 또한, 기본 제공 지속성 관리자는 크기가 구성 가능한 임계값(`imq.persist.file.message.max_record_size`)을 초과하는 메시지를 가변 크기 레코드 파일이 아니라 각자 해당 파일에 저장합니다. 이러한 개별 파일에서는 파일을 재사용할 수 있도록 파일 풀이 유지 관리됩니다. 메시지 파일이 더 이상 필요하지 않은 경우 삭제되지 않고 대신 대상 디렉토리의 사용 가능 파일 풀에 메시지 파일이 추가되어 새 메시지를 저장하는 데 사용됩니다.

대상 파일 풀의 최대 파일 수(`imq.persist.file.destination.message.filepool.limit`)를 구성할 수 있습니다. 또한 단순히 재사용 태그를 표시하는 것이 아니라 0으로 잘라내어 파일 풀에서 지울 사용 가능 파일 비율을 지정할 수 있습니다 (`imq.persist.file.message.filepool.cleanratio`). 지운 파일의 비율이 높을수록 디스크 공간은 더 줄어들고 파일 풀 관리에 필요한 오버헤드는 커집니다. 더 많은 오버헤드가 필요합니다.

또한 종료 시 태그가 표시된 파일을 지울 것인지 여부도 지정할 수 있습니다(`imq.persist.file.message.cleanup`). 파일을 지우면 디스크 공간은 덜 차지하지만, 브로커 종료 속도가 느려집니다.

모든 다른 지속성 데이터(대상, 영구 가입 및 트랜잭션)는 별도의 파일에 저장됩니다. 모든 대상이 한 파일에 저장되고, 모든 영구 가입이 다른 하나의 파일에 저장되는 식입니다.

안정성을 최대화하려면 `imq.persist.file.sync.enabled` 속성을 사용하여 지속성 작업이 메모리 상태에서 물리적 저장 장치와 동기화되도록 지정할 수 있습니다. 이렇게 하면 시스템 중단으로 인한 데이터 손실을 방지할 수 있지만 성능은 저하됩니다. Message Queue를 Sun Cluster 환경에서 실행하는 경우 클러스터의 모든 노드에 대해 이 속성을 `true`로 설정해야 합니다.

데이터 저장소에는 중요 정보나 소유 정보를 포함하는 메시지가 있을 수 있기 때문에 `instances/instanceName/fs350/` 디렉토리를 인증되지 않은 액세스로부터 보호해야 합니다. 자세한 지침은 [104페이지의 "지속성 데이터 보안"](#)을 참조하십시오.

## 플러그인 지속성

JDBC 드라이버를 통해 액세스 가능한 데이터 저장소를 모두 액세스하도록 브로커를 설정할 수 있습니다. 이 경우 여러 JDBC 관련 브로커 구성 등록 정보를 설정하고 데이터베이스 관리자 유틸리티(`imqdbmgr`)를 사용하여 적합한 체계를 갖는 데이터 저장소를 만들어야 합니다. 절차 및 관련 구성 등록 정보는 [99페이지의 "영구 저장소 설정"](#)을 참조하십시오.

## 지속성 관리자 등록 정보

이 등록 정보는 다음과 같이 사용 중인 지속성 유형을 지정합니다.

- `imq.persist.store`. 브로커가 기본 제공 파일 기반(file) 지속성을 사용하고 있는지 플러그인 JDBC 호환(jdbc) 지속성을 사용하고 있는지를 지정합니다.

이러한 등록 정보는 다음과 같이 기본 제공 지속성에 속합니다.

- `imq.persist.file.sync.enabled`. 지속성 작업이 메모리 상태에서 물리적 저장 장치와 동기화될 것인지 여부를 지정합니다.
- `imq.persist.file.message.max_record_size`. 메시지 저장소 파일에 추가될 최대 메시지 크기를 지정합니다.
- `imq.persist.file.destination.message.filepool.limit`. 대상 파일 풀에서 재사용할 수 있는 최대 사용 가능 파일 수를 지정합니다.
- `imq.persist.file.message.filepool.cleanratio`. *clean* 상태(0으로 잘림)로 유지되는 대상 파일 풀의 사용 가능 파일 비율을 지정합니다.
- `imq.persist.file.message.cleanup`. 브로커가 종료 시 대상 파일 풀에서 사용 가능 파일을 지울지 여부를 지정합니다.

이러한 등록 정보에 대한 자세한 내용은 [317페이지의 표 14-6](#)을 참조하십시오.

이러한 등록 정보는 다음과 같이 JDBC 기반 지속성에 속합니다.

- `imq.persist.jdbc.brokerid`. 데이터베이스에서 여러 브로커 인스턴스에 사용되는 테이블의 이름에 추가할 브로커 인스턴스 식별자를 지정합니다.
- `imq.persist.jdbc.driver`. 데이터베이스에 연결할 JDBC 드라이버의 Java 클래스 이름을 지정합니다.
- `imq.persist.jdbc.opendburl`. 기존 데이터베이스에 대한 연결을 여는 데이터베이스 URL을 지정합니다.



- `imq.persist.jdbc.createdburl`. 데이터베이스를 만들기 위해 연결을 여는 데이터베이스 URL을 지정합니다.
- `imq.persist.jdbc.closedburl`. 브로커를 종료할 때 현재 데이터베이스 연결을 종료하는 데이터베이스 URL을 지정합니다.
- `imq.persist.jdbc.user`. 필요한 경우 데이터베이스 연결을 열 때 사용되는 사용자 아이디를 지정합니다.
- `imq.persist.jdbc.needpassword`. 브로커에 액세스할 때 데이터베이스에 비밀번호가 필요한지 여부를 지정합니다.
- `imq.persist.jdbc.password`. 필요한 경우 데이터베이스 연결을 여는 비밀번호를 지정합니다.
- `imq.persist.jdbc.table.IMQSV35`. 버전 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQCCREC35`. 구성 변경 레코드 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQDEST35`. 대상 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQINT35`. 인터레스트 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQMSG35`. 메시지 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQPROPS35`. 등록 정보 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQILIST35`. 인터레스트 상태 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQTXN35`. 트랜잭션 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQTACK35`. 트랜잭션 확인 테이블을 만들 때 사용하는 SQL 명령

이러한 등록 정보에 대한 자세한 내용은 [317페이지의 표 14-7](#)을 참조하십시오.

## 보안 관리자

Message Queue는 인증 및 권한 부여(액세스 제어) 기능을 제공하며 암호화 기능도 지원 합니다.

인증 및 권한 부여 기능은 사용자 저장소(89페이지의 [그림 4-4](#) 참조), 즉 메시징 시스템 사용자에게 대한 정보(예: 아이디, 비밀번호, 그룹 멤버십)를 포함하는 파일, 디렉토리 또는 데이터베이스에 따라 달라집니다. 아이디와 비밀번호는 브로커와의 연결 요청 시 사용자를 인증할 때 사용됩니다. 대상에 대한 메시지 생성/사용과 같은 작업 권한을 부여할 때 아이디와 그룹 멤버십이 액세스 제어 파일과 함께 사용됩니다.

Message Queue 관리자는 Message Queue 제공 사용자 저장소(142페이지의 ["플랫 파일 사용자 저장소 사용"](#) 참조)를 채우거나 기존 LDAP 사용자 저장소를 보안 관리자 구성 요소에 플러그 인합니다(149페이지의 ["사용자 저장소에 LDAP 서버 사용"](#) 참조).

### 인증

Message Queue 보안은 비밀번호 기반의 인증을 지원합니다. 클라이언트가 브로커와의 연결을 요청할 경우 이 클라이언트는 아이디와 비밀번호를 제출해야 합니다.

보안 관리자는 클라이언트가 제출한 아이디와 비밀번호를 사용자 저장소에 저장된 정보와 비교합니다. 클라이언트가 브로커에게 비밀번호를 전송할 때 이 비밀번호는 기본 64 인코딩이나 메시지 다이제스트(MD5) 중 한 가지를 사용하여 암호화됩니다. 보다 안전한 전송에 대해서는 90페이지의 ["암호화"](#)를 참조하십시오. 별도로 각 연결 서비스가 사용하는 인코딩 유형을 구성하거나 브로커 전체에 대한 인코딩을 설정할 수 있습니다.

모든 보안 관리자 등록 정보는 90페이지의 ["보안 관리자 등록 정보"](#)에 나열되어 있으며 90페이지의 ["보안 관리자 등록 정보"](#)에서 자세히 설명합니다.

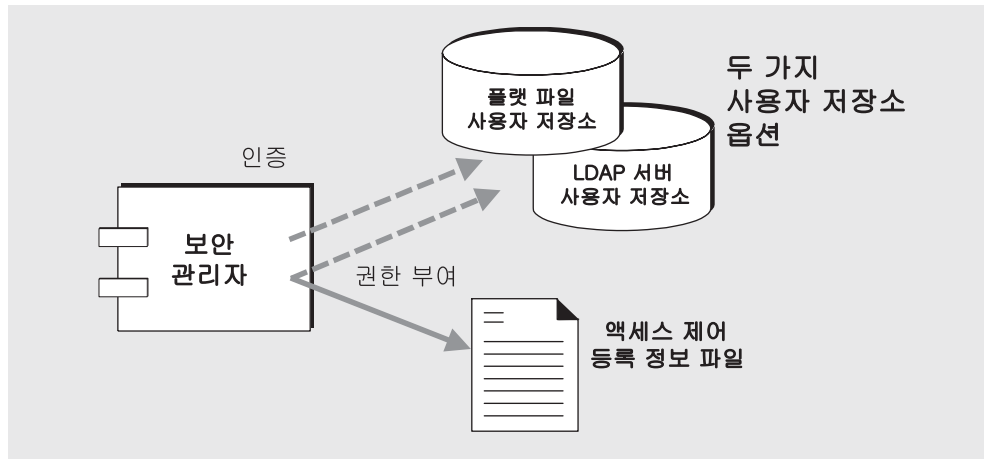
### 권한 부여

클라이언트 응용 프로그램 사용자가 인증되면 이 사용자는 여러 Message Queue 관련 작업을 수행할 권한을 갖습니다. 보안 관리자는 사용자 기반 액세스 제어와 그룹 기반 액세스 제어를 모두 지원합니다. 사용자 저장소에 있는 아이디 또는 해당 사용자가 속한 그룹에 따라 이 사용자는 특정 Message Queue 작업을 수행할 권한을 갖습니다. 액세스 제어 등록 정보 파일에서 이 액세스 제어를 지정합니다([그림 4-4](#) 참조).

사용자가 어떤 작업을 수행하려 하면 보안 관리자는 (액세스 제어 등록 정보 파일에 있는) 해당 작업 액세스를 위해 지정된 아이디/그룹 구성원을 (사용자 저장소에 있는) 아이디/그룹 구성원과 대조 확인합니다. 액세스 제어 등록 정보 파일은 다음 작업에 대한 권한을 지정합니다.

- 브로커와의 연결 설정
- 대상에 액세스: 사용자, 생성자 또는 특정 대상이나 모든 대상에 대한 대기열 브라우저 작성
- 대상 자동 작성

그림 4-4 보안 관리자 지원



기본 액세스 제어 등록 정보 파일이 *admin*이라는 단일 그룹만 명시적으로 참조합니다 (145페이지의 "그룹" 참조). *admin* 그룹의 사용자는 관리 서비스 연결 권한을 갖습니다. 관리 서비스 사용자는 대상 작성, 브로커 모니터링 및 제어와 같은 관리 기능을 수행할 수 있습니다. 다른 그룹으로 정의된 사용자는 기본적으로 관리 서비스를 연결할 수 없습니다.

Message Queue 관리자는 사용자 저장소에서 그룹을 정의하고 사용자를 해당 그룹에 연결할 수 있습니다(단, 플랫 파일 사용자 저장소에서는 그룹 기능이 완전히 지원되지 않음).

액세스 제어 등록 정보 파일을 편집하여 메시지 생성 및 사용 또는 대기열 대상에서의 메시지 찾아보기 목적의 대상 액세스 권한을 사용자 및 그룹별로 지정할 수 있습니다. 개별 대상이나 모든 대상에 대한 액세스 권한을 특정 사용자나 그룹에게만 부여할 수 있습니다. 대상 자동 생성을 허용하도록 브로커가 구성되어 있는 경우 액세스 제어 등록 정보 파일을 편집하여 브로커가 대상을 자동 생성할 수 있는 사용자와 그룹을 제어할 수 있습니다.

모든 보안 관리자 등록 정보는 90페이지의 "보안 관리자 등록 정보"에 나열되어 있으며 90페이지의 "보안 관리자 등록 정보"에서 자세히 설명합니다.

## 암호화

클라이언트와 브로커 사이에 전송되는 메시지를 암호화하려면 SSL(Secure Socket Layer) 표준 기반의 연결 서비스를 사용해야 합니다. SSL은 SSL 사용 가능 브로커와 SSL 사용 가능 클라이언트 사이에 암호화된 연결을 설정함으로써 연결 수준에서의 보안을 제공합니다.

Message Queue SSL 기반 연결 서비스를 사용하려면 키 도구 유틸리티(`imqkeytool`)를 사용하여 개인 키/공용 키 쌍을 생성합니다. 이 유틸리티는 자체 서명된 인증서에 공용 키를 내장하고 이를 Message Queue 키 저장소에 저장합니다. Message Queue 키 저장소 자체도 비밀번호로 보호됩니다. 이 잠금을 해제하려면 시작할 때 키 저장소 비밀번호를 제공해야 합니다. [159페이지의 "SSL 기반 서비스 작업"](#)을 참조하십시오.

키 저장소 잠금이 해제되면 브로커는 연결을 요청하는 모든 클라이언트에게 인증서를 전달할 수 있습니다. 그리고 나서 클라이언트는 인증서를 사용하여 브로커에게 보낼 암호화된 연결을 설정합니다.

모든 보안 관리자 등록 정보는 다음 절에 나열되어 있으며 [90페이지의 "보안 관리자 등록 정보"](#)에서 자세히 설명합니다.

## 보안 관리자 등록 정보

다음은 인증, 권한 부여, 암호화 및 기타 보안 통신에 대해 구성 가능한 등록 정보입니다.

- `imq.authentication.type`. 비밀번호를 기본 64 코딩(basic)으로 전달할지 MD5 다이제스트(digest)로 전달할지 여부를 지정합니다.
- `imq.service_name.authentication.type`. 비밀번호를 기본 64 코딩(basic)으로 전달할지 MD5 다이제스트(digest)로 전달할지 여부를 지정합니다.
- `imq.authentication.basic.user_repository`. 기본 64 코딩에 대해 인증에 사용할 사용자 저장소 유형을 지정하며 파일 기반 또는 LDAP 중 하나입니다.
- `imq.authentication.client.response.timeout`. 클라이언트가 브로커로부터의 인증 요청에 응답할 때까지 시스템이 대기할 시간(초)을 지정합니다.
- `imq.accesscontrol.enabled`. 액세스 제어 등록 정보 파일에 지정된 대로 인증된 사용자가 연결 서비스를 사용하거나 특정 대상에 대해 특정 Message Queue 작업을 수행할 권한이 있는지에 대해 시스템이 확인하는지 여부를 표시합니다.
- `imq.service_name.accesscontrol.enabled`. 명명된 연결 서비스의 액세스 제어를 설정하며(true/false), 브로커 전체 설정에 우선합니다.

- `imq.accesscontrol.file.filename`. 브로커 인스턴스가 지원하는 모든 연결 서비스에 대한 액세스 제어 등록 정보 파일의 이름을 지정합니다.
- `imq.service_name.accesscontrol.file.filename`. 브로커 인스턴스의 명명된 연결 서비스에 대한 액세스 제어 등록 정보 파일의 이름을 지정합니다.
- `imq.passfile.enabled`. 보안 통신을 위한 사용자 비밀번호(SSL, LDAP, JDBC™)가 파일에 지정되는지 여부를 지정합니다.
- `imq.passfile.dirpath`. `passfile`을 포함하는 디렉토리 경로를 지정합니다.
- `imq.passfile.name`. `passfile` 이름을 지정합니다.
- `imq.keystore.property_name`. SSL 기반 서비스에 대해 SSL 키 저장소와 관련된 보안 등록 정보를 지정합니다. [323페이지의 표 14-9](#)를 참조하십시오.

이러한 등록 정보에 대한 자세한 내용은 [320페이지의 표 14-8](#)을 참조하십시오.

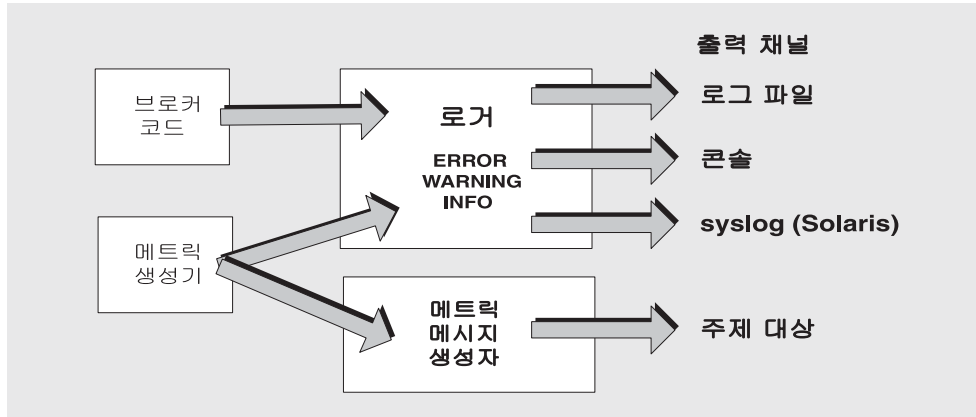
## 모니터링 서비스

브로커는 로그 작업을 모니터링하고 진단할 다양한 구성 요소를 포함합니다. 다음은 그러한 구성 요소의 예입니다.

- 데이터를 생성하는 구성 요소(이벤트를 기록하는 브로커 코드 및 메트릭 생성자)
- 여러 출력 채널을 통해 정보를 기록하는 로거 구성 요소("로거" 참조)
- 메트릭 정보를 포함하는 JMS 메시지를 JMS 모니터링 클라이언트가 사용할 수 있도록 주체 대상에게 보내는 메시지 생성자

[그림 4-5](#)는 그러한 일반 체계를 보여줍니다.

그림 4-5 모니터링 서비스 지원



### 메트릭 생성자

메트릭 생성자는 브로커 내부 및 외부로의 메시지 흐름, 브로커 메모리의 메시지 수 및 이 메시지가 사용하는 메모리, 열려 있는 연결 수, 사용 중인 스레드 수 등과 같은 브로커 활동 정보를 제공합니다.

메트릭 데이터 생성을 설정 또는 해제하고 메트릭 보고서 생성 빈도를 지정할 수 있습니다.

### 로거

Message Queue 로거는 브로커 코드 및 메트릭 생성자가 생성한 정보를 가져와서 표준 출력(콘솔), 로그 파일, syslog 데몬 프로세스(Solaris™ 운영 체제인 경우) 등과 같은 여러 출력 채널에 기록합니다.

로거에서 수집된 정보의 유형과 각 출력 채널에 기록된 유형을 지정할 수 있습니다.

예를 들어, 로거 수준을 지정하여 오류(ERROR), 오류 및 경고(WARNING) 또는 오류, 경고 및 정보(INFO) 등, 로거가 수집하는 정보 유형을 결정할 수 있습니다.

각 출력 채널마다 로거에 대해 설정된 범주 중 어느 것이 해당 채널에 기록될 것인지 지정할 수 있습니다. 예를 들어, 로거 수준이 INFO로 설정된 경우 콘솔에는 오류와 경고만, 로그 파일에는 정보(메트릭 데이터)만 기록하게 지정할 수 있습니다.

로그 파일을 사용하는 경우 로그 파일을 닫고 출력을 새 파일로 롤오버하는 지점을 지정할 수 있습니다. 새 롤오버 로그 파일을 작성하면서 가장 최신의 로그 파일 9개로 구성된 아카이브를 보존합니다.

로거 구성에 대한 자세한 내용은 [205페이지](#)의 "[브로커 로깅 구성 및 사용](#)"을 참조하십시오. Solaris syslog 구성 및 사용에 대한 자세한 내용은 `syslog(1M)`, `syslog.conf(4)` 및 `syslog(3C)` 설명서 페이지를 참조하십시오.

### 메트릭 메시지 생성자(엔터프라이즈판)

메시지 생성자 구성 요소는 메시지 생성자 구성 요소로부터 정기적으로 정보를 받아서 메시지에 기록한 다음 메트릭 주제 대상에게 보냅니다. 메트릭 메시지를 받는 대상은 메시지에 포함된 정보의 유형에 따라 달라집니다.

다섯 개의 메트릭 주제 대상이 있으며 [표 4-3](#)에 그 이름과 함께 각 대상에 전달되는 메트릭 메시지 유형이 표시되어 있습니다.

**표 4-3** 메트릭 주제 대상

주제 대상 이름	메트릭 메시지 유형
<code>mq.metrics.broker</code>	브로커 메트릭
<code>mq.metrics.jvm</code>	Java 가상 머신 메트릭
<code>mq.metrics.destination_list</code>	대상 및 대상 유형 목록
<code>mq.metrics.destination.queue. monitoredDestinationName</code>	지정된 이름의 대기열에 대한 대상 메트릭
<code>mq.metrics.destination.topic. monitoredDestinationName</code>	지정된 이름의 주제에 대한 대상 메트릭

이러한 메트릭 주제 대상에 가입한 Message Queue 클라이언트는 대상의 메시지를 사용하고 메트릭 정보를 처리합니다. 예를 들어, 클라이언트가 `mq.metrics.broker` 대상에 가입하여 브로커의 총 메시지 수와 같은 정보를 받아서 처리할 수 있습니다.

메트릭 메시지 생성자는 메트릭 데이터에 해당하는 이름-값 쌍이 포함된 메시지 (`MapMessage` 유형)를 만드는 내부 Message Queue 클라이언트입니다. 이러한 메시지는 해당 메트릭 주제 대상에 한 명 이상의 가입자가 있는 경우에만 생성됩니다.

메트릭 메시지 생성자가 생성하는 메시지는 `MapMessage` 유형입니다. 이 메시지는 포함된 메트릭 유형에 따라 여러 이름/값 쌍으로 구성됩니다. 각 이름/값 쌍은 메트릭 수량과 값에 해당합니다.

예를 들어, 브로커 메트릭 메시지는 브로커를 오가는 메시지 수와 이 메시지의 크기, 현재 메모리에 있는 메시지의 수 및 크기 등에 대한 값을 포함합니다. 각 메트릭 메시지 유형에서 보고하는 메트릭 수량에 대한 자세한 내용은 *Message Queue Developer's Guide for Java Clients*를 참조하십시오. 이 설명서에서는 메트릭 메시지를 사용할 Message Queue 클라이언트 작성 방법을 설명합니다.

메트릭 메시지 본문에 포함된 메트릭 정보 이외에 각 메시지의 헤더에는 다음과 같은 정보를 제공하는 등록 정보가 있습니다.

- 메시지 유형
- 메시지를 보낸 브로커의 호스트, 포트 및 주소
- 메트릭 샘플을 가져온 시간

이 등록 정보는 유형이 다르거나 서로 다른 브로커에서 가져온 메트릭 메시지를 처리하는 Message Queue 클라이언트 응용 프로그램에 유용합니다.

## 모니터링 서비스 등록 정보

다음은 브로커의 정보 생성, 로깅 및 메트릭 메시지 생성을 설정할 때 구성 가능한 등록 정보입니다.

- `imq.metrics.enabled`. 메트릭 정보를 로거에 기록할지 여부를 지정합니다.
- `imq.metrics.interval`. 메트릭 로깅이 사용 가능한 경우 메트릭 정보가 로거에 기록되는 간격(초)을 지정합니다.
- `imq.log.level`. 로거 수준, 즉 출력 채널에 기록 가능한 출력 범주를 지정합니다.
- `imq.log.file.output`. 로그 파일에 기록하는 로깅 정보의 범주를 지정합니다.
- `imq.log.file.dirpath`. 로그 파일을 포함하는 디렉토리 경로를 지정합니다.
- `imq.log.file.filename`. 로그 파일의 이름을 지정합니다.
- `imq.log.file.rolloverbytes`. 새 로그 파일로 출력을 롤오버할 로그 파일 크기 (바이트)를 지정합니다.
- `imq.log.file.rolloversecs`. 새 로그 파일로 출력을 롤오버할 로그 파일 표시 시간 (초)을 지정합니다.
- `imq.log.console.output`. 콘솔에 기록하는 로깅 정보의 범주를 지정합니다.
- `imq.log.console.stream`. 콘솔 출력을 `stdout (OUT)` 또는 `stderr (ERR)`에 기록할 것인지 지정합니다.
- `imq.log.syslog.facility`. (Solaris에만 적용) Message Queue 브로커가 어떤 syslog 기능으로 로깅할 것인지 지정합니다.



- `imq.log.syslog.logpid`. (Solaris에만 적용) 브로커 프로세스 아이디를 메시지와 함께 기록할지 여부를 지정합니다.
- `imq.log.syslog.logconsole`. (Solaris에만 적용) 메시지를 `syslog`에 보낼 수 없을 경우 시스템 콘솔에 메시지를 기록할 것인지 여부를 지정합니다.
- `imq.log.syslog.identity`. (Solaris에만 적용) `syslog`에 기록되는 모든 메시지의 앞에 추가될 아이디 문자열을 지정합니다.
- `imq.log.syslog.output`. (Solaris에만 적용) `syslogd(1M)`에 기록하는 로깅 정보의 범주를 지정합니다.
- `imq.log.timezone`. 로그 타임스탬프의 표준 시간대를 지정합니다.
- `imq.metrics.topic.enabled`. 메트릭 메시지 생성이 가능하지 여부를 지정합니다.
- `imq.metrics.topic.interval`. 메트릭 메시지를 생성하는 간격(초)을 지정합니다.
- `imq.metrics.topic.persist`. 메트릭 메시지가 지속성인지 여부를 지정합니다.
- `imq.metrics.topic.timetolive`. 메트릭 주제 대상에 보낸 메트릭 메시지의 수명(초)을 지정합니다.
- `imq.destination.logDeadMsgs`. 브로커가 사용 불가능 메시지를 삭제하거나 사용 불가능 메시지를 사용 불가능 메시지 대기열에 넣을 때마다 로그에 메시지를 기록할지 여부를 지정합니다.

이러한 등록 정보에 대한 자세한 내용은 [324페이지의 표 14-10](#)을 참조하십시오.

## 구성 파일 정보

브로커 구성 파일은 브로커를 구성하는 데 사용됩니다. [부록 A, "운영 체제별 Message Queue 데이터의 위치"](#)에는 운영 체제별로 브로커 구성 파일이 있는 디렉토리가 나열되어 있습니다.

이 디렉토리에는 다음과 같은 파일이 저장됩니다.

- 시작할 때 로드되는 **기본 구성 파일**. 이 파일은 `default.properties`이며 편집할 수 없습니다. 이 파일을 참조하여 기본 설정을 확인하고 변경할 등록 정보의 정확한 이름을 찾을 수 있습니다.
- Message Queue를 설치할 때 지정한 모든 등록 정보를 포함하는 **설치 구성 파일**. 이 파일은 `install.properties`이며 설치하고 나면 편집할 수 없습니다.

## 인스턴스 구성 파일

브로커를 처음으로 실행하면 인스턴스 구성 파일이 생성됩니다. 이 인스턴스 구성 파일을 사용하여 브로커의 해당 인스턴스에 대한 구성 등록 정보를 지정할 수 있습니다.

인스턴스 구성 파일은 다음과 같이 구성 파일이 연결되어 있는 브로커 인스턴스의 이름 (*instanceName*)으로 식별되는 디렉토리에 저장됩니다.

```
.../instances/instanceName/props/config.properties
```

`instances` 디렉토리의 위치는 [부록 A, "운영 체제별 Message Queue 데이터의 위치"](#)를 참조하십시오.

---

**주**            `.../instances/instanceName` 디렉토리 및 인스턴스 구성 파일은 해당 브로커 인스턴스를 만든 사람이 소유합니다. 브로커 인스턴스는 항상 이 사용자가 다시 시작해야 합니다.

---

인스턴스 구성 파일은 브로커 인스턴스에 의해 유지 관리됩니다. 관리 도구를 사용하여 구성을 변경하면 인스턴스 구성 파일이 수정됩니다. 인스턴스 구성 파일을 직접 편집하여 구성을 변경할 수도 있습니다([98페이지의 "인스턴스 구성 파일 편집"](#) 참조). 그렇게 하려면 `.../instances/instanceName` 디렉토리의 소유자이거나 루트로 로그인하여 디렉토리에 관한 권한을 변경해야 합니다.

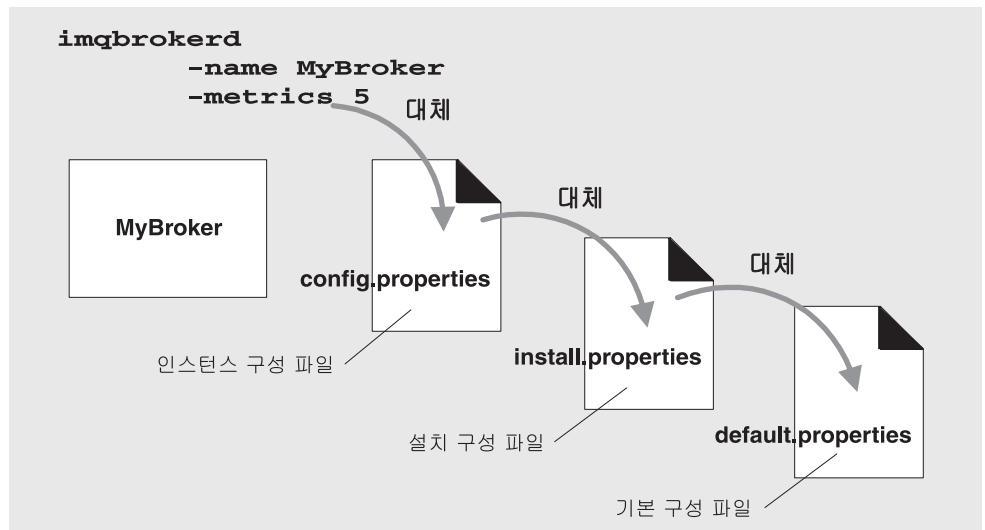
클러스터에서 브로커 인스턴스를 연결하는 경우에는 *클러스터 구성 파일*을 사용하여 클러스터 구성 정보를 지정해야 할 수도 있습니다. 자세한 내용은 [326페이지의 "클러스터 구성 등록 정보"](#)를 참조하십시오.

## 등록 정보 값 병합

시작할 때 브로커가 다른 구성 파일에 있는 등록 정보 값을 병합합니다. 설치 및 인스턴스 구성 파일에 있는 값을 사용하여 기본 구성 파일에 지정된 값을 대체합니다.

`imqbrokerd` 명령 옵션을 사용하면 결과 값을 대체할 수 있습니다. 이 체계가 [그림 4-6](#)에 표시되어 있습니다.

그림 4-6 브로커 구성 파일



## 등록 정보 이름 지정 구문

구성 파일의 모든 Message Queue 등록 정보 정의에는 다음과 같은 이름 지정 구문이 사용됩니다.

```
propertyName=value [ , value1 ] ...
```

예를 들어, 다음 항목은 브로커가 추가 메시지를 거부하기 전까지 메모리 및 영구 저장소에 최대 50,000개의 메시지를 저장하도록 지정합니다.

```
imq.system.max_count=50000
```

다음 항목은 매일(86400초) 새 로그 파일을 작성하도록 지정합니다.

```
imq.log.file.rolloversecs=86400
```

307페이지의 14장, "브로커 등록 정보 참조"에는 브로커 구성 등록 정보와 해당 기본값이 나열되어 있습니다.

## 인스턴스 구성 파일 편집

브로커 인스턴스를 처음으로 실행하면 config.properties 파일이 자동으로 작성됩니다. 이 인스턴스 구성 파일을 편집하여 해당 브로커 인스턴스의 동작과 자원을 사용자 정의할 수 있습니다.

브로커 인스턴스는 시작 시에만 config.properties 파일을 읽습니다. config.properties 파일을 영구히 변경하려면 다음 중 하나를 수행할 수 있습니다.

- 관리 도구를 사용합니다. imqcmd를 사용해서 설정할 수 있는 등록 정보에 대한 내용은 308페이지의 표 14-1을 참조하십시오.
- 브로커 인스턴스가 종료된 동안에 config.properties 파일을 편집한 후 인스턴스를 다시 시작합니다(Solaris와 Linux 운영 체제에서는 브로커 인스턴스를 처음으로 시작한 사용자만 config.properties 파일을 편집할 권한을 가짐).

표 14-1에는 브로커 인스턴스 구성 등록 정보와 기본값이 알파벳순으로 나와 있습니다. 각 등록 정보의 의미와 사용에 대한 자세한 내용은 지정된 상호 참조 부분을 보십시오.

## 명령줄에 구성 옵션 입력

브로커를 시작할 때 또는 시작한 후에 명령줄에 브로커 구성 옵션을 입력할 수 있습니다.

시작할 때 `mqbrokerd` 명령을 사용하여 브로커 인스턴스를 시작합니다. 명령의 `-D` 옵션을 사용하여 브로커 구성 등록 정보와 값을 지정할 수 있습니다. 브로커를 Windows 서비스로 시작하는 경우 `mqsvcadm` 명령에서 `-args` 옵션을 사용하여 시작 구성 등록 정보를 지정합니다.

브로커 인스턴스가 실행 중일 때 특정 브로커 등록 정보를 설정할 수도 있습니다. 실행 중인 브로커의 구성을 수정하려면 `mqcmd update bkr` 명령을 사용합니다.

시작 구성에 대한 자세한 내용은 3장, "브로커 및 클라이언트 시작"의 67페이지의 "대화식으로 브로커 시작"에 있는 예제를 참조하십시오.

실행 중인 브로커의 구성을 수정하는 방법은 5장, "브로커 관리" 및 14장, "브로커 등록 정보 참조"를 참조하십시오.

## 영구 저장소 설정

Message Queue 브로커에는 지속성 정보의 기록과 검색을 관리하는 지속성 관리자 구성 요소가 있습니다. 지속성 관리자는 기본적으로 기본 제공되는 파일 기반 데이터 저장소에 액세스하도록 구성되어 있지만 JDBC 호환 드라이버를 통해 액세스할 수 있는 데이터 저장소를 플러그 인으로 사용하도록 재구성할 수도 있습니다.

Message Queue 데이터 저장소에는 트랜잭션, 메시지, 영구 가입 및 물리적 대상에 대한 정보가 있으며 확인에 대한 메시지 상태 정보도 있습니다.

이 장에서는 영구 저장소를 사용하도록 브로커를 구성하는 방법에 대해 설명합니다.

이 장은 다음 항목으로 구성되어 있습니다.

- 100페이지의 "파일 시스템 저장소 구성"
- 100페이지의 "JDBC 저장소 구성"
- 104페이지의 "지속성 데이터 보안"

## 파일 시스템 저장소 구성

브로커 인스턴스를 만들면 파일 시스템 데이터 저장소가 자동으로 생성됩니다. 이 저장소는 해당 브로커의 인스턴스 디렉토리에 있습니다. 운영 체제별로 위치가 다르므로 영구 저장소의 정확한 위치는 [부록 A, "운영 체제별 Message Queue 데이터의 위치"](#)를 참조하십시오.

기본적으로 Message Queue는 디스크에 대한 비동기 쓰기 작업을 수행합니다. 운영 체제는 성능 향상을 위해 이러한 작업을 버퍼링합니다. 하지만 쓰기 작업 중에 운영 체제에 예기치 않은 오류가 발생할 경우 메시지가 손실될 수 있습니다. 안정성 향상을 위해 Message Queue에서 디스크에 동기식으로 기록하도록 지정할 수 있지만 이 옵션을 사용할 경우 성능이 떨어집니다. 디스크에 대한 동기식 기록을 지정하려면 `imq.persist.file.sync` 브로커 등록 정보를 설정합니다. 이 등록 정보에 대한 자세한 내용은 [317페이지의 표 14-6](#)을 참조하십시오.

브로커 인스턴스를 시작할 때 `imqbrokerd -reset` 옵션을 사용하여 파일 시스템 저장소를 지울 수 있습니다. 이 옵션 및 하위 옵션에 대한 자세한 내용은 [282페이지의 표 13-2](#)를 참조하십시오.

## JDBC 저장소 구성

JDBC 기반 지속성을 사용하도록 브로커를 구성하려면 브로커 인스턴스 구성 파일에서 JDBC 관련 등록 정보를 설정하고 적절한 데이터베이스 스키마를 만듭니다. Message Queue 데이터베이스 관리자 유틸리티(`imqdbmgr`)는 JDBC 드라이버와 브로커 구성 등록 정보를 사용하여 데이터베이스를 만들고 관리합니다.

이 장의 절차에서는 Java 2 Platform, Enterprise Edition(J2EE) SDK와 함께 제공되는 PointBase DBMS를 예로 사용하여 설명합니다. 버전 1.4는 [java.sun.com](http://java.sun.com)에서 다운로드할 수 있습니다. 예에서는 클라이언트/서버 버전 대신 PointBase 내장 버전을 사용합니다. 절차에서는 PointBase 예의 경로 이름과 등록 정보 이름을 사용하여 지침을 소개합니다. 이러한 지침은 "Example:"로 표시됩니다.

Oracle 및 PointBase의 예제 구성을 사용할 수 있습니다. 예제 파일 위치는 [부록 A, "운영 체제별 Message Queue 데이터의 위치"](#)를 참조하십시오. 운영 체제별 정보가 나열된 표에서 "예제 응용 프로그램 및 구성"의 위치를 찾으십시오.

또한 PointBase 내장 버전, PointBase 서버 버전 및 Oracle의 예는 인스턴스 구성 파일 `config.properties`에서 주석 처리된 값으로 제공됩니다.

## JDBC로 액세스할 수 있는 데이터 저장소 플러그인

JDBC로 액세스할 수 있는 데이터 저장소를 플러그인하는 작업은 단 몇 단계로 이루어집니다.

### ▶ JDBC로 액세스할 수 있는 데이터 저장소를 플러그인하는 방법

1. 브로커의 구성 파일에서 JDBC 관련 등록 정보를 설정합니다.

[317페이지의 "JDBC 기반 지속성"](#)에 설명된 등록 정보를 참조하십시오.

2. 다음 경로에 위치한 JDBC 드라이버 jar 파일의 사본 또는 심볼릭 링크를 넣습니다.

`/usr/share/lib/imq/ext/(Solaris)`

`/opt/sun/mq/share/lib/(Linux)`

`IMQ_VARHOME\lib\ext(Windows)`

복사 예(Solaris):

```
% cp j2eeSDK_install_directory/pointbase/lib/pointbase.jar
/usr/share/lib/imq/ext
```

심볼릭 링크 예(Solaris):

```
% ln -s j2eeSDK_install_directory/lib/pointbase/pointbase.jar
/usr/share/lib/imq/ext
```

3. Message Queue 지속성에 필요한 데이터베이스 스키마를 만듭니다.

`imqdbmgr create all` 명령(내장 데이터베이스의 경우) 또는 `imqdbmgr create tbl` 명령(외부 데이터베이스의 경우)을 사용합니다. [104페이지의 "데이터베이스 관리자 유틸리티\(imqdbmgr\)"](#)를 참조하십시오.

예/:

- a. `imqdbmgr`가 위치한 디렉토리로 변경합니다.

`cd /usr/bin(Solaris)`

`cd /opt/sun/mq/bin(Linux)`

`cd IMQ_HOME/bin(Windows)`

- b. `imqdbmgr` 명령을 입력합니다.

`imqdbmgr create all`

---

**주** 내장 데이터베이스를 사용하는 경우 다음 디렉토리에 해당 데이터베이스를 만드는 것이 좋습니다.

`.../instances/instanceName/dbstore/databasename`

내장 데이터베이스가 사용자 아이디와 비밀번호로 보호되지 않는 경우에는 파일 시스템 권한으로 보호합니다. 브로커에서 데이터베이스를 읽고 쓸 수 있게 하려면 브로커를 실행하는 사용자가 `imqdbmgr` 명령을 사용해서 내장 데이터베이스를 만든 사용자와 같아야 합니다([104페이지의 "데이터베이스 관리자 유틸리티\(imqdbmgr\)"](#) 참조).

---

## JDBC 관련 브로커 등록 정보

브로커의 인스턴스 구성 파일은 다음과 같이 구성 파일과 연관된 브로커 인스턴스 이름으로 식별되는 디렉토리에 있습니다([부록 A, "운영 체제별 Message Queue 데이터의 위치"](#) 참조).

`.../instances/instanceName/props/config.properties`

파일이 아직 없는 경우에는 Message Queue에서 파일을 만들 수 있도록 `-name instanceName` 옵션을 사용하여 브로커를 시작해야 합니다.

[317페이지의 "JDBC 기반 지속성"](#)에는 JDBC로 액세스할 수 있는 데이터 저장소를 플러그인할 때 설정해야 하는 구성 등록 정보가 있습니다. 이 절의 끝에는 이러한 등록 정보에 대한 요약 설명이 있습니다. 플러그인 지속성을 사용하는 각 브로커 인스턴스의 구성 파일(`config.properties`)에서 이 등록 정보를 설정합니다.

인스턴스 구성 등록 정보를 사용하면 Message Queue 데이터베이스 스키마를 만드는 SQL 코드를 사용자 정의할 수 있습니다. 각 데이터베이스 테이블을 만드는 SQL 코드를 지정하는 구성 가능한 등록 정보가 있습니다. 플러그인 데이터베이스에서 사용하는 데이터 유형을 제대로 지정하려면 이러한 등록 정보가 필요합니다.

데이터베이스 공급업체 간에 정확한 SQL 구문에 대해 호환되지 않는 점이 있으므로 데이터베이스 공급업체의 해당 설명서를 확인하여 [317페이지의 표 14-7](#)의 등록 정보를 적절하게 조정해야 합니다. 예를 들어, PointBase 데이터베이스의 경우 `IMQMSG35` 테이블에서 `MSG` 열(`imq.persist.jdbc.table.IMQMSG35` 등록 정보 참조)의 최대 허용 길이를 조정해야 하는 경우가 있습니다.

브로커 구성 등록 정보와 마찬가지로, `-D` 명령줄 옵션을 사용해서 값을 설정할 수 있습니다. 데이터베이스에 데이터베이스 특정 등록 정보를 설정해야 하는 경우에도 브로커(`imqbrokerd`) 또는 데이터베이스 관리자 유틸리티(`imqdbmgr`)를 시작할 때 `-D` 명령줄 옵션을 사용하여 설정할 수 있습니다.



예:

PointBase 내장 데이터베이스의 예에서는 데이터베이스 연결 URL에 데이터베이스의 절대 경로를 지정하는 대신 다음과 같이 `-D` 명령줄 옵션을 사용하여 PointBase 시스템 디렉토리를 정의할 수 있습니다.

```
-Ddatabase.home=IMQ_VARHOME/instances/instanceName/dbstore
```

이 경우 다음과 같이 URL을 지정하여 데이터베이스를 만들 수 있습니다.

```
img.persist.jdbc.createdburl=jdbc:pointbase:embedded:dbName;new
```

다음과 같이 URL을 지정하여 데이터베이스를 열 수 있습니다.

```
img.persist.jdbc.opendburl=jdbc:pointbase:embedded:dbName
```

다음은 JDBC 관련 등록 정보에 대한 요약 설명입니다.

- `img.persist.store`. 파일 기반 또는 JDBC 기반 데이터 저장소를 지정합니다.
- `img.persist.jdbc.brokerid`. 데이터베이스 테이블 이름에 추가하여 고유한 이름으로 만들어 주는 브로커 인스턴스 식별자를 지정합니다.
- `img.persist.jdbc.driver`. 데이터베이스에 연결할 JDBC 드라이버의 Java 클래스 이름을 지정합니다.
- `img.persist.jdbc.opendburl`. 기존 데이터베이스에 대한 연결을 여는 데이터베이스 URL을 지정합니다.
- `img.persist.jdbc.createdburl`. 데이터베이스를 만들기 위해 연결을 여는 데이터베이스 URL을 지정합니다.
- `img.persist.jdbc.closedburl`. 브로커를 종료할 때 현재 데이터베이스 연결을 종료하는 데이터베이스 URL을 지정합니다.
- `img.persist.jdbc.user`. 필요한 경우 데이터베이스 연결을 열 때 사용되는 사용자 아이디를 지정합니다.
- `img.persist.jdbc.needpassword`. 브로커에 액세스할 때 데이터베이스에 비밀번호가 필요한지 여부를 지정합니다.
- `img.persist.jdbc.password`. 필요한 경우 데이터베이스 연결을 여는 비밀번호를 지정합니다.
- `img.persist.jdbc.table.IMQSV35`. 버전 테이블을 만들 때 사용하는 SQL 명령
- `img.persist.jdbc.table.IMQCCREC35`. 구성 변경 레코드 테이블을 만들 때 사용하는 SQL 명령
- `img.persist.jdbc.table.IMQDEST35`. 대상 테이블을 만들 때 사용하는 SQL 명령

- `imq.persist.jdbc.table.IMQINT35`. 인터레스트 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQMSG35`. 메시지 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQPROPS35`. 등록 정보 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQILIST35`. 인터레스트 상태 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQTXN35`. 트랜잭션 테이블을 만들 때 사용하는 SQL 명령
- `imq.persist.jdbc.table.IMQTACK35`. 트랜잭션 확인 테이블을 만들 때 사용하는 SQL 명령

이러한 등록 정보에 대한 자세한 내용은 [14장, "브로커 등록 정보 참조"](#)를 참조하십시오.

### 데이터베이스 관리자 유틸리티(imqdbmgr)

Message Queue에는 지속성에 필요한 스키마를 설정하는 데이터베이스 관리자 유틸리티(imqdbmgr)가 포함되어 있습니다. 이 유틸리티는 테이블이 손상되거나 데이터 저장소로 다른 데이터베이스를 사용하는 경우에 Message Queue 데이터베이스 테이블을 삭제하는 용도로도 사용할 수 있습니다.

`imqcmd` 명령의 구문, 하위 명령 및 옵션에 대한 자세한 내용은 [13장, "명령 참조"](#)를 참조하십시오.

## 지속성 데이터 보안

영구 저장소는 정보 가운데 일시적으로 저장되는 메시지 파일을 포함할 수 있습니다. 이러한 메시지에는 소유 정보가 포함될 수 있기 때문에 인증되지 않은 액세스로부터 데이터 저장소를 보호해야 합니다. 이 절에서는 기본 제공 파일 저장소 또는 JDBC 저장소에서 데이터를 보호하는 방법에 대해 설명합니다.

## 기본 제공(파일 기반) 영구 저장소

기본 제공 지속성을 사용하는 브로커는 다음과 같이 운영 체제에 따라 다른 위치에 있는 플랫폼 파일 데이터 저장소에 지속성 데이터를 작성합니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

```
.../instances/instanceName/fs350/
```

여기서 *instanceName*은 브로커 인스턴스를 식별하는 이름입니다.

*instanceName/filestore/* 디렉토리는 브로커 인스턴스가 처음으로 시작될 때 생성됩니다. 이 디렉토리 보안 절차는 브로커가 실행 중인 운영 체제에 따라 다릅니다.

**Solaris 및 Linux** `IMQ_VARHOME/instances/instanceName/filestore/` 디렉토리에 대한 사용 권한은 브로커 인스턴스를 시작한 사용자의 `umask`에 따라 다릅니다. 따라서, `umask`를 적절하게 설정하여 브로커 인스턴스 시작 권한과 지속성 파일 읽기 권한을 제한할 수 있습니다. 또는, 관리자(슈퍼유저)가 `IMQ_VARHOME/instances` 디렉토리에 대한 사용 권한을 700으로 설정하여 지속성 데이터를 안전하게 보호할 수 있습니다.

**Windows** 사용 중인 Windows 운영 체제에서 제공하는 메커니즘을 사용하여 `IMQ_VARHOME/instances/instanceName/filestore/` 디렉토리에 대한 사용 권한을 설정할 수 있습니다. 여기에는 일반적으로 디렉토리에 대한 등록 정보 대화 상자 열기가 포함됩니다.

## 플러그 인(JDBC) 영구 저장소

플러그 인 지속성을 사용하는 브로커는 JDBC 호환 데이터베이스에 지속성 데이터를 작성합니다.

데이터베이스 서버에서 관리하는 데이터베이스(예: Oracle 데이터베이스)의 경우 Message Queue 데이터베이스 테이블(이름이 "IMQ"로 시작하는 테이블)에 액세스할 아이디와 비밀번호를 만드는 것이 좋습니다. 데이터베이스에서 개별 테이블을 보호할 수 없는 경우 Message Queue 브로커 전용 데이터베이스를 만듭니다. 사용자 아이디/비밀번호 액세스 작성 방법에 대한 자세한 내용은 데이터베이스 공급업체 설명서를 참조하십시오.

브로커가 데이터베이스 연결을 여는 데 필요한 사용자 아이디와 비밀번호는 브로커 구성 등록 정보에서 제공될 수 있습니다. 브로커 시작 시 사용자 아이디와 비밀번호를 명령줄 옵션으로 제공하는 것이 보다 안전합니다(Message Queue 관리 설명서, 부록 A, "플러그 인 지속성 설정" 참조).

브로커가 데이터베이스의 JDBC™ 드라이버를 통해 직접 액세스하는 내장 데이터베이스의 경우 "기본 제공(파일 기반) 영구 저장소"에서 설명한 대로 대개는 지속성 데이터가 저장되는 디렉토리에 대한 파일 권한을 설정하여 보안을 제공합니다. 브로커와 imqdbmgr 유틸리티를 모두 사용하여 데이터베이스를 읽기 및 쓰기 가능하게 하려면 해당 브로커와 유틸리티를 모두 동일한 사용자가 실행해야 합니다.

# 브로커 관리

이 장에서는 브로커 및 해당 서비스 관리에 관련된 기본 작업을 수행하는 방법을 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 108페이지의 "전제 조건"
- 108페이지의 "imqcmd 명령 유틸리티 사용"
- 110페이지의 "도움말 표시"
- 111페이지의 "제품 버전 표시"
- 111페이지의 "브로커 정보 표시"
- 112페이지의 "브로커 등록 정보 업데이트"
- 113페이지의 "브로커 일시 중지 및 다시 시작"
- 114페이지의 "브로커 종료 및 다시 시작"
- 115페이지의 "브로커 메트릭 표시"
- 116페이지의 "연결 서비스 관리"
- 121페이지의 "연결 정보 얻기"
- 122페이지의 "영구 가입 관리"
- 123페이지의 "트랜잭션 관리"

이 장에서 브로커 관리와 관련된 모든 항목을 설명하지는 않습니다. 많은 추가 항목들은 다음 별도의 장에서 설명합니다.

- 브로커의 물리적 대상 관리. 물리적 대상 만들기, 표시, 업데이트 및 완전 삭제 방법과 사용 불능 메시지 대기열 사용 방법은 [6장](#), "물리적 대상 관리"를 참조하십시오.
- 브로커에 대한 보안 설정. 사용자 인증, 액세스 제어, 암호화, 비밀번호 파일, 감사 로깅 등과 같은 항목에 대한 자세한 내용은 [7장](#), "보안 관리"를 참조하십시오.

## 전제 조건

imqcmd 및 imqusermgr 명령을 사용하여 브로커를 관리합니다. 브로커를 관리하기 전에 다음을 수행해야 합니다.

- imqbrokerd 명령을 사용해서 브로커를 시작합니다. 브로커가 실행될 때까지는 다른 명령을 사용할 수 없습니다.
- Message Queue 관리 사용자를 설정할지 아니면 기본 계정을 사용할지를 결정합니다. 관리 명령을 사용하려면 아이디와 비밀번호를 지정해야 합니다.

Message Queue를 설치하면 기본 플랫폼 파일 사용자 저장소가 설치됩니다. 저장소는 admin 및 guest 사용자의 두 가지 기본 항목과 함께 제공됩니다. Message Queue를 테스트하는 경우 기본 아이디와 비밀번호(admin/admin)를 사용하여 imqcmd 유틸리티를 실행할 수 있습니다.

작업 시스템을 설정하는 경우 관리 사용자에 대한 인증과 권한 부여를 설정해야 합니다. 파일 기반 사용자 저장소 설정 또는 LDAP 디렉토리 서버 사용 구성에 대한 자세한 내용은 [7장](#), "[보안 관리](#)"를 참조하십시오. 작업 환경에서는 보안을 위해 기본 아이디와 비밀번호를 사용하지 않는 것이 좋습니다.

- 브로커에 대한 보안 연결을 사용하려면 대상 브로커 인스턴스에서 ssladmin 서비스를 설정하고 활성화합니다. 자세한 내용은 [159페이지의 "SSL 기반 서비스 작업"](#)을 참조하십시오.

## imqcmd 명령 유틸리티 사용

imqcmd 명령 유틸리티를 사용하여 브로커와 해당 서비스를 관리할 수 있습니다.

imqcmd 명령의 구문, 하위 명령 및 옵션에 대한 자세한 내용은 [279페이지의 13장](#), "[명령 참조](#)"를 참조하십시오. 물리적 대상 관리에 대한 자세한 내용은 별도의 장으로 제공되는 [329페이지의 15장](#), "[물리적 대상 등록 정보 참조](#)"를 참조하십시오.

## 아이디 및 비밀번호 지정

각 imqcmd 명령은 사용자 저장소에 대해 인증되므로 아이디와 비밀번호가 필요합니다. 다음과 같은 경우만 예외입니다.

- `-h` 또는 `-H` 옵션을 사용하여 도움말 명령을 표시하는 명령
- `-v` 옵션을 사용하여 제품 버전을 표시하는 명령

### 아이디 지정

`-u` 옵션을 사용하여 관리자 아이디를 지정합니다. 아이디를 생각하면 아이디를 묻는 명령 프롬프트가 표시됩니다.

이 장의 예제에서는 쉽게 이해할 수 있도록 기본 아이디 `admin`을 `-u` 옵션의 인수로 표시합니다. 작업 환경에서는 사용자 정의 아이디를 사용하게 됩니다.

### 비밀번호 지정

다음 중 한 방법으로 비밀번호를 지정합니다.

- 비밀번호 파일(`passfile`)을 만들고 해당 파일에 비밀번호를 입력합니다. 명령줄에서 `-passfile` 옵션을 사용하여 `passfile`의 이름을 제공합니다.
- 비밀번호를 묻는 명령 프롬프트를 표시하게 합니다. 이 방법은 입력하는 내용을 다른 사람이 볼 수 없는 한 가장 안전한 비밀번호 지정 방법입니다.

이전 버전 Message Queue에서는 명령줄에서 `-p` 옵션을 사용하여 비밀번호를 지정했습니다. 이 옵션은 부정적 의견이 많아서 이후 버전에서는 제거됩니다.

## 브로커 이름 및 포트 지정

imqcmd의 기본 브로커는 로컬 호스트에서 실행되고 있는 브로커이며 기본 포트는 7676입니다.

원격 호스트에서 실행 중인 브로커 및/또는 기본 포트가 아닌 포트에 명령을 실행하는 경우 `-b` 옵션을 사용하여 브로커의 호스트와 포트를 지정해야 합니다.

## 예

이 절의 예에서는 `imqcmd` 사용 방법을 설명합니다.

첫 번째 예에서는 `localhost`의 포트 7676에서 실행 중인 브로커의 등록 정보를 나열하므로 `-b` 옵션이 필요하지 않습니다. 이 명령에서는 기본 관리 아이디(`admin`)를 사용하고 비밀번호를 생략했기 때문에 비밀번호를 묻는 명령 프롬프트가 표시됩니다.

```
imqcmd query bkr -u admin
```

다음 예에서는 `myserver`의 포트 1564에서 실행 중인 브로커의 등록 정보를 나열합니다. 아이디는 `aladdin`입니다. 이 명령은 아이디 `aladdin`이 `admin` 그룹에 할당되도록 사용자 저장소를 업데이트해야 합니다.

```
imqcmd query bkr -b myserver:1564 -u aladdin
```

다음 예에서는 `localhost`의 포트 7676에서 실행 중인 브로커의 등록 정보를 나열합니다. 명령의 초기 시간 제한은 20초로 설정되고 시간 초과 이후의 시도 횟수는 7로 설정됩니다. 사용자의 비밀번호는 명령을 호출할 때 해당 디렉토리에 있는 `myPassfile`이라는 비밀번호 파일에 있습니다.

```
imqcmd query bkr -u admin -passfile myPassfile -rtm 20 -rtr 7
```

브로커에 보안 연결을 위해 이 예에 `-secure` 옵션을 포함시킬 수 있습니다. `-secure` 옵션은 서비스가 구성되어 시작된 경우 `imqcmd`가 `ssladmin` 서비스를 사용하게 합니다.

## 도움말 표시

`imqcmd` 명령 유틸리티에서 도움말을 표시하려면 `-h` 또는 `-H` 옵션을 사용합니다. 하위 명령은 사용하지 마십시오. 특정 하위 명령에 대한 도움말은 볼 수 없습니다.

예를 들어, 다음 명령은 `imqcmd`에 대한 도움말을 표시합니다.

```
imqcmd -H
```

`-h` 또는 `-H` 옵션을 포함하는 명령줄을 하위 명령 또는 다른 옵션과 함께 입력할 경우 명령 유틸리티는 `-h` 또는 `-H` 옵션만 처리합니다. 명령줄의 모든 다른 항목은 무시됩니다.



## 제품 버전 표시

Message Queue 제품 버전을 표시하려면 `-v` 옵션을 사용합니다. 예를 들면 다음과 같습니다.

```
imqcmd -v
```

`-v` 옵션을 포함하는 명령줄을 하위 명령 또는 다른 옵션과 함께 입력할 경우 명령 유틸리티는 `-v` 옵션만 처리합니다. 명령줄의 모든 다른 항목은 무시됩니다.

## 브로커 정보 표시

단일 브로커에 대한 정보를 쿼리 및 표시하려면 `query bkr` 하위 명령을 사용합니다.

다음은 `query bkr` 하위 명령의 구문입니다.

```
imqcmd query bkr -b hostName:port
```

이 하위 명령은 기본 브로커 또는 지정된 호스트 및 포트의 브로커에 해당하는 현재 등록 정보 설정을 나열합니다. 지정된 브로커에 연결된 실행 중인 브로커(멀티 브로커 클러스터에서)의 목록도 표시합니다.

예를 들면 다음과 같습니다.

```
imqcmd query bkr -u admin
```

이 명령은 비밀번호를 묻는 메시지를 표시한 후 다음과 같은 출력을 생성합니다.

Version	3.6
Instance Name	imgbroker
Primary Port	7676
Current Number of Messages in System	0
Current Total Message Bytes in System	0
Current Number of Messages in Dead Message Queue	0
Current Total Message Bytes in Dead Message Queue	0
Log Dead Messages	true
Truncate Message Body in Dead Message Queue	false
Max Number of Messages in System	unlimited (-1)
Max Total Message Bytes in System	unlimited (-1)
Max Message Size	70m
Auto Create Queues	true
Auto Create Topics	true
Auto Created Queue Max Number of Active Consumers	1

```

Auto Created Queue Max Number of Backup Consumers    0

Cluster Broker List (active)
Cluster Broker List (configured)
Cluster Master Broker
Cluster URL

Log Level                                             INFO
Log Rollover Interval (seconds)                     604800
Log Rollover Size (bytes)                           unlimited (-1)
    
```

## 브로커 등록 정보 업데이트

update bkr 하위 명령을 사용하여 다음과 같은 브로커 등록 정보를 업데이트할 수 있습니다.

- imq.autocreate.queue
- imq.autocreate.topic
- imq.autocreate.queue.maxNumActiveConsumers
- imq.autocreate.queue.maxNumBackupConsumers
- imq.cluster.url
- imq.destination.DMQ.truncateBody
- imq.destination.logDeadMsgs
- imq.log.level
- imq.log.file.rolloversecs
- imq.log.file.rolloverbytes
- imq.system.max\_count
- imq.system.max\_size
- imq.message.max\_size
- imq.portmapper.port

다음은 update bkr 하위 명령의 구문입니다.

```
imqcmd update bkr [-b hostName:port] -o attribute=value [-o attribute=value1]...
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에 해당하는 지정된 속성을 변경합니다.

이 등록 정보는 [14장](#), "[브로커 등록 정보 참조](#)"에서 자세히 설명합니다.

예를 들어, 다음 명령은 대기열 대상의 자동 작성을 해제합니다.

```
imqcmd update bkr -o "imq.autocreate.queue=false" -u admin
```

## 브로커 일시 중지 및 다시 시작

브로커를 시작한 후에는 `imqcmd` 하위 명령을 사용하여 브로커의 상태를 제어할 수 있습니다.

### 브로커 일시 중지

브로커를 일시 중지하면 브로커의 연결 서비스 스레드가 지연되고 브로커에서는 연결 포트의 수신을 중지합니다. 따라서 브로커가 더 이상 새로운 연결을 받아들이거나, 메시지를 수신하거나, 메시지를 디스패치할 수 없습니다.

그러나 브로커를 일시 중지하더라도 관리 연결 서비스는 지연되지 않으므로 브로커에 대한 메시지 흐름을 규제하는 데 필요한 관리 작업은 수행할 수 있습니다. 예를 들어, 특정 물리적 대상으로 메시지가 폭주하는 경우에는 브로커를 일시 중지한 후 다음과 같은 작업을 수행하여 문제 해결을 도울 수 있습니다.

- 메시지 소스 추적
- 물리적 대상 크기 제한
- 물리적 대상 완전 삭제

또한 브로커를 일시 중지해도 클러스터 연결 서비스는 지연되지 않습니다. 그러나 클러스터 내의 메시지 전달은 클러스터의 여러 브로커에서 수행하는 전달 기능에 따라 다릅니다.

다음은 `pause bkr` 하위 명령의 구문입니다.

```
imqcmd pause bkr [-b hostName:port]
```

이 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커를 일시 중지합니다.

다음 명령은 `myhost`의 포트 1588에서 실행 중인 브로커를 일시 중지합니다.

```
imqcmd pause bkr -b myhost:1588 -u admin
```

개별 연결 서비스와 개별 물리적 대상을 일시 중지할 수도 있습니다. 자세한 내용은 [120페이지](#)의 "[연결 서비스 일시 중지 및 다시 시작](#)" 및 [133페이지](#)의 "[물리적 대상 일시 중지 및 다시 시작](#)"을 참조하십시오.

## 브로커 다시 시작

브로커를 다시 시작하면 브로커의 서비스 스레드가 다시 활성화되고 브로커가 포트 수신을 다시 시작합니다.

다음은 `resume bkr` 하위 명령의 구문입니다.

```
imqcmd resume bkr [-b hostName:port]
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커를 다시 시작합니다.

다음 명령은 `localhost`의 포트 7676에서 실행 중인 브로커를 다시 시작합니다.

```
imqcmd resume bkr -u admin
```

## 브로커 종료 및 다시 시작

브로커를 종료하면 브로커 프로세스가 자연스럽게 종료됩니다. 브로커가 새로운 연결 및 메시지 수락을 중지하고 기존 메시지 전달을 완료한 다음 브로커 프로세스를 종료합니다.

다음은 `shutdown bkr` 하위 명령의 구문입니다.

```
imqcmd shutdown bkr [-b hostName:port]
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커를 종료합니다.

다음 명령은 `ctrlsrv`의 포트 1572에서 실행 중인 브로커를 종료합니다.

```
imqcmd shutdown bkr -b ctrlsrv:1572 -u admin
```

브로커를 종료하고 다시 시작할 수 있습니다. 다음은 `restart bkr` 하위 명령의 구문입니다.

```
imqcmd restart bkr [-b hostName:port]
```

이 하위 명령은 브로커를 처음 시작할 때 지정한 옵션을 사용하여 기본 브로커 또는 지정한 호스트 및 포트의 브로커를 종료하고 다시 시작합니다. 다른 옵션을 선택하려면 원하는 옵션을 지정하여 브로커를 종료하였다가 다시 시작합니다.

다음 명령은 `localhost`의 포트 7676에서 실행 중인 브로커를 다시 시작합니다.

```
imqcmd restart bkr -u admin
```

# 브로커 메트릭 표시

브로커에 대한 메트릭 정보를 표시하려면 `metrics bkr` 하위 명령을 사용합니다.

다음은 `metrics bkr` 하위 명령의 구문입니다.

```
imqcmd metrics bkr [-b hostName:port]
                  [-m metricType] [-int interval] [-msp numSamples]
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커 메트릭을 표시합니다.

`-m` 옵션을 사용하여 다음 메트릭 유형 중 하나를 지정하여 표시합니다.

- **ttl**      브로커에 유입되고 유출되는 메시지와 패킷에 대한 메트릭을 표시합니다(기본 메트릭 유형).
- **rts**      브로커에 유입 및 유출되는 메시지와 패킷의 메트릭을 초당 속도로 표시합니다.
- **cxn**      연결, 가상 메모리 힙 및 스레드를 표시합니다.

`-int` 옵션을 사용하여 메트릭 표시 간격(초)을 지정합니다. 기본값은 5초입니다.

`-msp` 옵션을 사용하여 출력에 표시되는 샘플 수를 지정합니다. 기본값은 무제한 수입니다(무한).

예를 들어, 다음 명령은 브로커에 메시지가 유입 및 유출되는 속도를 10초 간격으로 표시합니다.

```
imqcmd metrics bkr -m rts -int 10 -u admin
```

다음과 같은 출력이 표시됩니다.

Msgs/sec		Msg Bytes/sec		Pkts/sec		Pkt Bytes/sec	
In	Out	In	Out	In	Out	In	Out
0	0	27	56	0	0	38	66
10	0	7365	56	10	10	7457	1132
0	0	27	56	0	0	38	73
0	10	27	7402	10	20	1400	8459
0	0	27	56	0	0	38	73

`imqcmd`를 사용하여 브로커 메트릭을 보고하는 방법에 대한 자세한 내용은 [350페이지](#)의 "[브로커 전체 메트릭](#)"을 참조하십시오.

# 연결 서비스 관리

명령 유틸리티의 하위 명령을 사용하여 다음과 같은 연결 서비스 관리 작업을 수행할 수 있습니다.

- 연결 서비스 나열
- 연결 서비스 정보 표시
- 연결 서비스 등록 정보 업데이트
- 연결 서비스 메트릭 표시
- 연결 서비스 일시 중지 및 다시 시작

브로커는 응용 프로그램 클라이언트 및 관리 클라이언트 모두와의 연결을 지원합니다. 현재 Message Queue 브로커에서 사용할 수 있는 연결 서비스가 표 5-1에 표시되어 있습니다. 서비스 이름 옆에 있는 값은 -n 옵션의 서비스 이름을 지정할 때 사용되는 값입니다. 표에 있는 것과 같이 각 서비스는 NORMAL(응용 프로그램 클라이언트) 또는 ADMIN(관리 클라이언트) 중에서 사용하는 서비스 유형과 기본 전송 프로토콜에 연결됩니다.

**표 5-1** 브로커가 지원하는 연결 서비스

서비스 이름	서비스 유형	프로토콜 유형
jms	NORMAL	tcp
ssljms(엔터프라이즈판)	NORMAL	tls(SSL 기반 보안)
httpjms(엔터프라이즈판)	NORMAL	http
httpsjms(엔터프라이즈판)	NORMAL	https(SSL 기반 보안)
admin	ADMIN	tcp
ssladmin(엔터프라이즈판)	ADMIN	tls(SSL 기반 보안)

## 연결 서비스 나열

브로커에서 사용 가능한 연결 서비스를 나열하려면 `list svc` 하위 명령을 사용합니다.

다음은 `list svc` 하위 명령의 구문입니다.

```
imqcmd list svc [-b hostName:port]
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에 있는 모든 연결 서비스를 나열합니다.

명령줄에서 다음과 같은 하위 명령을 사용합니다.

```
imqcmd list svc [-b hostName:portNumber] -u admin
```

예를 들어, 다음 명령은 `myServer` 호스트의 포트 6565에서 실행 중인 브로커에서 사용할 수 있는 서비스를 나열합니다.

```
imqcmd list svc -b MyServer:6565 -u admin
```

다음 명령은 `localhost`의 포트 7676에서 실행 중인 브로커의 모든 서비스를 나열합니다.

```
imqcmd list svc -u admin
```

이 명령은 다음과 같은 정보를 출력합니다.

Service Name	Port Number	Service State
admin	41844 (dynamic)	RUNNING
httpjms	-	UNKNOWN
httpsjms	-	UNKNOWN
jms	41843 (dynamic)	RUNNING
ssladmin	dynamic	UNKNOWN
ssljms	dynamic	UNKNOWN

## 연결 서비스 정보 표시

단일 서비스에 대한 정보를 쿼리 및 표시하려면 `query` 하위 명령을 사용합니다.

다음은 `query svc` 하위 명령의 구문입니다.

```
imqcmd query svc -n serviceName [-b hostName:port]
```

이 하위 명령은 기본 브로커 또는 지정된 호스트 및 포트의 브로커에서 실행 중인, 지정된 서비스에 대한 정보를 표시합니다.

예를 들면 다음과 같습니다.

```
imqcmd query svc -n jms -u admin
```

이 명령은 비밀번호를 묻는 메시지를 표시한 후 다음과 같은 출력을 생성합니다.

Service Name	jms
Service State	RUNNING
Port Number	60920 (dynamic)
Current Number of Allocated Threads	0
Current Number of Connections	0
Min Number of Threads	10
Max Number of Threads	1000

## 연결 서비스 등록 정보 업데이트

`update` 하위 명령을 사용하여 표 5-2에 나열된 서비스 등록 정보 중 하나 이상을 변경할 수 있습니다.

**표 5-2** imqcmd가 업데이트하는 연결 서비스 등록 정보

등록 정보	설명
port	업데이트할 서비스에 할당된 포트입니다(httpjms 또는 httpsjms에는 적용되지 않음). 값이 0인 경우 포트 매퍼가 포트를 동적으로 할당합니다.
minThreads	서비스에 할당된 최소 스레드 수입니다.
maxThreads	서비스에 할당된 최대 스레드 수입니다.



다음은 update 하위 명령의 구문입니다.

```
imqcmd update svc -n serviceName [-b hostName:port]
-o attribute=value [-o attribute=value1]...
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 실행 중인, 지정된 서비스의 지정된 속성을 업데이트합니다. 서비스 속성에 대한 설명은 [311페이지의 "연결 서비스 등록 정보"](#)를 참조하십시오.

다음 명령은 `jms` 서비스에 할당된 최소 스레드 수를 20으로 변경합니다.

```
imqcmd update svc -n jms -o "minThreads=20" -u admin
```

## 연결 서비스 메트릭 표시

단일 서비스에 대한 메트릭 정보를 표시하려면 `metrics` 하위 명령을 사용합니다.

`metrics` 하위 명령의 구문입니다.

```
imqcmd metrics svc -n serviceName [-b hostName:port] [-m metricType]
[-int interval] [-msp numSamples]
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 지정된 서비스의 메트릭을 표시합니다.

`-m` 옵션을 사용하여 표시할 메트릭 유형을 지정합니다.

- **t1** 브로커에 유입 및 유출되는 메시지와 패킷의 메트릭을 지정된 연결 서비스를 통해 표시합니다(기본 메트릭 유형).
- **rts** 브로커에 메시지와 패킷이 유입 및 유출되는 초당 속도에 대한 메트릭을 지정된 연결 서비스를 통해 표시합니다.
- **cxn** 연결, 가상 메모리 힙 및 스레드를 표시합니다.

`-int` 옵션을 사용하여 메트릭 표시 간격(초)을 지정합니다. 기본값은 5초입니다.

`-msp` 옵션을 사용하여 출력에 표시되는 샘플 수를 지정합니다. 기본값은 무제한 수입니다(무한).

예를 들어, 다음 명령은 `jms` 연결 서비스에서 처리된 메시지 및 패킷의 누적 총 수를 구합니다.

```
imqcmd metrics svc -n jms -m t1 -u admin
```

이 명령은 비밀번호를 묻는 메시지를 표시한 후 다음과 같은 출력을 생성합니다.

Msgs		Msg Bytes		Pkts		Pkt Bytes	
In	Out	In	Out	In	Out	In	Out
164	100	120704	73600	282	383	135967	102127
657	100	483552	73600	775	876	498815	149948

imqcmd를 사용하여 연결 서비스 메트릭을 보고하는 방법에 대한 자세한 내용은 [352페이지](#)의 "연결 서비스 메트릭"을 참조하십시오.

## 연결 서비스 일시 중지 및 다시 시작

일시 중지할 수 없는 **admin** 서비스를 제외한 모든 서비스를 일시 중지하려면 `pause svc` 및 `resume svc` 하위 명령을 사용합니다.

다음은 `pause svc` 하위 명령의 구문입니다.

```
imqcmd pause svc -n serviceName [-b hostName:port]
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 실행 중인, 지정된 서비스를 일시 중지합니다. 관리 서비스는 일시 중지할 수 없습니다.

다음과 같은 명령줄을 사용합니다.

```
imqcmd pause svc -n serviceName -u admin
```

서비스를 일시 중지하면 다음과 같이 됩니다.

- 브로커는 일시 중지된 서비스에서 새 클라이언트 연결 수신을 멈춥니다. **Message Queue** 클라이언트가 새 연결을 열려고 하면 예외가 발생합니다.
- 일시 중지된 서비스의 기존 연결은 모두 그대로 유지되지만 브로커는 서비스가 다시 시작될 때까지 이러한 연결의 모든 메시지 처리를 지연합니다(예를 들어, 클라이언트가 메시지를 보내려고 하면 서비스가 다시 시작될 때까지 `send()` 메소드가 차단됨).
- 브로커가 이미 수신한 메시지의 메시지 전달 상태는 그대로 유지됩니다(예를 들어, 트랜잭션이 중단되지 않고 서비스가 다시 시작되면 메시지 전달이 재개됨).

서비스를 다시 시작하려면 `resume svc` 하위 명령을 사용합니다.

다음은 `resume svc` 하위 명령의 구문입니다.

```
imqcmd resume svc -n serviceName[-b hostName:port]
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트에서 실행 중인, 지정된 서비스를 다시 시작합니다.

다음과 같은 명령줄을 사용합니다.

```
imqcmd resume svc -n serviceName -u admin
```

## 연결 정보 얻기

명령 유틸리티의 하위 명령을 사용하여 연결 정보를 나열하고 가져올 수 있습니다.

`list cxn` 하위 명령은 지정된 서비스 이름의 모든 연결을 나열합니다. 다음은 `list cxn` 하위 명령의 구문입니다.

```
imqcmd list cxn [-svn serviceName] [-b hostName:port]
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 지정된 서비스 이름의 연결을 모두 나열합니다. 서비스 이름을 지정하지 않는 경우 모든 연결이 나열됩니다.

예를 들면 다음과 같습니다.

```
imqcmd list cxn -u admin
```

이 명령은 비밀번호를 묻는 메시지를 표시한 후 다음과 같은 출력을 생성합니다.

```
Listing all the connections on the broker specified by:
-----
Host                Primary Port
-----
localhost           7676

-----
Connection ID      User      Service  Producers  Consumers  Host
-----
1964412264455443200  guest    jms       0           1          127.0.0.1
1964412264493829311  admin    admin     1           1          127.0.0.1
Successfully listed connections.
```

단일 연결 서비스에 대한 정보를 쿼리 및 표시하려면 `query` 하위 명령을 사용합니다.

```
query cxn -n connectionID [-b hostName:port]
```

이 하위 명령은 기본 브로커 또는 지정된 호스트 및 포트의 브로커에서 지정된 연결에 대한 정보를 표시합니다.

예를 들면 다음과 같습니다.

```
imqcmd query cxn -n 421085509902214374 -u admin
```

이 명령은 비밀번호를 묻는 메시지를 표시한 후 다음과 같은 출력을 생성합니다.

Connection ID	421085509902214374
User	guest
Service	jms
Producers	0
Consumers	1
Host	111.22.333.444
Port	60953
Client ID	
Client Platform	

## 영구 가입 관리

`imqcmd` 하위 명령을 사용해서 브로커의 영구 가입을 관리할 수 있습니다. 영구 가입은 클라이언트에 영구로 등록된 주제에 가입하는 것입니다. 여기에는 고유한 아이디가 있으며, 사용자가 비활성 상태인 동안에도 브로커에서 가입에 해당하는 메시지를 보존해야 합니다. 보통 브로커는 메시지가 만료될 때 영구 가입자에 대해 보존된 메시지만 삭제할 수 있습니다.

지정된 물리적 대상에 대한 영구 가입을 나열하려면 `list dur` 하위 명령을 사용합니다. 다음은 `list dur` 하위 명령의 구문입니다.

```
imqcmd list dur -d destName
```

예를 들어, 다음 명령은 로컬 호스트의 기본 포트에서 브로커를 사용하여 `SPQuotes` 주제에 대한 모든 영구 가입을 나열합니다.

```
imqcmd list dur -d SPQuotes
```

`list dur` 하위 명령은 주제의 각 영구 가입에 대해 영구 가입의 이름과 사용자의 클라이언트 아이디, 이 주제의 대기열에 들어 있는 메시지의 수, 영구 가입 상태(활성/비활성)를 반환합니다. 예를 들면 다음과 같습니다.

Name	Client ID	Number of Messages	Durable Sub State
-----	-----	-----	-----
myDurable	myClientID	1	INACTIVE

`list dur` 하위 명령에서 반환된 정보를 사용하여 완전 삭제하거나 메시지를 제거할 영구 가입을 확인할 수 있습니다.

`destroy dur` 하위 명령은 지정된 클라이언트 식별자를 사용하여 지정된 영구 가입을 완전 삭제합니다. 다음은 `destroy dur` 하위 명령의 구문입니다.

```
imqcmd destroy dur -n subscrName -c client_id
```

가입 이름과 클라이언트 아이디를 사용하여 가입을 확인합니다. 예를 들면 다음과 같습니다.

```
imqcmd destroy dur -n myDurable -c myClientID
```

`purge dur` 하위 명령은 지정된 클라이언트 식별자를 사용하여 지정된 영구 가입에 대한 모든 메시지를 제거합니다. 다음은 `purge dur` 하위 명령의 구문입니다.

```
imqcmd purge dur -n subscrName -c client_id
```

## 트랜잭션 관리

클라이언트 응용 프로그램에서 시작하는 모든 트랜잭션은 브로커에서 추적됩니다. 이는 단순한 Message Queue 트랜잭션일 수도 있고 분산 트랜잭션(XA 자원) 관리자가 관리하는 분산 트랜잭션일 수도 있습니다.

모든 트랜잭션에는 브로커의 트랜잭션을 고유하게 나타내는 64비트 숫자, 즉 Message Queue 트랜잭션 아이디가 있습니다. 분산 트랜잭션에도 분산 트랜잭션 관리자가 할당하는 최대 128바이트의 분산 트랜잭션 아이디(XID)가 있습니다. Message Queue에서는 Message Queue 트랜잭션 아이디와 XID 사이의 연결을 보존합니다.

분산 트랜잭션에 오류가 발생하면 그 트랜잭션은 PREPARED 상태로 남아 완결되지 않을 수도 있습니다. 따라서 관리자는 준비된 상태로 남아 있는 트랜잭션을 모니터링하고 롤백 또는 완결해야 합니다.

브로커가 추적 중인 모든 트랜잭션을 나열하려면 `list txn` 명령을 사용합니다. 다음은 `list tx` 하위 명령의 구문입니다.

```
imqcmd list txn
```

예를 들어, 다음 명령은 브로커에 있는 모든 트랜잭션을 나열합니다.

```
imqcmd list txn
```

`list` 하위 명령은 각 트랜잭션에 대해 트랜잭션 아이디, 상태, 아이디, 메시지 또는 확인 응답의 수, 작성 시간을 반환합니다. 예를 들면 다음과 같습니다.

Transaction ID	State	User name	# Msgs/ # Acks	Creation time
64248349708800	PREPARED	guest	4/0	1/30/02 10:08:31 AM
64248371287808	PREPARED	guest	0/4	1/30/02 10:09:55 AM

명령은 브로커에 있는 로컬 및 분산 트랜잭션을 모두 표시합니다. PREPARED 상태인 트랜잭션만을 완결 또는 롤백할 수 있습니다. 트랜잭션이 오류로 인해 준비 상태에 있고 분산 트랜잭션 관리자에서 완결을 진행하고 있지 않다는 것이 확인된 경우에만 완결 또는 롤백할 수 있습니다.

예를 들어, 브로커의 자동 롤백 등록 정보가 `false`로 설정되어 있으면(313페이지의 표 14-3 참조) 브로커를 시작할 때 PREPARED 상태인 트랜잭션을 수동으로 완결 또는 롤백해야 합니다.

`list` 하위 명령도 트랜잭션에서 생성된 메시지의 수와 트랜잭션에서 확인 응답된 (#Msgs/#Acks) 메시지의 수를 표시합니다. 트랜잭션이 완결될 때까지는 이런 메시지가 전달되지 않으며 확인 응답이 처리되지도 않습니다.

`query` 하위 명령을 사용하면 같은 정보와 클라이언트 아이디, 연결 아이디, 분산 트랜잭션 아이디(XID) 등의 여러 추가 값을 볼 수 있습니다. 다음은 `query txn` 하위 명령의 구문입니다.

```
imqcmd query txn -n transaction_id
```

예를 들어, 다음 예는 아래와 같은 출력을 생성합니다.

```
imqcmd query txn -n 64248349708800
```

이 명령에 의해 생성되는 출력은 다음과 같습니다.

```
Client ID
Connection          guest@192.18.116.219:62209->jms:62195
Creation time       1/30/02 10:08:31 AM
Number of acknowledgments 0
Number of messages 4
State               PREPARED
Transaction ID      64248349708800
User name           guest
XID
6469706F6C7369646577696E6465723130313234313431313030373230
```

commit 및 rollback 하위 명령은 분산 트랜잭션을 완결 또는 롤백할 때 사용할 수 있습니다. 앞에서 설명한 것과 같이, PREPARED 상태에 있는 트랜잭션만 완결 또는 롤백할 수 있습니다.

다음은 commit 하위 명령의 구문입니다.

```
imqcmd commit txn -n transaction_id
```

예를 들면 다음과 같습니다.

```
imqcmd commit txn -n 64248349708800
```

브로커를 시작할 때 PREPARED 상태에 있는 트랜잭션을 자동으로 롤백하도록 브로커를 구성할 수도 있습니다.

다음은 rollback 하위 명령의 구문입니다.

```
imqcmd rollback txn -n transaction_id
```

자세한 내용은 [313페이지의 표 14-3](#)의 imq.transaction.autorollback 등록 정보를 참조하십시오.





## 물리적 대상 관리

Message Queue 메시지는 브로커의 물리적 대상을 통해 사용자 클라이언트로 라우팅 됩니다. 브로커는 물리적 대상에 연결된 메모리와 영구 저장소를 관리하고 해당 동작을 설정합니다.

클러스터에서 브로커 하나에 물리적 대상을 만들면 클러스터가 해당 물리적 대상을 모든 브로커에 전파합니다. 브로커는 공동 작업을 통해 클러스터 전체에서 메시지 경로를 지정하기 때문에 응용 프로그램 클라이언트는 항목에 가입하거나 클러스터의 모든 브로커에 있는 대기열을 사용할 수 있습니다. 하지만 메시지가 처음 생성된 브로커에서만 해당 메시지에 대한 지속성과 확인을 관리할 수 있습니다.

이 장에서는 다음 작업을 수행하는 방법에 대해 설명합니다.

- 128페이지의 "imqcmd 명령 유틸리티 사용"
- 129페이지의 "물리적 대상 만들기"
- 131페이지의 "물리적 대상 나열"
- 131페이지의 "물리적 대상 정보 표시"
- 133페이지의 "물리적 대상 등록 정보 업데이트"
- 133페이지의 "물리적 대상 일시 중지 및 다시 시작"
- 134페이지의 "물리적 대상 제거"
- 135페이지의 "물리적 대상 완전 삭제"
- 136페이지의 "물리적 대상 압축"
- 138페이지의 "사용 불능 메시지 대기열 사용 구성"

표 13-5는 물리적 대상 관리와 이러한 작업 수행에 사용되는 imqcmd 하위 명령에 대한 전체 참조 정보를 제공합니다.

---

**주** 클라이언트 응용 프로그램은 물리적 대상과 상호 작용할 때마다 Destination 객체를 사용합니다. 공급자 독립성과 이식성을 위해 클라이언트는 대상 관리 대상 객체라는 관리자 생성 대상 객체를 일반적으로 사용합니다. [8장, "관리 대상 객체 관리"](#)에서 설명한 대로 클라이언트 응용 프로그램에 사용하도록 관리 대상 객체를 구성할 수 있습니다.

---

## imqcmd 명령 유틸리티 사용

imqcmd 명령 유틸리티를 사용하여 물리적 대상을 관리할 수 있습니다. imqcmd 명령 구문은 다른 브로커 서비스 관리에 사용할 때의 구문과 동일합니다.

imqcmd, 해당 하위 명령 및 옵션에 대한 자세한 내용은 [279페이지의 13장, "명령 참조"](#)를 참조하십시오.

### 하위 명령

[표 6-1](#)에는 이 장에서 설명하는 imqcmd 하위 명령이 나열되어 있습니다. 이러한 하위 명령에 대한 자세한 내용은 [290페이지의 "물리적 대상 관리 하위 명령"](#)을 참조하십시오.

**표 6-1** imqcmd 명령 유틸리티의 물리적 대상 하위 명령

하위 명령 및 인수	설명
compact dst	기본 제공 파일 기반 데이터 저장소에서 하나 이상의 물리적 대상을 압축합니다.
create dst	물리적 대상을 만듭니다.
destroy dst	물리적 대상을 완전 삭제합니다.
list dst	브로커의 물리적 대상을 나열합니다.
metrics dst	물리적 대상 메트릭을 표시합니다.
pause dst	브로커에서 하나 이상의 물리적 대상을 일시 중지합니다.
purge dst	물리적 대상을 완전 삭제하지 않고 물리적 대상에 있는 모든 메시지를 제거합니다.
query dst	물리적 대상의 정보를 쿼리 및 표시합니다.
resume dst	브로커에서 일시 중지된 하나 이상의 물리적 대상을 다시 시작합니다.
update dst	대상의 속성을 업데이트합니다.

---

# 물리적 대상 만들기

물리적 대상을 만들려면 `mqcmd create` 하위 명령을 사용합니다. 다음은 `create` 하위 명령의 구문입니다.

```
create dst -t destType -n destName [-o property=value] [-o property=value1]...
```

물리적 대상을 만들 때 다음을 지정해야 합니다.

- 물리적 대상의 유형, `t`(주제) 또는 `q`(대기열)
- 물리적 대상 이름. 이름 지정 규칙은 다음과 같습니다.
  - 이름은 영숫자만 포함할 수 있으며 공백을 포함할 수 없습니다.
  - 이름은 영문자, 밑줄 문자(`_`) 또는 달러 기호(`$`)로 시작할 수 있습니다. `"mq"` 문자열로는 시작할 수 없습니다.
- 기본값을 제외한 모든 물리적 대상 등록 정보 값

물리적 대상을 업데이트할 때 등록 정보도 설정할 수 있습니다.

많은 물리적 대상 등록 정보는 브로커 메모리 자원과 메시지 흐름을 관리합니다. 예를 들어, 물리적 대상에 보낼 수 있는 생성자 수, 생성자가 보낼 수 있는 메시지 수와 크기, 물리적 대상 제한에 도달할 때 브로커가 수행해야 하는 응답 등을 지정할 수 있습니다. 제한은 브로커 구성 등록 정보에서 제어하는 브로커 전체 제한과 비슷합니다.

다음 등록 정보는 대기열 대상과 주제 대상 모두에 사용됩니다.

- `maxNumMsgs`. 물리적 대상에 허용되는 사용되지 않은 최대 메시지 수를 지정합니다.
- `maxTotalMsgBytes`. 물리적 대상의 사용되지 않은 메시지에 허용되는 최대 메모리 합계(바이트)를 지정합니다.
- `limitBehavior`. 메모리 제한 임계값에 도달할 경우에 브로커가 응답하는 방법을 지정합니다.
- `maxBytesPerMsg`. 물리적 대상에 허용되는 단일 메시지의 최대 크기(바이트)를 지정합니다.
- `maxNumProducers`. 물리적 대상의 최대 생성자 수를 지정합니다.
- `consumerFlowLimit`. 일괄적으로 사용자에게 전달되는 최대 메시지 수를 지정합니다.

- `isLocalOnly`. 브로커 클러스터에만 적용됩니다. 물리적 대상이 다른 브로커에 복제되지 않도록 지정합니다. 따라서, 메시지 전달이 로컬 사용자(물리적 대상이 생성되는 브로커에 연결된 사용자)에게만 제한됩니다.
- `useDMQ`. 물리적 대상의 사용 불능 메시지를 제거할지, 사용 불능 메시지 대기열에 넣을지를 지정합니다.

다음은 대기열 대상에만 사용되는 등록 정보입니다.

- `maxNumActiveConsumers`. 대기열 대상으로부터의 로드 균형 조정 전달에서 활성 상태가 될 수 있는 최대 사용자 수를 지정합니다.
- `maxNumBackupConsumers`. 대기열 대상으로부터의 로드 균형 조정 전달 중에 오류가 발생할 경우 활성 사용자를 대신할 수 있는 백업 사용자의 최대 수를 지정합니다.
- `localDeliveryPreferred`. 브로커 클러스터의 로드 균형 조정된 대기열 전달에만 적용됩니다. 로컬 브로커에 사용자가 없는 경우에만 원격 사용자에게 메시지를 전달하도록 지정합니다.

물리적 대상 등록 정보에 대한 자세한 내용은 [329페이지의 15장, "물리적 대상 등록 정보 참조"](#)를 참조하십시오.

자동 생성 대상의 경우 브로커의 인스턴스 구성 파일에 기본 등록 정보 값을 설정합니다. 자동 생성 등록 정보에 대한 자세한 내용은 [314페이지의 표 14-4](#)를 참조하십시오.

### ▶ 물리적 대상 생성 방법

- 대기열 대상을 만들려면 다음과 같이 명령을 입력합니다.

```
imqcmd create dst -n myQueue -t q -o "maxNumActiveConsumers=5"
```

- 주제 대상을 만들려면 다음과 같이 명령을 입력합니다.

```
imqcmd create dst -n myTopic -t t -o "maxBytesPerMsg=5000"
```

## 물리적 대상 나열

물리적 대상의 현재 속성 값, 물리적 대상과 연관된 생성자 또는 사용자 수 및 메시징 메트릭(물리적 대상의 메시지 수 및 크기 등)에 대한 정보를 얻을 수 있습니다.

정보를 얻으려는 물리적 대상을 찾으려면 브로커의 모든 물리적 대상을 나열합니다. 그렇게 하려면 `list dst` 하위 명령을 사용합니다. 다음은 `list dst` 하위 명령의 구문입니다.

```
list dst [-t destType] [-tmp]
```

이 명령은 지정된 유형의 물리적 대상을 나열합니다. 대상 유형(-t) 옵션 값은 q(대기열) 또는 t(주제) 값을 가질 수 있습니다.

대상 유형을 생략하면 모든 유형의 물리적 대상이 나열됩니다.

`list dst` 하위 명령에서는 나열할 대상의 유형을 지정하거나 임시 대상을 포함하도록 지정할 수도 있습니다(-tmp 옵션 사용). 임시 대상은 일반적으로 다른 클라이언트에 보낸 메시지에 대한 응답을 수신하기 위해 클라이언트가 만듭니다.

예를 들어, `myHost`의 포트 4545에서 실행 중인 브로커의 모든 물리적 대상을 나열하려면 다음 명령을 입력합니다.

```
imqcmd list dst -b myHost:4545
```

주제만 포함하도록 t 대상 유형을 지정하지 않으면 `mq.sys.dmq` 사용 불능 메시지 대기열이 다른 물리적 대상과 함께 항상 표시됩니다.

## 물리적 대상 정보 표시

물리적 대상의 현재 등록 정보 값에 대한 정보를 보려면, `query dst` 하위 명령을 사용합니다. 다음은 `query dst` 하위 명령의 구문입니다.

```
query dst -t destType -n destName
```

이 명령은 지정한 유형과 이름의 대상에 대한 정보를 나열합니다. 예를 들면 다음과 같습니다.

```
imqcmd query dst -t q -n XQueue -u admin
```

이 명령은 다음과 같은 출력을 표시합니다.

```

-----
Destination Name      Destination Type
-----
XQueue                Queue

On the broker specified by:

-----
Host                  Primary Port
-----
localhost            7676

Destination Name      XQueue
Destination Type      Queue
Destination State     RUNNING
Created Administratively true

Current Number of Messages      0
Current Total Message Bytes     0
Current Number of Producers     0
Current Number of Active Consumers 0
Current Number of Backup Consumers 0

Max Number of Messages      unlimited (-1)
Max Total Message Bytes     unlimited (-1)
Max Bytes per Message       unlimited (-1)
Max Number of Producers     100
Max Number of Active Consumers 1
Max Number of Backup Consumers 0

Limit Behavior           REJECT_NEWEST
Consumer Flow Limit      1000
Is Local Destination     false
Local Delivery is Preferred false
Use Dead Message Queue   true

```

출력에는 해당 대상에 연관되어 있는 생성자와 사용자 수도 표시됩니다. 대기열 대상의 경우 이 수에는 활성 사용자와 백업 사용자가 포함됩니다.

update dst 하위 명령을 사용하여 하나 이상의 등록 정보 값을 변경할 수 있습니다(133페이지의 "물리적 대상 등록 정보 업데이트" 참조).

## 물리적 대상 등록 정보 업데이트

update dst 하위 명령과 업데이트할 등록 정보를 지정하는 -o 옵션을 사용하여 물리적 대상의 등록 정보를 변경할 수 있습니다. 다음은 update dst 하위 명령의 구문입니다.

```
update dst -t destType -n destName -o property=value [-o property=value1]...
```

이 명령은 지정한 대상에서 지정한 등록 정보의 값을 업데이트합니다. 등록 정보 이름은 [표 15-1](#)에 설명된 모든 등록 정보가 될 수 있습니다.

-o 옵션을 여러 번 사용하여 여러 등록 정보를 업데이트할 수 있습니다. 예를 들어, 다음 명령은 maxBytesPerMsg 등록 정보를 1000으로 변경하고 MaxNumMsgs 등록 정보를 2000으로 변경합니다.

```
imqcmd update dst -t q -n myQueue -o "maxBytesPerMsg=1000"
-o "maxNumMsgs=2000" -u admin
```

업데이트할 수 있는 등록 정보의 목록을 보려면 [15장](#), "[물리적 대상 등록 정보 참조](#)"를 참조하십시오.

update dst 하위 명령을 사용해서 물리적 대상의 *유형*을 업데이트하거나 isLocalOnly 등록 정보를 업데이트할 수 없습니다.

---

**주**            사용 불가능 메시지 대기열은 등록 정보가 다른 대상과 약간 차이가 나는 특수한 물리적 대상입니다. 자세한 내용은 [138페이지](#)의 "[사용 불가능 메시지 대기열 사용 구성](#)"을 참조하십시오.

---

## 물리적 대상 일시 중지 및 다시 시작

물리적 대상을 일시 중지하여 생성자에서 대상으로, 대상에서 사용자로 또는 두 가지 모두에 대한 메시지 전달을 제어할 수 있습니다. 특히, 대상으로의 메시지 흐름을 일시 중지하면 메시지 생성이 사용보다 훨씬 빠를 때 대상에서 메시지가 넘치는 것을 방지할 수 있습니다.

물리적 대상에서 주고 받는 메시지 전달을 일시 중지하려면 pause dst 하위 명령을 사용합니다. 다음은 pause dst 하위 명령의 구문입니다.

```
pause dst [-t destType -n destName] [-pst pauseType]
```

지정된 유형과 이름의 대상에서 사용자에게 메시지 전달(-pst CONSUMERS), 생성자로부터 메시지 전달(-pst PRODUCERS) 또는 두 가지 모두(-pst ALL)를 일시 중지합니다. 대상 유형과 이름을 지정하지 않은 경우 모든 물리적 대상이 일시 중지됩니다. 기본값은 ALL입니다.

예:

```
imqcmd pause dst -n myQueue -t q -pst PRODUCERS -u admin
```

```
imqcmd pause dst -n myTopic -t t -pst CONSUMERS -u admin
```

일시 중지된 대상으로의 전달을 다시 시작하려면 `resume dst` 하위 명령을 사용합니다. 다음은 `resume dst` 하위 명령의 구문입니다.

```
resume dst [-t destType -n destName]
```

이 하위 명령은 지정된 유형과 이름의 일시 중지된 대상으로의 메시지 전달을 다시 시작합니다. 대상 유형과 이름을 지정하지 않으면 모든 대상이 다시 시작됩니다.

예:

```
imqcmd resume dst -n myQueue -t q
```

브로커 클러스터에서 물리적 대상 인스턴스는 클러스터의 각 브로커에 있습니다. 각 인스턴스를 개별적으로 일시 중지해야 합니다.

## 물리적 대상 제거

현재 물리적 대상의 대기열에 들어 있는 모든 메시지를 제거할 수 있습니다. 물리적 대상을 제거하는 것은 대상의 대기열에 들어 있는 모든 메시지를 삭제하는 것을 의미합니다.

누적된 메시지가 시스템의 자원을 너무 많이 차지하는 경우에 메시지를 제거할 수 있습니다. 이런 상황은 대기열에 등록된 사용자 클라이언트가 없는 상태에서 너무 많은 메시지를 받으면 발생할 수 있습니다. 주제의 비활성 영구 가입자가 활성화되지 않는 경우에도 발생할 수 있습니다. 두 경우 모두 불필요한 메시지가 보존됩니다.

물리적 대상의 메시지를 제거하려면 `purge dst` 하위 명령을 사용합니다. 다음은 `purge dst` 하위 명령의 구문입니다.

```
purge dst -t destType -n destName
```

이 하위 명령은 지정한 유형과 이름의 물리적 대상에서 메시지를 제거합니다.



예:

```
imqcmd purge dst -n myQueue -t q -u admin
```

```
imqcmd purge dst -n myTopic -t t -u admin
```

브로커를 종료했다가 다시 시작할 때 오래된 메시지가 전달되지 않게 하려면 `-reset messages` 옵션을 사용하여 오래된 메시지를 제거합니다. 예를 들면 다음과 같습니다.

```
imqbrokerd -reset messages -u admin
```

그러면 브로커를 다시 시작한 후에 대상을 제거할 필요가 없습니다.

브로커 클러스터에서 물리적 대상 인스턴스는 클러스터의 각 브로커에 있습니다. 이러한 대상을 개별적으로 제거해야 합니다.

## 물리적 대상 완전 삭제

물리적 대상을 완전 삭제하려면 `destroy dst` 하위 명령을 사용합니다. 다음은 `destroy dst` 하위 명령의 구문입니다.

```
destroy dst -t destType -n destName
```

이 하위 명령은 지정한 유형과 이름의 물리적 대상을 완전 삭제합니다.

예:

```
imqcmd destroy dst -t q -n myQueue -u admin
```

물리적 대상을 완전 삭제하면 대상에 있는 모든 메시지가 제거되고 대상이 브로커에서 제거됩니다. 이 작업은 다시 되돌릴 수 없습니다.

사용 불가능 메시지 대기열은 완전 삭제할 수 없습니다.

## 물리적 대상 압축

플러그인 JDBC 호환 데이터 저장소와 반대되는 기본 제공 파일 기반 데이터 저장소를 메시지의 영구 저장소로 사용하는 경우 디스크 사용률을 모니터링하고 필요한 경우 디스크를 압축할 수 있습니다.

파일 기반 메시지 저장소는 메시지가 보관될 물리적 대상에 따라 다른 디렉토리에 저장되도록 구성됩니다. 각 물리적 대상 디렉토리에서 대부분의 메시지는 가변 크기 레코드로 구성되는 단일 파일인 가변 크기 레코드 파일에 저장됩니다(단편화를 줄이기 위해 크기가 구성 가능한 임계값을 초과하는 메시지는 자체의 개별 파일에 저장).

가변 크기 레코드 파일에서 다양한 크기의 메시지가 지속되다가 제거될 때 파일에서 사용 가능한 레코드가 다시 사용되지 않는 공간이 생길 수 있습니다.

사용되지 않은 사용 가능한 레코드를 관리하려면 명령 유틸리티의 하위 명령을 사용하여 물리적 대상별 디스크 사용률을 모니터링하고 사용률이 떨어지면 사용 가능한 디스크 공간을 확보합니다.

### 물리적 대상 디스크 사용률 모니터링

물리적 대상의 디스크 사용률을 모니터링하려면 다음과 같은 명령을 사용합니다.

```
imqcmd metrics dst -t q -n myQueue -m dsk -u admin
```

다음과 같은 출력이 표시됩니다.

Reserved	Used	Utilization Ratio
806400	804096	99
1793024	1793024	100
2544640	2518272	98

하위 명령 출력에서 각 열의 의미는 다음과 같습니다.

**표 6-2** 물리적 대상 디스크 사용률 메트릭

메트릭	설명
예약됨	활성 메시지가 보관된 레코드와 재사용 대기 중인 사용 가능한 레코드를 포함한 모든 레코드가 사용하는 디스크 공간(바이트)
사용됨	활성 메시지가 보관된 레코드에서 사용하는 디스크 공간(바이트)
사용률	예약된 디스크 공간에서 사용되는 디스크 공간의 비율. 비율이 높을수록 활성 메시지를 보관하는 데 사용되고 있는 디스크 공간이 많은 것입니다.

## 사용되지 않은 물리적 대상 디스크 공간 확보

특정 물리적 대상을 사용하는 메시징 응용 프로그램의 특성에 따라 디스크 사용률 패턴이 달라집니다. 물리적 대상에 유입 및 유출되는 메시지의 상대적 흐름과 상대적 메시지 크기에 따라 예약된 디스크 공간이 점점 더 커질 수 있습니다.

메시지 생성 속도가 메시지 사용 속도보다 큰 경우 사용 가능한 레코드를 다시 사용하고 사용률을 높은 수준으로 유지해야 합니다. 그러나 메시지 생성 속도가 메시지 사용 속도보다 작거나 비슷한 경우 사용률이 낮아도 됩니다.

일반적으로 예약된 디스크 공간은 안정적으로 유지하고 사용률은 높게 유지해야 합니다. 일반적으로 시스템에서 예약된 디스크 공간은 매우 일정하게 유지되고 사용률이 높은(75% 이상) 안정적인 상태에 도달하는 경우 사용되지 않는 디스크 공간을 확보할 필요가 없습니다. 시스템에서 안정적인 상태에 도달하고 사용률이 낮은(50% 이하) 경우 디스크를 압축하여 사용 가능한 레코드가 사용 중인 디스크 공간을 확보할 수 있습니다.

`compact dst` 하위 명령을 사용하여 데이터 저장소를 압축합니다.

다음은 `compact dst` 하위 명령의 구문입니다.

```
compact dst [-t destType -n destName]
```

이 하위 명령은 기본 제공 파일 기반 데이터 저장소에서 지정된 유형과 이름의 물리적 대상을 압축합니다. 대상 유형과 이름을 지정하지 않으면 모든 대상이 압축됩니다. 물리적 대상을 압축하려면 먼저 일시 중지해야 합니다.

예약된 디스크 공간이 점점 증가하는 경우 대상 메모리 제한 등록 정보와 제한 동작(329 페이지의 표 15-1 참조)을 설정하여 대상의 메모리 관리를 다시 구성해야 합니다.

▶ **사용되지 않는 물리적 대상 디스크 공간을 확보하는 방법**

1. 대상을 일시 중지합니다.

```
imqcmd pause dst -t q -n myQueue -u admin
```

2. 디스크를 압축합니다.

```
imqcmd compact dst -t q -n myQueue -u admin
```

3. 물리적 대상을 다시 시작합니다.

```
imqcmd resume dst -t q -n myQueue -u admin
```

대상 유형과 이름을 지정하지 않으면 이 작업이 모든 물리적 대상에 대해 수행됩니다.

## 사용 불능 메시지 대기열 사용 구성

사용 불능 메시지 대기열 `mq.sys.dmq`는 브로커 및 다른 물리적 대상의 사용 불능 메시지를 보관하는 시스템 생성 물리적 대상입니다. 사용 불능 메시지 대기열은 모니터링, 시스템 효율성 조정 및 문제 해결을 위한 도구입니다. "사용 불능 메시지"의 용어 정의와 사용 불능 메시지 대기열에 대한 자세한 내용은 *Message Queue 기술 개요*를 참조하십시오.

브로커는 시작될 때 사용 불능 메시지 대기열을 자동으로 생성합니다. 브로커는 메시지를 처리할 수 없거나 수명이 만료된 경우 해당 메시지를 대기열에 넣습니다. 다른 물리적 대상도 사용 불능 메시지 대기열을 사용하여 제거된 메시지를 보관합니다. 사용 불능 메시지 대기열을 사용하면 시스템 문제 해결에 유용한 정보를 확인할 수 있습니다.

## 사용 불능 메시지 대기열 사용 구성

기본적으로 물리적 대상은 사용 불능 메시지 대기열을 사용하도록 구성됩니다. 물리적 대상 등록 정보 `useDMQ`를 설정하여 물리적 대상에서 사용 불능 메시지 대기열을 사용 불가능하게 하거나 사용 가능하게 할 수 있습니다.

다음 예에서는 기본적으로 사용 불능 메시지 대기열을 사용하는 `myDist` 대기열을 만듭니다.

```
imqcmd create dst -n -myDist -t q
```

다음 예에서는 동일한 대기열에 대해 사용 불능 메시지 대기열을 사용 불가능하게 합니다.

```
imqcmd update dst -n myDist -t q -o useDMQ=false
```

`imq.autocreate.destination.useDMQ` 브로커 등록 정보를 설정하여 브로커의 모든 자동 생성 물리적 대상이 사용 불능 메시지 대기열을 사용 가능하게 하거나 사용 불가능하게 할 수 있습니다.

## 사용 불능 메시지 대기열 구성 및 관리

imqcmd 명령 유틸리티는 사용 불능 메시지 대기열을 관리합니다. 약간의 차이는 있지만 사용 불능 메시지 대기열을 관리하는 방법은 다른 대기열을 관리하는 방법과 비슷합니다. 예를 들어, 사용 불능 메시지 대기열은 시스템에서 생성되기 때문에 사용자가 생성, 일시 중지 또는 완전 삭제할 수 없습니다.

### 사용 불능 메시지 대기열 등록 정보

사용 불능 메시지 대기열은 다른 대기열과 같은 방법으로 구성하지만 특정 물리적 대상 등록 정보가 적용되지 않거나 다른 기본값을 갖습니다. 표 6-3에는 사용 불능 메시지 대기열에서 고유한 방법으로 처리하는 대기열 등록 정보가 나열되어 있습니다.

**표 6-3** 사용 불능 메시지 대기열에서 처리하는 표준 물리적 대상 등록 정보

등록 정보	사용 불능 메시지 대기열의 고유한 처리
limitBehavior	사용 불능 메시지 대기열의 기본값은 REMOVE_OLDEST입니다. 다른 대기열의 기본값은 REJECT_NEWEST입니다. 사용 불능 메시지 대기열에는 흐름 제어가 지원되지 않습니다.
localDeliveryPreferred	사용 불능 메시지 대기열에 적용되지 않습니다.
maxNumMsgs	사용 불능 메시지 대기열의 기본값은 1000이고, 다른 대기열의 기본값은 -1(제한 없음)입니다.
maxNumProducers	사용 불능 메시지 대기열에 적용되지 않습니다.
maxTotalMsgBytes	사용 불능 메시지 대기열의 기본값은 10MB입니다. 다른 대기열의 기본값은 -1(제한 없음)입니다.
isLocalOnly	브로커 클러스터에서 사용 불능 메시지 대기열은 항상 로컬 물리적 대상이고 이 등록 정보는 true로 영구 설정됩니다. 하지만 로컬 브로커의 사용 불능 메시지 대기열은 로컬 브로커가 메시지를 사용 불능으로 표시할 경우 클러스터에 있는 다른 브로커의 클라이언트가 생성한 메시지를 포함할 수 있습니다.

## 메시지 내용

브로커는 헤더와 등록 정보 데이터만 유지하고 전체 메시지를 사용 불능 메시지 대기열에 넣거나 메시지 본문 내용을 제거할 수 있습니다. 기본적으로 사용 불능 메시지 대기열은 전체 메시지를 저장합니다.

대기열 크기를 줄이고 사용 불능 메시지를 복원하지 않으려면 본문 내용 제거를 고려해보십시오.

아래 예에 표시된 것처럼 본문 내용을 제거하고 헤더와 등록 정보 데이터만 유지하려면 `imq.destination.DMQ.truncateBody` 브로커 등록 정보를 `true`로 설정합니다.

```
imqcmd update bkr -o imq.destination.DMQ.truncateBody=true
```

## 사용 불능 메시지 로깅 사용

표준 대기열 모니터링 및 로깅 옵션 이외에 브로커가 사용 불능으로 분류한 메시지를 기록할 수 있습니다.

사용 불능 메시지 로깅을 사용하는 경우 브로커는 다음과 같은 이벤트 유형을 기록합니다.

- 브로커가 메시지를 사용 불능 메시지 대기열로 이동합니다.
- 브로커가 사용 불능 메시지 대기열에서 그리고 사용 불능 메시지 대기열을 사용하지 않는 모든 물리적 대상에서 메시지를 제거합니다.
- 물리적 대상이 한계에 도달했습니다.

사용 불능 메시지 로깅은 기본적으로 비활성화됩니다. 다음 예는 사용 불능 메시지 로깅을 활성화합니다.

```
imqcmd update bkr -o imq.destination.logDeadMsgs=true
```

사용 불능 메시지 로깅은 사용 불능 메시지 대기열을 사용하는 모든 물리적 대상에 적용됩니다. 개별 물리적 대상에 대해 로깅을 활성화하거나 비활성화할 수 없습니다.

# 보안 관리

관리자는 사용자 인증에 사용할 사용자 저장소를 구성하고 액세스 제어를 정의하며 클라이언트-브로커 통신을 암호화하는 SSL(Secure Socket Layer) 연결 서비스를 구성하고 브로커를 시작할 때 사용할 passfile을 설정합니다.

이 장은 다음 내용으로 구성되어 있습니다.

- 142페이지의 "사용자 인증"
- 152페이지의 "사용자 권한 부여: 액세스 제어 등록 정보 파일"
- 159페이지의 "SSL 기반 서비스 작업"
- 169페이지의 "Passfile 사용"
- 171페이지의 "감사 로그 만들기"

## 사용자 인증

사용자 저장소에서 사용자 목록, 해당 그룹, 비밀번호를 관리해야 합니다. 브로커 인스턴스마다 다른 사용자 저장소를 사용할 수 있습니다. 이 절에서는 이러한 저장소를 만들고 채우고 관리하는 방법에 대해 설명합니다.

사용자가 브로커에 연결하려고 시도하면 브로커는 제공된 아이디와 비밀번호를 검사하여 사용자를 인증합니다. 아이디와 비밀번호가 각 브로커가 참조하도록 구성된 브로커별 사용자 저장소의 아이디 및 비밀번호와 일치하면 브로커는 연결을 허용합니다.

저장소 유형은 다음 중 하나가 될 수 있습니다.

- **Message Queue**와 함께 제공되는 플랫폼 파일 저장소

이 사용자 저장소 유형은 사용하기가 매우 쉽습니다. 사용자 관리자 유틸리티 (`imqusermgr`)를 사용하면 저장소를 채우고 관리할 수 있습니다. 인증을 사용하려면 각 사용자의 아이디, 비밀번호, 사용자 그룹 이름으로 사용자 저장소를 채웁니다.

사용자 저장소의 설정 및 관리에 대한 자세한 내용은 ["플랫폼 파일 사용자 저장소 사용"](#)을 참조하십시오.

- **LDAP 서버**

LDAP v2 또는 v3 프로토콜을 사용하는 기존 또는 새 LDAP 디렉토리 서버가 될 수 있습니다. 플랫폼 파일 저장소만큼 사용이 쉽지는 않지만 더 확장 가능하므로 작업 환경에 적합합니다.

LDAP 사용자 저장소를 사용하는 경우에는 LDAP 공급업체에서 제공하는 도구를 사용하여 사용자 저장소를 채우고 관리해야 합니다. 자세한 내용은 [149페이지의 "사용자 저장소에 LDAP 서버 사용"](#)을 참조하십시오.

## 플랫폼 파일 사용자 저장소 사용

Message Queue에는 플랫폼 파일 사용자 저장소와 플랫폼 파일 사용자 저장소를 채우고 관리할 때 사용할 수 있는 명령줄 도구인 Message Queue 사용자 관리자(`imqusermgr`)가 제공됩니다. 다음 절에서는 플랫폼 파일 사용자 저장소에 대해 설명하고 Message Queue 사용자 관리자 유틸리티(`imqusermgr`)를 사용하여 해당 저장소를 채우고 관리하는 방법을 설명합니다.



## 사용자 저장소 만들기

플랫 파일 사용자 저장소는 인스턴스별로 고유합니다. 기본 사용자 저장소(passwd)가 시작하는 각 브로커 인스턴스에 대해 만들어집니다. 이 사용자 저장소는 저장소와 연관된 브로커 인스턴스의 이름으로 식별되는 디렉토리에 저장됩니다([부록 A, "운영 체제별 Message Queue 데이터의 위치"](#) 참조).

```
.../instances/instanceName/etc/passwd
```

저장소는 아래의 [표 7-1](#)과 같이 두 개의 항목(행)으로 만들어집니다.

**표 7-1** 사용자 저장소 초기 항목

사용자 아이디	비밀번호	그룹	상태
admin	admin	admin	active
guest	guest	anonymous	active

이러한 초기 항목을 사용하면 관리자가 개입하지 않아도 설치 후 바로 Message Queue 브로커를 사용할 수 있습니다. 초기 아이디/비밀번호 설정은 Message Queue 브로커를 사용하는 데 필요하지 않습니다.

초기 guest 사용자 항목을 사용하면 클라이언트에서 기본 guest 사용자 아이디와 비밀번호를 사용하여 브로커 인스턴스에 연결할 수 있습니다(예를 들어 테스트 목적으로).

초기 admin 사용자 항목을 사용하면 imqcmd 명령을 사용하여 기본 admin 사용자 아이디와 비밀번호로 브로커 인스턴스를 관리할 수 있습니다. 비밀번호를 변경하려면 이 초기 항목을 업데이트해야 합니다([148페이지의 "기본 관리자 비밀번호 변경"](#) 참조).

다음 절에서는 플랫 파일 사용자 저장소를 채우고 관리하는 방법을 설명합니다.

## 사용자 관리자 유틸리티(imqusermgr)

사용자 관리자 유틸리티(imqusermgr)를 사용하면 플랫 파일 사용자 저장소를 편집하거나 채울 수 있습니다. 이 절에서는 사용자 관리자 유틸리티에 대해 소개합니다. 다음 절에서는 imqusermgr 하위 명령을 사용하여 특정 작업을 수행하는 방법에 대해 설명합니다.

imqusermgr 명령에 대한 자세한 내용은 [13장, "명령 참조"](#)를 참조하십시오.

imqusermgr를 사용하기 전에 주의해야 할 사항은 다음과 같습니다.

- 브로커별 사용자 저장소가 없는 경우 해당 브로커 인스턴스를 시작하여 브로커별 사용자 저장소를 만들어야 합니다.
- imqusermgr 명령은 브로커가 설치된 호스트에서 실행해야 합니다.
- 저장소에 쓸 수 있는 적절한 권한이 있어야 합니다. 즉, Solaris 및 Linux의 경우 해당 브로커 인스턴스를 처음으로 만든 사용자나 루트 사용자여야 합니다.

---

**주**            다음 절의 예에서는 기본 브로커 인스턴스인 경우를 가정합니다.

---

### 하위 명령

imqusermgr 명령에는 add, delete, list 및 update 하위 명령이 있습니다.

**add 하위 명령** add 하위 명령은 지정한(또는 기본) 브로커 인스턴스 저장소에 사용자와 관련 비밀번호를 추가하고 선택적으로 사용자 그룹을 지정합니다. 하위 명령의 구문은 다음과 같습니다.

```
add [-i instanceName] -u userName -p passwd [-g group] [-s]
```

**delete 하위 명령** delete 하위 명령은 지정한(또는 기본) 브로커 인스턴스 저장소에서 지정한 사용자를 삭제합니다. 하위 명령의 구문은 다음과 같습니다.

```
delete [-i instanceName] -u userName [-s] [-f]
```

**list 하위 명령** list 하위 명령은 지정한(또는 기본) 브로커 인스턴스 저장소의 지정한 사용자 또는 모든 사용자에 대한 정보를 표시합니다. 하위 명령의 구문은 다음과 같습니다.

```
list [-i instanceName] [-u userName]
```

**update 하위 명령** update 하위 명령은 지정한(또는 기본) 브로커 인스턴스 저장소에 있는 지정한 사용자의 비밀번호 및/또는 상태를 업데이트합니다. 하위 명령의 구문은 다음과 같습니다.

```
update [-i instanceName] -u userName -p passwd [-a state] [-s] [-f]
```

```
update [-i instanceName] -u userName -a state [-p passwd] [-s] [-f]
```

## 명령 옵션

표 7-2에는 imqusermgr 명령의 옵션이 나열되어 있습니다.

**표 7-2** imqusermgr 옵션

옵션	설명
-a <i>active_state</i>	사용자 상태의 활성화 여부를 지정합니다(true/false). true 값은 활성화 상태를 나타내며 기본값입니다.
-f	사용자의 확인 없이 작업을 수행합니다.
-h	사용 도움말을 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.
-i <i>instanceName</i>	명령이 적용될 브로커 인스턴스 사용자 저장소를 지정합니다. 지정하지 않으면 기본 인스턴스 이름 imqbroker인 것으로 가정합니다.
-p <i>passwd</i>	사용자의 비밀번호를 지정합니다.
-g <i>group</i>	사용자 그룹을 지정합니다. 유효한 값에는 admin, user, anonymous가 있습니다.
-s	자동 모드를 설정합니다.
-u <i>userName</i>	사용자 아이디를 지정합니다.
-v	버전 정보를 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.

## 그룹

브로커 인스턴스의 사용자 저장소에 사용자 항목을 추가할 때, 사전 정의된 admin, user, anonymous의 세 가지 그룹 중 하나를 지정할 수 있습니다. 그룹을 지정하지 않으면 기본 그룹인 user가 할당됩니다.

- **admin 그룹.** 브로커 관리자에 사용됩니다. 이 그룹에 할당된 사용자는 기본적으로 브로커를 구성하고 관리할 수 있습니다. admin 그룹에 사용자를 두 명 이상 할당할 수도 있습니다.
- **user 그룹.** 관리자가 아닌 일반 Message Queue 클라이언트 사용자에게 사용됩니다. 대부분의 클라이언트 사용자는 user 그룹에 속합니다. 기본적으로 이 그룹의 사용자는 모든 주제와 대기열에 메시지를 생성하고 모든 주제와 대기열에서 메시지를 사용하고 모든 대기열에서 메시지를 찾아볼 수 있습니다.

- **anonymous 그룹.** 브로커에 알려진 사용자 아이디를 사용하지 않는(클라이언트 응용 프로그램에서 사용해야 할 실제 사용자의 아이디를 모르기 때문일 수 있음) Message Queue 클라이언트에 사용됩니다. 이 계정은 대부분의 FTP 서버에 있는 anonymous 계정과 유사합니다. anonymous 그룹에는 한 번에 한 사용자만 할당할 수 있습니다. 이 그룹의 액세스 권한을 user 그룹에 비해 제한하거나 배포할 때 이 그룹에서 사용자를 제거해야 합니다.

사용자의 그룹을 변경하려면, 사용자 항목을 삭제한 후 그 사용자에게 다른 항목을 추가하여 새 그룹을 지정해야 합니다.

시스템에서 생성한 그룹의 경우 이름을 변경하거나 삭제할 수 없습니다. 또한 새 그룹을 만들 수도 없습니다. 하지만 해당 그룹의 구성원이 수행할 수 있는 작업을 정의하는 액세스 규칙을 지정할 수 있습니다. 자세한 내용은 152페이지의 "사용자 권한 부여: 액세스 제어 등록 정보 파일"을 참조하십시오.

## 사용자 상태

사용자를 저장소에 추가하면 그 사용자는 기본적으로 활성 상태가 됩니다. 사용자를 비활성 상태로 만들려면 update 명령을 사용해야 합니다. 예를 들어, 다음 명령은 사용자 JoeD를 비활성 상태로 만듭니다.

```
imqusermgr update -u JoeD -a false
```

비활성 상태인 사용자의 항목은 저장소에 보존되지만, 비활성 상태인 사용자가 새 연결을 열 수는 없습니다. 사용자가 비활성 상태일 때 같은 이름을 가진 다른 사용자를 추가하면 작업이 실패합니다. 비활성 사용자 항목을 삭제하거나, 새 사용자의 이름을 변경하거나, 새 사용자에게 다른 이름을 사용해야 합니다. 그러면 중복되는 사용자 아이디를 추가하지 않게 됩니다.

## 사용자 아이디 및 비밀번호 형식

사용자 아이디와 비밀번호는 다음과 같은 지침을 따라야 합니다.

- 사용자 이름은 별표(\*), 쉼표(,), 콜론(:), 줄바꿈 또는 캐리지 리턴을 포함할 수 없습니다.
- 사용자 아이디 또는 비밀번호는 한 문자 이상이어야 합니다.
- 사용자 아이디 또는 비밀번호에 공백이 포함된 경우에는 아이디 또는 비밀번호 전체를 따옴표로 묶어야 합니다.
- 명령 셸에서 명령줄에 입력할 수 있는 최대 문자 수가 제한되어 있는 것 외에는 비밀번호 또는 사용자 아이디에 적용되는 길이 제한은 없습니다.

## 사용자 저장소 채우기 및 관리

저장소에 사용자를 추가하려면 `add` 하위 명령을 사용합니다. 예를 들어, 다음 명령은 기본 브로커 인스턴스 사용자 저장소에 비밀번호가 `sesame`인 사용자 `Katharine`을 추가합니다.

```
imqusermgr add -u Katharine -p sesame -g user
```

저장소에서 사용자를 삭제하려면 `delete` 하위 명령을 사용합니다. 예를 들어, 다음 명령은 `Bob`이라는 사용자를 삭제합니다.

```
imqusermgr delete -u Bob
```

사용자의 비밀번호 또는 상태를 변경하려면 `update` 하위 명령을 사용합니다. 예를 들어, 다음 명령은 `Katharine`의 비밀번호를 `aladdin`으로 변경합니다.

```
imqusermgr update -u Katharine -p aladdin
```

하나 이상의 사용자에 대한 정보를 나열하려면 `list` 명령을 사용합니다. 다음 명령은 `isa`라는 사용자에 대한 정보를 표시합니다.

```
imqusermgr list -u isa
```

```
% imqusermgr list -u isa

User repository for broker instance: imqbroker
-----
User Name      Group      Active State
-----
isa            admin      true
```

다음 명령은 모든 사용자에 대한 정보를 나열합니다.

```
imqusermgr list
```

```
% imqusermgr list

User repository for broker instance: imqbroker
-----
User Name          Group             Active State
-----
admin              admin             true
guest              anonymous          true
isa                admin             true
testuser1          user              true
testuser2          user              true
testuser3          user              true
testuser4          user              false
testuser5          user              false
```

### 기본 관리자 비밀번호 변경

보안을 위해 admin의 기본 비밀번호를 혼자만 아는 값으로 변경해야 합니다. 그러려면 imqusermgr 도구를 사용해야 합니다.

다음 명령은 mybroker 브로커 인스턴스의 기본 관리자 비밀번호를 admin에서 grandpoobah로 변경합니다.

```
imqusermgr update mybroker -u admin -p grandpoobah
```

브로커 인스턴스가 실행 중일 때 명령줄 도구 중 하나를 실행하여 이러한 변경 사항의 적용 여부를 빠르게 확인할 수 있습니다. 예를 들어, 다음 명령은 비밀번호를 묻습니다.

```
imqcmd list svc mybroker -u admin
```

새 비밀번호(grandpoobah)를 입력할 수 있어야 하고 이전 비밀번호는 실패해야 합니다.

비밀번호를 변경한 후에 관리 콘솔을 포함한 Message Queue 관리 도구를 사용하려면 새 비밀번호를 제공해야 합니다.

## 사용자 저장소에 LDAP 서버 사용

사용자 저장소에 LDAP 서버를 사용하려면 다음 작업을 수행합니다.

- 인스턴스 구성 파일 편집
- 관리자에 대한 액세스 제어 설정

### 인스턴스 구성 파일 편집

브로커가 디렉토리 서버를 사용하게 하려면 브로커 인스턴스 구성 파일 `config.properties`에서 특정 등록 정보의 값을 설정해야 합니다. 이러한 등록 정보를 사용하면 브로커 인스턴스가 LDAP 서버에 사용자 및 그룹에 대한 정보를 쿼리할 수 있습니다. 브로커는 사용자가 브로커 인스턴스 연결을 시도하거나 특정 메시징 작업을 수행할 때마다 LDAP 서버에 쿼리합니다.

인스턴스 구성 파일은 브로커 인스턴스 디렉토리 아래의 디렉토리에 있습니다. 경로 형식은 다음과 같습니다.

```
.../instances/instanceName/props/config.properties
```

운영 체제별 인스턴스 디렉토리 위치에 대한 자세한 내용은 [부록 A, "운영 체제별 Message Queue 데이터의 위치"](#)를 참조하십시오.

#### ▶ LDAP 서버를 사용하도록 구성 파일을 편집하는 방법

1. 다음 등록 정보를 설정하여 LDAP 사용자 저장소를 사용하고 있음을 지정합니다.

```
imq.authentication.basic.user_repository=ldap
```

2. `imq.authentication.type` 등록 정보를 설정하여 클라이언트에서 브로커로 전달되는 비밀번호를 base64 인코딩(basic) 또는 MD5 다이제스트(digest) 중 어느 쪽으로 할 것인지 결정합니다. 사용자 저장소에 LDAP 디렉토리를 사용하는 경우에는 인증 유형을 basic으로 설정해야 합니다. 예를 들면 다음과 같습니다.

```
imq.authentication.type=basic
```

- LDAP 액세스를 제어하는 브로커 등록 정보도 설정해야 합니다. 이러한 등록 정보는 브로커의 인스턴스 구성 파일에 저장됩니다. 등록 정보에 대해서는 이 절의 뒷부분에서 요약하여 설명합니다.

Message Queue는 JNDI API를 사용하여 LDAP 디렉토리 서버와 통신합니다. 이러한 등록 정보에 사용되는 구문과 용어에 대한 자세한 내용은 JNDI 설명서를 참조하십시오. Message Queue에서는 Sun JNDI LDAP 공급자를 사용하며 단순 인증을 사용합니다.

Message Queue는 LDAP 인증 페일오버를 지원하므로 인증을 시도할 LDAP 디렉토리 서버 목록을 지정할 수 있습니다(자세한 내용은 `imq.user.repos.ldap.server` 등록 정보 참조).

LDAP 사용자 저장소 관련 등록 정보를 설정하는 방법은 브로커의 `config.properties` 파일을 참조하십시오.

- 필요한 경우에는 액세스 제어 등록 정보 파일에서 사용자/그룹과 규칙을 편집해야 합니다. 액세스 제어 등록 정보 파일 사용에 대한 자세한 내용은 [152페이지의 "사용자 권한 부여: 액세스 제어 등록 정보 파일"](#)을 참조하십시오.
- 연결 인증 및 그룹 검색 중 브로커가 SSL을 통해 LDAP 디렉토리 서버와 통신하도록 하려면 LDAP 서버에서 SSL을 활성화한 후 브로커 구성 파일에서 다음 등록 정보를 설정해야 합니다.
  - LDAP 서버가 SSL 통신에 사용하는 포트를 지정합니다. 예를 들면 다음과 같습니다.
 

```
imq.user_repository.ldap.server=myhost:7878
```
  - 브로커 등록정보 `imq.user_repository.ldap.ssl.enabled`를 `true`로 설정합니다.

LDAP 관련 등록 정보는 다음과 같습니다.

- `imq.user_repository.ldap.server`. LDAP 서버의 *host:port*
- `imq.user_repository.ldap.principal`. 브로커에서 검색할 디렉토리 서버에 바인드할 때 사용하는 고유 이름입니다.
- `imq.user_repository.ldap.password`. 브로커에서 사용하는 고유 이름에 연결된 비밀번호입니다.
- `imq.user_repository.ldap.base`. 사용자 항목에 사용되는 디렉토리 기반입니다.
- `imq.user_repository.ldap.uidattr`. 사용자를 고유하게 식별하는 값을 가진 공급자별 속성 식별자입니다. `uid`, `cn` 등이 있습니다.



- `imq.user_repository.ldap.usrfilter`. 사용자에게 사용할 JNDI 검색 필터입니다.
- `imq.user_repository.ldap.grpsearch`. 그룹 검색의 사용 여부를 지정하는 부울 값입니다.
- `imq.user_repository.ldap.grpbase`. 그룹 항목에 사용되는 디렉토리 기반입니다.
- `imq.user_repository.ldap.gidattr`. 그룹 이름을 값으로 가진 공급자별 속성 식별자입니다.
- `imq.user_repository.ldap.memattr`. 그룹 구성원의 고유 이름을 값으로 가진 그룹 항목 내의 속성 식별자입니다.
- `imq.user_repository.ldap.grpfiltler`. 그룹에 사용할 JNDI 검색 필터입니다.
- `imq.user_repository.ldap.timeout`. 검색의 시간 제한(초)을 지정하는 정수입니다.
- `imq.user_repository.ldap.ssl.enabled`. 브로커가 LDAP 서버와 통신할 때 SSL 프로토콜을 사용해야 하는지 여부를 지정하는 부울 값입니다.

이러한 등록 정보에 대한 자세한 내용은 [320페이지의 "보안 관리자 등록 정보"](#)를 참조하십시오.

### 관리자에 대한 액세스 제어 설정

관리 사용자를 만들려면 액세스 제어 등록 정보 파일을 사용하여 ADMIN 연결을 생성할 수 있는 사용자와 그룹을 지정합니다. 이러한 사용자와 그룹을 LDAP 디렉토리에서 사전 정의해야 합니다.

ADMIN 연결을 만들 수 있는 모든 사용자나 그룹은 관리 명령을 실행할 수 있습니다.

#### ▶ 관리 사용자 설정 방법

1. 브로커 등록 정보 `imq.accesscontrol.enabled`를 `true`(기본값)로 설정하여 액세스 제어 파일 사용을 가능하게 합니다.

`imq.accesscontrol.enabled` 등록 정보는 액세스 제어 파일을 사용 가능하게 합니다.

2. 액세스 제어 파일 `accesscontrol.properties`를 엽니다. 파일 위치는 **부록 A, "운영 체제별 Message Queue 데이터의 위치"**에 나와 있습니다.

파일에는 다음과 같은 항목이 포함되어 있습니다.

```
service connection access control
#####
connection.NORMAL.allow.user=*
connection.ADMIN.allow.group=admin
```

나열된 항목은 예입니다. `admin` 그룹은 파일 기반 사용자 저장소에 있지만 기본적으로 LDAP 디렉토리에는 없습니다. LDAP 디렉토리에 정의된 그룹의 이름을 Message Queue 관리자 권한을 부여할 그룹 이름으로 대체해야 합니다.

3. 사용자에게 Message Queue 관리자 권한을 부여하려면 다음과 같이 사용자 아이디를 입력합니다.

```
connection.ADMIN.allow.user=userName [ , userName2 , .. ]
```

4. 그룹에 Message Queue 관리자 권한을 부여하려면 다음과 같이 그룹 아이디를 입력합니다.

```
connection.ADMIN.allow.group=groupName [ , groupName2 , .. ]
```

## 사용자 권한 부여: 액세스 제어 등록 정보 파일

액세스 제어 등록 정보 파일(ACL 파일)에는 사용자와 사용자 그룹이 수행할 수 있는 작업을 지정하는 규칙이 포함되어 있습니다. ACL 파일을 편집하여 이런 작업을 특정 사용자 및 그룹으로 제한합니다. 브로커 인스턴스마다 다른 ACL 파일을 사용할 수 있습니다.

브로커는 클라이언트 응용 프로그램이 다음 작업 중 하나를 수행할 때 ACL 파일을 확인합니다.

- 연결 만들기
- 생성자 만들기
- 사용자 만들기
- 대기열 찾아보기

브로커는 ACL 파일을 확인하여 작업 수행 권한을, 요청을 생성한 사용자에게 부여할지 해당 사용자가 속하는 그룹에 부여할지를 결정합니다.

ACL 파일을 편집한 경우 브로커가 다음에 파일을 검사하여 권한 부여를 확인할 때 새 설정이 적용됩니다. 파일을 편집한 후 브로커를 다시 시작할 필요가 없습니다.

ACL 파일은 사용자 정보가 플랫폼 파일 사용자 저장소(142페이지의 "플랫폼 파일 사용자 저장소 사용" 참조)에 있는지 LDAP 사용자 저장소(149페이지의 "사용자 저장소에 LDAP 서버 사용" 참조)에 있는 지에 관계 없이 모두 사용할 수 있습니다.

## 액세스 제어 등록 정보 파일 작성

ACL 파일은 인스턴스에 고유합니다. 브로커 인스턴스를 시작할 때마다 기본 파일 `accesscontrol.properties`가 인스턴스 디렉토리에 생성됩니다. 파일에 대한 경로 형식은 다음과 같습니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

```
.../instances/brokerInstanceName/etc/accesscontrol.properties
```

ACL 파일은 Java 등록 정보 파일과 같은 형식으로 구성됩니다. 먼저 파일 버전을 정의하는 것으로 시작하여 다음 세 개 섹션에서 액세스 제어 규칙을 지정합니다.

- 연결 액세스 제어
- 물리적 대상 액세스 제어
- 물리적 대상 자동 작성 액세스 제어

`version` 등록 정보는 ACL 등록 정보 파일의 버전을 정의하며 이 항목은 변경할 수 없습니다.

```
version=JMQFileAccessControlModel/100
```

액세스 규칙의 기본 구문, 권한을 계산하는 방법 및 ACL 파일에서 액세스 제어를 지정하는 세 개 섹션에 대한 설명이 아래에 나와 있습니다.

## 액세스 규칙 구문

ACL 등록 정보 파일에서 액세스 제어는 물리적 대상 및 연결 서비스와 같이 보호된 자원에 대해 특정 사용자 또는 그룹이 갖는 액세스 권한을 정의합니다. 액세스 제어는 규칙 또는 규칙 집합으로 나타내며 각 규칙은 Java 등록 정보로 표시됩니다.

이 규칙의 기본 구문은 다음과 같습니다.

```
resourceType.resourceVariant.operation.access.principalType = principals
```

표 7-3에서는 구문 규칙의 요소에 대해 설명합니다.

**표 7-3** 액세스 규칙의 구문 요소

요소	설명
<i>resourceType</i>	connection, queue 또는 topic 중 하나입니다.
<i>resourceVariant</i>	<i>resourceType</i> 에서 지정한 유형의 인스턴스입니다. 예를 들면 myQueue가 있습니다. 와일드카드 문자(*)를 사용하여 모든 연결 서비스 유형 또는 모든 물리적 대상을 나타낼 수 있습니다.
<i>operation</i>	값은 표현할 액세스 규칙의 종류에 따라 달라집니다.
<i>access</i>	allow 또는 deny 중 하나입니다.
<i>principalType</i>	user 또는 group 중 하나입니다. 자세한 내용은 145페이지의 "그룹"을 참조하십시오.
<i>principals</i>	규칙의 왼쪽에 지정한 액세스를 갖는 사람입니다. principalType이 user이면 개별 사용자 또는 사용자 목록(침표로 구분)이 될 수 있고, principalType이 group이면 한 그룹 또는 그룹 목록(침표로 구분)이 될 수 있습니다. 와일드카드 문자(*)를 사용하여 모든 사용자 또는 모든 그룹을 나타낼 수 있습니다.

다음은 액세스 규칙의 몇 가지 예입니다.

- 다음 규칙은 모든 사용자가 q1이라는 이름의 대기열에 메시지를 보낼 수 있다는 것을 나타냅니다.  
queue.q1.produce.allow.user=\*
- 다음 규칙은 모든 사용자가 모든 대기열에 메시지를 보낼 수 있다는 것을 나타냅니다.  
queue.\*.produce.allow.user=\*

---

**주** ASCII가 아닌 사용자, 그룹 또는 대상 이름을 지정하려면 유니코드 제어 (\uXXXX) 표기를 사용해야 합니다. ACL 파일을 편집한 후 ASCII 인코딩이 아닌 이름으로 저장한 경우에는 `Java native2ascii` 도구를 사용해서 해당 파일을 ASCII로 변환할 수 있습니다. 자세한 내용은 <http://java.sun.com/j2se/1.4/docs/guide/intl/faq.html>을 참조하십시오.

---

## 사용 권한 계산 방법

파일에 여러 액세스 규칙이 있는 경우 사용 권한은 다음과 같이 계산됩니다.

- 구체적인 액세스 규칙은 일반적인 액세스 규칙을 대체합니다. 다음 두 규칙을 적용하고 나면 모든 사용자가 모든 대기열에 전송할 수 있지만 Bob은 tq1에 전송할 수 없습니다.

```
queue.*.produce.allow.user=*
queue.tq1.produce.deny.user=Bob
```

- 명시된 *principal*로 지정한 액세스는 \* *principal*로 지정한 액세스를 대체합니다. 다음 규칙은 tq1에서 메시지를 생성할 권한을 Bob에게는 허용하지 않지만 다른 모든 사람에게는 허용합니다.

```
queue.tq1.produce.allow.user=*
queue.tq1.produce.deny.user=Bob
```

- 사용자에 대한 \* *principal* 규칙은 그룹에 대한 해당 \* *principal*을 대체합니다. 예를 들어, 다음 두 규칙은 인증된 모든 사용자가 tq1에 메시지를 보낼 수 있도록 허용합니다.

```
queue.tq1.produce.allow.user=*
queue.tq1.produce.deny.group=*
```

- 사용자에게 부여된 액세스 권한은 사용자 그룹에 부여된 액세스 권한을 대체합니다. 다음 예에서 Bob은 User의 구성원이지만 tq1에 메시지를 생성할 수 없습니다. 다른 모든 User 구성원은 메시지를 생성할 수 있습니다.

```
queue.tq1.produce.allow.group=User
queue.tq1.produce.deny.user=Bob
```

- 액세스 규칙을 통해 명확하게 부여되지 않은 모든 액세스 권한은 거부됩니다. 예를 들어, ACL 파일에 액세스 규칙이 없으면 모든 사용자의 모든 작업이 거부됩니다.

- 같은 사용자 또는 그룹에 권한을 거부하고 허용하면 설정이 서로 상쇄됩니다. 예를 들어, 다음과 같은 두 규칙이 있으면 Bob은 q1을 찾아볼 수 없습니다.

```
queue.q1.browse.allow.user=Bob
```

```
queue.q1.browse.deny.user=Bob
```

다음 두 규칙은 User 그룹이 q5에서 메시지를 사용하지 못하게 합니다.

```
queue.q5.consume.allow.group=User
```

```
queue.q5.consume.deny.group=User
```

- 왼쪽의 같은 규칙이 여러 개 있는 경우에는 마지막 항목만 적용됩니다.

## 연결 서비스에 대한 액세스 제어

ACL 등록 정보 파일의 연결 액세스 제어 섹션에는 브로커의 연결 서비스에 대한 액세스 제어 규칙이 있습니다. 연결 액세스 제어 규칙의 구문은 다음과 같습니다.

```
connection.resourceVariant.access.principalType = principals
```

*resourceVariant*에 정의할 수 있는 값은 NORMAL과 ADMIN 두 가지입니다. 이러한 사전 정의된 값은 사용자가 액세스 권한을 부여할 수 있는 유일한 연결 서비스 유형입니다.

기본 ACL 등록 정보 파일은 모든 사용자에게 NORMAL 연결 서비스에 대한 액세스 권한을 부여하고 admin 그룹에 속하는 사용자에게는 ADMIN 연결 서비스에 대한 액세스 권한을 부여합니다.

```
connection.NORMAL.allow.user=*
```

```
connection.ADMIN.allow.group=admin
```

파일 기반 사용자 저장소를 사용하는 경우 기본 그룹 admin이 imqusermgr에 의해 생성됩니다. LDAP 사용자 저장소를 사용하는 경우 다음 중 하나를 수행하여 기본 ACL 등록 정보 파일을 사용할 수 있습니다.

- LDAP 디렉토리에서 admin 그룹을 정의합니다.
- ACL 등록 정보 파일의 이름 admin을 LDAP 디렉토리에 정의된 하나 이상의 그룹 이름으로 교체합니다.

연결 액세스 권한을 제한할 수 있습니다. 예를 들어, 다음 규칙에서는 NORMAL에 대한 Bob의 액세스를 거부하지만 다른 모든 사람은 허용합니다.

```
connection.NORMAL.deny.user=Bob
```

```
connection.NORMAL.allow.user=*
```

별표(\*) 문자를 사용해서 인증된 모든 사용자 또는 그룹을 지정할 수 있습니다.

ACL 등록 정보 파일을 사용하여 ADMIN 연결에 대한 액세스 권한을 부여하는 방법은 파일 기반 사용자 저장소와 LDAP 사용자 저장소에서 다음과 같이 다릅니다.

- **파일 기반 사용자 저장소**
  - 액세스 제어가 비활성화된 경우 admin 그룹의 사용자가 ADMIN 연결 권한을 갖습니다.
  - 액세스 제어가 활성화된 경우 ACL 파일을 편집합니다. ADMIN 연결 서비스에 대한 액세스 권한을 사용자 또는 그룹에 명시적으로 부여합니다.
- **LDAP 사용자 저장소.** LDAP 사용자 저장소를 사용하는 경우 다음을 모두 수행합니다.
  - 액세스 제어를 활성화합니다.
  - ACL 파일을 편집하고 ADMIN 연결을 만들 수 있는 사용자 또는 그룹의 이름을 제공합니다. LDAP 디렉토리 서버에 정의된 사용자 또는 그룹을 지정합니다.

## 물리적 대상에 대한 액세스 제어

액세스 제어 등록 정보 파일의 대상 액세스 제어 섹션에는 물리적 대상 기반 액세스 제어 규칙이 있습니다. 이 규칙은 누가(사용자/그룹) 어디에서(물리적 대상) 무엇을(작업) 할 수 있는지 결정합니다. 이 규칙으로 규제되는 액세스 유형에는 대기열로 메시지 보내기, 주제에 메시지 게시, 대기열에서 메시지 받기, 주제에 가입, 대기열에서 메시지 찾아보기가 포함됩니다.

기본적으로 모든 사용자 또는 그룹은 모든 물리적 대상에 대해 모든 유형의 액세스 권한을 갖습니다. 특정 대상 액세스 규칙을 추가하거나 기본 규칙을 편집할 수 있습니다. 이 절의 나머지 부분에서는 규칙을 직접 작성하기 위해 알아야 하는 물리적 대상 액세스 규칙의 구문에 대해 설명합니다.

대상 규칙의 구문은 다음과 같습니다.

*resourceType.resourceVariant.operation.access.principalType = principals*

표 7-4에서는 이러한 요소에 대해 설명합니다.

**표 7-4** 물리적 대상 액세스 제어 규칙의 요소

구성 요소	설명
<i>resourceType</i>	queue 또는 topic일 수 있습니다.
<i>resourceVariant</i>	물리적 대상 이름 또는 모든 대기열이나 모든 주제를 나타내는 모든 물리적 대상(*)입니다.

**표 7-4** 물리적 대상 액세스 제어 규칙의 요소(계속)

구성 요소	설명
<i>operation</i>	produce, consume 또는 browse 중 하나일 수 있습니다.
<i>access</i>	allow 또는 deny일 수 있습니다.
<i>principalType</i>	user 또는 group일 수 있습니다.

하나 이상의 사용자 또는 하나 이상의 그룹에 액세스를 허용할 수 있습니다.

다음 예에서는 다양한 종류의 물리적 대상 액세스 제어 규칙을 보여줍니다.

- 모든 사용자가 모든 대기열 대상에 메시지를 보낼 수 있도록 허용합니다.  
queue.\*.produce.allow.user=\*
- user 그룹의 어떤 구성원도 Admissions 주제에 가입하지 못하게 합니다.  
topic.Admissions.consume.deny.group=user

## 자동 작성 물리적 대상에 대한 액세스 제어

ACL 등록 정보 파일의 마지막 섹션에는 브로커에서 물리적 대상을 자동 작성하는 사용자 또는 그룹을 지정하는 액세스 규칙이 있습니다.

사용자가 아직 존재하지 않는 물리적 대상에서 생성자 또는 사용자를 만든 경우 브로커의 자동 작성 등록 정보가 활성화되어 있으면 브로커가 해당 대상을 만듭니다.

기본적으로 모든 사용자 또는 그룹은 브로커가 물리적 대상을 자동 작성하도록 할 수 있습니다. 이 권한은 다음과 같은 규칙으로 지정합니다.

```
queue.create.allow.user=*
```

```
topic.create.allow.user=*
```

ACL 파일을 편집하여 이 유형의 액세스를 제한할 수도 있습니다.

물리적 대상 자동 작성 액세스 규칙의 일반 구문은 다음과 같습니다.

```
resourceType.create.access.principalType = principals
```

여기서 *resourceType*은 queue 또는 topic입니다.



예를 들어, 다음 규칙은 브로커가 Snoopy를 제외한 모든 사람에 대해 주제 대상을 자동 작성하도록 허용합니다.

```
topic.create.allow.user=*
topic.create.deny.user=Snoopy
```

물리적 대상 자동 작성 규칙의 효과와 물리적 대상 액세스 규칙의 효과 사이에 모순이 없도록 해야 합니다. 예를 들어, 1) 어떤 사용자도 대상으로 메시지를 보낼 수 없도록 대상 액세스 규칙을 변경했는데 2) 대상의 자동 작성은 허용했다면, 브로커는 물리적 대상이 없는 경우 물리적 대상을 만들기는 *하지만* 메시지를 그 물리적 대상으로 전달하지는 *않습니다*.

## SSL 기반 서비스 작업

SSL(Secure Socket Layer) 표준을 기반으로 하는 연결 서비스에서는 클라이언트와 브로커가 주고 받는 암호화된 메시지를 보냅니다. 이 절에서는 SSL 기반 연결 서비스 설정 방법에 대해 설명합니다.

Message Queue는 SSL(Secure Socket Layer) 표준을 기반으로 하는 다음과 같은 연결 서비스를 지원합니다.

- ssljms, ssladmin 및 cluster가 TCP/IP를 통해 사용됩니다.
- httpsjms는 HTTP를 통해 사용됩니다.

이러한 연결 서비스를 사용하면 클라이언트와 브로커가 주고 받는 메시지를 암호화할 수 있습니다. Message Queue는 자체 서명된 서버 인증서나 서명된 인증서를 기반으로 SSL 암호화를 지원합니다.

SSL 기반 연결 서비스를 사용하려면 키 도구 유틸리티(imqkeytool)를 사용하여 개인 키/공용 키 쌍을 생성합니다. 이 유틸리티는 브로커에 대해 연결을 요청하는 모든 클라이언트로 전달되는, 자체 서명된 인증서에 공용 키를 포함합니다. 그러면 클라이언트에서 이 인증서를 사용하여 암호화된 연결을 설정합니다.

Message Queue의 SSL 기반 연결 서비스가 개념은 비슷하지만, 설정 방법에는 몇 가지 차이점이 있습니다.

이 절의 나머지 부분에서는 TCP/IP를 통해 보안 연결을 설정하는 방법에 대해 설명합니다.

HTTP를 통한 SSL 기반 연결 서비스(httpsjms)를 사용하면 클라이언트와 브로커가 HTTPS 터널 서블릿을 통해 보안 연결을 설정할 수 있습니다. HTTP를 통한 보안 연결 설정에 관한 자세한 내용은 [369페이지의 부록 C, "HTTP/HTTPS 지원"](#)을 참조하십시오.

## TCP/IP에 대한 보안 연결 서비스

다음 SSL 기반 연결 서비스는 TCP/IP를 통해 직접 보안 연결을 제공합니다.

- `ssljms` 서비스는 클라이언트와 브로커 간에 암호화된 보안 연결을 통해 메시지를 전달합니다.
- `ssladmin` 서비스는 Message Queue 명령 유틸리티(`imqcmd`)와 브로커 사이에서 암호화된 보안 연결을 생성합니다. 관리 콘솔(`imqadmin`)에서는 보안 연결이 지원되지 않습니다.
- `cluster` 서비스는 동일 클러스터의 브로커 간에 암호화된 보안 연결을 통해 메시지를 전달하고 브로커 간 통신을 제공합니다(199페이지의 "브로커 간 보안 연결" 참조).

## 자체 서명된 인증서 사용 구성

이 절에서는 자체 서명된 인증서를 사용하여 SSL 기반 서비스를 설정하는 방법을 설명합니다.

인증 수준을 강화하려면 인증 기관이 확인한 서명된 인증서를 사용할 수 있습니다. 이 절의 단계를 수행한 다음 166페이지의 "서명된 인증서 사용 구성"으로 이동하여 추가 단계를 수행하십시오.

### ▶ SSL 기반 연결 서비스를 설정하는 방법

1. 자체 서명된 인증서를 생성합니다.
2. 브로커에서 `ssljms`, `ssladmin` 또는 `cluster` 연결 서비스를 활성화합니다.
3. 브로커를 시작합니다.
4. 클라이언트를 구성하고 실행합니다(`ssljms` 연결 서비스에만 적용).

`ssljms`와 `ssladmin` 연결 서비스를 설정하는 절차는 4단계의 클라이언트 구성 및 실행을 제외하면 서로 같습니다.

각 단계에 대해서는 뒤에서 자세히 설명합니다.

### 1단계. 자체 서명된 인증서 생성

Message Queue 자체 서명된 인증서를 사용한 SSL 지원은 클라이언트가 알려지고 신뢰할 수 있는 서버와 통신한다는 가정 하에 전송 데이터를 보호하기 위한 것입니다.

`imqkeytool` 명령을 실행하여 브로커에 자체 서명된 인증서를 생성합니다. Unix® 시스템에서 키 저장소를 만들 권한을 가지려면 `imqkeytool`을 슈퍼유저(`root`)로 실행해야 할 수 있습니다.

ssljms, ssladmin 또는 cluster 연결 서비스에 동일한 인증서를 사용할 수 있습니다.

명령 프롬프트에서 다음을 입력합니다.

```
imqkeytool -broker
```

유틸리티가 키 저장소 비밀번호를 묻습니다.

```
Generating keystore for the broker ...
```

```
Enter keystore password:
```

그런 다음 유틸리티가 인증서를 소유하는 브로커를 식별하기 위해 정보를 묻는 메시지를 표시합니다. 입력한 정보를 기반으로 X.500 고유 이름이 생성됩니다. 다음 표에서는 프롬프트를 나열 및 설명하고 각 프롬프트에 대한 예를 제공합니다. 굵은 대소문자가 구분되고 공백을 포함할 수 있습니다.

**표 7-5** 자체 서명된 인증서에 필요한 고유 이름 정보

프롬프트	설명	예
성과 이름은 무엇입니까?	X.500 commonName(CN). 브로커를 실행하는 서버의 정규화된 이름을 입력합니다.	myhost.sun.com
조직 단위의 이름은 무엇입니까?	X.500 organizationalUnit(OU). 부서 이름을 입력합니다.	purchasing
조직의 이름은 무엇입니까?	X.500 organizationName(ON). 회사, 정부 기관 등과 같이 더 큰 조직의 이름을 입력합니다.	My Company, Inc.
구/군/시 이름은 무엇입니까?	X.500 localityName(L).	San Francisco
시/도 이름은 무엇입니까?	X.500 stateName(ST). 약어를 사용하지 않고 시/도의 전체 이름을 입력합니다.	California
이 단위에 대한 2자 국가 코드는 무엇입니까?	X.500 country(C).	US

정보 입력이 끝나면 **imqkeytool**이 확인을 위해 해당 정보를 표시합니다. 예를 들면 다음과 같습니다.

```
Is CN=mqserver.sun.com, OU=purchasing, O=My Company, Inc., L=San Francisco, ST=California, C=US correct?
```

값을 다시 입력하려면 기본값을 적용하거나 no를 입력합니다. 현재 값을 적용하고 계속하려면 yes를 입력합니다. 확인이 끝나면 **imqkeytool**이 키 쌍을 생성하는 동안 일시 중지됩니다.

그런 다음 `imgkeytool`은 특정 키 쌍을 잠글 비밀번호(키 비밀번호)를 묻습니다. 키 비밀번호 및 키 저장소 비밀번호와 동일한 비밀번호를 사용하려면 이 프롬프트에서 **Return** 키를 누릅니다.

---

**주**      입력한 비밀번호를 기억해 두십시오. 브로커를 시작할 때 이 비밀번호를 입력해야 브로커가 키 저장소를 열 수 있습니다. 키 저장소 비밀번호를 `passfile`에 저장할 수 있습니다(169페이지의 "[Passfile 사용](#)" 참조).

---

`imgkeytool`을 실행하면 `JDK keytool` 유틸리티가 실행되어 자체 서명된 인증서를 생성하고 생성된 인증서를 [부록 A, "운영 체제별 Message Queue 데이터의 위치"](#)에 표시된 대로 운영 체제에 따라 다른 디렉토리에 있는 `Message Queue`의 키 저장소에 넣습니다.

키 저장소의 형식은 `JDK1.2 keytool` 유틸리티에서 지원하는 것과 같습니다.

`Message Queue` 키 저장소에 대해 구성 가능한 등록 정보는 다음과 같습니다.

- `img.keystore.file.dirpath`. SSL 기반 서비스의 경우: 키 저장소 파일이 있는 디렉토리의 경로를 지정합니다. 기본값은 [부록 A, "운영 체제별 Message Queue 데이터의 위치"](#)를 참조하십시오.
- `img.keystore.file.name`. SSL 기반 서비스의 경우: 키 저장소 파일의 이름을 지정합니다.
- `img.keystore.password`. SSL 기반 서비스의 경우: 키 저장소 비밀번호를 지정합니다.

특정 문제를 해결하기 위해 키 쌍을 다시 생성해야 할 수도 있습니다. 예를 들면 다음과 같습니다.

- 키 저장소 비밀번호를 잊은 경우
- 브로커를 시작할 때 SSL 기반 서비스가 시작되지 않고  
`java.security.UnrecoverableKeyException: Cannot recover key` 예외가 발생합니다.

이 예외는 [160페이지의 "1단계. 자체 서명된 인증서 생성"](#)에서 자체 서명된 인증서를 생성할 때 지정한 키 저장소 비밀번호와 입력한 키 저장소 비밀번호가 다른 경우에 발생할 수 있습니다.

### ▶ 키 쌍을 다시 생성하는 방법

1. 부록 A, "운영 체제별 Message Queue 데이터의 위치"에 나와 있는 위치에 있는 브로커의 키 저장소를 제거합니다.
2. `imgkeytool`을 다시 실행하여 160페이지의 "1단계. 자체 서명된 인증서 생성"의 설명과 같이 키 쌍을 생성합니다.

### 2단계. 브로커에서 SSL 기반 서비스 활성화

브로커에서 SSL 기반 서비스를 활성화하려면 `img.service.activelist` 등록 정보에 `ssljms` 또는 `ssladmin`을 추가해야 합니다.

---

**주** SSL 기반 클러스터 연결 서비스는 `img.service.activelist` 등록 정보 대신 `img.cluster.transport` 등록 정보를 사용하여 활성화합니다. 199페이지의 "브로커 간 보안 연결"을 참조하십시오.

---

### ▶ 브로커에서 SSL 기반 서비스를 활성화하는 방법

1. 브로커의 인스턴스 구성 파일을 엽니다.

인스턴스 구성 파일은 구성 파일이 연관된 브로커 인스턴스의 이름(`instanceName`)으로 식별되는 디렉토리에 있습니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

```
.../instances/instanceName/props/config.properties
```

2. `img.service.activelist` 등록 정보에 항목을 추가하고(항목이 아직 없는 경우) 목록에 SSL 기반 서비스를 포함시킵니다.

기본적으로 이 등록 정보에는 `jms` 및 `admin` 연결 서비스가 포함됩니다. 활성화할 서비스에 따라 `ssljms` 및/또는 `ssladmin` 연결 서비스를 추가해야 합니다.

```
img.service.activelist=jms,admin,ssljms,ssladmin
```

### 3단계. 브로커 시작

브로커를 시작하며 키 저장소 비밀번호를 제공합니다. 비밀번호는 다음 중 한 가지 방법으로 제공할 수 있습니다.

- 브로커를 시작할 때 비밀번호를 묻는 프롬프트가 표시되게 합니다.

```
imgbrokerd
Please enter Keystore password: mypassword
```

- 169페이지의 "Passfile 사용"에서 설명한 대로 passfile에 비밀번호를 저장합니다. passfile에 비밀번호를 저장하고 등록 정보 imq.passfile.enabled=true를 설정한 경우 다음 중 하나를 수행합니다.
  - passfile의 위치를 imqbrokerd 명령에 전달합니다.
 

```
imqbrokerd -passfile /tmp/mypassfile
```
  - -passfile 옵션 없이 브로커를 시작하되, 다음의 두 브로커 구성 등록 정보를 사용하여 passfile의 위치를 지정합니다.
 

```
imq.passfile.dirpath=/tmp
imq.passfile.name=mypassfile
```

SSL을 사용하는 브로커나 클라이언트를 시작할 때 몇 초간 CPU 사이클을 과도하게 사용하는 경우가 있을 수 있습니다. 이것은 Message Queue에서 SSL 구현에 JSSE(Java Secure Socket Extension)를 사용하기 때문입니다. JSSE에서는 java.security.SecureRandom()을 사용하여 난수를 생성합니다. 이 방법은 초기 난수 시드를 만드는 동안 상당한 시간을 소비하기 때문에 CPU 사용량이 많아지게 됩니다. 시드를 만들고 나면 CPU 수준이 정상으로 돌아갑니다.

#### 4단계. SSL기반 클라이언트 구성 및 실행

마지막으로 보안 연결 서비스를 사용하도록 클라이언트를 구성합니다. TCP/IP를 통한 보안 연결 시나리오의 유형은 다음 두 가지입니다.

- ssljms를 사용하는 응용 프로그램 클라이언트
- ssladmin을 사용하는 Message Queue 관리 클라이언트(예: imqcmd)

다음 절에서 이 두 가지에 대해 별도로 설명합니다.

##### *ssljms를 사용하는 응용 프로그램 클라이언트*

클라이언트의 클래스 경로에 필요한 Secure Socket Extension(JSSE) jar 파일이 있는지 확인하고 ssljms 연결 서비스를 사용하도록 지정해야 합니다.

1. 클라이언트에서 (JSSE 및 JNDI 지원이 기본적으로 제공되는) J2SDK1.4를 사용하지 않는 경우에는 클라이언트의 클래스 경로에 다음 jar 파일이 있는지 확인합니다.
 

```
jsse.jar, jnet.jar, jcert.jar, jndi.jar
```
2. 클라이언트의 클래스 경로에 다음 Message Queue jar 파일이 있는지 확인합니다.
 

```
imq.jar, jms.jar
```

3. 클라이언트를 시작하고 브로커의 `ssljms` 서비스에 연결합니다. 이 작업을 수행하는 방법 중 하나는 다음과 같은 명령을 입력하는 것입니다.

```
java -DmqConnectionType=TLS clientAppName
```

`mqConnectionType`을 설정하면 연결에서 SSL을 사용하게 됩니다.

클라이언트 응용프로그램에서의 `ssljms` 연결 서비스 사용에 대한 자세한 내용은 *Message Queue Developer's Guide for Java Clients*의 관리 대상 객체 사용에 대한 장을 참조하십시오.

### *ssladmin*을 사용하는 관리 클라이언트(*imqcmd*)

`imqcmd`를 사용할 때 `-secure` 옵션을 포함하면 보안 관리 연결을 설정할 수 있습니다.

예를 들면 다음과 같습니다.

```
imqcmd list svc -b hostName:port -u adminName -secure
```

여기서 `adminName`은 Message Queue 사용자 저장소에 유효한 항목이고 이 명령은 비밀번호를 묻는 프롬프트를 표시합니다(플랫 파일 저장소를 사용하는 경우 [148페이지의 "기본 관리자 비밀번호 변경"](#) 참조).

연결 서비스를 나열하는 것은 `ssladmin` 서비스가 실행 중이며 다음 출력과 같이 보안 관리 연결을 성공적으로 설정할 수 있다는 것을 보여주는 방법입니다.

Listing all the services on the broker specified by:

Host	Primary Port	
localhost	7676	
Service Name	Port Number	Service State
admin	33984 (dynamic)	RUNNING
httpjms	-	UNKNOWN
httpsjms	-	UNKNOWN
jms	33983 (dynamic)	RUNNING
ssladmin	35988 (dynamic)	RUNNING
ssljms	dynamic	UNKNOWN

Successfully listed services.

## 서명된 인증서 사용 구성

서명된 인증서는 자체 서명된 인증서보다 더 높은 수준의 서버 인증을 제공합니다. 서명된 인증서를 구현하려면 키 저장소에 서명된 인증서를 설치한 다음 `mqbrokerd`에 SSL 연결을 설정할 때 서명된 인증서를 요구하도록 `Message Queue` 클라이언트를 구성합니다.

서명된 인증서는 클라이언트와 브로커 간에만 구현 가능하고 동일 클러스터 내의 브로커 간에는 구현할 수 없습니다.

아래 지침에서는 160페이지의 "자체 서명된 인증서 사용 구성"에서 설명한 단계를 이미 수행한 것으로 가정합니다. 지침을 따르면 <http://java.sun.com>에서 J2SE 키 도구 및 X.509 인증서에 대한 정보에 액세스하는 데 도움이 됩니다.

### 1단계: 서명된 인증서 가져오기 및 설치

#### ▶ 서명된 인증서를 가져오는 방법

1. J2SE 키 도구를 사용하여 방금 생성한 자체 서명된 인증서에 대한 CSR(Certificate Signing Request)을 생성합니다.

예를 들면 다음과 같습니다.

```
keytool -certreq -keyalg RSA -alias imq -file certreq.csr
        -keystore /etc/imq/keystore -storepass myStorePassword
```

CSR은 `certreq.csr` 파일에서 인증서를 캡슐화합니다.

2. 다음 중 하나의 방법으로 서명된 인증서를 생성하거나 요청합니다.
  - Thawte 또는 Verisign과 같이 공신력 있는 인증 기관(CA)에서 서명한 인증서를 가져옵니다. 이 프로세스에 대한 자세한 내용은 해당 CA의 설명서를 참조하십시오.
  - SSL 서명 소프트웨어 패키지를 사용하여 인증서에 직접 서명합니다.

결과적으로 만들어지는 서명된 인증서는 ASCII 문자 시퀀스입니다. CA로부터 서명된 인증서를 받을 때 해당 인증서는 전자 메일 첨부 파일이나 메시지 텍스트로 전달될 수 있습니다.

3. 서명된 인증서를 받은 다음 인증서를 파일에 저장합니다.

이 지침에서는 `broker.cer`이라는 예제 이름을 사용하여 브로커 인증서를 나타냅니다.



### ▶ 서명된 인증서 설치 방법

1. 다음과 같이 \$JAVA\_HOME/lib/security/cacerts를 검사하여 J2SE가 기본적으로 사용자 CA를 지원하는지 여부를 확인합니다.

```
keytool -v -list -keystore $JAVA_HOME/lib/security/cacerts
```

이 명령은 시스템 키 저장소에 있는 루트 CA를 나열합니다.

해당 CA가 목록에 있는 경우 다음 단계를 건너뛸니다.

2. 해당 CA가 J2SE에서 지원되지 않는 경우 인증 기관 루트 인증서를 imqbrokerd 키 저장소로 가져옵니다.

예를 들면 다음과 같습니다.

```
keytool -import -alias ca -file ca.cer -noprompt -trustcacerts
      -keystore /etc/imq/keystore -storepass myStorePassword
```

ca.cer 값은 CA에서 가져온 CA 루트 인증서입니다.

CA 테스트 인증서를 사용하는 경우 테스트 CA 루트 인증서를 가져와야 할 수 있습니다. CA에서 테스트 CA 루트 복사본을 가져오는 방법에 대한 지침을 제공해야 합니다.

3. 서명된 인증서를 키 저장소로 가져와서 원본 자체 서명된 인증서를 대체합니다.

예를 들면 다음과 같습니다.

```
keytool -import -alias imq -file broker.cer -noprompt -trustcacerts
      -keystore /etc/imq/keystore -storepass myStorePassword
```

broker.cer 값은 CA로부터 받은 서명된 인증서가 들어 있는 파일입니다.

이제 imqbrokerd 키 저장소에 SSL 연결에 사용할 서명된 인증서가 있습니다.

## 2단계: 서명된 인증서를 요구하도록 클라이언트 런타임 구성

### ▶ Java 클라이언트 런타임 구성 방법

기본적으로 Message Queue 클라이언트 런타임은 imqbrokerd를 신뢰하므로 제시하는 모든 인증서를 승인합니다. 이제 서명된 인증서를 요구하도록 클라이언트 런타임을 구성하고 클라이언트가 인증서를 서명한 CA를 신뢰하는지 확인합니다.

1. 클라이언트가 imqbrokerd로부터 유효한 서명된 인증서를 요구하도록 구성하려면 클라이언트의 ConnectionFactory 객체에 대해 imqSSLIsHostTrusted 속성을 false로 설정합니다.

2. 164페이지의 "4단계. SSL기반 클라이언트 구성 및 실행"에서 설명한 대로 imqbrokrd에 SSL 연결을 설정해 봅니다.

브로커의 인증서가 공신력 있는 CA에 의해 서명된 경우 연결에 성공할 것이므로 다음 단계를 건너뛸 수 있습니다. 인증서 검증 오류로 인해 연결에 실패할 경우 다음 단계를 수행합니다.

3. 다음 절에서 설명한 대로 클라이언트의 truststore에 서명 CA의 루트 인증서를 설치합니다.

truststore를 사용하여 클라이언트를 구성하는 세 가지 옵션이 있습니다.

- 기본 시스템 cacerts 파일에 루트 CA를 설치합니다.
- 대체 시스템 파일 jssecacerts에 루트 CA를 설치합니다. 이 방법이 권장 옵션입니다.
- 키 저장소 파일에 루트 CA를 설치하고 해당 CA를 truststore로 사용하도록 클라이언트를 구성합니다.

다음 절에서는 이러한 옵션을 사용하여 Verisign Test Root CA를 설치하는 방법을 예를 들어 설명합니다. 루트 CA는 testrootca.cer 파일에 포함되어 있습니다. 이 예에서는 J2SE가 /usr/j2se에 설치되어 있다고 가정합니다.

#### **기본 시스템 cacerts 파일에 설치**

이 예에서는 루트 CA를 \$JAVA\_HOME/usr/jre/lib/security/cacerts 파일에 설치합니다.

```
keytool -import -keystore /usr/j2se/jre/lib/security/cacerts
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

기본적으로 클라이언트가 이 키 저장소를 검색하기 때문에 추가 클라이언트 구성은 필요하지 않습니다.

#### **jssecacerts에 설치**

이 예에서는 루트 CA를 \$JAVA\_HOME/usr/jre/lib/security/jssecacerts 파일에 설치합니다.

```
keytool -import -keystore /usr/j2se/jre/lib/security/jssecacerts
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

기본적으로 클라이언트가 이 키 저장소를 검색하기 때문에 추가 클라이언트 구성은 필요하지 않습니다.

### 다른 파일에 설치

이 예에서는 루트 CA를 /home/smith/.keystore 파일에 설치합니다.

```
keytool -import -keystore /home/smith/.keystore
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

기본적으로 클라이언트는 이 키 저장소를 검색하지 않기 때문에 클라이언트에 **truststore** 위치를 직접 제공해야 합니다. 그렇게 하려면 클라이언트가 실행된 다음 Java 시스템 등록 정보 `javax.net.ssl.trustStore`를 설정합니다. 예를 들면 다음과 같습니다.

```
javax.net.ssl.trustStore=/home/smith/.keystore
```

## Passfile 사용

여러 가지 명령 유형에 비밀번호가 필요합니다. 표 7-6에서 첫 번째 열에는 비밀번호가 필요한 명령이 나열되고 두 번째 열에는 비밀번호가 필요한 이유가 나열됩니다.

**표 7-6** 비밀번호를 사용하는 명령

명령	목적	비밀번호의 목적
<code>imqbrokerd</code>	브로커를 시작합니다.	플러그인 지속성 데이터 저장소, SSL 인증서 키 저장소 또는 LDAP 사용자 저장소에 액세스합니다.
<code>imqcmd</code>	브로커를 관리합니다.	명령 사용 권한이 있는 관리 사용자를 인증합니다.
<code>imqdbmgr</code>	플러그인 데이터 저장소를 관리합니다.	데이터 저장소에 액세스합니다.

비밀번호 파일(*passfile*)에서 비밀번호를 지정하고 `-passfile` 옵션을 사용하여 파일의 이름을 지정할 수 있습니다. `-passfile` 옵션의 형식은 다음과 같습니다.

```
imqbrokerd -passfile myPassfile
```

**주** 이전 릴리스에서는 `-p`, `-password`, `-dbpassword` 및 `-ldappassword` 옵션을 사용하여 명령줄에서 비밀번호를 지정할 수 있었습니다. 이러한 옵션은 더 이상 사용되지 않으며 향후 릴리스에서는 제거됩니다. 이 릴리스에서는 이러한 옵션 중 하나에 대한 명령줄의 값이 `password` 파일의 관련 값보다 우선합니다.

## 보안 문제

모니터를 다른 사람이 보고 있지 않다면 프롬프트에 응답하여 비밀번호를 대화식으로 지정하는 것이 가장 안전한 비밀번호 지정 방법입니다. 명령줄에서 `passfile`을 지정할 수도 있습니다. 명령을 비대화식으로 사용할 경우 `passfile`을 사용해야 합니다.

`passfile`은 암호화되지 않기 때문에 사용 권한을 설정하여 인증되지 않은 액세스로부터 보호해야 합니다. 브로커를 시작하는 사용자가 파일을 볼 수 있지만 읽기 액세스만 허용하도록 사용 권한을 설정합니다.

## Passfile 내용

`passfile`은 일련의 등록 정보와 값이 들어 있는 간단한 텍스트 파일입니다. 각 값은 명령에 사용되는 비밀번호입니다.

`passfile`에는 표 7-7에 있는 것과 같은 비밀번호를 포함할 수 있습니다.

**표 7-7** Passfile의 비밀번호

비밀번호	영향 받는 명령	설명
<code>imq.imqcmd.password</code>	<code>imqcmd</code>	<code>imqcmd</code> 명령줄에 대한 관리자 비밀번호를 지정합니다. 비밀번호는 각 명령에 대해 인증됩니다.
<code>imq.keystore.password</code>	<code>imqbrokerd</code>	SSL 기반 서비스의 키 저장소 비밀번호를 지정합니다.
<code>imq.persist.jdbc.password</code>	<code>imqbrokerd</code> <code>imdbmgr</code>	필요한 경우 데이터베이스 연결을 여는 비밀번호를 지정합니다.
<code>imq.user_repository.ldap.password</code>	<code>imqbrokerd</code>	구성된 LDAP 사용자 저장소에 바인드할 수 있도록 브로커에 할당된 고유 이름에 연결되는 비밀번호를 지정합니다.

샘플 `passfile`은 Message Queue 제품의 일부입니다. 샘플 파일의 위치는 부록 A, "운영 체제별 Message Queue 데이터의 위치"를 참조하십시오.

# 감사 로그 만들기

Message Queue는 엔터프라이즈판에서만 감사 로깅을 지원합니다. 감사 로깅이 활성화 되면 Message Queue는 다음과 같은 이벤트 유형에 대한 레코드를 생성합니다.

- 브로커 인스턴스 시작, 종료, 재시작 및 제거
- 사용자 인증 및 권한 부여
- 영구 저장소 재설정
- 물리적 대상 만들기, 제거 및 완전 삭제
- 관리상 영구 가입자 완전 삭제

Message Queue 브로커 로그 파일에 감사 레코드를 기록하려면 `imq.audit.enabled` 브로커 등록 정보를 `true`로 설정합니다. 로그의 모든 감사 레코드에는 `AUDIT` 키워드가 있습니다.

`imq.audit.enabled` 등록 정보에 대한 자세한 내용은 [320페이지의 "보안 관리자 등록 정보"](#)를 참조하십시오.

감사 로그 만들기

## 관리 대상 객체 관리

관리 대상 객체를 사용하면 다른 JMS 공급자에 이식할 수 있는 클라이언트 응용 프로그램을 개발할 수 있습니다. 관리 대상 객체는 공급자별 구성 및 이름 지정 정보를 캡슐화합니다.

Message Queue 관리자는 일반적으로 클라이언트 응용 프로그램이 브로커에 연결하는데 사용하는 관리 대상 객체를 만듭니다. 클라이언트 응용 프로그램은 연결을 사용하여 물리적 대상과 메시지를 주고 받습니다.

이 장에서는 객체 관리자 유틸리티(imqobjmgr)를 사용해서 이런 작업을 수행하는 방법을 설명합니다. 이 작업을 수행하려면 사용할 객체 저장소와 만들 관리 대상 객체의 속성을 이해하고 있어야 하므로 이 장에서 imqobjmgr을 사용하여 관리 대상 객체를 관리하는 방법을 설명하기 전에 이 두 항목에 대한 배경 정보를 제공합니다.

이 장은 다음 내용으로 구성되어 있습니다.

- [174페이지의 "객체 저장소 정보"](#)
- [176페이지의 "관리 대상 객체 속성 정보"](#)
- [185페이지의 "객체 관리자 유틸리티\(imqobjmgr\) 사용"](#)
- [189페이지의 "관리 대상 객체 추가 및 삭제"](#)
- [193페이지의 "관리 대상 객체 나열"](#)
- [193페이지의 "단일 객체 관련 정보 얻기"](#)
- [194페이지의 "관리 대상 객체 업데이트"](#)

## 객체 저장소 정보

관리 대상 객체는 클라이언트 응용 프로그램에서 JNDI 조회를 통해 액세스할 수 있도록 미리 만들어진 객체 저장소에 있습니다. 사용할 수 있는 객체 저장소의 유형에는 표준 LDAP 디렉토리 서버 또는 파일 시스템 객체 저장소의 두 가지가 있습니다.

### LDAP 서버 객체 저장소

LDAP 서버는 작업 메시징 시스템에 권장되는 객체 저장소입니다. LDAP 구현은 여러 공급업체에서 제공하며 분산 시스템에서 사용할 수 있도록 디자인되어 있습니다. LDAP 서버는 작업 환경에 유용한 보안 기능도 제공합니다.

Message Queue 관리 도구는 LDAP 서버의 객체 저장소를 관리할 수 있습니다. 그러나 먼저 LDAP 서버 설명서에 나와 있는 대로 `java` 객체를 저장하고 JNDI 조회를 수행하도록 LDAP 서버를 구성해야 합니다.

LDAP 서버를 객체 저장소로 사용하는 경우 표 8-1에 나와 있는 속성들을 지정해야 합니다. 이 속성들은 다음 범주로 구분됩니다.

- **초기 컨텍스트:** LDAP 서버 객체 저장소의 이 속성은 고정되어 있습니다.
- **위치:** LDAP 서버에 설정된 대로 관리 대상 객체를 저장할 URL 및 디렉토리 경로를 지정합니다. 특히 지정된 경로가 존재하는지 확인해야 합니다.
- **보안 정보:** LDAP 공급자에 따라 다릅니다. 보안 정보가 모든 작업에서 필요한지 아니면 저장된 데이터를 변경하는 작업에서만 필요한지 확인하려면 LDAP 구현에 제공된 설명서를 참조하십시오.

**표 8-1** LDAP 객체 저장소 속성

속성	설명
<code>java.naming.factory.initial</code>	LDAP 서버에서 JNDI 조회의 초기 컨텍스트 <code>com.sun.jndi.ldap.LdapCtxFactory</code>
<code>java.naming.provider.url</code>	LDAP 서버 URL 및 디렉토리 경로 정보. 예를 들면 다음과 같습니다. <code>ldap://mydomain.com:389/ou=mqobjjs,o=myapp</code> 여기서 관리 대상 객체는 <code>/myapp/mqobjjs</code> 디렉토리에 저장됩니다.



표 8-1 LDAP 객체 저장소 속성(계속)

속성	설명
java.naming.security.principal	LDAP 서버에 대해 호출자를 인증할 때 사용하는 기본 사용자의 아이디. 그 형식은 인증 방법에 따라 달라지며, 예를 들면 다음과 같습니다. uid=fooUser, ou=People, o=mcq 이 등록 정보를 지정하지 않으면 LDAP 서비스 공급자가 동작을 결정합니다.
java.naming.security.credentials	LDAP 서버에 대해 호출자를 인증할 때 사용하는 기본 사용자의 자격 증명. 이 등록 정보 값은 인증 방법에 따라 다르며 해시 비밀번호, 단순 텍스트 비밀번호, 키, 인증서 등이 될 수 있습니다. 예를 들면 다음과 같습니다. fooPasswd 이 등록 정보를 지정하지 않으면 LDAP 서비스 공급자가 동작을 결정합니다.
java.naming.security.authentication	사용할 보안 수준입니다. 값은 none, simple, strong의 세 키워드 중 하나입니다. 예를 들어, simple을 지정하면 누락된 기본 사용자 또는 자격 증명 값을 묻는 프롬프트가 나타납니다. 그러면 아이디 정보를 좀더 안전하게 제공할 수 있습니다. 이 등록 정보를 지정하지 않으면 LDAP 서비스 공급자가 동작을 결정합니다.

## 파일 시스템 객체 저장소

Message Queue에서는 파일 시스템 객체 저장소 구현도 지원합니다. 파일 시스템 객체 저장소는 완전히 테스트되지 않았기 때문에 작업 시스템에는 권장되지 않지만, 개발 환경에서는 매우 사용하기가 쉽다는 장점이 있습니다. LDAP 서버를 설정할 필요 없이 로컬 파일 시스템에 디렉토리를 만들기만 하면 됩니다.

그러나 클라이언트가 여러 컴퓨터 노드에 배포된 경우 이 클라이언트들이 객체 저장소가 위치한 디렉토리에 액세스할 수 없으면 파일 시스템 저장소를 중앙 집중식 객체 저장소로 사용할 수 없습니다. 또한 해당 디렉토리에 액세스할 수 있는 모든 사용자는 Message Queue 관리 도구를 사용하여 관리 대상 객체를 만들고 관리할 수 있습니다.

파일 시스템 객체 저장소를 사용하는 경우 표 8-2에 나와 있는 속성들을 지정해야 합니다. 이 속성들은 다음 범주로 구분됩니다.

- **초기 컨텍스트:** 파일 시스템 객체 저장소의 이 속성 값은 고정되어 있습니다.
- **위치:** 이 속성 값은 관리 대상 객체를 저장할 디렉토리 경로를 지정합니다. 해당 디렉토리가 존재해야 하며 Message Queue 관리 도구 사용자와 이 저장소에 액세스할 클라이언트 응용 프로그램 사용자에게 적절한 액세스 권한이 있어야 합니다.

**표 8-2** 파일 시스템 객체 저장소 속성

속성	설명
java.naming.factory.initial	파일 시스템 객체 저장소에서 JNDI 조회의 초기 컨텍스트: com.sun.jndi.fscontext. RefFSContextFactory
java.naming.provider.url	디렉토리 경로 정보. 예를 들면 다음과 같습니다. file:///C:/myapp/mqobjs

## 관리 대상 객체 속성 정보

Message Queue 관리 대상 객체에는 기본적으로 다음과 같은 두 가지 종류가 있습니다.

- **연결 팩토리** 관리 대상 객체는 클라이언트 응용 프로그램에서 브로커에 연결하는 데 사용됩니다.
- **대상** 관리 대상 객체는 생성자가 메시지를 보내거나 사용자가 메시지를 검색하는 대상을 클라이언트 응용 프로그램에서 식별하는 데 사용됩니다.

Message Queue에는 관리 대상 객체를 만들고 관리할 때 사용되는 두 개의 관리 도구가 있습니다. 하나는 명령줄 객체 관리자 유틸리티(imqobjmgr)이고 다른 하나는 GUI 관리 콘솔입니다. 이 장에서는 명령줄을 사용하는 방법에 대해서만 설명합니다.

관리 대상 객체의 속성은 속성-값 쌍을 사용해서 지정합니다.

## 연결 팩토리 속성

연결 팩토리 구성은 연결 팩토리가 클라이언트 응용 프로그램을 대신하여 만드는 모든 연결에 전달됩니다. 메시지를 보내거나 받을 때 포함되는 대상을 정의하고 클라이언트 런타임에서 메시지 흐름을 처리하는 방법을 지정하며 연결을 통해 보낸 모든 메시지에 대한 특정 정보를 자동으로 설정하도록 연결을 구성합니다.

연결 팩토리 객체에는 다음과 같은 두 가지 유형이 있습니다.

- `ConnectionFactory`는 일반 메시징과 비분산 트랜잭션을 지원합니다.
- `XAConnectionFactory`는 분산 트랜잭션을 지원합니다.

`ConnectionFactory`와 `XAConnectionFactory` 객체는 동일한 속성 집합을 공유합니다.

연결 팩토리 객체는 관리자 또는 응용 프로그램(프로토타입 제작 또는 테스트용)에서 만들고 구성할 수 있습니다. `imqobjmgr` 도구 또는 관리 콘솔을 사용하여 연결 팩토리 속성을 설정합니다.

이 절에서는 연결 팩토리 속성에 대해 설명하며 속성의 영향을 받는 동작을 기준으로 다음 내용으로 구성됩니다.

- [178페이지의 "연결 처리"](#)
- [180페이지의 "클라이언트 아이디"](#)
- [182페이지의 "안정성 및 흐름 제어"](#)
- [183페이지의 "대기열 브라우저 동작 및 서버 세션"](#)
- [184페이지의 "메시지 헤더 무시"](#)
- [183페이지의 "JMS 정의 등록 정보 지원"](#)

여기에서 주로 고려해야 할 속성은 `imqAddressList`로 클라이언트가 연결을 설정할 브로커를 지정하는 데 사용됩니다. [189페이지의 "연결 팩토리 추가"](#)에서는 객체 저장소에 연결 팩토리 관리 대상 객체를 추가할 때 속성을 지정하는 방법을 설명합니다.

연결 팩토리 속성에 대한 자세한 내용은 [16장, "관리 대상 객체 속성 참조"](#) 및 JavaDoc API 설명서의 `Message Queue` 클래스 `com.sun.messaging.ConnectionConfiguration` 부분을 참조하십시오.

## 연결 처리

연결 처리 속성을 사용하여 연결할 메시지 서버 주소를 지정하고 다시 연결해야 하는 경우 재연결 시도 횟수와 시도 간격을 지정합니다.

클라이언트는 `imqAddressList` 속성 값으로 지정된 메시지 서버 주소에 있는 메시지 서버에 연결합니다. 메시지 서버 주소는 브로커 호스트 이름, 포트 번호 및 연결 서비스 유형으로 구성됩니다.

포트 번호는 브로커의 포트 매핑이 있는 포트 또는 특정 연결 서비스가 있는 포트입니다. 포트 매핑 포트를 지정하면 포트 매핑이 연결 포트 번호를 동적으로 할당합니다. 메시지 서버 주소 지정에 대한 자세한 내용은 [335페이지의 "imqAddressList 속성 값 구분"](#)을 참조하십시오.

## 자동 재연결

단일 브로커 환경이나 멀티 브로커 클러스터 환경에서 연결이 실패할 경우 클라이언트가 브로커에 자동으로 다시 연결하도록 해주는 연결 처리 속성을 설정할 수 있습니다. 재연결 프로세스를 구성할 수도 있습니다.

재연결 기능에서는 연결 페일오버는 제공하지만 데이터 페일오버는 제공하지 않으므로 클라이언트가 다른 브로커 인스턴스에 다시 연결되는 경우 실패했거나 연결이 끊어진 브로커에 보관된 지속성 메시지 및 기타 상태 정보는 손실될 수 있습니다.

자동 재연결이 가능한 경우 `Message Queue`는 연결에 실패하면 클라이언트가 다시 연결하여 액세스할 수 있기 때문에 임시 대상을 지속합니다. 클라이언트가 다시 연결하여 이러한 대상을 사용할 수 있는 시간을 준 다음 브로커는 해당 대상을 삭제합니다.

재연결 처리 방법은 클라이언트가 단일 브로커에 연결되는지 클러스터에 속한 브로커에 연결되는지에 따라 달라집니다. 다음 절에서는 이러한 각각의 가능성에 대해 설명합니다.

**단일 브로커에 재연결** 연결에 실패할 경우 클라이언트가 브로커에 자동으로 다시 연결하게 하려면 다음과 같은 연결 팩토리 속성을 설정합니다.

- `imqReconnectEnabled`. 자동 재연결 동작을 사용 가능하게 합니다.
- `imqReconnectAttempts`. 클라이언트 런타임이 클라이언트 재연결을 시도하는 횟수를 지정합니다.
- `imqReconnectInterval`. 클라이언트 런타임이 대기하는 클라이언트 재연결 시도 간격을 지정합니다.

이러한 속성에 대한 자세한 내용은 [334페이지의 "연결 처리"](#)를 참조하십시오.

**클러스터 내의 브로커에 재연결** 멀티 브로커 클러스터 환경에서 `imqAddressList` 속성에 여러 주소를 지정한 경우 브로커 목록을 통해 자동 재연결이 반복됩니다. 목록에 있는 모든 브로커가 **Message Queue Enterprise Edition**을 설치해야 합니다.

목록의 첫 번째 주소에 대한 클라이언트 연결이 실패할 경우 클라이언트 런타임은 클라이언트를 목록의 다른 브로커에 다시 연결하려고 시도합니다. 이 시도도 실패할 경우 클라이언트 런타임은 클라이언트를 다시 연결할 수 있을 때까지 목록에 있는 브로커를 차례로 연결합니다.

모든 시도가 실패할 경우 사용 가능한 브로커를 찾거나 찾기에 실패할 때까지 클라이언트 런타임은 목록을 지정된 횟수만큼 순환합니다. `imqAddressListBehavior` 속성 설정에 따라 다시 연결할 때 주소 목록에 제공된 주소 순서로 다음 브로커를 선택할지 아니면 해당 목록에서 임의로 선택할지 여부를 결정합니다.

클라이언트가 클러스터 내의 브로커에 다시 연결하게 하려면 다음 속성을 사용합니다.

- `imqReconnectEnabled`. 자동 재연결 동작을 사용 가능하게 합니다.
- `imqReconnectAttempts`. 다음 브로커 주소로 넘어가기 전에 각 브로커 주소에 대해 연결을 시도하는 횟수를 지정합니다.
- `imqReconnectInterval`. 대기해야 할 시도 간격을 지정합니다.
- `imqAddressListIterations`. 목록을 통해 반복하는 횟수를 지정합니다.
- `imqAddressListBehavior`. 연결을 주소 목록에 있는 주소 순서로 시도할지 임의의 순서로 시도할지를 지정합니다.

이러한 속성에 대한 자세한 내용은 [334페이지의 "연결 처리"](#)를 참조하십시오.

### 연결 핑

`imqPingInterval` 속성은 클라이언트 런타임에서 브로커로의 핑 작업 빈도를 지정합니다. 클라이언트 런타임은 연결을 주기적으로 테스트하여 실패한 연결을 우선적으로 감지할 수 있습니다. 핑 작업이 실패할 경우 클라이언트 런타임은 클라이언트 응용 프로그램의 예외 수신기 객체에 대해 예외를 발생시킵니다. 응용 프로그램에 예외 수신기가 없는 경우 응용 프로그램의 다음 연결 사용 시도가 실패합니다.

메시지 수신을 대기하는 동안 메시지를 보내지 않는 사용자 클라이언트 응용 프로그램에서는 반드시 핑을 사용해야 합니다. 그러한 응용 프로그램에서 핑을 사용하지 않을 경우 연결이 실패해도 알 수 없습니다. 또한 이 기능을 사용하면 실패한 연결을 미리 처리하게 되어 메시지를 보낼 필요가 없어지므로 클라이언트가 메시지를 거의 생성하지 않아도 된다는 장점이 있습니다.

기본적으로 핑 간격은 30초로 설정됩니다. 값을 -1로 설정하면 핑 작업을 안 하게 됩니다.

끊어진 연결에 대한 응답은 운영 체제별로 다릅니다. 예를 들어, 일부 운영 체제에서는 ping마다 실패를 즉시 보고하기도 하고 다른 운영 체제에서는 ping이 성공하거나 버퍼 오버플로가 발생할 때까지 연속적으로 ping을 버퍼링하여 브로커에 대한 연결을 계속해서 시도하기도 합니다.

imqPingInterval 속성에 대한 자세한 내용은 [334페이지의 "연결 처리"](#)를 참조하십시오.

## 클라이언트 아이디

Message Queue는 영구 가입자에게 필요한 고유 클라이언트 아이디 설정과 클라이언트 인증을 지원하도록 연결 팩토리 속성 집합을 정의합니다.

브로커 연결을 시도하는 클라이언트는 인증되어야 합니다. 클라이언트가 연결할 때 아이디 또는 비밀번호를 지정하지 않을 경우 다음 중 하나가 발생합니다.

- imqDefaultUsername 및 imqDefaultPassword 연결 팩토리 속성을 설정하지 않은 경우 클라이언트 런타임은 guest/guest 값을 브로커에 전달하고 브로커는 해당 값을 사용하여 클라이언트를 인증합니다.

사용자 저장소는 guest/guest 항목과 함께 제공되므로 클라이언트가 연결됩니다.

- imqDefaultUsername 및 imqDefaultPassword 연결 팩토리 속성이 설정되어 있는 경우 클라이언트 런타임은 해당 값을 브로커에 전달하고 브로커는 해당 값을 사용하여 클라이언트를 인증합니다.

사용자/비밀번호 쌍이 사용자 저장소에 있는 경우 클라이언트가 연결됩니다.

이 인증 방법에서는 모든 사용자의 연결을 허용하므로 개발 및 테스트 작업에 편리합니다. 작업 시스템에서는 사용자 저장소에 추가된 사용자로 연결 액세스를 제한해야 합니다.

연결을 요청하는 클라이언트의 브로커 인증 이외에 JMS 사양에서는 클라이언트에 대한 상태를 유지하도록 연결 시 고유 클라이언트 식별자를 제공해야 합니다. Message Queue에서는 클라이언트 아이디를 사용하여 영구 가입자를 추적합니다. 영구 가입자가 비활성 상태인 경우 브로커는 해당 가입자에 대한 메시지를 보관하였다가 가입자가 다시 활성화 되면 메시지를 전달합니다. 브로커는 클라이언트 아이디로 가입자를 식별합니다.

클라이언트 아이디는 사용자가 관리상 설정하거나 클라이언트가 프로그래밍 방식으로 설정할 수 있습니다. 여러 클라이언트가 동일한 연결 팩토리 객체로부터 연결하는 경우 연결 팩토리에 대한 클라이언트 아이디를 설정합니다. 그러면 Message Queue가 해당 팩토리로부터 설정되는 각 연결에 대해 고유한 클라이언트 아이디를 제공할 수 있습니다.

고유한 클라이언트 아이디 값을 보장하려면 다음과 같은 형식으로 `imqConfiguredClientID` 속성을 설정합니다.

```
imqConfiguredClientID=${u}string
```

`{u}`는 속성 값의 처음 네 문자입니다. "u" 이외의 문자가 있는 경우 연결할 때 JMS 예외가 발생합니다.

`string` 값은 이 연결 팩토리에서 생성된 연결과 연관시킬 수 있는 모든 값(예: Xconn)입니다. 사용자 인증 단계에서 Message Queue는 `u`를 `u:userName`으로 대체합니다. 예를 들어, 연결에 연관된 사용자가 Athena이고 연결에 지정된 문자열이 `{u}Xconn`인 경우 클라이언트 아이디는 `u:AthenaXconn`이 됩니다.

이 방법을 사용하면 모든 다른 방법과 마찬가지로 연결 팩토리에 의해 생성되는 연결마다 고유한 클라이언트 아이디를 갖게 됩니다.

이 방법이 적용되지 않는 한 가지 경우는 두 클라이언트가 `guest`와 같은 기본 아이디를 사용하여 연결하는 경우로 각 클라이언트는 동일한 `{u}` 구성 요소가 있는 클라이언트 아이디를 갖게 됩니다. 런타임에서는 MQ가 고유하지 않은 클라이언트 아이디로 연결을 생성할 수 없으므로 연결을 먼저 요청하는 클라이언트는 연결되지만 나중에 요청하는 클라이언트는 연결되지 않습니다.

연결 팩토리를 사용하는 클라이언트가 구성된 클라이언트 아이디를 프로그래밍 방식으로 변경하지 못하도록 `imqDisableSetClientID` 속성을 설정할 수 있습니다.

응용 프로그램 코드에서 `setClientId()` 메소드를 사용하지 않는 경우 영구 가입을 위해 `imqConfiguredClientID` 속성을 설정해야 합니다.

요컨대 다음은 클라이언트 아이디에 영향을 미치는 속성입니다.

- `imqDefaultUsername`. 클라이언트가 연결할 때 아이디를 지정하지 않는 경우에 브로커를 인증하는 데 사용할 기본 아이디를 지정합니다.
- `imqDefaultPassword`. 클라이언트가 연결할 때 비밀번호를 지정하지 않는 경우에 브로커를 인증하는 데 사용할 기본 비밀번호를 지정합니다.
- `imqConfiguredClientID`. 관리상 구성된 클라이언트 아이디 값을 지정합니다.
- `imqDisableSetClientID`. 연결 팩토리를 사용하는 클라이언트가 클라이언트 아이디를 프로그래밍 방식으로 변경할 수 있는지 여부를 지정합니다.

이러한 속성에 대한 자세한 내용은 338페이지의 "클라이언트 아이디"를 참조하십시오.

## 안정성 및 흐름 제어

클라이언트가 주고 받는 메시지와 Message Queue에 사용되는 제어 메시지는 동일한 클라이언트-브로커 연결을 통해 전달됩니다. 따라서 JMS 메시지 전달에 의해 브로커 확인과 같은 제어 메시지가 일시적으로 중단되어 전달이 지연될 수 있습니다.

클라이언트 메시지의 흐름에 따라 제어 메시지의 흐름을 관리할 수 있도록 연결 팩토리 속성을 설정할 수 있습니다. 두 가지 유형의 메시지 흐름 제어는 안정성과 처리 능력의 절충과 관련됩니다. 이러한 속성을 사용하여 흐름 제어와 안정성을 관리하는 방법은 [244페이지](#)의 "클라이언트 런타임 메시지 흐름 조정"을 참조하십시오.

다음 속성은 클라이언트 및 제어 메시지의 흐름에 영향을 줍니다.

- `imqAckTimeout`. 클라이언트 런타임이 브로커 응답을 대기하는 최대 시간(밀리초)을 지정합니다.
- `imqConnectionFlowCount`. 측정된 일괄 처리에서 JMS 메시지 수를 지정합니다.
- `imqConnectionFlowLimitEnabled`. 연결 수준에서 메시지 흐름을 제한합니다.
- `imqConnectionFlowLimit`. 연결을 통해 전달될 수 있고 클라이언트 런타임에서 버퍼링될 수 있는 사용 대기 중인 메시지 수를 제한합니다.
- `imqConsumerFlowLimit`. 연결을 통해 전달될 수 있고 클라이언트 런타임에서 버퍼링될 수 있는 사용 대기 중인 메시지 수를 사용자별로 제한합니다.
- `imqConsumerFlowThreshold`. 클라이언트 런타임에 버퍼링된 사용자당 메시지의 수를 `imqConsumerFlowLimit`에 대한 백분율로 지정하며 이 값보다 낮으면 해당 사용자에 대한 메시지 전달이 다시 시작됩니다.

이러한 속성에 대한 자세한 내용은 [339페이지](#)의 "안정성 및 흐름 제어"를 참조하십시오.



## 대기열 브라우저 동작 및 서버 세션

클라이언트 대기열 찾아보기에 영향을 주는 속성은 다음과 같습니다.

- `imqQueueBrowserMaxMessagesPerRetrieve`. 대기열 대상의 내용을 찾아볼 때 클라이언트가 한 번에 검색하는 메시지의 최대 수를 지정합니다.
- `imqQueueBrowserRetrieveTimeout`. 대기열 대상의 내용을 찾아볼 때 클라이언트가 메시지 검색을 대기하는 시간을 지정합니다.
- `imqLoadMaxToServerSession`. JMS 응용 프로그램 서버 기능의 경우 `Message Queue Connection Consumer`가 `ServerSession` 세션에 메시지를 `maxMessages`까지 로드할지 아니면 한 번에 하나씩 로드할지를 지정합니다.

이러한 속성에 대한 자세한 내용은 340페이지의 "대기열 브라우저 동작 및 서버 세션"을 참조하십시오.

## JMS 정의 등록 정보 지원

연결 팩토리 속성을 사용하여 연결에서 생성하는 메시지에 대한 JMS 정의 등록 정보를 자동으로 설정할 수 있습니다. JMS 등록 정보는

<http://java.sun.com/products/jms/docs.html>에 있는 JMS 사양에 정의됩니다.

다음 속성을 사용하여 JMS 정의 등록 정보를 설정합니다.

- `imqSetJMSXUserID`. 생성된 메시지에 대해 `Message Queue`에서 JMS 정의 등록 정보 `JMSXUserID`(메시지를 보내는 사용자의 아이디)를 설정할지 여부를 지정합니다.
- `imqSetJMSXAppID`. 생성된 메시지에 대해 `Message Queue`에서 JMS 정의 등록 정보 `JMSXAppID`(메시지를 보내는 응용 프로그램의 아이디)를 설정할지 여부를 지정합니다.
- `imqSetJMSXProducerTXID`. 생성된 메시지에 대해 `Message Queue`에서 JMS 정의 등록 정보 `JMSXProducerTXID`(메시지를 생성한 트랜잭션의 트랜잭션 식별자)를 설정할지 여부를 지정합니다.
- `imqSetJMSXConsumerTXID`. 사용된 메시지에 대해 `Message Queue`에서 JMS 정의 등록 정보 `JMSXConsumerTXID`(메시지를 사용한 트랜잭션의 트랜잭션 식별자)를 설정할지 여부를 지정합니다.
- `imqSetJMSXRcvTimestamp`. 사용된 메시지에 대해 `Message Queue`에서 JMS 정의 등록 정보 `JMSXRcvTimestamp`(메시지가 사용자에게 전달되는 시간)를 설정할지 여부를 지정합니다.

이러한 속성에 대한 자세한 내용은 341페이지의 "JMS 정의 등록 정보 지원"을 참조하십시오.

## 메시지 헤더 무시

연결 팩토리 속성을 설정하여 메시지의 지속성, 수명 및 우선 순위를 지정하는 JMS 메시지 헤더 필드를 무시할 수 있습니다. 이러한 설정은 연결 팩토리에서 가져온 연결에서 생성된 모든 메시지에 사용됩니다.

다음 JMS 필드의 값을 무시할 수 있습니다.

- JMSDeliveryMode(메시지 지속성/비지속성)
- JMSExpiration(메시지 수명)
- JMSPriority(메시지 우선 순위 - 0부터 9까지의 정수)

이러한 필드에 대한 자세한 내용은 <http://java.sun.com/products/jms/docs.html>의 JMS 사양을 참조하십시오.

메시지 헤더를 무시하면 응용 프로그램 요구 사항에 위배될 수 있으므로 이 기능을 사용할 경우 응용 프로그램 사용자 또는 설계자와 반드시 상의하십시오.

다음 목록에는 메시지 무시를 처리하는 연결 팩토리 속성이 포함되어 있습니다. 이러한 속성은 대부분 쌍으로 구성됩니다. 각 쌍에서 첫 번째 속성은 지정된 헤더 필드를 무시할 수 있는지 여부를 지정하고 두 번째 속성은 무시하는 값을 지정합니다.

- `imqOverrideJMSDeliveryMode` 및 `imqJMSDeliveryMode`. 첫 번째 속성은 클라이언트에서 설정한 `JMSDeliveryMode` 필드를 무시할 수 있는지 여부를 지정하고 두 번째 속성은 무시하는 값을 지정합니다.
- `imqOverrideJMSExpiration` 및 `imqJMSExpiration`. 첫 번째 속성은 클라이언트에서 설정한 `JMSExpiration` 필드를 무시할 수 있는지 여부를 지정하고 두 번째 속성은 무시하는 값을 지정합니다.
- `imqOverrideJMSPriority` 및 `imqJMSPriority`. 첫 번째 속성은 클라이언트에서 설정한 `JMSPriority` 필드를 무시할 수 있는지 여부를 지정하고 두 번째 속성은 무시하는 값을 지정합니다.
- `imqOverrideJMSHeadersToTemporaryDestinations`. 무시가 임시 대상에 적용되는지 여부를 지정합니다.

이러한 속성에 대한 자세한 내용은 338페이지의 "메시지 헤더 무시"를 참조하십시오.

## 대상 관리 대상 객체 속성

물리적 주제 또는 대기열 대상을 나타내는 대상 관리 대상 객체는 [333페이지의 표 16-1](#)에 나열된 것과 같은 속성을 갖습니다. [190페이지의 "주제 또는 대기열 추가"](#) 절에서는 객체 저장소에 대상 관리 대상 객체를 추가할 때 이런 속성을 지정하는 방법을 설명합니다.

여기에서 주로 고려해야 할 속성은 `imqDestinationName`입니다. 이 속성은 주제 또는 대기열 관리 대상 객체에 해당하는 물리적 대상에 지정하는 이름입니다. 대상에 설명을 입력해 두면 여러 응용 프로그램을 지원하기 위해 만드는 다른 대상과 구분하기가 쉽습니다.

자세한 내용은 JavaDoc API 설명서에서 `Message Queue` 클래스 `com.sun.messaging.DestinationConfiguration` 부분을 참조하십시오.

## 객체 관리자 유틸리티(imqobjmgr) 사용

객체 관리자 유틸리티를 사용하면 `Message Queue` 관리 대상 객체를 만들고 관리할 수 있습니다. 이 유틸리티를 사용하여 다음 작업을 수행할 수 있습니다.

- 객체 저장소에서 관리 대상 객체를 추가 또는 삭제합니다.
- 기존 관리 대상 객체를 나열합니다.
- 관리 대상 객체 관련 정보를 쿼리 및 표시합니다.
- 객체 저장소에 있는 기존 관리 대상 객체를 수정합니다.

`imqobjmgr` 명령의 구문, 하위 명령 및 옵션에 대한 자세한 내용은 [13장, "명령 참조"](#)를 참조하십시오. 다음 절에서는 `imqobjmgr` 하위 명령으로 작업할 때 제공해야 하는 정보에 대해 설명합니다.

### 필요한 정보

관리 대상 객체와 관련된 대부분의 작업을 수행할 때 `imqobjmgr` 하위 명령의 옵션으로 다음 정보를 지정해야 합니다.

- **관리 대상 객체 유형**  
허용되는 유형은 [297페이지의 표 13-11](#)에 나와 있습니다.
- **관리 대상 객체의 JNDI 조회 이름:**  
클라이언트 코드에서 객체 저장소에 있는 관리 대상 객체를 참조할 때(JNDI 사용) 사용되는 논리적 이름입니다.

- **관리 대상 객체 속성**(특히 add 및 update 하위 명령에서 필요):
  - 대상의 경우: 브로커에 있는 물리적 대상의 이름. imqcmd create dst 하위 명령의 -n 옵션으로 지정했던 이름입니다. 이름을 지정하지 않으면 기본 이름인 Untitled\_Destination\_Object가 사용됩니다.
  - 연결 팩토리의 경우: 가장 많이 사용하는 속성은 클라이언트가 연결을 시도할 하나 이상의 메시지 서버 주소를 지정하는 주소 목록(imqAddressList). 이 정보를 지정하지 않으면 로컬 호스트와 기본 포트 번호(7676)를 사용합니다. 즉 클라이언트는 로컬 호스트의 포트 7676에 있는 브로커에 연결을 시도합니다. [189페이지의 "연결 팩토리 추가"](#) 절에서는 객체 속성을 지정하는 방법에 대해 설명합니다.  
추가 속성에 대한 내용은 [177페이지의 "연결 팩토리 속성"](#)을 참조하십시오.

- **객체 저장소 속성**

이 정보는 파일 시스템 저장소 또는 LDAP 서버 중 어느 것을 사용하는지에 따라 달라지지만 다음 속성을 포함해야 합니다.

  - JNDI 구현 유형(initial context 속성). 예를 들어, 파일 시스템 또는 LDAP입니다.
  - 객체 저장소의 관리 대상 객체 위치(공급자 URL 속성), 즉 "폴더"
  - 객체 저장소 액세스에 필요한 사용자 아이디, 비밀번호, 권한 부여 유형(있는 경우)

객체 저장소 속성에 대한 자세한 내용은 [174페이지의 "LDAP 서버 객체 저장소"](#) 및 [175페이지의 "파일 시스템 객체 저장소"](#)를 참조하십시오.

## 명령 파일 사용

imqobjmgr 명령을 사용하면 imqobjmgr 하위 명령 절의 일부 또는 전부에 Java 등록 정보 파일 구문을 사용하는 명령 파일의 이름을 지정할 수 있습니다.

객체 관리자 유틸리티(imqobjmgr)에 명령 파일을 사용하면 여러 imqobjmgr 실행에서 동일하게 사용되면서 입력할 내용이 많은 객체 저장소 속성을 지정할 때 유용합니다. 명령 파일을 사용하면 명령줄에 허용된 최대 문자 수를 초과하는 상황을 방지할 수도 있습니다.

imqobjmgr 명령 파일의 일반 구문은 다음과 같습니다(버전 등록 정보는 Message Queue 제품이 아니라 명령 파일의 버전을 나타내며(명령줄 옵션이 아님), 그 값을 2.0으로 설정해야 함).

```
version=2.0
cmdtype=[ add | delete | list | query | update ]
obj.type=[ q | t | qf | tf | cf | xqf | xtf | xcf | e ]
obj.lookupName=lookup name
obj.attrs.objAttrName1=value1
obj.attrs.objAttrName2=value2
obj.attrs.objAttrNameN=valueN
...
objstore.attrs.objStoreAttrName1=value1
objstore.attrs.objStoreAttrName2=value2
objstore.attrs.objStoreAttrNameN=valueN
...
```

아래에서는 imqobjmgr 명령을 예로 들어 명령 파일의 사용 방법을 설명합니다.

```
imqobjmgr add
-t qf
-l "cn=myQCF"
-o "imqAddressList=mq://foo:777/jms"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

이 명령은 다음과 같은 내용을 가진 MyCmdFile과 같은 파일에 캡슐화할 수 있습니다.

```
version=2.0
cmdtype=add
obj.type=qf
obj.lookupName=cn=myQCF
obj.attrs.imqAddressList=mq://foo:777/jms
objstore.attrs.java.naming.factory.initial=\
    com.sun.jndi.ldap.LdapCtxFactory
objstore.attrs.java.naming.provider.url=\
    ldap://mydomain.com:389/o=imq
objstore.attrs.java.naming.security.principal=\
    uid=fooUser, ou=People, o=imq
objstore.attrs.java.naming.security.credentials=fooPasswd
objstore.attrs.java.naming.security.authentication=simple
```

-i 옵션을 사용하면 이 파일을 객체 관리자 유틸리티(imqobjmgr)로 전달할 수 있습니다.

```
imqobjmgr -i MyCmdFile
```

명령 파일에 몇 가지 옵션을 지정하고 명령줄을 사용해서 또 다른 옵션을 지정할 수도 있습니다. 이 경우 하위 명령 줄 중 유틸리티가 실행될 때마다 동일하게 유지되는 부분을 지정하는 데 명령 파일을 사용할 수 있습니다. 예를 들어, 다음 명령은 관리 대상 객체의 저장 위치를 제외하고 연결 팩토리 관리 대상 객체를 추가할 때 필요한 모든 옵션을 지정합니다.

```
imqobjmgr add
-t qf
-l "cn=myQCF"
-o "imqAddressList=mq://foo:777/jms"
-i MyCmdFile
```

이 경우 MyCmdFile 파일에는 다음과 같은 정의가 포함됩니다.

```
version=2.0
objstore.attrs.java.naming.factory.initial=\
    com.sun.jndi.ldap.LdapCtxFactory
objstore.attrs.java.naming.provider.url=\
    ldap://mydomain.com:389/o=imq
objstore.attrs.java.naming.security.principal=\
```

```
uid=fooUser, ou=People, o=imq
objstore.attrs.java.naming.security.credentials=fooPasswd
objstore.attrs.java.naming.security.authentication=simple
```

다음 위치에서 명령 파일의 예를 추가로 확인할 수 있습니다.

```
/usr/demo/imq/imqobjmgr(Solaris)
/opt/sun/mq/examples/imqobjmgr(Linux)
IMQ_HOME/demo/imqobjmgr(Windows)
```

## 관리 대상 객체 추가 및 삭제

이 절에서는 연결 팩토리 및 주제 또는 대기열 대상의 관리 대상 객체를 객체 저장소에 추가하는 방법을 설명합니다.

---

**주** 객체 관리자 유틸리티(imqobjmgr)는 Message Queue 관리 대상 객체만을 나열하고 표시합니다. 객체 저장소에 추가할 관리 대상 객체와 동일한 조회 이름을 가진 비 Message Queue 객체를 포함해야 하는 경우에 이 객체를 추가하려고 하면 오류 메시지가 표시됩니다.

---

## 연결 팩토리 추가

클라이언트 응용 프로그램에서 브로커에 대한 연결을 설정할 수 있게 하려면 클라이언트 응용 프로그램에서 원하는 연결 유형(주제 연결 팩토리 또는 대기열 연결 팩토리)을 나타내는 관리 대상 객체를 추가합니다.

대기열 연결 팩토리를 추가하려면 다음과 같은 명령을 사용합니다.

```
imqobjmgr add
-t qf
-l "cn=myQCF"
-o "imqAddressList=mq://myHost:7272/jms"
-j "java.naming.factoryinitial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

이전 명령은 조회 이름이 cn=myQCF이고 myHost에서 실행 중인 브로커에 연결하여 포트 7272를 수신하는 관리 대상 객체를 만듭니다. 관리 대상 객체는 LDAP 서버에 저장됩니다. imqobjmgr 명령의 인수로 명령 파일을 지정해도 같은 결과를 얻을 수 있습니다. 자세한 내용은 [186페이지의 "명령 파일 사용"](#)을 참조하십시오.

---

**주**      **이름 지정 규약:** LDAP 서버를 사용하여 관리 대상 객체를 저장할 경우 위에에서처럼 접두어 "cn="에 있는 조회 이름을 지정해야 합니다 (cn=myQCF). -l 옵션을 사용하여 조회 이름을 지정합니다. 파일 시스템 객체 저장소를 사용하는 경우 cn 접두어를 사용할 필요가 없습니다. 그러나 "/"가 포함된 조회 이름은 사용하지 마십시오. [표 8-3](#)을 참조하십시오.

---

**표 8-3**      이름 지정 규약 예

객체 저장소 유형	올바른 이름	잘못된 이름
LDAP 서버	cn=myQCF	myQCF
파일 시스템	myTopic	myObjects/myTopic

## 주제 또는 대기열 추가

클라이언트 응용 프로그램에서 브로커의 물리적 대상에 액세스할 수 있게 하려면 이 대상에 해당하는 관리 대상 객체를 객체 저장소에 추가합니다.

객체 저장소에 해당 관리 대상 객체를 추가하기 전에 물리적 대상을 먼저 만드는 것이 좋습니다. 명령 유틸리티(imqcmd)를 사용하여 브로커에 객체 저장소의 대상 관리 대상 객체에 해당하는 물리적 위치를 만듭니다. 물리적 대상을 만드는 자세한 내용은 [121페이지의 "연결 정보 얻기"](#)를 참조하십시오.

다음 명령은 조회 이름이 myTopic이고 물리적 대상 이름이 TestTopic인 주제 대상에 해당하는 관리 대상 객체를 추가합니다. 관리 대상 객체는 LDAP 서버에 저장됩니다.



```

imgobjmgr add
-t t
-l "cn=myTopic"
-o "imgDestinationName=TestTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"

```

같은 명령이지만 관리 대상 객체만 Solaris 파일 시스템에 저장됩니다.

```

imgobjmgr add
-t t
-l "cn=myTopic"
-o "imgDestinationName=TestTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.fscontext.RefFSContextFactory"
-j "java.naming.provider.url=
    file:///home/foo/img_admin_objects"

```

예를 들어, LDAP 서버 사례에서 명령 파일 MyCmdFile을 사용하여 하위 명령 절을 지정할 수 있습니다. 파일에는 다음과 같은 텍스트가 포함됩니다.

```

version=2.0
cmdtype=add
obj.type=t
obj.lookupName=cn=myTopic
obj.attrs.imgDestinationName=TestTopic
objstore.attrs.java.naming.factory.initial=
    com.sun.jndi.fscontext.RefFSContextFactory
objstore.attrs.java.naming.provider.url=
    file:///home/foo/img_admin_objects
objstore.attrs.java.naming.security.principal=
    uid=fooUser, ou=People, o=img
objstore.attrs.java.naming.security.credentials=fooPasswd
objstore.attrs.java.naming.security.authentication=simple

```

-i 옵션을 사용해서 해당 파일을 imqobjmgr 명령에 전달합니다.

```
imqobjmgr -i MyCmdFile
```

---

**주** LDAP 서버를 사용하여 관리 대상 객체를 저장할 경우 위 예에서처럼 접두어 "cn="이 있는 조회 이름을 지정해야 합니다. -l 옵션을 사용하여 조회 이름을 지정합니다. 파일 시스템 객체 저장소를 사용하는 경우 이 접두어를 사용하지 않아도 됩니다.

---

-t 옵션에 q를 지정한다는 점만 제외하면 대기열 객체를 추가하는 것도 이와 같습니다.

## 관리 대상 객체 삭제

관리 대상 객체를 삭제하려면 delete 하위 명령을 사용합니다. 객체의 조회 이름과 유형, 위치를 지정해야 합니다.

다음 명령은 조회 이름이 cn=myTopic이고 LDAP 서버에 저장되는 주제의 관리 대상 객체를 삭제합니다.

```
imqobjmgr delete
-t t
-l "cn=myTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

## 관리 대상 객체 나열

모든 관리 대상 객체의 목록을 보거나 특정 유형에 해당하는 모든 관리 대상 객체의 목록을 보려면 `list` 하위 명령을 사용합니다. 다음 샘플 코드에서는 관리 대상 객체가 LDAP 서버에 저장되어 있다고 가정합니다.

다음 명령은 모든 객체를 나열합니다.

```
imgobjmgr list
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

다음 명령은 `queue` 유형의 모든 객체를 나열합니다.

```
imgobjmgr list
-t q
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

## 단일 객체 관련 정보 얻기

특정 관리 대상 객체에 대한 정보를 얻으려면 `query` 하위 명령을 사용합니다. 객체의 조회 이름과 관리 대상 객체를 포함하는 객체 저장소의 속성(초기 컨텍스트, 위치 등)을 지정해야 합니다.

다음 예에서는 `query` 하위 명령을 사용하여 조회 이름이 `cn=myTopic`인 객체에 관련된 정보를 표시합니다.

```

imgobjmgr query
-l "cn=myTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"

```

## 관리 대상 객체 업데이트

관리 대상 객체의 속성을 수정하려면 update 명령을 사용합니다. 조회 이름과 객체 위치를 지정해야 합니다. 속성 값을 수정하려면 -o 옵션을 사용합니다.

이 명령은 주제 연결 팩토리를 나타내는 관리 대상 객체의 속성을 변경합니다.

```

imgobjmgr update
-t tf
-l "cn=MyTCF"
-o imgReconnectAttempts=3
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"

```

## 브로커 클러스터 작업

Message Queue Enterprise Edition은 메시지 전달 서비스를 클라이언트에 제공하기 위해 함께 작업하는 브로커 그룹인 *브로커 클러스터*의 사용을 지원합니다. 클러스터를 사용하여 메시지 서버는 다중 브로커 간에 클라이언트 연결을 분배하는 방식으로 메시지 트래픽의 볼륨에 대한 작업 크기를 조절할 수 있습니다. 클러스터 및 클러스터 작업 방법에 대한 자세한 내용은 *Message Queue 기술 개요*를 참조하십시오.

이 장에서는 브로커 클러스터를 관리하고 브로커를 브로커 클러스터에 연결하는 방법 및 브로커 클러스터 구성 방법을 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- [196페이지의 "클러스터 구성 등록 정보"](#)
- [198페이지의 "클러스터 관리"](#)
- [201페이지의 "마스터 브로커"](#)

## 클러스터 구성 등록 정보

각 클러스터 구성원 브로커에 대한 *클러스터 구성 등록 정보*를 지정하여 클러스터를 정의합니다. 이러한 등록 정보는 클러스터의 각 브로커에 대해 개별적으로 설정할 수도 있지만 모든 브로커가 참조하는 중앙 *클러스터 구성 파일*에 모으는 것이 일반적으로 더 편리합니다. 이렇게 하면 설정이 달라지지 않으므로 클러스터의 모든 브로커가 동일하고 일관된 구성 정보를 공유하게 됩니다.

클러스터 구성 등록 정보는 [326페이지의 표 14-11](#)에서 자세히 설명합니다. 여기에는 다음 항목이 포함됩니다.

- `imq.cluster.brokerlist` 클러스터에 속하는 모든 브로커의 호스트 이름과 포트 번호를 지정합니다.
- `imq.cluster.masterbroker` 상태 변경을 추적하는 마스터 브로커에 해당하는 브로커(있는 경우)를 지정합니다.
- `imq.cluster.url` 클러스터 구성 파일(있는 경우)의 위치를 지정합니다.
- `imq.cluster.hostname` 클러스터 내의 브로커 간 내부 통신에 사용되는 클러스터 연결 서비스의 호스트 이름 또는 IP 주소를 지정합니다. 이 설정은 컴퓨터에 여러 개의 네트워크 인터페이스 카드가 있을 때처럼 사용 가능한 호스트가 두 개 이상인 경우에 유용합니다.
- `imq.cluster.port` 클러스터 연결 서비스에 포트 번호를 제공합니다.
- `imq.cluster.transport` 클러스터 연결 서비스에 사용되는 전송 프로토콜(예: tcp 또는 ssl)을 지정합니다.

`hostname` 및 `port` 등록 정보는 각 개별 브로커에 대해 독립적으로 설정할 수 있지만 `brokerlist`, `masterbroker`, `url` 및 `transport`는 클러스터 내의 모든 브로커에 대해 동일한 값을 가져야 합니다.

다음 절에서는 클러스터 구성 파일을 사용하여 브로커의 클러스터 구성 등록 정보를 설정하는 방법(클러스터의 각 브로커에 대해 개별 설정 또는 중앙 설정)에 대해 설명합니다.

## 개별 브로커에 대한 클러스터 등록 정보 설정

인스턴스 구성 파일(또는 브로커를 시작할 때 명령줄에서) 브로커 클러스터 구성 등록 정보를 설정할 수 있습니다. 예를 들어, host1의 9876 포트, host2의 5000 포트 및 ctrlhost의 기본 포트(7676)에서 여러 브로커로 구성된 클러스터를 만들려면 세 브로커 모두의 인스턴스 구성 파일에 다음 등록 정보를 포함시켜야 합니다.

```
imq.cluster.brokerlist=host1:9876,host2:5000,ctrlhost
```

이 방법을 사용하면 클러스터 구성을 변경해야 하는 경우 클러스터 내 모든 브로커의 인스턴스 구성 파일을 업데이트해야 합니다.

## 클러스터 구성 파일 사용

일관성과 간편한 유지 관리를 위해서는 클러스터 구성 등록 정보를 브로커별로 개별 설정하는 대신 모든 공유 클러스터 구성 등록 정보를 단일 클러스터 구성 파일에 모으는 것이 좋습니다. 이 방법에서 각 브로커의 인스턴스 구성 파일은 클러스터 구성 파일의 위치를 가리키도록 `imq.cluster.url` 등록 정보를 설정해야 합니다. 예를 들면, 다음과 같습니다.

```
imq.cluster.url=file:/home/cluster.properties
```

그런 다음 클러스터 구성 파일은 클러스터 내의 모든 브로커에 대해 연결할 브로커 목록(`imq.cluster.brokerlist`), 클러스터 연결 서비스에 사용할 전송 프로토콜(`imq.cluster.transport`) 및 마스터 브로커의 주소(`imq.cluster.masterbroker`, 선택 사항)와 같은 공유 구성 등록 정보를 정의합니다. 다음 코드는 마스터 브로커 역할을 하는 `ctrlhost`에서 실행되는 브로커와 함께 이전 예에서와 같은 클러스터를 정의합니다.

```
imq.cluster.brokerlist=host1:9876,host2:5000,ctrlhost
imq.cluster.masterbroker=ctrlhost
```

# 클러스터 관리

이 절에서는 브로커 집합을 연결하여 클러스터를 형성하고 기존 클러스터에 새 브로커를 추가하고 클러스터에서 브로커를 제거하는 방법에 대해 설명합니다.

## 브로커 연결

명령줄에서 `-cluster` 옵션을 사용하거나 클러스터 구성 파일에서 `imq.cluster.brokerlist` 등록 정보를 설정하는 두 가지 방법으로 브로커를 클러스터에 연결합니다. 사용하는 방법에 관계없이 시작한 각 브로커는 5분마다 다른 브로커에 연결하려고 시도합니다. 이 때 마스터 브로커(구성되어 있는 경우)가 시작되면 연결에 성공한 것입니다. 클러스터의 브로커가 마스터 브로커보다 먼저 시작된 경우 일시 중지된 상태를 유지하여 클라이언트 연결을 거부합니다. 일시 중지된 브로커는 마스터 브로커가 시작되면 자동으로 다시 작동합니다.

각 브로커를 시작할 때 클러스터 구성 파일을 사용하는 대신 `imqbrokerd` 명령에서 `-cluster` 옵션을 사용하여 클러스터 내의 전체 브로커 목록을 지정할 수 있습니다. 예를 들어, 다음 명령은 새 브로커를 시작하여 `host1`의 기본 포트(7676), `host2`의 5000 포트 및 기본 호스트(`localhost`)의 9876 포트에서 실행하는 브로커에 연결합니다.

```
imqbrokerd -cluster host1,host2:5000,:9876
```

대체 방법으로 `imq.cluster.brokerlist` 등록 정보를 사용하여 연결할 브로커 목록을 지정하는 클러스터 구성 파일을 만들 수 있습니다. 이 방법은 작업 시스템에 더 적합한 방법입니다. 그런 다음 클러스터의 각 브로커에서 이 클러스터 구성 파일을 가리키도록 자체 `imq.cluster.url` 등록 정보를 설정해야 합니다.

### Linux 전제 조건: IP 주소 설정

Linux 시스템에서는 브로커를 클러스터에 연결하기 위한 특별한 전제 조건이 있습니다. 일부 Linux 설치 프로그램은 `localhost` 항목을 네트워크 역순환 IP 주소(127.0.0.1)로 자동으로 설정합니다. 사용자는 클러스터의 모든 브로커 주소가 제대로 지정될 수 있도록 시스템의 IP 주소를 설정해야 합니다.

클러스터에 참여하는 모든 Linux 시스템의 경우 클러스터 설정 과정의 일부로 `/etc/hosts` 파일을 확인합니다. 시스템에서 정적 IP 주소를 사용하는 경우 `/etc/hosts` 파일을 편집하여 올바른 `localhost` 주소를 지정합니다. 주소가 DNS(Domain Name Service)에 등록되면 `/etc/nsswitch.conf` 파일을 편집하여 시스템이 로컬 `hosts` 파일을 참조하기 전에 DNS 조회를 수행하도록 항목의 순서를 변경합니다. `/etc/nsswitch.conf` 파일의 행은 다음과 같아야 합니다.

```
hosts: dns files
```



## 브로커 간 보안 연결

클러스터 내의 브로커 간에 암호화된 보안 메시지를 전달하려면 SSL 기반 전송 프로토콜을 사용하도록 클러스터 연결 서비스를 구성합니다. 159페이지의 "SSL 기반 서비스 작업"에 설명된 것처럼 클러스터 내의 각 브로커에 대해 SSL 기반 연결 서비스를 설정합니다. 그런 다음 클러스터 구성 파일에서 또는 브로커별로 개별적으로 각 브로커의 `imq.cluster.transport` 등록 정보를 `ssl`로 설정합니다.

## 클러스터에 브로커 추가

클러스터에 새 브로커를 추가하는 절차는 클러스터가 클러스터 구성 파일을 사용하는지 여부에 따라 달라집니다.

### ▶ 클러스터 구성 파일을 사용하여 클러스터에 새 브로커를 추가하는 방법

1. 클러스터 구성 파일의 `imq.cluster.brokerlist` 등록 정보에 새 브로커를 추가합니다.

2. 클러스터의 모든 브로커에 대해 다음 명령을 실행합니다.

```
imqcmd reload cls
```

이 명령은 각 브로커가 클러스터 구성을 다시 로드하게 하여 클러스터에 있는 브로커의 모든 영구 정보를 최신 상태로 유지합니다.

3. (선택 사항) 클러스터 구성 파일을 가리키도록 브로커의 `config.properties` 파일에서 `imq.cluster.url` 등록 정보 값을 설정합니다.

4. 새 브로커를 시작합니다.

단계 3을 수행하지 않은 경우 `imqbrokerd` 명령줄에서 `-D` 옵션을 사용하여 `imq.cluster.url` 값을 설정합니다.

### ▶ 클러스터 구성 파일을 사용하지 않고 클러스터에 새 브로커를 추가하는 방법

`config.properties` 파일을 편집하거나 `imqbrokerd` 명령줄에서 `-D` 옵션을 사용하여 다음 등록 정보 값을 설정합니다.

- `imq.cluster.brokerlist`
- `imq.cluster.masterbroker`(필요한 경우)
- `imq.cluster.transport`(보안 클러스터 연결 서비스를 사용하는 경우)

## 클러스터에서 브로커 제거

클러스터에서 브로커를 제거하는 데 사용하는 방법은 처음에 클러스터를 만들 때 명령줄을 사용했는지 중앙 클러스터 구성 파일을 사용했는지에 따라 달라집니다.

### 명령줄을 사용하여 브로커 제거

명령줄에서 `imgbrokerd` 명령을 사용하여 클러스터에 브로커를 연결한 경우 명령줄에서 새 클러스터 구성원 집합을 지정하여 각 브로커를 중지한 다음 다시 시작해야 합니다. 절차는 다음과 같습니다.

#### ▶ 명령줄을 사용하여 클러스터에서 브로커를 제거하는 방법

1. `imgcmd` 명령을 사용하여 클러스터에서 각 브로커를 중지합니다.
2. `imgbrokerd` 명령의 `-cluster` 옵션을 사용하여 남아 있는 브로커만 지정하여 클러스터에서 해당 브로커를 다시 시작합니다.

예를 들어, 처음에 다음 명령을 사용하여 *A*, *B*, *C* 브로커 각각을 시작하여 세 개의 브로커로 구성된 클러스터를 만들었다고 가정합니다.

```
imgbrokerd -cluster A,B,C
```

클러스터에서 *A* 브로커를 제거하려면 다음 명령을 사용하여 *B*와 *C* 브로커를 다시 시작합니다.

```
imgbrokerd -cluster B,C
```

### 클러스터 구성 파일을 사용하여 브로커 제거

처음에 중앙 클러스터 구성 파일의 `img.cluster.brokerlist` 등록 정보를 사용하여 구성원 브로커를 지정하여 클러스터를 만든 경우 브로커 중 하나를 제거하기 위해 브로커를 중지할 필요가 없습니다. 대신 구성 파일을 편집하여 제거할 브로커를 제외시키고 나머지 클러스터 구성원이 클러스터 구성을 다시 로드하게 한 다음 제외된 브로커가 해당 클러스터 구성 파일을 더 이상 가리키지 않도록 다시 구성하기만 하면 됩니다. 절차는 다음과 같습니다.

#### ▶ 클러스터 구성 파일을 사용하여 클러스터에서 브로커를 제거하는 방법

1. 클러스터 구성 파일을 편집하여 `img.cluster.brokerlist` 등록 정보에 지정된 목록에서 제외된 브로커를 제거합니다.
2. 클러스터에 남아 있는 각 브로커에 대해 다음 명령을 실행합니다.

```
imgcmd reload cls
```

이렇게 하면 브로커가 클러스터 구성을 다시 로드합니다.

3. 클러스터에서 제거할 브로커를 중지합니다.
4. 해당 브로커의 `config.properties` 파일을 편집하여 `imq.cluster.url` 등록 정보에 대한 값을 제거하거나 다른 값을 지정합니다.

## 마스터 브로커

각 클러스터는 클러스터의 지속성 상태에 대한 모든 변경을 추적하기 위해 *구성 변경 기록*을 관리하는 *마스터 브로커* 하나를 선택적으로 가질 수 있습니다. 마스터 브로커는 클러스터 구성 파일이나 개별 브로커의 인스턴스 구성 파일에 있는 `imq.cluster.masterbroker` 구성 등록 정보에서 식별됩니다.

구성 변경 기록에는 영구 가입, 관리자가 만든 물리적 대상 등과 같이 클러스터에 연결된 지속성 항목에 대한 변경 관련 정보가 포함되어 있습니다. 클러스터의 모든 브로커는 시작할 때 마스터 브로커를 참조하여 이러한 지속성 항목에 대한 정보를 업데이트합니다. 마스터 브로커에 오류가 발생하면 이러한 동기화가 불가능해집니다. 자세한 내용은 [202 페이지의 "마스터 브로커를 사용할 수 없는 경우"](#)를 참조하십시오.

## 구성 변경 기록 관리

구성 변경 기록에는 중요한 정보가 포함되어 있으므로 오류가 발생할 경우 복원할 수 있도록 정기적으로 백업해야 합니다. 백업으로부터 복원하더라도 백업을 수행한 이후에 발생한 클러스터 지속성 상태의 변경 사항은 손실되지만 백업을 자주 수행하면 이러한 잠재적 정보 손실을 최소화할 수 있습니다. 백업 및 복원 작업을 수행하면 시간이 지나면 상당히 커질 수 있는, 구성 변경 레코드에 포함된 변경 기록을 압축하고 최적화할 수 있는 이점도 있습니다.

### ▶ 구성 변경 기록을 백업하는 방법

`imqbrokerd` 명령의 `-backup` 옵션을 사용하여 백업 파일의 이름을 지정합니다. 예를 들면 다음과 같습니다.

```
imqbrokerd -backup mybackuplog
```

▶ 구성 변경 기록 복원 방법

1. 클러스터에 있는 모든 브로커를 종료합니다.
2. 다음 명령을 사용하여 백업 파일로부터 마스터 브로커의 구성 변경 기록을 복원합니다.

```
imgbrokerd -restore mybackuplog
```

3. 마스터 브로커에 새 이름이나 포트 번호를 할당하는 경우 클러스터 구성 파일에서 `img.cluster.brokerlist` 및 `img.cluster.masterbroker` 등록 정보를 각각 업데이트합니다.
4. 클러스터에 있는 모든 브로커를 다시 시작합니다.

## 마스터 브로커를 사용할 수 없는 경우

클러스터의 모든 브로커는 마스터 브로커가 있어야 지속성 작업을 수행할 수 있기 때문에 마스터 브로커를 사용할 수 없을 때 클러스터의 브로커에 대해 다음 `imgcmd` 하위 명령을 수행하면 오류가 반환됩니다.

- `create dst`
- `destroy dst`
- `update dst`
- `destroy dur`

자동 생성된 물리적 대상과 임시 대상에는 영향을 주지 않습니다.

마스터 브로커가 없는 경우 클라이언트 응용 프로그램이 영구 가입자를 만들거나 영구 가입을 취소하려고 시도하면 오류가 발생합니다. 하지만 클라이언트는 기존 영구 가입을 지정하여 상호 작용할 수 있습니다.

# 메시지 서버 모니터링

이 장에서는 메시지 서버를 모니터링하는 데 사용 가능한 도구와 메트릭 데이터를 가져오는 방법에 대해 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 203페이지의 "모니터링 도구 소개"
- 205페이지의 "브로커 로깅 구성 및 사용"
- 210페이지의 "대화식 메트릭 표시"
- 215페이지의 "브로커를 모니터링하는 응용 프로그램 작성"

특정 메트릭에 대한 자세한 내용은 18장, "메트릭 참조"를 참조하십시오.

## 모니터링 도구 소개

Message Queue 정보에 대해서는 메트릭을 얻을 수 있는 로그 파일, 대화식 명령, 클라이언트 API 등 세 가지 모니터링 인터페이스가 있습니다. 각 인터페이스는 다음과 같은 장점과 단점이 있습니다.

- 로그 파일은 장기간의 메트릭 데이터 기록을 제공하지만 구문 분석하기가 어렵습니다.
- 명령을 사용하면 요구에 맞는 정보를 빠르게 샘플링할 수 있지만 기록 정보를 살펴거나 데이터를 프로그래밍 방식으로 조작할 수 없습니다.
- 클라이언트 API를 사용하면 정보 추출 및 처리, 데이터 조작, 그래프 표현, 경고 보내기 등을 할 수 있습니다. 하지만 클라이언트 API를 사용하려면 데이터를 캡처하고 분석하기 위한 사용자 정의 응용 프로그램을 작성해야 합니다.

표 10-1에서는 다른 도구들을 비교합니다.

**표 10-1**      메트릭 모니터링 도구의 장점 및 제한

메트릭 모니터링 도구	장점	제한
imqcmd metrics	원격 모니터링 스팟 체크에 적합 명령 옵션에 설정된 보고 간격을 실행 중에 변경 가능 원하는 특정 데이터를 선택하기 쉬움 보기 쉬운 테이블 형식으로 데이터 제시	하나의 명령으로 모든 데이터를 얻을 수 없음 데이터를 프로그래밍 방식으로 분석하기 어려움 기록 레코드를 작성하지 않음 기록 추세를 보기 어려움
로그 파일	정기적인 샘플링 기록 레코드 작성	브로커 등록 정보를 구성해야 하며 적용하려면 브로커를 종료하고 다시 시작해야 함 로컬 모니터링 전용 데이터 형식이 읽거나 구문 분석하기가 아주 어려우며 구문 분석 도구 없음 보고 간격을 실행 중에 변경할 수 없으며 모든 메트릭 데이터도 마찬가지임 데이터 선택에 융통성이 없음 브로커 메트릭 전용이며 대상 및 연결 서비스 메트릭은 포함되어 있지 않음 간격을 너무 짧게 설정하면 성능에 악영향을 줄 수 있음
클라이언트 API	원격 모니터링 원하는 특정 데이터를 선택하기 쉬움 데이터를 프로그램 방식으로 분석하고 모든 형식으로 표시할 수 있음	브로커 등록 정보를 구성해야 하며 적용하려면 브로커를 종료하고 다시 시작해야 함 사용자 고유의 메트릭 모니터링 클라이언트를 작성해야 함 보고 간격을 실행 중에 변경할 수 없으며 모든 메트릭 데이터도 마찬가지임

표에 나와 있는 차이점 이외에 각 도구는 브로커가 생성한 메트릭 정보 중 약간씩 다른 하위 집합을 수집합니다. 각 모니터링 도구가 수집하는 메트릭 데이터에 대한 자세한 내용은 [349페이지의 18장, "메트릭 참조"](#)를 참조하십시오.

# 브로커 로깅 구성 및 사용

Message Queue 로거는 브로커 코드, 디버거, 메트릭 생성기에서 생성한 정보를 가져와서 이 정보를 표준 출력(콘솔), 로그 파일, syslog 데몬 프로세스(Solaris™ 운영 체제인 경우) 등과 같은 여러 출력 채널에 기록합니다.

로거에서 수집된 정보의 유형과 각 출력 채널에 기록된 유형을 지정할 수 있습니다. 특히 메트릭 정보가 로그 파일에 기록되도록 지정할 수 있습니다.

이 절에서는 브로커의 기본 로깅 구성에 대해 설명하며, 로그 정보를 대체 출력 채널로 리디렉션하는 방법, 로그 파일 롤오버 기준 변경 방법 및 메트릭 데이터를 로그 파일로 보내는 방법을 설명합니다.

## 기본 로깅 구성

브로커는 로그 출력을 로그 파일 집합에 저장하도록 자동으로 구성됩니다. 로그 파일은 연결된 브로커 인스턴스의 이름으로 식별되는 디렉토리에 있습니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

```
.../instances/instanceName/log/
```

로그 파일은 단순 텍스트 파일입니다. 이름은 다음과 같으며 이 순서대로 지정됩니다.

```
log.txt
log_1.txt
log_2.txt
...
log_9.txt
```

기본적으로 로그 파일은 한 주에 한 번씩 롤오버되며, 시스템에서는 아홉 개의 백업 파일을 보존합니다.

- 로그 파일을 보존하는 디렉토리를 변경하려면 `imq.log.file.dirpath` 등록 정보를 원하는 경로로 설정합니다.
- 로그 파일의 기본 이름을 `log`가 아닌 다른 이름으로 변경하려면 `imq.log.file.filename` 등록 정보를 설정합니다.

브로커는 ERROR, WARNING, INFO 등 세 가지 로그 수준을 지원합니다. 각 수준은 표 10-2에 설명되어 있습니다.

**표 10-2**      로깅 수준

수준	설명
ERROR	시스템 오류가 발생할 수 있는 문제에 대한 메시지
WARNING	주의해야 하지만 시스템 오류는 발생하지 않을 경고
INFO	메트릭 및 기타 정보 메시지 보고

로깅 수준을 설정하면 해당 수준 이상의 메시지를 수집합니다. 기본 로그 수준은 INFO이므로 ERROR, WARNING 및 INFO 메시지가 기본적으로 모두 기록됩니다.

## 로그 메시지 형식

기록된 메시지는 타임스탬프, 메시지 코드, 메시지 자체로 이루어집니다. 정보의 양은 설정한 로그 수준에 따라 달라집니다. 다음은 INFO 메시지의 예입니다.

```
[13/Sep/2000:16:13:36 PDT] B1004 Starting the broker service using tcp [
25374,100] with min threads 50 and max threads of 500
```

타임스탬프 표준 시간대를 변경하려면 [324페이지의 표 14-10](#)에 설명되어 있는 `imq.log.timezone` 등록 정보에 대한 정보를 참조하십시오.

## 로거 구성 변경

로거 관련 등록 정보에 대한 설명은 [324페이지의 표 14-10](#)에 나와 있습니다.

### ▶ 브로커의 로거 구성을 변경하는 방법

1. 로그 수준을 설정합니다.
2. 로깅 범주 하나 이상에 해당하는 출력 채널(파일, 콘솔, 또는 둘 다)을 설정합니다.
3. 출력을 파일에 기록하는 경우에는 파일의 롤오버 기준을 구성합니다.



이 단계들은 로거 등록 정보를 설정하여 완료합니다. 이 작업은 두 방법 중 한 가지를 사용하여 수행할 수 있습니다.

- 브로커를 시작하기 전에 브로커의 `config.properties` 파일에 있는 로거 등록 정보를 변경 또는 추가합니다.
- 브로커를 시작하는 `imqbrokerd` 명령에서 로거 명령줄 옵션을 지정합니다. 브로커 옵션 `-D`를 사용해서 로거 등록 정보(또는 모든 브로커 등록 정보)를 변경할 수도 있습니다.

명령줄에 전달되는 옵션은 브로커 인스턴스 구성 파일에서 지정한 등록 정보를 대체합니다. 표 10-3에는 로깅에 영향을 주는 `imqbrokerd` 옵션이 나와 있습니다.

**표 10-3** `imqbrokerd` 로거 옵션 및 해당 등록 정보

<b>imqbrokerd</b> 옵션	설명
<code>-metrics interval</code>	메트릭 정보가 로거에 기록되는 간격(초)을 지정합니다.
<code>-loglevel level</code>	로그 수준을 ERROR, WARNING, INFO 중 하나로 설정합니다.
<code>-silent</code>	콘솔에 대한 로깅을 끕니다.
<code>-tty</code>	모든 메시지를 콘솔로 보냅니다. 기본적으로 WARNING 및 ERROR 수준 메시지만 표시됩니다.

다음 절에서는 기본 구성을 변경하여 다음을 수행하는 방법을 설명합니다.

- 출력 채널(로그 메시지 대상) 변경
- 롤오버 기준 변경

### 출력 채널 변경

기본적으로 오류 및 경고 메시지는 로그 파일에 기록될 뿐 아니라 터미널에도 표시됩니다. Solaris의 경우에는 오류 메시지가 시스템의 `syslog` 데몬에도 기록됩니다.

로그 메시지의 출력 채널은 다음과 같은 방법으로 변경할 수 있습니다.

- 모든 로그 범주(주어진 수준에서)의 출력이 화면에 표시되게 하려면 `imqbrokerd` 명령에 `-tty` 옵션을 사용합니다.
- 로그 출력이 화면에 표시되지 않게 하려면 `imqbrokerd` 명령에 `-silent` 옵션을 사용합니다.

- 로그 파일에 기록할 로깅 정보의 범주를 지정하려면 `imq.log.file.output` 등록 정보를 사용합니다. 예를 들면 다음과 같습니다.  
`imq.log.file.output=ERROR`
- 콘솔에 기록할 로깅 정보의 범주를 지정하려면 `imq.log.console.output` 등록 정보를 사용합니다. 예를 들면 다음과 같습니다.  
`imq.log.console.output=INFO`
- Solaris의 경우 Solaris `syslog`에 기록할 로깅 정보의 범주를 지정하려면 `imq.log.syslog.output` 등록 정보를 사용합니다. 예를 들면 다음과 같습니다.  
`imq.log.syslog.output=NONE`

---

**주** 로거 출력 채널을 변경하기 전에 출력 채널에 매핑할 정보를 지원하는 수준으로 로깅을 설정해야 합니다. 예를 들어, 로그 수준을 `ERROR`로 설정하고 `imq.log.console.output` 등록 정보를 `WARNING`으로 설정한 경우에는 `WARNING` 메시지의 로깅을 활성화하지 않았기 때문에 메시지가 기록되지 않습니다.

---

## 로그 파일 롤오버 기준 변경

로그 파일의 롤오버 기준에는 시간과 크기의 두 가지가 있습니다. 기본값은 시간 기준을 사용하고 7일마다 파일을 롤오버하는 것입니다.

- 시간 간격을 변경하려면 `imq.log.file.rolloversecs` 등록 정보를 변경해야 합니다. 예를 들어, 다음과 같은 등록 정보 정의를 사용하면 시간 간격을 10일로 변경할 수 있습니다.  
`imq.log.file.rolloversecs=864000`
- 롤오버 기준을 파일 크기로 변경하려면 `imq.log.file.rolloverbytes` 등록 정보를 설정해야 합니다. 예를 들어, 다음 정의는 500,000바이트 제한에 도달하면 파일을 롤오버하도록 브로커를 설정합니다.  
`imq.log.file.rolloverbytes=500000`

시간 및 크기 관련 롤오버 등록 정보를 모두 설정한 경우에는 먼저 도달한 제한에 의해 롤오버가 발생합니다. 앞에서 설명했듯이 브로커는 아홉 개까지의 롤오버 파일을 보존합니다.

브로커가 실행되고 있을 때 로그 파일 롤오버 등록 정보를 설정 또는 변경할 수 있습니다. 이러한 등록 정보를 설정하려면 `imqcmd update bkr` 명령을 사용합니다.

## 로그 파일로 메트릭 데이터 보내기

이 절에서는 브로커 로그 파일을 사용하여 메트릭 정보를 보고하는 절차를 설명합니다. 로거 구성에 대한 자세한 내용은 [205페이지의 "브로커 로그 구성 및 사용"](#)을 참조하십시오.

### ▶ 로그 파일을 사용하여 메트릭 정보를 보고하는 방법

#### 1. 브로커의 메트릭 생성 기능을 구성합니다.

- a. `imq.metrics.enabled=true`인지 확인합니다.

로깅을 위한 메트릭 생성은 기본적으로 설정되어 있습니다.

- b. 메트릭 생성 간격을 원하는 시간(초)으로 설정합니다.

```
imq.metrics.interval=interval
```

이 값은 `config.properties` 파일에서 설정하거나 브로커를 시작할 때 `-metrics interval` 명령줄 옵션을 사용하여 설정할 수 있습니다.

#### 2. 로거가 메트릭 정보를 수집하는지 확인합니다.

```
imq.log.level=INFO
```

이것이 기본값입니다. 이 값은 `config.properties` 파일에서 설정하거나 브로커를 시작할 때 `-loglevel level` 명령줄 옵션을 사용하여 설정할 수 있습니다.

#### 3. 로거가 메트릭 정보를 로그 파일에 기록하도록 설정되어 있는지 확인합니다.

```
imq.log.file.output=INFO
```

이것이 기본값입니다. 이 값은 `config.properties` 파일에서 설정할 수 있습니다.

#### 4. 브로커를 시작합니다.

다음에는 로그 파일로의 샘플 브로커 메트릭 출력이 나와 있습니다.

```
[21/Jul/2004:11:21:18 PDT]
Connections: 0   JVM Heap: 8323072 bytes (7226576 free) Threads: 0 (14-1010)
  In: 0 msgs (0bytes) 0 pkts (0 bytes)
  Out: 0 msgs (0bytes) 0 pkts (0 bytes)
Rate In: 0 msgs/sec (0 bytes/sec) 0 pkts/sec (0 bytes/sec)
Rate Out: 0 msgs/sec (0 bytes/sec) 0 pkts/sec (0 bytes/sec)
```

메트릭 데이터에 대한 자세한 내용은 [18장, "메트릭 참조"](#)를 참조하십시오.

## 사용 불능 메시지 기록

브로커에 대한 사용 불능 메시지 로깅을 사용하여 물리적 대상을 모니터링할 수 있습니다. 사용 불능 메시지 대기열을 사용하는지 여부에 관계 없이 사용 불능 메시지를 기록할 수 있습니다.

사용 불능 메시지 로깅을 사용하는 경우 브로커는 다음과 같은 이벤트 유형을 기록합니다.

- 물리적 대상이 최대 크기를 초과했습니다.
- 브로커가 다음과 같은 이유로 물리적 대상에서 메시지를 제거했습니다.
  - 대상 크기 제한에 도달했습니다.
  - 메시지 활성 시간이 만료되었습니다.
  - 메시지가 너무 큼니다.
  - 브로커가 메시지를 처리하는 중에 오류가 발생했습니다.

사용 불능 메시지 대기열을 사용하는 경우 다음과 같은 이벤트 유형도 로깅에 포함됩니다.

- 브로커가 메시지를 사용 불능 메시지 대기열로 이동했습니다.
- 브로커가 메시지를 사용 불능 메시지 대기열에서 제거하여 삭제했습니다.

사용 불능 메시지 로깅은 기본적으로 비활성화됩니다. 사용 불능 메시지 로깅을 사용하려면 `imq.destination.logDeadMsgs` 브로커 속성을 설정합니다.

# 대화식 메트릭 표시

Message Queue 브로커는 다음과 같은 이벤트 유형을 보고할 수 있습니다.

- **Java Virtual Machine(JVM) 메트릭.** JVM 힙 크기에 대한 정보.
- **브로커 전체 메트릭.** 브로커에 저장되어 있는 메시지, 브로커에 유입 및 유출되는 메시지, 메모리 사용 등에 대한 정보. 메시지는 메시지 수와 바이트 수라는 측면에서 추적됩니다.
- **연결 서비스 메트릭.** 연결 및 연결 스레드 자원 정보, 특정 연결 서비스에 대한 메시지 흐름 정보.
- **대상 메트릭.** 특정 물리적 대상에 유입 및 유출되는 메시지 정보, 물리적 대상 사용자에 대한 정보, 메모리 및 디스크 공간 사용에 대한 정보

imqcmd 명령은 브로커 전체, 개별 연결 서비스 및 개별 물리적 대상에 대한 메트릭 정보를 얻을 수 있습니다. 메트릭 데이터를 얻으려면 일반적으로 imqcmd의 metrics 하위 명령을 사용합니다. 메트릭 데이터는 지정한 간격이나 지정한 횟수에 콘솔 화면에 기록됩니다.

query 하위 명령을 사용하여 구성 정보를 포함하는 유사한 데이터를 볼 수도 있습니다. 자세한 내용은 [214페이지의 "imqcmd query"](#)를 참조하십시오.

## imqcmd 메트릭

imqcmd metrics의 구문과 옵션은 각각 [표 10-4](#)와 [표 10-5](#)에 나타나 있습니다.

**표 10-4** imqcmd metrics 하위 명령 구문

하위 명령 구문	제공되는 메트릭 데이터
metrics bkr [-b <i>hostName:port</i> ] [-m <i>metricType</i> ] [-int <i>interval</i> ] [-msp <i>numSamples</i> ]	기본 브로커 또는 지정한 호스트 및 포트의 브로커 메트릭을 표시합니다.
또는	
metrics svc -n <i>serviceName</i> [-b <i>hostName:port</i> ] [-m <i>metricType</i> ] [-int <i>interval</i> ] [-msp <i>numSamples</i> ]	기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 지정된 서비스의 메트릭을 표시합니다.
또는	
metrics dst -t <i>destType</i> -n <i>destName</i> [-b <i>hostName:port</i> ] [-m <i>metricType</i> ] [-int <i>interval</i> ] [-msp <i>numSamples</i> ]	지정한 유형과 이름의 물리적 대상에 대한 메트릭 정보를 표시합니다.

**표 10-5** imqcmd metrics 하위 명령 옵션

하위 명령 옵션	설명
-b <i>hostName:port</i>	메트릭 데이터를 보고하는 브로커의 호스트 이름과 포트를 지정합니다. 기본값은 localhost:7676입니다.
-int <i>interval</i>	메트릭을 표시할 간격(초)을 지정합니다. 기본값은 5초입니다.
-m <i>metricType</i>	표시할 메트릭의 유형을 지정합니다. <b>tt1</b> 브로커, 서비스 또는 대상에 유입되고 유출되는 메시지와 패킷에 대한 메트릭을 표시합니다(기본 메트릭 유형). <b>rts</b> 브로커, 연결 서비스 또는 대상에 유입 및 유출되는 메시지와 패킷의 메트릭을 초당 속도로 표시합니다. <b>cxm</b> 연결, 가상 메모리 힘 및 스레드를 표시합니다(브로커 및 연결 서비스에만 해당). <b>oon</b> 사용자 관련 메트릭을 표시합니다(대상에만 해당). <b>disk</b> 디스크 사용 메트릭을 표시합니다(대상에만 해당).
-msp <i>numSamples</i>	출력에 표시되는 샘플 수를 지정합니다. 기본값은 무제한 수입니다(무한).
-n <i>destName</i>	메트릭 데이터가 보고되는 물리적 대상(있는 경우)의 이름을 지정합니다. 기본값이 없습니다.
-n <i>serviceName</i>	메트릭 데이터가 보고되는 연결 서비스(있는 경우)를 지정합니다. 기본값이 없습니다.
-t <i>destTyp</i>	메트릭 데이터가 보고되는 물리적 대상(있는 경우)의 유형(대기열 또는 주제)을 지정합니다. 기본값이 없습니다.

## 메트릭 하위 명령을 사용하여 메트릭 데이터 표시

이 절에서는 metrics 하위 명령을 사용하여 메트릭 정보를 보고하는 절차를 설명합니다.

### ▶ 메트릭 하위 명령을 사용하는 방법

1. 메트릭 정보가 필요한 브로커를 시작합니다.  
 67페이지의 "대화식으로 브로커 시작"을 참조하십시오.
2. 표 10-4와 표 10-5에 표시된 대로 적절한 imqcmd metrics 하위 명령과 옵션을 실행합니다.

## 메트릭 출력: imqcmd metrics

이 절에서는 imqcmd metrics 하위 명령의 출력 예를 보여줍니다. 예로는 브로커 전체, 연결 서비스 및 물리적 대상 메트릭이 나와 있습니다.

### 브로커 전체 메트릭

메시지와 패킷이 브로커에 유입 및 유출되는 속도를 10초 간격으로 구하려면 다음과 같이 metrics bkr 하위 명령을 사용합니다.

```
imqcmd metrics bkr -m rts -int 10 -u admin
```

이 명령은 다음과 유사한 출력을 생성합니다(350페이지의 표 18-2의 데이터 설명 참조).

-----							
Msgs/sec		Msg Bytes/sec		Pkts/sec		Pkt Bytes/sec	
In	Out	In	Out	In	Out	In	Out
-----							
0	0	27	56	0	0	38	66
10	0	7365	56	10	10	7457	1132
0	0	27	56	0	0	38	73
0	10	27	7402	10	20	1400	8459
0	0	27	56	0	0	38	73

### 연결 서비스 메트릭

jms 연결 서비스가 처리한 메시지와 패킷의 누적 총 수를 구하려면 다음과 같이 metrics svc 하위 명령을 사용합니다.

```
imqcmd metrics svc -n jms -m ttl -u admin
```

이 명령은 다음과 유사한 출력을 생성합니다(352페이지의 표 18-3의 데이터 설명 참조).

-----							
Msgs		Msg Bytes		Pkts		Pkt Bytes	
In	Out	In	Out	In	Out	In	Out
-----							
164	100	120704	73600	282	383	135967	102127
657	100	483552	73600	775	876	498815	149948

## 물리적 대상 메트릭

물리적 대상에 대한 메트릭 정보를 얻으려면 다음과 같이 `metrics dst` 하위 명령을 사용하십시오.

```
imqcmd metrics dst -t q -n XQueue -m ttl -u admin
```

이 명령은 다음과 유사한 출력을 생성합니다(354페이지의 표 18-4의 데이터 설명 참조).

Msgs		Msg Bytes		Msg Count			Total Msg Bytes (k)			Largest
In	Out	In	Out	Current	Peak	Avg	Current	Peak	Avg	Msg (k)
200	200	147200	147200	0	200	0	0	143	71	0
300	200	220800	147200	100	200	10	71	143	64	0
300	300	220800	220800	0	200	0	0	143	59	0

물리적 대상의 사용자에 대한 정보를 얻으려면 다음과 같이 `metrics dst` 하위 명령을 사용합니다.

```
imqcmd metrics dst -t q -n SimpleQueue -m con -u admin
```

이 명령은 다음과 유사한 출력을 생성합니다(354페이지의 표 18-4의 데이터 설명 참조).

Active Consumers			Backup Consumers			Msg Count		
Current	Peak	Avg	Current	Peak	Avg	Current	Peak	Avg
1	1	0	0	0	0	944	1000	525

## imqcmd query

`imqcmd query`의 구문과 옵션은 이 명령이 제공하는 메트릭 데이터에 대한 설명과 함께 표 10-6에 나와 있습니다.



**표 10-6** imqcmd query 하위 명령 구문

하위 명령 구문	제공되는 메트릭 데이터
<pre>query bkr   [-b hostName:port]</pre> <p>또는</p> <pre>query svc -n serviceName   [-b hostName:port]</pre> <p>또는</p> <pre>query dst -t destType   -n destName   [-b hostName:port]</pre>	<p>브로커 메모리와 영구 저장소에 저장되어 있는 현재의 메시지 수와 메시지 바이트에 대한 정보(111페이지의 "브로커 정보 표시" 참조)</p> <p>지정한 연결 서비스에 대한 현재의 할당된 스레드 수와 연결 수에 대한 정보(118페이지의 "연결 서비스 정보 표시" 참조)</p> <p>지정한 대상의 메모리와 영구 저장소에 저장되어 있는 생성자, 활성 및 백업 사용자, 메시지 및 메시지 바이트의 현재 수에 대한 정보(131페이지의 "물리적 대상 정보 표시" 참조)</p>

**주** imqcmd query에서 제공하는 메트릭 데이터는 제한되어 있기 때문에 이 도구는 349페이지의 18장, "메트릭 참조"에 제시된 표에 설명되어 있지 않습니다.

## 브로커를 모니터링하는 응용 프로그램 작성

Message Queue는 브로커가 메트릭 데이터를 JMS 메시지에 기록한 다음 메시지에 포함된 메트릭 정보 유형에 따라 여러 메트릭 주제 대상 중 하나에 보낼 수 있는 메트릭 모니터링 기능을 제공합니다.

메트릭 주제 대상에 가입하고 이러한 대상에서 메시지를 사용하며 메시지에 포함된 메트릭 정보를 처리하는 클라이언트 응용 프로그램을 작성하여 이러한 메트릭 정보에 액세스할 수 있습니다.

다섯 개의 메트릭 주제 대상이 있으며 표 10-7에 그 이름과 함께 각 대상에 전달되는 메트릭 메시지 유형이 표시되어 있습니다.

**표 10-7**      메트릭 주제 대상

주제 이름	메트릭 메시지 유형
mq.metrics.broker	브로커 메트릭
mq.metrics.jvm	Java 가상 머신 메트릭
mq.metrics.destination_list	대상 및 해당 유형 목록
mq.metrics.destination.queue. <i>monitoredDestinationName</i>	지정된 이름의 대기열에 대한 대상 메트릭
mq.metrics.destination.topic. <i>monitoredDestinationName</i>	지정된 이름의 주제에 대한 대상 메트릭

## 메시지 기반 모니터링 설정

이 절에서는 메시지 기반 모니터링 기능을 사용하여 메트릭 정보를 수집하는 절차를 설명합니다. 이 절차는 클라이언트 개발과 관리 작업을 모두 포함합니다.

### ▶ 메시지 기반 모니터링을 설정하는 방법

1. 메트릭 모니터링 클라이언트를 작성합니다.

메트릭 주제 대상에 가입하고 메트릭 메시지를 사용하며 이러한 메시지에서 메트릭 데이터를 추출하는 클라이언트를 프로그래밍하는 데 대한 지침은 *Message Queue Developer's Guide for Java Clients*를 참조하십시오.

2. `config.properties` 파일에서 브로커 등록 정보 값을 설정하여 브로커의 메트릭 메시지 생성자를 구성합니다.

- a. 메트릭 메시지 생성을 활성화합니다.

`imq.metrics.topic.enabled=true`를 설정합니다.

기본값은 `true`입니다.

- b. 메트릭 메시지가 생성되는 간격(초)을 설정합니다.

`imq.metrics.topic.interval=interval`을 설정합니다.

기본값은 60초입니다.

- c. 메트릭 메시지가 지속되는지(즉, 브로커에 오류가 발생해도 메시지가 보존되는지) 여부를 지정합니다.  
`img.metrics.topic.persist`를 설정합니다.  
 기본값은 `false`입니다.
  - d. 메트릭 메시지가 삭제되기 전까지 해당 대상에 남아 있는 기간을 지정합니다.  
`img.metrics.topic.timetolive`를 지정합니다.  
 기본값은 300초입니다.
3. 메트릭 주제 대상에 대한 액세스 제어가 필요한 경우 설정합니다.  
 아래의 "[보안 및 액세스 고려 사항](#)"에 있는 설명을 참조하십시오.
  4. 메트릭 모니터링 클라이언트를 시작합니다.  
 사용자가 메트릭 주제에 가입하면 메트릭 주제 대상이 자동으로 만들어집니다. 메트릭 주제가 만들어지면 브로커 메트릭 메시지 생성자가 메트릭 메시지를 메트릭 주제로 보내기 시작합니다.

## 보안 및 액세스 고려 사항

메트릭 주제 대상에 대한 액세스를 제한하는 이유는 두 가지입니다.

- 메트릭 데이터에는 브로커와 그 자원에 대한 중요한 정보가 포함되어 있을 수 있습니다.
- 메트릭 주제 대상에 대한 가입 수가 과도한 경우 브로커 오버헤드가 늘어나고 성능에 부정적인 영향을 미칠 수 있습니다.

이러한 사항을 고려할 때 메트릭 주제 대상에 대한 액세스를 제한하는 것이 좋습니다.

모니터링 클라이언트는 다른 클라이언트와 같은 인증 및 권한 부여 제어를 받습니다.

Message Queue 사용자 저장소에서 유지 관리되는 사용자만 브로커에 연결할 수 있습니다.

152페이지의 "[사용자 권한 부여: 액세스 제어 등록 정보 파일](#)"에 설명되어 있는 액세스 제어 등록 정보 파일을 통해 특정 메트릭 주제 대상에 대한 액세스를 제한하여 추가 보호를 제공할 수 있습니다.

예를 들어 `accesscontrol.properties` 파일의 다음 항목은 `user1`과 `user2`를 제외한 모든 사람에 대해 `mq.metrics.broker` 메트릭 주제에 대한 액세스를 거부합니다.

```
topic.mq.metrics.broker.consume.deny.user=*
topic.mq.metrics.broker.consume.allow.user=user1,user2
```

다음 항목은 사용자 user3만 주제 t1을 모니터링할 수 있도록 합니다.

```
topic.mq.metrics.destination.topic.t1.consume.deny.user=*  
topic.mq.metrics.destination.topic.t1.consume.allow.user=user3
```

메트릭 데이터의 중요도에 따라 암호화된 연결을 사용하여 메트릭 모니터링 클라이언트를 브로커에 연결할 수도 있습니다. 암호화된 연결 사용에 대한 자세한 내용은 [159페이지](#)의 "SSL 기반 서비스 작업"을 참조하십시오.

## 메트릭 출력: 메트릭 메시지

메시지 기반 모니터링 API를 사용하여 얻는 메트릭 데이터 출력은 사용자가 어떤 메트릭 모니터링 클라이언트를 작성하는지에 따라 다릅니다. 단지 브로커의 메트릭 생성기에서 어떤 데이터를 제공하는지에 따라 제한을 받습니다. 이 데이터의 전체 목록을 보려면 [349페이지](#)의 "메트릭 참조"를 참조하십시오.

# 메시지 서비스 분석 및 조정

이 장에서는 메시징 응용 프로그램의 성능을 최적화하기 위해 Message Queue 서비스를 분석하고 조정하는 방법에 대한 여러 항목을 다룹니다. 이 장은 다음 항목으로 구성되어 있습니다.

- [219페이지의 "성능 정보"](#)
- [223페이지의 "성능에 영향을 미치는 요소"](#)
- [237페이지의 "성능 향상을 위한 구성 조정"](#)

## 성능 정보

이 절에서는 성능 조정에 대한 배경 정보를 제공합니다.

### 성능 조정 프로세스

메시징 응용 프로그램의 성능은 응용 프로그램과 Message Queue 서비스 사이의 상호 작용에 따라 달라집니다. 따라서 성능을 최대화하려면 응용 프로그램 개발자와 관리자가 함께 노력해야 합니다.

성능 최적화 프로세스는 응용 프로그램 설계에서 시작되어 응용 프로그램을 배포 이후 메시지 서비스 조정에 이르기까지 계속됩니다. 성능 조정 프로세스에는 다음 단계가 포함됩니다.

- 응용 프로그램의 성능 요구 사항 정의
- 성능에 영향을 미치는 요소(특히 안정성과 성능 간의 균형)를 고려하여 응용 프로그램 설계
- 기본 성능 측정 설정
- 성능 최적화를 위해 메시지 서비스 조정 또는 재구성

위에서 설명한 프로세스는 자주 반복됩니다. 응용 프로그램 배포 중에 Message Queue 관리자는 메시지 서버가 응용 프로그램의 일반 성능 요구 사항에 적합한지 평가합니다. 벤치마크 테스트가 이러한 요구 사항을 충족시키는 경우 관리자는 이 장에서 설명한 대로 시스템을 조정할 수 있습니다. 하지만 벤치마크 테스트가 성능 요구 사항을 충족시키지 못하는 경우 응용 프로그램을 재설계하거나 배포 구조를 수정해야 합니다.

## 성능 요소

일반적으로 성능은 메시지 서비스가 생성자의 메시지를 사용자에게 전달하는 속도와 효율성에 대한 측정입니다. 하지만 사용자의 필요에 따라 중요할 수 있는 여러 다른 성능 요소가 있습니다.

**연결 로드** 메시지 생성자나 메시지 사용자 수 또는 시스템이 지원할 수 있는 동시 연결 수입입니다.

**메시지 처리량** 메시징 시스템을 통해 전달될 수 있는 초당 메시지 수 또는 메시지 바이트입니다.

**대기 시간** 메시지 생성자로부터 메시지 사용자에게 특정 메시지를 전달하는 데 걸리는 시간입니다.

**안정성** 메시지 서비스의 전체적인 가용성 또는 로드량이 많거나 장애가 발생하는 경우 메시지 서비스 성능이 유연하게 감소되는 정도입니다.

**효율성** 메시지 전달의 효율성, 즉 사용된 컴퓨팅 자원을 기준으로 메시지 처리량을 측정하는 것입니다.

이러한 성능의 여러 요소들은 일반적으로 상호 연관됩니다. 메시지 처리량이 높으면 메시지가 메시지 서버에서 백로그될 가능성이 적어지므로 대기 시간이 짧아집니다(단일 메시지가 매우 빠르게 전달될 수 있음). 하지만 대기 시간은 통신 연결 속도, 메시지 서버 처리 속도, 클라이언트 처리 속도 등 많은 요소에 따라 달라질 수 있습니다.

어느 경우든 여러 다른 성능 요소가 있습니다. 어떤 성능 요소가 가장 중요한지는 일반적으로 특정 응용 프로그램의 요구 사항에 따라 달라집니다.

## 벤치마크

벤치마크는 메시징 응용 프로그램을 위한 테스트 프로그램을 만들고 이 테스트 프로그램의 메시지 처리량이나 기타 성능 요소를 측정하는 프로세스입니다.

예를 들어, 일정 수의 생성자 클라이언트가 일정 수의 연결, 세션 및 메시지 생성자를 사용하여 메시징 응용 프로그램 설계에 따라 다른 특정 수의 대기열이나 주제에 표준 크기의 지속성 또는 비지속성 메시지를 지정된 속도로 보내는 테스트 프로그램을 만들 수 있습니다. 마찬가지로 테스트 프로그램에는 일정 수의 연결, 세션 및 특정 확인 모드로 테스트 프로그램 물리적 대상에서 메시지를 사용하는 특정 유형의 메시지 사용자를 사용하는 일정 수의 사용자 클라이언트가 포함됩니다.

표준 테스트 프로그램을 사용하면 메시지 생성에서 사용까지 걸리는 시간이나 평균 메시지 처리 속도를 측정할 수 있으며 시스템을 모니터링하여 연결 스레드 사용, 메시지 저장소 데이터, 메시지 흐름 데이터 및 기타 관련 메트릭을 살펴볼 수 있습니다. 그런 다음 성능에 부정적인 영향을 미칠 때까지 메시지 생성 속도, 메시지 생성자 수 또는 기타 변수를 증가시켜볼 수 있습니다. 달성할 수 있는 최대 처리량이 메시지 서비스 구성에 대한 벤치마크입니다.

이 벤치마크를 사용하여 테스트 프로그램의 일부 특성을 수정할 수 있습니다. 성능에 영향을 미칠 수 있는 모든 요소([224페이지의 "성능에 영향을 미치는 응용 프로그램 설계 요소" 참조](#))를 조심스럽게 제어하여 이러한 요소들 중 일부를 변경했을 때 벤치마크에 미치는 영향을 살펴볼 수 있습니다. 예를 들어, 연결 수나 메시지 크기를 5배나 10배 증가시켜 성능에 미치는 영향을 살펴볼 수 있습니다.

이와 반대로 응용 프로그램 기반 요소를 일정하게 유지하면서 제어 가능한 방식으로 브로커 구성을 변경하고(예: 연결 등록 정보, 스레드 풀 등록 정보, JVM 메모리 제한, 제한 동작, 기본 제공 지속성 대 플러그인 지속성 등을 변경) 이러한 변경이 성능에 미치는 영향을 살펴볼 수 있습니다.

이러한 응용 프로그램 벤치마크는 메시지 서비스를 조정하여 배포된 응용 프로그램의 성능을 높이고자 할 때 유용한 정보를 제공합니다. 벤치마크를 사용하면 한 가지 변경 사항이나 일련의 변경 사항이 미치는 영향을 좀 더 정확하게 예상할 수 있습니다.

일반적으로 벤치마크는 제어된 테스트 환경에서 메시지 서비스가 안정될 정도의 충분한 기간 동안 실행해야 합니다. Java 코드를 기계 코드로 변환하는 JIT(Just-In-Time) 컴파일 과정은 시작 시 성능에 부정적인 영향을 미칩니다.

## 기본 사용 패턴

일단 메시징 응용 프로그램이 배포되어 실행 중이면 기본 사용 패턴을 설정하는 것이 중요합니다. 최대 수요가 발생하는 시기를 알아 해당 수요를 수량화할 수 있습니다. 예를 들어, 수요는 일반적으로 최종 사용자 수, 활동 수준, 시간 또는 이러한 모든 것에 의해 변동됩니다.

기본 사용 패턴을 설정하려면 메시지 서버를 오랫동안 모니터링하여 다음과 같은 데이터를 조사해야 합니다.

- 연결 수
- 브로커 또는 특정 물리적 대상에 저장된 메시지 수
- 브로커 또는 특정 물리적 대상에 유입 및 유출되는 메시지
- 활성 사용자 수

메트릭 데이터에서 제공하는 평균 값과 최대 값을 사용할 수도 있습니다.

이러한 기본 메트릭을 설계 예측과 비교하여 확인하는 것이 중요합니다. 이렇게 함으로써 클라이언트 코드가 제대로 작동하는지 확인할 수 있습니다. 예를 들어, 연결이 열려 있는 상태로 남아 있지 않은지 또는 사용된 메시지가 인식할 수 없는 상태로 남아 있지 않은지 확인할 수 있습니다. 이러한 코딩 오류는 메시지 서버 자원을 사용하므로 성능에 상당한 영향을 미칠 수 있습니다.

기본 사용 패턴을 통해 최적 성능을 위해 시스템을 조정하는 방법을 결정할 수 있습니다. 예를 들면 다음과 같습니다.

- 한 물리적 대상이 다른 물리적 대상에 비해 훨씬 더 많이 사용되는 경우 해당 물리적 대상에 다른 물리적 대상보다 더 높은 메시지 메모리 제한을 설정하거나 상황에 맞게 제한 동작을 조정할 수 있습니다.
- 필요한 연결 수가 최대 스레드 풀 크기에서 허용하는 연결 수보다 훨씬 더 큰 경우 스레드 풀 크기를 늘리거나 공유 스레드 모델을 적용할 수 있습니다.
- 최대 메시지 흐름이 평균 흐름보다 훨씬 큰 경우 메모리가 부족할 때 사용하는 제한 동작에 영향을 미칠 수 있습니다.

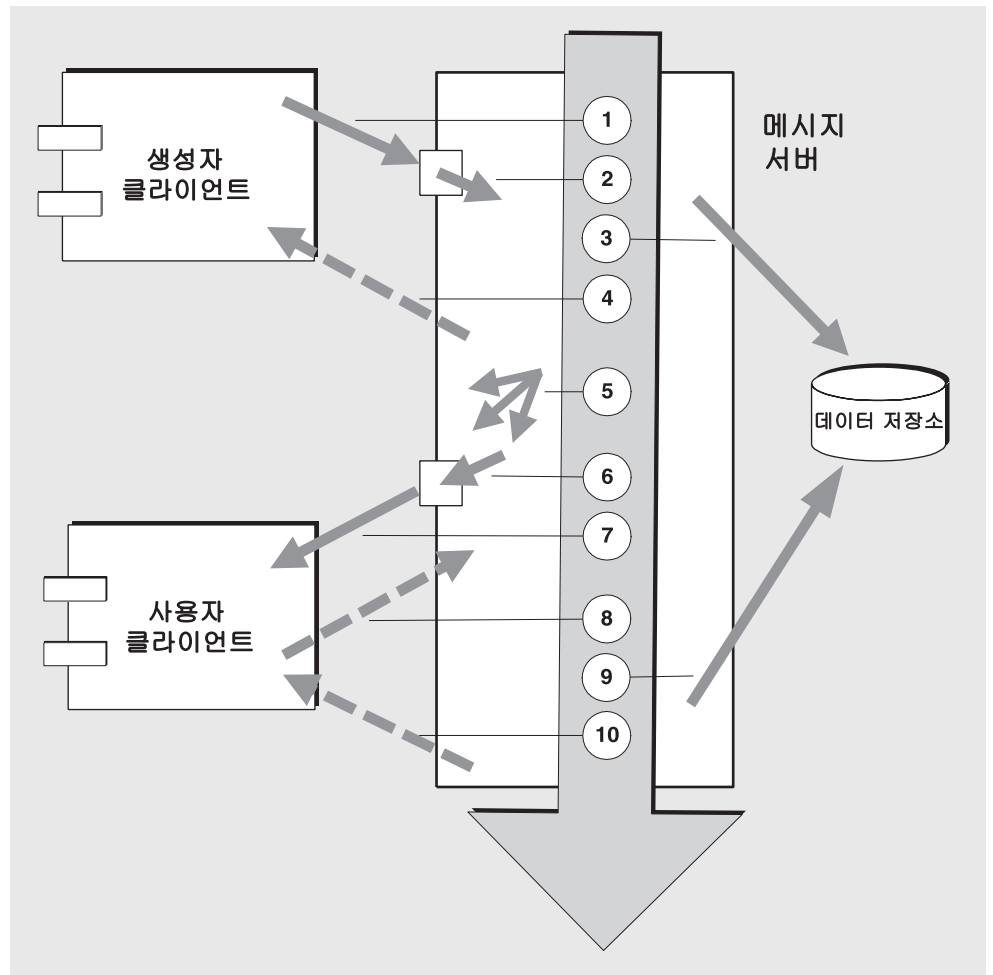
일반적으로 사용 패턴에 대해 더 많이 알수록 이러한 패턴에 맞춰 시스템을 조정하고 향후 요구에 더 잘 대비할 수 있습니다.



## 성능에 영향을 미치는 요소

두 가지 주요 성능 표시기인 메시지 대기 시간과 메시지 처리량은 일반 메시지가 메시지 전달 프로세스의 여러 단계를 완료하는 데 걸리는 시간에 따라 달라집니다. 아래에서 안정적으로 전달되는 지속성 메시지의 경우에 대해 이러한 단계를 보여 줍니다. 그 단계는 다음 그림과 같습니다.

그림 11-1 Message Queue 서비스를 통한 메시지 전달



1. 메시지가 생성자 클라이언트에서 메시지 서버로 전달됩니다.
2. 메시지 서버가 메시지를 읽습니다.
3. 안정성을 위해 메시지가 영구 저장소에 저장됩니다.
4. 안정성을 위해 메시지 서버가 메시지 수신을 확인합니다.
5. 메시지 서버가 메시지를 라우팅합니다.
6. 메시지 서버가 메시지를 씁니다.
7. 메시지가 메시지 서버에서 사용자 클라이언트로 전달됩니다.
8. 안정성을 위해 사용자 클라이언트가 메시지 수신을 확인합니다.
9. 안정성을 위해 메시지 서버가 클라이언트 확인을 처리합니다.
10. 메시지 서버가 클라이언트 확인이 처리되었는지 확인합니다.

이러한 단계는 순차적이므로 어느 단계든 생성자 클라이언트에서 사용자 클라이언트로 메시지를 전달할 때 병목 현상을 일으킬 수 있습니다. 이러한 단계들은 대부분 네트워크 대역폭, 컴퓨터 처리 속도, 메시지 서버 구조 등 메시징 시스템의 물리적 특성에 영향을 받습니다. 하지만 일부 단계는 메시징 응용 프로그램의 특성과 메시징 응용 프로그램이 요구하는 안정성 수준에 따라서도 달라집니다.

다음 하위 절에서는 응용 프로그램 설계 요소와 메시징 시스템 요소가 성능에 미치는 영향을 설명합니다. 응용 프로그램 설계와 메시징 시스템 요소가 메시지 전달에서 긴밀하게 상호 작용하지만 각 범주에 대해 별도로 고려합니다.

## 성능에 영향을 미치는 응용 프로그램 설계 요소

응용 프로그램 설계 결정은 전체 메시징 성능에 상당한 영향을 미칠 수 있습니다.

성능에 영향을 미치는 가장 중요한 요소는 메시지 전달의 안전성에 영향을 미치는 요소입니다. 이러한 요소는 다음과 같습니다.

- 전달 모드(지속성/비지속성 메시지)
- 트랜잭션 사용
- 확인 모드
- 영구 가입 및 비영구 가입

성능에 영향을 미치는 다른 응용 프로그램 설계 요소는 다음과 같습니다.

- 선택기 사용(메시지 필터링)
- 메시지 크기
- 메시지 본문 유형

다음 절에서는 이러한 요소 각각이 메시징 성능에 미치는 영향을 설명합니다. 일반적으로 성능과 안정성은 동시에 얻을 수 없습니다. 즉, 안정성을 높이는 요소가 성능을 저하시키는 경향이 있습니다.

**표 11-1**은 여러 응용 프로그램 설계 요소가 일반적으로 메시징 성능에 어떤 영향을 미치는지 보여 줍니다. 이 표에서는 두 가지 시나리오(높은 안정성/낮은 성능 시나리오와 높은 성능/낮은 안정성 시나리오)와 각각을 특징 지우는 응용 프로그램 설계 요소의 선택을 보여 줍니다. 이러한 양극단 사이에는 안정성과 성능 모두에 영향을 미치는 여러 선택 사항과 상충이 있습니다.

**표 11-1**      높은 안정성 및 높은 성능 시나리오 비교

응용 프로그램 설계 요소	높은 안정성 낮은 성능 시나리오	높은 성능 낮은 안정성 시나리오
전달 모드	지속성 메시지	비지속성 메시지
트랜잭션 사용	트랜잭션된 세션	트랜잭션 없음
확인 모드	AUTO_ACKNOWLEDGE 또는 CLIENT_ACKNOWLEDGE	DUPS_OK_ACKNOWLEDGE
영구/비영구 가입	영구 가입	비영구 가입
선택기 사용	메시지 필터링	메시지 필터링 없음
메시지 크기	많은 수의 작은 메시지	적은 수의 큰 메시지
메시지 본문 유형	복합 본문 유형	단순 본문 유형

---

**주**        다음 그래프의 성능 데이터는 1002Mhz CPU가 두 개 있는 Solaris 8 시스템에서 파일 기반 지속성을 사용하여 생성되었습니다. 성능 테스트는 JIT(Just-In-Time) 컴파일러가 시스템을 최적화하고 지속성 데이터베이스를 사용할 수 있도록 먼저 Message Queue 브로커를 준비했습니다.

      브로커가 준비되면 30초 동안 단일 생성자와 단일 사용자를 만들고 메시지를 생성했습니다. 사용자가 생성된 모든 메시지를 받는 데 필요한 시간을 기록했고 처리 속도(초당 메시지)를 계산했습니다. 표 11-1에 표시된 응용 프로그램 설계 요소의 다른 조합에 대해 이 시나리오를 반복했습니다.

---

### 전달 모드(지속성/비지속성 메시지)

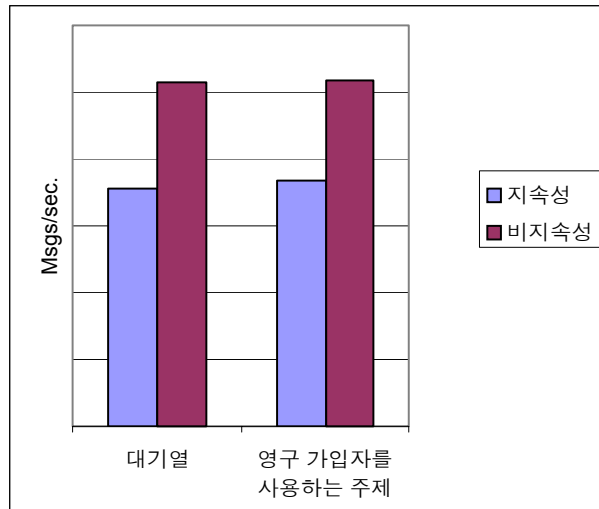
지속성 메시지는 메시지 서버에 오류가 발생하는 경우 메시지 전달을 보장합니다. 브로커는 의도된 모든 사용자가 메시지 사용을 확인할 때까지 메시지를 영구 저장소에 저장합니다.

다음과 같은 이유로 브로커의 지속성 메시지 처리가 비지속성 메시지 처리보다 느립니다.

- 브로커는 브로커에 오류가 발생하더라도 지속성 메시지가 손실되지 않도록 안정적으로 메시지를 저장해야 합니다.
- 브로커는 수신되는 각 지속성 메시지의 수신을 확인해야 합니다. 메시지를 생성하는 메소드가 예외 없이 반환되면 브로커로의 전달이 보장됩니다.
- 클라이언트 확인 모드에 따라 브로커는 지속성 메시지를 사용자 클라이언트가 확인했는지 확인해야 할 수 있습니다.

지속성 및 비지속성 모드간 성능 차이가 상당할 수 있습니다. 그림 11-2는 두 가지 안정적 전달 사례(대기열과 영구 가입을 사용하는 주제 양쪽에 전달되는 10k 크기의 메시지)에서 지속성 메시지와 비지속성 메시지에 대한 처리량을 비교합니다. 두 가지 경우 모두 AUTO\_ACKNOWLEDGE 확인 모드를 사용합니다.

그림 11-2 전달 모드의 성능 영향



## 트랜잭션 사용

트랜잭션은 트랜잭션된 세션에서 생성된 모든 메시지와 트랜잭션된 세션에서 사용된 모든 메시지가 하나의 단위로 처리되거나 처리되지 않도록(롤백되도록) 보장합니다.

Message Queue는 로컬 트랜잭션과 분산 트랜잭션을 모두 지원합니다.

트랜잭션된 세션에서의 메시지 생성이나 확인은 다음과 같은 이유 때문에 트랜잭션되지 않은 세션보다 느립니다.

- 생성된 각 메시지와 함께 추가 정보를 저장해야 합니다.
- 가입이 없는 주제 대상에 전달되는 지속성 메시지는 일반적으로 삭제되지만 트랜잭션이 시작될 때 가입에 대한 정보를 사용할 수 없는 경우와 같이 일반적으로 그럴 수 없는 데도 트랜잭션의 메시지가 저장되는 경우가 있습니다.
- 트랜잭션이 완료될 때 트랜잭션 내의 메시지 사용과 확인에 대한 정보를 저장하고 처리해야 합니다.

## 확인 모드

JMS 메시지 전달의 안정성을 보장하는 한 가지 방법은 Message Queue 메시지 서버가 클라이언트에 전달한 메시지 사용에 대해 클라이언트가 확인 응답을 보내는 것입니다.

클라이언트의 메시지 확인 없이 세션이 닫히거나 확인이 처리되기 전에 메시지 서버에 오류가 발생하는 경우 브로커는 해당 메시지를 재전송하여 JMSRedelivered 플래그를 설정합니다.

트랜잭션되지 않은 세션의 경우 클라이언트는 각각 고유한 성능 특성을 가지는 다음과 같은 세 가지 확인 모드 중 하나를 선택할 수 있습니다.

- **AUTO\_ACKNOWLEDGE.** 사용자가 메시지를 처리하면 시스템이 자동으로 메시지를 확인합니다. 이 모드는 공급자 오류 후 최대 한 개의 재전송 메시지를 보장합니다.
- **CLIENT\_ACKNOWLEDGE.** 응용 프로그램이 메시지가 확인되는 시점을 제어합니다. 이전 확인 이후 해당 세션에서 처리된 모든 메시지가 확인됩니다. 일련의 확인을 처리하는 동안 메시지 서버가 실패하는 경우 해당 그룹에서 하나 이상의 메시지가 재전송될 수 있습니다.
- **DUPS\_OK\_ACKNOWLEDGE.** 이 모드는 시스템에게 메시지를 느리게 확인하도록 명령합니다. 공급자 오류 후 여러 메시지가 재전송될 수 있습니다.

(CLIENT\_ACKNOWLEDGE 모드를 사용하는 것은 처리 중에 공급자 오류가 발생하는 경우 모든 확인이 함께 처리되도록 보장하지 않는다는 점을 제외하고는 트랜잭션 사용과 유사합니다.)

확인 모드는 다음과 같은 이유로 성능에 영향을 미칩니다.

- **AUTO\_ACKNOWLEDGE**와 **CLIENT\_ACKNOWLEDGE** 모드에서는 브로커와 클라이언트 사이에 추가 제어 메시지가 필요합니다. 추가 제어 메시지는 처리 오버헤드를 추가하고 **JMS** 페이로드 메시지를 방해할 수 있으므로 처리가 지연됩니다.
- **AUTO\_ACKNOWLEDGE**와 **CLIENT\_ACKNOWLEDGE** 모드에서는 클라이언트가 추가 메시지를 사용할 수 있으려면 브로커가 클라이언트 확인을 처리했다고 확인할 때까지 대기해야 합니다. (이와 같은 브로커 확인은 브로커가 실수로 이 메시지를 재전송하지 않도록 보장합니다.)
- 사용자가 받은 모든 지속성 메시지에 대한 확인 정보로 **Message Queue** 영구 저장소를 업데이트해야 하므로 성능이 감소됩니다.

## 영구 가입 및 비영구 가입

주제 대상의 가입자는 영구 가입자와 비영구 가입자의 두 가지 범주로 구분됩니다.

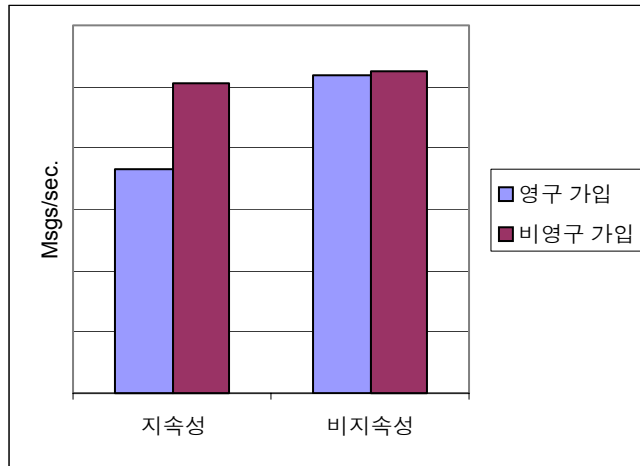
영구 가입은 다음과 같은 이유로 안정성이 향상되지만 처리 능력이 더 떨어집니다.

- **Message Queue** 메시지 서버는 메시지 서버에 오류가 발생하더라도 복구 후 목록을 사용할 수 있도록 각 영구 가입에 할당된 메시지 목록을 영구 저장해야 합니다.
- 메시지 서버에 오류가 발생하더라도 복구 후 해당 사용자가 활성화되면 메시지를 계속 전달할 수 있도록 영구 가입의 지속성 메시지가 영구적으로 저장됩니다. 반면 비영구 가입의 지속성 메시지는 영구 저장되지 않습니다. 메시지 서버에 오류가 발생하는 경우 해당 사용자 연결이 끊기며 메시지가 전달되지 않습니다.

그림 11-3은 지속성 및 비지속성 10k 크기 메시지의 두 가지 경우에서 영구 및 비영구 가입을 사용하는 주제 대상의 처리량을 비교합니다. 두 가지 경우 모두 AUTO\_ACKNOWLEDGE 확인 모드를 사용합니다.

그림 11-3에서 지속성 메시지인 경우에만 영구 가입 사용의 성능 영향이 확인한 것을 알 수 있으며 그 이유는 위에서 설명한 대로 지속성 메시지가 영구 가입에 대해서만 영구 저장되기 때문입니다.

그림 11-3 가입 유형의 성능 영향



## 선택기 사용(메시지 필터링)

응용 프로그램 개발자가 특정 사용자들을 메시지 집합의 대상으로 지정할 수 있습니다. 이는 메시지 집합마다 고유 물리적 대상을 지정하거나 단일 물리적 대상을 사용하여 각 사용자에게 대해 하나 이상의 선택기를 등록하여 수행할 수 있습니다.

선택기는 해당 문자열과 일치하는 등록 정보 값을 갖는 메시지만 특정 사용자에게 전달되도록 요청하는 문자열입니다. 예를 들어 선택기 `NumberOfOrders>1`은 `NumberOfOrders` 등록 정보 값이 2이상인 메시지만 전달합니다.

선택기를 사용하여 사용자를 등록하면 각 메시지를 처리하는 데 추가 처리가 필요하므로 여러 물리적 대상을 사용하는 것에 비해 성능이 떨어집니다. 선택기를 사용하는 경우 선택기가 이후의 메시지와 일치될 수 있도록 구문 분석되어야 합니다. 또한 각 메시지가 라우팅될 때 각 메시지의 메시지 등록 정보를 검색하고 선택기와 비교해야 합니다. 그러나 선택기를 사용하면 메시지 응용 프로그램의 융통성이 증가합니다.

## 메시지 크기

생성자 클라이언트에서 브로커로 그리고 브로커에서 사용자 클라이언트로 더 많은 데이터가 전달되어야 하고 지속성 메시지의 경우 더 큰 메시지를 저장해야 하므로 메시지 크기는 성능에 영향을 미칩니다.

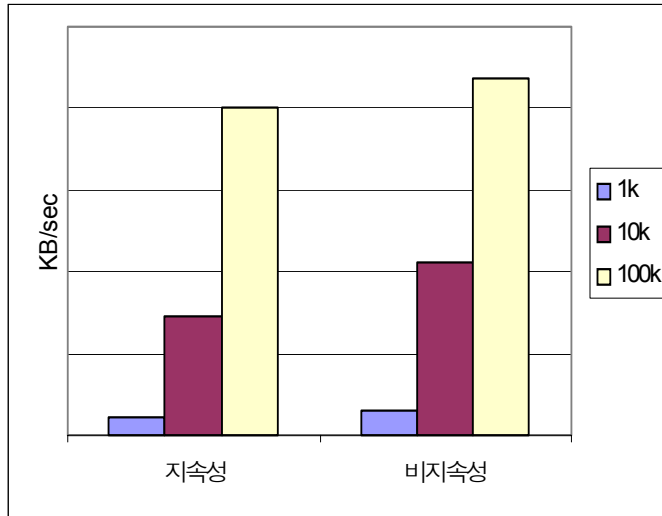
그러나 작은 메시지들을 단일 메시지로 일괄 처리하면 개별 메시지의 라우팅과 처리를 최소화하여 전체적 성능 향상을 제공할 수 있습니다. 이 경우 개별 메시지의 상태에 대한 정보는 손실됩니다.

**그림 11-4**는 지속성 및 비지속성 메시지의 두 가지 경우에서 1k, 10k 및 100k 크기의 메시지에 대한 처리량(초당 KB)을 비교합니다. 모든 경우에서 메시지를 대기열 대상으로 보내며 `AUTO_ACKNOWLEDGE` 확인 모드를 사용합니다.

**그림 11-4**는 두 경우 모두에서 작은 메시지보다 큰 메시지를 전달할 때 오버헤드가 적다는 것을 보여 줍니다. 또한 크기가 1k 및 10k인 메시지에서는 지속성 메시지가 비지속성 메시지보다 약 50%의 성능 향상을 보였지만 100k 크기의 메시지에서는 아마도 네트워크 대역폭 때문에 해당 경우의 메시지 처리량에 병목 현상이 발생하여 이러한 성능 향상이 유지되지 않았음을 알 수 있습니다.



그림 11-4 메시지 크기의 성능 영향



### 메시지 본문 유형

JMS는 복잡성 순서에 따라 아래에 대략적으로 표시된 다섯 개의 메시지 본문 유형을 지원합니다.

- **BytesMessage:** 응용 프로그램에서 지정하는 형식의 바이트 집합을 포함합니다.
- **TextMessage:** 단순한 `java.lang.String`입니다.
- **StreamMessage:** Java 프리미티브 값의 스트림을 포함합니다.
- **MapMessage:** 일련의 이름-값 쌍을 포함합니다.
- **ObjectMessage:** Java 일련화 객체를 포함합니다.

일반적으로 메시지 유형은 응용 프로그램의 필요에 따라 제어되지만 좀 더 복잡한 유형 (**MapMessage** 및 **ObjectMessage**)은 성능 저하(데이터 일련화 및 일련화 해제로 인한 저하)를 수반합니다. 성능 저하는 데이터의 단순성이나 복잡성에 따라 달라집니다.

## 성능에 영향을 미치는 메시지 서비스 요소

메시징 응용 프로그램의 성능은 응용 프로그램 설계뿐만 아니라 메시지 라우팅 및 전달을 수행하는 메시지 서비스의 영향도 받습니다.

다음 절에서는 성능에 영향을 미칠 수 있는 여러 메시지 서비스 요소를 설명합니다. 이러한 요소의 영향을 이해하는 것은 메시지 서비스 크기를 지정하고 배포된 응용 프로그램에서 발생할 수 있는 성능 병목 현상을 진단하고 해결하는 데 중요합니다.

Message Queue 서비스에서 성능에 영향을 미치는 가장 중요한 요소는 다음과 같습니다.

- 하드웨어
- 운영 체제
- Java 가상 머신(JVM)
- 연결
- 브로커 제한 및 동작
- 메시지 서비스 구조
- 데이터 저장소 성능
- 클라이언트 런타임 구성

아래의 절에서는 이러한 각 요소가 메시징 성능에 미치는 영향을 설명합니다.

### 하드웨어

Message Queue 메시지 서버나 클라이언트 응용 프로그램 모두 CPU 처리 속도와 사용 가능한 메모리가 메시지 서비스 성능의 주요 결정 요소입니다. 처리량을 높이면 많은 소프트웨어 제한 사항을 제거할 수 있고 메모리를 추가하면 처리 속도와 용량을 모두 늘릴 수 있습니다. 그러나 하드웨어 업그레이드만으로 병목 현상을 극복하는 것은 일반적으로 비용이 높습니다.

## 운영 체제

하드웨어 플랫폼이 같은 경우라도 각기 다른 운영 체제의 효율성으로 인해 성능은 다양할 수 있습니다. 예를 들어, 운영 체제에서 사용하는 스레드 모델은 메시지 서버가 지원할 수 있는 동시 연결 수에 중요한 영향을 미칠 수 있습니다. 일반적으로 모든 하드웨어가 동일한 경우 Solaris가 Linux보다 빠르며 Linux가 Windows보다 빠릅니다.

## Java 가상 머신(JVM)

메시지 서버는 호스트 JVM에서 실행되고 지원되는 Java 프로세스입니다. 결과적으로 JVM 처리는 메시지 서버가 얼마나 빠르고 효율적으로 메시지를 라우팅하고 전달할 수 있는지 결정하는 중요 요소입니다.

특히 JVM의 메모리 자원 관리는 중요할 수 있습니다. 메모리 로드 증가를 수용하기 위해 충분한 메모리를 JVM에 할당해야 합니다. 또한 JVM은 주기적으로 사용되지 않은 메모리를 재생 이용하며 이러한 메모리 재생 이용은 메시지 처리를 지연시킬 수 있습니다. JVM 메모리 힙이 클수록 메모리 재생 이용 중에 경험할 수 있는 잠재적 지연은 더 길어집니다.

## 연결

클라이언트와 브로커간 연결의 수와 속도는 메시지 서버가 처리할 수 있는 메시지 수와 메시지 전달 속도에 영향을 미칠 수 있습니다.

### 메시지 서버 연결 제한

메시지 서버에 대한 모든 액세스는 연결을 통해서 이루어집니다. 동시 연결 수에 대한 제한은 메시지 서버를 동시에 사용할 수 있는 생성자나 사용자 클라이언트 수에 영향을 미칠 수 있습니다.

일반적으로 메시지 서버에 대한 연결 수는 사용 가능한 스레드 수에 의해 제한됩니다. Message Queue는 전용 스레드 모델이나 공유 스레드 모델 중 하나를 지원하도록 구성할 수 있는 스레드 풀 관리자를 사용합니다(77페이지의 "스레드 풀 관리자" 참조).

전용 스레드 모델은 각 연결이 전용 스레드를 가지므로 아주 빠르지만 연결 수가 사용 가능한 스레드 수에 의해 제한됩니다(연결당 하나의 입력 스레드와 하나의 출력 스레드 필요). 공유 스레드 모델은 연결 수에 대한 제한이 없지만 많은 연결 사이에서 스레드를 공유하며 이러한 연결이 사용 중인 경우에는 상당한 오버헤드와 처리량 지연이 발생합니다.

### 전송 프로토콜

Message Queue 소프트웨어를 사용하면 클라이언트가 다양한 저급 전송 프로토콜을 사용하여 메시지 서버와 통신할 수 있습니다. Message Queue는 75페이지의 "연결 서비스"에 설명된 연결 서비스와 해당 프로토콜을 지원합니다.

프로토콜은 응용 프로그램 요구 사항(암호화, 방화벽을 통한 액세스 가능)을 기반으로 선택하지만 이 선택은 전체 성능에 영향을 미칩니다.

그림 11-5 전송 프로토콜 속도



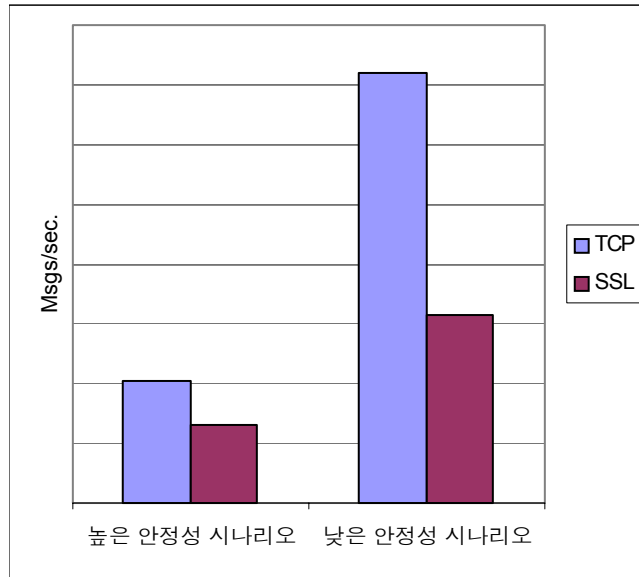
그림 11-5는 다양한 프로토콜 기술의 성능 특성을 반영합니다.

- TCP는 브로커와 통신할 수 있는 가장 빠른 방법을 제공합니다.
- SSL은 메시지를 보내고 받을 때 TCP보다 50-70퍼센트 더 느립니다(지속성 메시지의 경우 50퍼센트, 비지속성 메시지의 경우 70퍼센트에 근접). 또한 클라이언트와 브로커(또는 HTTPS의 경우 웹 서버)가 전송할 데이터를 암호화할 때 사용될 개인 키를 설정해야 하므로 SSL을 사용하면 초기 연결 설정이 더 느려집니다(몇 초 걸릴 수 있음). 각 저급 TCP 패킷을 암호화하고 해독하는 데 필요한 추가 처리에 의해 성능이 저하됩니다.

그림 11-6은 높은 안정성 시나리오(영구 가입이 있고 AUTO\_ACKNOWLEDGE 확인 모드를 사용하는 주제 대상에 1k의 지속성 메시지를 보냄)와 높은 성능 시나리오(영구 가입이 없고 DUPES\_OK\_ACKNOWLEDGE 확인 모드를 사용하는 주제 대상에 1k의 비지속성 메시지를 보냄)의 두 가지 경우에 대해 TCP와 SSL의 처리 능력을 비교합니다.

그림 11-6은 프로토콜이 높은 안정성 사례에서 영향을 덜 미침을 보여 줍니다. 이는 아마도 높은 안정성 사례에서 필요한 지속성 오버헤드가 처리량을 제한하는 데 프로토콜 속도보다 더 중요한 요소이기 때문일 것입니다.

그림 11-6 전송 프로토콜의 성능 영향



- HTTP는 TCP나 SSL보다 느립니다. HTTP는 웹 서버에서 실행되는 서블릿을 클라이언트와 브로커 사이의 프록시로 사용합니다. 성능 오버헤드는 HTTP 요청의 패킷 캡슐화와 메시지가 브로커에 도달하기 위해 두 개의 홉(클라이언트에서 서블릿으로, 서블릿에서 브로커로)을 통과해야 한다는 점과 관련됩니다.
- 클라이언트와 서블릿 사이 그리고 서블릿과 브로커 사이에서 패킷을 암호화하는 데 필요한 추가 오버헤드로 인해 HTTPS는 HTTP보다 더 느립니다.

### 메시지 서비스 구조

Message Queue 메시지 서버는 단일 브로커로 구현되거나 다중 상호 연결 브로커 인스턴스, 즉 브로커 클러스터로 구현될 수 있습니다.

브로커에 연결된 클라이언트 수가 늘어나고 전달되는 메시지 수가 늘어나면 파일 설명자, 스레드, 메모리 제한 같은 브로커의 자원 제한 사항이 초과하게 됩니다. 늘어나는 로드를 수용하는 한 가지 방법은 Message Queue 메시지 서버에 브로커 인스턴스를 추가하여 클라이언트 연결과 메시지 라우팅 및 전달을 여러 브로커에 걸쳐 분산시키는 것입니다.

일반적으로 이러한 확장은 클라이언트, 특히 메시지 생성자 클라이언트가 클러스터 전체에 균등하게 분산되어 있는 경우에 가장 잘 작동합니다. 클러스터에 있는 브로커 사이의 메시지 전달과 관련된 오버헤드로 인해 제한된 연결 수나 메시지 전달 비율을 갖는 클러스터는 단일 브로커보다 더 낮은 성능을 보일 수 있습니다.

또한 브로커 클러스터를 사용하여 네트워크 대역폭을 최적화할 수 있습니다. 예를 들어, 클러스터 내의 원격 브로커들 사이에는 느린 장거리 네트워크 링크를 사용하고 클라이언트와 해당 브로커 인스턴스의 연결에는 고속 링크를 사용할 수 있습니다.

클러스터에 대한 자세한 내용은 [9장](#), "[브로커 클러스터 작업](#)"을 참조하십시오.

## 브로커 제한 및 동작

메시지 서버가 처리해야 하는 메시지 처리량은 메시지 서버가 지원하는 메시징 응용 프로그램의 사용 패턴에 달려 있습니다. 하지만 메시지 서버는 메모리, CPU 사이클 등 자원이 제한되어 있습니다. 따라서 메시지 서버가 넘치게 되어 응답하지 않거나 불안정하게 될 수 있습니다.

Message Queue 메시지 서버에는 메모리 자원을 관리하고 브로커의 메모리 부족을 방지하기 위해 기본적으로 제공되는 메커니즘이 있습니다. 이러한 메커니즘은 브로커나 개별 물리적 대상이 보유할 수 있는 메시지 수나 메시지 바이트에 대한 구성 가능한 제한을 포함하며 물리적 대상 제한에 이르면 시작될 수 있는 동작 집합을 포함합니다.

이러한 구성 가능한 메커니즘은 세밀하게 모니터링하고 조정하면 메시지 유입과 유출의 균형을 유지하여 시스템 과부하가 발생할 수 없도록 하는 데 사용할 수 있습니다. 이 메커니즘은 오버헤드를 사용하고 메시지 처리량을 제한할 순 있지만 운영 무결성을 유지합니다.

## 데이터 저장소 성능

Message Queue는 기본 제공 지속성과 플러그인 지속성을 모두 지원합니다. 기본 제공 지속성은 파일 기반 데이터 저장소입니다. 플러그인 지속성은 JDBC™ (Java Database Connectivity) 인터페이스를 사용하며 JDBC 호환 데이터 저장소가 필요합니다.

기본 제공 지속성이 플러그인 지속성보다 상당히 빠르지만 JDBC 호환 데이터베이스 시스템은 응용 프로그램에 필요한 중복, 보안 및 관리 기능을 제공할 수 있습니다.

기본 제공 지속성의 경우 지속성 작업이 메모리 상태에서 데이터 저장소와 동기화되도록 지정하여 안정성을 최대화할 수 있습니다. 이렇게 하면 시스템 중단으로 인한 데이터 손실을 방지할 수 있지만 성능은 저하됩니다.

## 클라이언트 런타임 구성

Message Queue 클라이언트 런타임은 클라이언트 응용 프로그램에 Message Queue 메시지 서비스에 대한 인터페이스를 제공합니다. 클라이언트가 물리적 대상에게 메시지를 보내고 이러한 대상으로부터 메시지를 받는 데 필요한 모든 작업을 지원합니다. 클라이언트 런타임은 구성 가능하므로(연결 팩토리 속성 값을 설정하여) 일반적으로 성능과 메시지 처리량을 향상시킬 수 있는 등록 정보와 동작을 설정할 수 있습니다.

예를 들어 Message Queue 클라이언트 런타임은 다음과 같은 구성 가능한 동작을 지원합니다.

- JMS 메시지와 Message Queue 제어 메시지가 동일한 연결을 통해 흐르기 때문에 발생하는 혼잡을 막는 데 도움을 주는 연결 흐름 측정(imqConnectionFactoryCount)
- 클라이언트 런타임 연결을 통해 전달될 수 있는 메시지 수와 사용 대기 중인 메시지 수를 제한하여 클라이언트 자원 제한을 피하는 데 도움을 주는 연결 흐름 제한(imqConnectionFactoryLimit)
- 다중 사용자 대기열 전달의 경우 사용자 간의 로드 균형 조정을 향상시킬 수 있고(한 사용자에게 보내는 메시지 수의 불균형 방지), 연결된 한 사용자가 다른 연결된 사용자에게 과도하게 메시지를 보내는 것을 방지하는 데 도움을 주는 사용자 흐름 제한(imqConsumerFlowLimit). 이 등록 정보는 클라이언트 런타임 연결을 통해 전달될 수 있고 사용 대기 중인 사용자당 메시지 수를 제한합니다. 또한 이 등록 정보는 대기열 대상 등록 정보(consumerFlowLimit)로 구성될 수 있습니다.

이러한 동작들과 그 구성에 사용되는 속성에 대한 자세한 내용은 [244페이지의 "클라이언트 런타임 메시지 흐름 조정"](#)을 참조하십시오.

# 성능 향상을 위한 구성 조정

## 시스템 조정

다음 절에서는 운영 체제, JVM 및 통신 프로토콜에서 조정할 수 있는 내용을 설명합니다.

### Solaris 조정: CPU 사용률, 페이징/스왑/디스크 입출력

운영 체제 조정에 대해서는 시스템 설명서를 참조하십시오.

## Java 가상 머신 조정

기본적으로 브로커는 192MB의 JVM 힙을 사용합니다. 이 크기는 대량의 메시지 로드를 처리하기에 너무 작은 경우가 많으므로 늘리는 것이 좋습니다.

브로커가 Java 객체에서 사용되는 JVM 힙 공간을 소진하는 데 이르면 브로커는 흐름 제어, 메시지 스왑 등 여러 기술을 사용하여 메모리를 확보합니다. 아주 드물게 메모리를 확보하고 메시지 유입을 줄이기 위해 클라이언트 연결을 닫는 경우도 있습니다. 따라서, 이러한 경우가 발생하지 않도록 최대 JVM 힙 공간을 충분하게 설정하는 것이 좋습니다.

하지만, 최대 Java 힙 공간을 너무 높게 설정하면 브로커가 전체 시스템 메모리가 부족해질 때까지 Java 힙 공간을 계속해서 증가시킬 수 있습니다. 그렇게 되면 성능이 감소하거나, 예상치 않은 브로커 충돌이 발생하거나, 시스템에서 실행 중인 다른 응용 프로그램 및 서비스의 동작에 영향을 미칠 수 있습니다. 일반적으로 운영 체제 및 시스템에서 실행할 다른 응용 프로그램이 충분한 물리적 메모리를 사용할 수 있도록 해야 합니다.

일반적으로 정상 시스템 메모리와 최대 시스템 메모리를 평가하여 시스템 메모리 문제를 일으키지 않고 우수한 성능을 제공할 수 있도록 Java 힙 크기를 구성하는 것이 좋습니다.

브로커의 최소 및 최대 힙 크기를 변경하려면 브로커를 시작할 때 `-vmargs` 명령줄 옵션을 사용합니다. 예를 들면 다음과 같습니다.

```
/usr/bin/imqbrokerd -vmargs "-Xms256m -Xmx1024m"
```

이 명령은 시작 Java 힙 크기를 256MB로 설정하고 최대 Java 힙 크기를 1GB로 설정합니다.

- Solaris 또는 Linux에서 `/etc/rc*`(즉, `/etc/init.d/imq`)를 통해 브로커를 시작하는 경우 `/etc/imq/imqbrokerd.conf`(Solaris) 또는 `/etc/opt/sun/mq/imqbrokerd.conf`(Linux) 파일에 브로커 명령줄 인수를 지정합니다. 자세한 내용은 해당 파일의 주석을 참조하십시오.
- Windows에서 브로커를 Windows 서비스로 시작하는 경우 `imqsvcadm install` 명령에 `-vmargs` 옵션을 사용하여 JVM 인수를 지정합니다. 13장, "명령 참조"의 "`imqsvcadm`"을 참조하십시오.

어떤 경우든 브로커 로그 파일을 확인하거나

```
imqcmd metrics bkr -m cxn
```

 명령을 사용하여 설정을 확인합니다.

## 전송 프로토콜 조정

응용 프로그램 요구에 맞는 프로토콜을 선택했으면 선택한 프로토콜을 기반으로 추가 조정하여 성능을 향상시킬 수 있습니다.



다음 세 가지 브로커 등록 정보를 사용하여 프로토콜의 성능을 수정할 수 있습니다.

- `imq.protocol.protocol_type.nodelay`
- `imq.protocol.protocol_type.inbufsz`
- `imq.protocol.protocol_type.outbufsz`

TCP 및 SSL 프로토콜의 경우 이러한 등록 정보는 클라이언트와 브로커 사이의 메시지 전달 속도에 영향을 미칩니다. HTTP 및 HTTPS 프로토콜의 경우 이 등록 정보는 웹 서버에서 실행 중인 Message Queue 터널 서블릿과 브로커 사이의 메시지 전달 속도에 영향을 미칩니다. HTTP/HTTPS 프로토콜의 경우 성능에 영향을 미칠 수 있는 추가 등록 정보가 있습니다(241 페이지의 "[HTTP/HTTPS 조정](#)" 참조).

프로토콜 조정 등록 정보에 대해서는 다음 절에서 설명합니다.

### *nodelay*

`nodelay` 등록 정보는 지정된 프로토콜의 Nagle 알고리즘(TCP/IP에서 TCP\_NODELAY 소켓 수준 옵션 값)에 영향을 미칩니다. Nagle 알고리즘은 WAN(Wide-Area Network) 같은 느린 연결을 사용하는 시스템에서 TCP 성능을 향상시키는 데 사용됩니다.

이 알고리즘이 사용되면 TCP는 여러 개의 작은 데이터 청크가 원격 시스템으로 보내지지 않도록 데이터를 큰 패킷으로 묶으려고 합니다. 소켓에 기록된 데이터가 필요한 버퍼 크기를 채우지 못하는 경우 프로토콜은 버퍼가 채워지거나 특정 지연 시간이 경과될 때까지 패킷 전송을 지연합니다. 버퍼가 채워지거나 시간 초과가 발생하면 패킷을 보냅니다.

대부분의 메시징 응용 프로그램에서 패킷을 지연 없이 보내는 경우(Nagle 알고리즘을 사용하지 않는 경우) 최고의 성능을 냅니다. 이는 클라이언트와 브로커 사이의 상호 작용이 대부분 클라이언트가 데이터 패킷을 브로커에게 보내고 응답을 기다리는 요청/응답 상호 작용이기 때문입니다. 예를 들어, 일반적인 상호 작용에는 다음이 포함됩니다.

- 연결 만들기
- 생성자 또는 사용자 만들기
- 지속성 메시지 보내기(브로커가 메시지 수신 확인)
- `AUTO_ACKNOWLEDGE` 또는 `CLIENT_ACKNOWLEDGE` 세션에서 클라이언트 확인 보내기(브로커가 확인 처리 확인)

이러한 상호 작용에서 대부분의 패킷은 버퍼 크기보다 작습니다. 이는 Nagle 알고리즘이 사용되는 경우 브로커가 사용자에게 응답을 보내기 전에 몇 밀리초를 지연함을 의미합니다.

하지만 Nagle 알고리즘은 연결이 느리고 브로커 응답이 필요하지 않은 경우 성능을 향상시킬 수 있습니다. 이런 것으로는 클라이언트가 비지속성 메시지를 보내는 경우나 클라이언트 확인을 브로커에서 확인하지 않는 경우(DUPS\_OK\_ACKNOWLEDGE 세션)를 들 수 있습니다.

### *inbufsz/outbufsz*

inbufsz 등록 정보는 소켓에서 들어오는 데이터를 읽는 입력 스트림의 버퍼 크기를 설정합니다. 마찬가지로 outbufsz는 데이터를 소켓에 기록하기 위해 브로커가 사용하는 출력 스트림의 버퍼 크기를 설정합니다.

일반적으로 두 매개 변수 모두 보내거나 받는 평균 패킷보다 약간 큰 값으로 설정해야 합니다. 경험에 따르면 이 등록 정보 값을 평균 패킷에 1k를 더한 크기(가장 가까운 k 값으로 반올림)로 설정하는 것이 좋습니다.

예를 들어, 브로커가 본문 크기가 1k인 패킷을 받는 경우 패킷의 전체 크기(메시지 본문 + 헤더 + 등록 정보)는 약 1200바이트입니다. inbufsz가 2k(2048바이트)인 경우 적당한 성능을 제공합니다.

inbufsz나 outbufsz를 이 크기보다 더 키우면 성능은 약간 늘어날 수 있지만 각 연결에 필요한 메모리가 늘어나게 됩니다.

그림 11-7은 1k 패킷에 대해 inbufsz를 변경할 때의 결과를 보여 줍니다.

**그림 11-7** 1k(1024바이트) 패킷에 대한 inbufsz 변경의 결과

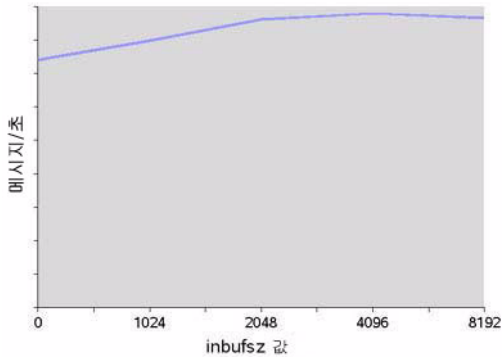
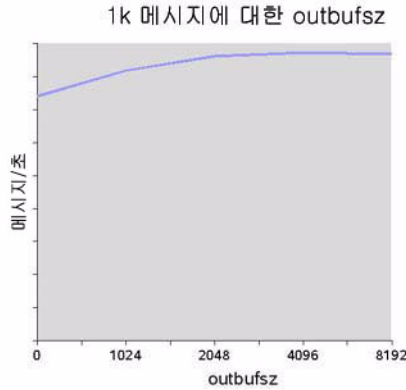


그림 11-8은 1k 패킷에 대해 outbufsz를 변경할 때의 결과를 보여 줍니다.

그림 11-8 1k(1024바이트) 패킷에 대한 `outbufsz` 변경의 결과



### HTTP/HTTPS 조정

앞의 두 절에서 설명한 일반 등록 정보 이외에도 HTTP/HTTPS 성능은 클라이언트가 Message Queue 터널 서버를 호스트하는 웹 서버에 대해 HTTP 요청을 얼마나 빨리 보낼 수 있는지에 따라 제한됩니다.

단일 소켓에서 다중 요청을 처리하도록 웹 서버를 최적화해야 할 수 있습니다. JDK 버전 1.4 이상에서는 웹 서버에 대한 HTTP 연결이 유지되어(웹 서버 소켓이 열린 상태로 유지) 다중 HTTP 요청을 처리할 때 웹 서버에서 사용되는 자원을 최소화합니다. JDK 버전 1.4를 사용하는 클라이언트 응용 프로그램의 성능이 이전 JDK 릴리스로 실행 중인 같은 응용 프로그램보다 느린 경우 성능을 향상시키기 위해 웹 서버의 연결 유지 구성 매개 변수를 조정해야 할 수 있습니다.

이러한 웹 서버 조정과 더불어 클라이언트가 웹 서버를 풀하는 간격도 조정할 수 있습니다. HTTP는 요청 기반 프로토콜입니다. 따라서 HTTP 기반 프로토콜을 사용하는 클라이언트는 웹 서버에서 메시지가 대기 중인지 주기적으로 확인해야 합니다.

`imq.httpjms.http.pullPeriod` 브로커 등록 정보(및 해당

`imq.httpsjms.https.pullPeriod` 등록 정보)는 Message Queue 클라이언트 런타임이 웹 서버를 풀하는 간격을 지정합니다.

`pullPeriod` 값이 -1(기본값)인 경우 클라이언트 런타임은 이전 요청이 반환되자마자 서버를 풀하여 개별 클라이언트의 성능을 최대화합니다. 따라서 각 클라이언트 연결이 웹 서버의 요청 스레드를 독점하여 웹 서버 자원이 고갈될 수 있습니다.

pullPeriod 값이 양수인 경우 클라이언트 런타임은 주기적으로 웹 서버에 요청을 보내 보류 중인 데이터가 있는지 확인합니다. 이 경우 클라이언트가 웹 서버의 요청 스레드를 독점하지 않습니다. 따라서 많은 수의 클라이언트가 웹 서버를 사용 중인 경우 pullPeriod를 양수로 설정하면 웹 서버 자원을 절약할 수 있습니다.

## 파일 기반 영구 저장소 조정

파일 기반 영구 저장소 조정에 대한 자세한 내용은 83페이지의 "[지속성 관리자](#)"를 참조하십시오.

## 브로커 조정

다음 절에서는 브로커 등록 정보를 조정하여 성능을 향상시키는 방법에 대해 설명합니다.

### 메모리 관리: 로드 하에서 브로커 안정성 증가

대상별 또는 시스템 전체 수준(모든 대상을 묶어서)에서 메모리 관리를 구성할 수 있습니다.

#### *물리적 대상 제한 사용*

물리적 대상 제한에 대한 자세한 내용은 6장, "[물리적 대상 관리](#)"를 참조하십시오.

#### *시스템 전체 제한 사용*

메시지 생성자가 메시지 사용자보다 넘치는 경향이 있는 경우 메시지가 브로커에 누적될 수 있습니다. 브로커에는 메모리가 부족한 경우 생성자를 억제하고 활성 메모리로부터 메시지를 스왑하는 메커니즘이 포함되어 있지만 브로커가 보관할 수 있는 전체 메시지 수와 메시지 바이트에 대해 엄격한 제한을 설정하는 것이 좋습니다.

imq.system.max\_count 및 imq.system.max\_size 브로커 등록 정보를 설정하여 이러한 제한을 제어합니다.

예를 들면 다음과 같습니다.

```
imq.system.max_count=5000
```

위에서 정의한 값은 브로커가 전달되지 않은/확인되지 않은 메시지를 5000개까지만 보관함을 의미합니다. 추가로 보내는 메시지는 브로커에서 거부됩니다. 메시지가 지속성인 경우 생성자가 메시지를 보내려고 하면 예외가 발생합니다. 메시지가 비지속성인 경우 브로커는 메시지를 자동으로 삭제합니다.

비지속성 메시지가 지속성 메시지처럼 예외를 반환하게 하려면 클라이언트가 사용하는 연결 팩토리 객체에 대해 다음 등록 정보를 설정합니다.

```
imqAckOnProduce = true
```

위의 설정은 비지속성 메시지를 브로커로 보낼 때의 성능을 저하시킬 수 있지만(클라이언트가 다음 메시지를 보내기 전에 응답을 기다림) 브로커로의 메시지 유입은 일반적으로 시스템 병목 현상이 아니므로 종종 이 설정이 허용되기도 합니다.

메시지를 보낼 때 예외가 반환되면 클라이언트는 잠시 동안 일시 중지하고 다시 보내 보아야 합니다.

## 다중 사용자 대기열 성능

여러 대기열 사용자가 단일 대기열 대상에서 메시지를 처리하는 효율성은 다음과 같은 구성 가능한 대기열 대상 속성에 따라 달라집니다.

- 활성 사용자 수(maxNumActiveConsumers)
- 일괄적으로 사용자에게 전달될 수 있는 최대 메시지 수(consumerFlowLimit)

최적의 메시지 처리량을 달성하려면 대기열의 메시지 생성 속도에 부합하는 충분한 활성 사용자의 수가 있어야 하며 대기열의 메시지를 그 사용 속도를 최대화할 수 있는 방식으로 라우팅한 다음 활성 사용자에게 전달해야 합니다. 여러 사용자 간에 메시지 전달의 균형을 조정하기 위한 일반 메커니즘에 대해서는 *Sun Java System Message Queue 기술 개요*에서 설명합니다.

메시지가 대기열에 누적되고 있는 경우 활성 사용자 수가 메시지 로드를 처리하기에 충분하지 않을 수 있습니다. 또는 사용자에서 메시지 정체를 일으키는 일괄 처리 크기로 메시지가 사용자에게 전달되고 있을 수 있습니다. 예를 들어 일괄 처리 크기(consumerFlowLimit)가 너무 큰 경우 한 사용자가 대기열의 모든 메시지를 받는 동안 다른 활성 사용자는 메시지를 전혀 받지 못할 수 있습니다. 사용자가 아주 빠른 경우 이것은 문제가 되지 않을 수 있습니다.

하지만 사용자가 비교적 느린 경우 메시지를 사용자에게 균등하게 분산시켜야 하므로 일괄 처리 크기가 작은 게 좋습니다. 일괄 처리 크기가 작을수록 메시지를 사용자에게 전달하는 데 더 많은 오버헤드가 필요합니다. 그럼에도 불구하고 느린 사용자의 경우 작은 일괄 처리 크기를 사용하는 것이 결과적으로 성능 향상이 있습니다.

## 클라이언트 런타임 메시지 흐름 조정

이 절에서는 성능에 영향을 미치는 흐름 제어 동작을 설명합니다(237페이지의 "클라이언트 런타임 구성" 참조). 이러한 동작은 연결 팩토리 관리 대상 객체의 속성으로 구성됩니다. 연결 팩토리 속성 설정에 대한 자세한 내용은 8장, "관리 대상 객체 관리"를 참조하십시오.

### 메시지 흐름 측정

클라이언트가 보내고 받는 메시지(JMS 메시지)와 Message Queue 제어 메시지는 같은 클라이언트-브로커 연결을 통해 전달됩니다. 제어 메시지가 JMS 메시지 전달로 인해 일시 중단되면 브로커 확인과 같은 제어 메시지 전달이 지연될 수 있습니다. 이러한 유형의 혼잡을 막기 위해 Message Queue는 연결에서 JMS 메시지의 흐름을 측정합니다.

JMS 메시지는 `imqConnectionFactory` 등록 정보에서 지정한 대로 설정된 수만 전달되도록 일괄 처리됩니다. 일괄 처리가 전달되면 JMS 메시지 전달이 일시 중지되고 보류 중인 제어 메시지만 전달됩니다. JMS 메시지의 다른 일괄 처리가 전달된 다음 보류 중인 제어 메시지가 전달되는 식으로 이러한 주기가 반복됩니다.

클라이언트가 브로커에 많은 응답을 요구하는 작업을 수행 중인 경우(예: 클라이언트가 `CLIENT_ACKNOWLEDGE` 또는 `AUTO_ACKNOWLEDGE` 모드, 지속성 메시지, 트랜잭션, 대기열 브라우저를 사용하고 있거나 클라이언트가 사용자를 추가 또는 제거하고 있는 경우) `imqConnectionFactory` 값을 낮게 유지해야 합니다. 반면 클라이언트에 연결에서 `DUPS_OK_ACKNOWLEDGE` 모드를 사용하는 단순 사용자만 있는 경우 성능을 저하시키지 않고 `imqConnectionFactory`를 늘릴 수 있습니다.

### 메시지 흐름 제한

메모리와 같은 로컬 자원 제한이 발생하기 전에 Message Queue 클라이언트 런타임이 처리할 수 있는 JMS 메시지 수에 대한 제한이 있습니다. 이 제한에 가까워지면 성능에 악영향을 줍니다. 따라서 Message Queue에서는 연결을 통해 전달되고 클라이언트 런타임에 버퍼링되어 사용 대기 중일 수 있는 사용자당 메시지(또는 연결당 메시지) 수를 제한할 수 있습니다.

#### 사용자 기반 제한

클라이언트 런타임에 전달된 JMS 메시지 수가 사용자의 `imqConsumerFlowLimit` 값을 초과하면 해당 사용자에 대해 메시지 전달이 중지됩니다. 해당 사용자의 사용되지 않은 메시지 수가 `imqConsumerFlowThreshold`에 설정된 값 이하로 떨어져야만 메시지 전달이 재개됩니다.

다음은 이러한 제한의 사용을 보여 주는 예입니다. 주제 사용자의 기본 설정을 고려해 보십시오.

```
imqConsumerFlowLimit=1000
imqConsumerFlowThreshold=50
```

사용자가 만들어지면 브로커는 초기 일괄 처리인 1000개의 메시지(있는 경우)를 일시 중지 없이 이 사용자에게 전달합니다. 1000개의 메시지를 보낸 다음 브로커는 클라이언트 런타임이 추가 메시지를 요청할 때까지 전달을 중지합니다. 클라이언트 런타임은 응용 프로그램이 메시지를 처리할 때까지 이 메시지들을 보관합니다. 그런 다음 클라이언트 런타임은 브로커에게 다음 일괄 처리를 보내도록 요청하기 전에 응용 프로그램이 적어도 메시지 버퍼 용량(imqConsumerFlowThreshold)의 50%(즉, 500개의 메시지)를 사용할 수 있도록 합니다.

같은 상황에서 임계값이 10%인 경우 클라이언트 런타임은 다음 일괄 처리를 요청하기 전에 응용 프로그램이 최소 900개의 메시지를 사용할 때까지 기다립니다.

다음 일괄 처리 크기는 다음과 같이 계산됩니다.

$imqConsumerFlowLimit - (\text{버퍼에서 현재 보유 중인 메시지 수})$

따라서 imqConsumerFlowThreshold가 50%인 경우 다음 일괄 처리 크기는 응용 프로그램의 메시지 처리 속도에 따라 500과 1000 사이에서 변동될 수 있습니다.

imqConsumerFlowThreshold가 너무 높게 설정된 경우(100% 가까이) 브로커는 더 작은 일괄 처리를 보내는 경향이 있으므로 메시지 처리량이 떨어질 수 있습니다. 이 값이 너무 낮게 설정된 경우(0% 가까이) 브로커가 다음 집합을 전달하기 전에 클라이언트가 버퍼링된 나머지 메시지를 모두 처리할 수 있으므로 메시지 처리량이 저하됩니다. 일반적으로 특정 성능이나 안정성 문제가 있는 경우가 아니라면 imqConsumerFlowThreshold 속성의 기본 값을 변경할 필요가 없습니다.

사용자 기반 흐름 제어(특히 imqConsumerFlowLimit)는 클라이언트 런타임의 메모리 관리에 가장 좋은 방법입니다. 일반적으로 클라이언트 응용 프로그램에 따라 연결에서 지원해야 할 사용자 수, 메시지 크기 및 클라이언트 런타임에서 사용할 수 있는 전체 메모리 양에 대해 알고 있습니다.

### **연결 기반 제한**

하지만 일부 클라이언트 응용 프로그램의 경우 최종 사용자의 선택에 따라 사용자 수가 불확실할 수 있습니다. 이러한 경우에도 연결 수준 흐름 제한을 사용하여 메모리를 관리할 수 있습니다.

연결 수준 흐름 제어는 연결의 모든 사용자에게 대해 버퍼링되는 전체 메시지 수를 제한합니다. 이 수가 `imqConnectionFlowLimit`를 초과하면 전체 수가 연결 제한 아래로 떨어질 때까지 해당 연결을 통한 메시지 전달이 중지됩니다(`imqConnectionFlowLimitEnabled` 등록 정보를 `true`로 설정하는 경우에만 `imqConnectionFlowLimit`를 사용할 수 있음).

세션의 대기열에 있는 메시지 수는 각 사용자의 메시지 로드와 세션을 사용하는 메시지 사용자 수의 함수입니다. 클라이언트가 메시지를 생성하거나 사용할 때 지연을 나타내는 경우 일반적으로 응용 프로그램을 재설계하여 메시지 생성자와 사용자를 여러 세션에 분산시키거나 세션을 여러 연결에 분산시킴으로써 성능을 향상시킬 수 있습니다.



## 문제 해결

이 장에서는 다음과 같은 문제를 이해하고 해결하는 방법을 설명합니다.

- 248페이지의 "클라이언트가 연결을 설정할 수 없음"
- 253페이지의 "연결 처리 속도가 너무 느림"
- 255페이지의 "클라이언트가 메시지 생성자를 만들 수 없음"
- 256페이지의 "메시지 생성이 지연되거나 느림"
- 259페이지의 "메시지가 백로그됨"
- 264페이지의 "메시지 서버 처리 능력이 일정하지 않음"
- 265페이지의 "메시지가 사용자에게 도달하지 않음"
- 269페이지의 "사용 불능 메시지 대기열에 메시지가 포함되어 있음"

문제가 발생할 경우 설치된 **Message Queue** 소프트웨어의 버전 번호를 확인하는 것이 좋습니다. 버전 번호를 사용하여 사용 중인 설명서 버전이 소프트웨어 버전과 일치하는 지 확인합니다. 문제를 Sun에 보고할 때도 버전 번호가 필요합니다. 버전 번호를 확인하려면 다음 명령을 실행합니다.

```
imqcmd -v
```

## 클라이언트가 연결을 설정할 수 없음

이 문제의 증상은 다음과 같습니다.

- 클라이언트가 새 연결을 설정할 수 없습니다.
- 클라이언트가 실패한 연결을 자동으로 다시 연결할 수 없습니다.

이 절에서는 다음과 같은 가능한 원인에 대해 설명합니다.

- 클라이언트 응용 프로그램이 연결을 닫을 수 없어 연결 수가 자원 제한을 초과합니다.
- 브로커가 실행 중이지 않거나 네트워크 연결에 문제가 있습니다.
- 연결 서비스가 비활성 상태이거나 일시 중지되어 있습니다.
- 필요한 연결 수에 비해 사용 가능한 스레드 수가 너무 적습니다.
- Solaris나 Linux 운영 체제에서 필요한 연결 수에 비해 파일 설명자가 너무 적습니다.
- TCP 백로그가 동시에 설정할 수 있는 새 연결 요청 수를 제한합니다.
- 운영 체제가 동시 연결 수를 제한합니다.
- 사용자의 인증 또는 권한 부여가 실패합니다.

**클라이언트 응용 프로그램이 연결을 닫을 수 없어 연결 수가 자원 제한을 초과합니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

브로커에 대한 연결을 모두 나열합니다.

```
imqcmd list cxn
```

출력에 모든 연결과 각 연결이 설정된 호스트가 나열되고 특정 클라이언트에 비정상적인 수의 연결이 열려 있는 것으로 표시됩니다.

문제를 해결하려면 다음 작업을 수행합니다.

문제가 있는 클라이언트를 다시 작성하여 사용되지 않는 연결을 닫습니다.

**브로커가 실행 중이지 않거나 네트워크 연결에 문제가 있습니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

- 브로커의 기본 포트(예: 기본값 7676)로 텔넷하고 브로커가 포트 매퍼 출력으로 응답하는지 확인합니다.
- 브로커 프로세스가 호스트에서 실행 중인지 확인합니다.

문제를 해결하려면 다음 작업을 수행합니다.

- 브로커를 시작합니다.
- 네트워크 연결 문제를 해결합니다.

**연결 서비스가 비활성 상태이거나 일시 중지되어 있습니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

모든 연결 서비스의 상태를 확인합니다.

```
imqcmd list svc
```

연결 서비스 상태가 unknown 또는 paused로 표시되어 있는 경우 클라이언트가 해당 서비스를 사용하여 연결을 설정할 수 없습니다.

문제를 해결하려면 다음 작업을 수행합니다.

- 연결 서비스의 상태가 unknown으로 표시되어 있는 경우 해당 연결 서비스가 활성 서비스 목록(imq.service.active)에 없는 것입니다. SSL 기반 서비스의 경우 서비스가 잘못 구성되어 있을 수도 있습니다. 이로 인해 브로커가 브로커 로그에 ERROR [B3009]: Unable to start service ssljms: [B4001]: Unable to open protocol tls for ssljms service... 라는 항목과 이 예외의 근본 원인에 대한 설명을 포함시키게 됩니다.

SSL 서비스를 올바르게 구성하려면 [159페이지](#)의 "SSL 기반 서비스 작업"을 참조하십시오.

- 연결 서비스의 상태가 paused로 표시되어 있는 경우 서비스를 다시 시작합니다([120페이지](#)의 "연결 서비스 일시 중지 및 다시 시작" 참조).

**필요한 연결 수에 비해 사용 가능한 스레드 수가 너무 적습니다.**  
문제의 원인을 확인하려면 다음 작업을 수행합니다.

브로커 로그에서 다음 항목이 있는지 확인합니다.

```
WARNING [B3004]: No threads are available to process a new connection on
service ... Closing the new connection.
```

또한 다음 형식 중 하나를 사용하여 연결 서비스의 연결 수와 현재 사용 중인 스레드 수를 확인합니다.

```
imqcmd query svc -n serviceName
imqcmd metrics svc -n serviceName -m cxn
```

각 연결에는 받는 메시지와 보내는 메시지에 하나씩 두 개의 스레드가 필요합니다(77페이지의 "스레드 풀 관리자" 참조).

문제를 해결하려면 다음 작업을 수행합니다.

- 전용 스레드 풀 모델을 사용하고 있는 경우(`imq.service_name.threadpool_model=dedicated`), 최대 연결 수는 스레드 풀의 최대 스레드 수의 반입니다. 따라서 연결 수를 늘리려면 스레드 풀의 크기를 늘리거나 (`imq.service_name.max_threads`) 공유 스레드 풀 모델로 전환합니다.
- 공유 스레드 풀 모델을 사용 중인 경우(`imq.service_name.threadpool_model=shared`), 최대 연결 수는 연결 모니터 제한 (`imq.service_name.connectionMonitor_limit`)과 최대 스레드 수 (`imq.service_name.max_threads`)의 두 가지 등록 정보를 곱한 수의 반입니다. 따라서 연결 수를 늘리려면 스레드 풀의 크기를 늘리거나 연결 모니터 제한을 늘립니다.
- 결국 지원 가능한 연결 수(또는 연결의 처리량)가 입출력 제한에 도달합니다. 그런 경우 다중 브로커 클러스터를 사용하여 클러스터 내의 브로커 인스턴스로 연결을 분산합니다.

**Solaris나 Linux 운영 체제에서 필요한 연결 수에 비해 파일 설명자가 너무 적습니다.**  
이 문제에 대한 자세한 내용은 66페이지의 "파일 설명자 제한 설정(Solaris 또는 Linux)"을 참조하십시오.

문제의 원인을 확인하려면 다음 작업을 수행합니다.

브로커 로그에 Too many open files와 유사한 항목이 있는지 검사합니다.

문제를 해결하려면 다음 작업을 수행합니다.

`ulimit` 설명서 페이지에 설명되어 있는 대로 파일 설명자 제한을 늘립니다.

**TCP 백로그가 동시에 설정할 수 있는 새 연결 요청 수를 제한합니다.**

TCP 백로그는 포트 매퍼가 추가 요청을 거부하기 전에 시스템 백로그(imq.portmapper.backlog)에 저장할 수 있는 동시 연결 요청 수를 제한합니다(Windows 운영 체제의 경우 하드 코딩된 백로그 제한이 있으며 Windows 데스크탑의 경우 5이고 Windows 서버의 경우 200임).

백로그 제한으로 인한 요청 거부는 비정상적으로 많은 동시 연결 요청 수로 인해 발생하는 일시적인 현상입니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

브로커 로그를 검사합니다. 먼저, 브로커가 특정 기간 동안 일부 연결은 수신하고 다른 일부 연결은 거부하는지 여부를 확인합니다. 그런 다음, 거부된 연결을 설명하는 메시지를 확인합니다. 그러한 메시지가 있다면 브로커는 TCP 백로그로 인한 연결 거부를 기록하지 않으므로 TCP 백로그 문제가 아닐 수 있습니다.

일부 성공적인 연결이 기록되어 있고 연결 거부는 기록되어 있지 않은 경우 TCP 백로그에 문제가 있을 수 있습니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

다음 방법을 사용하여 TCP 백로그 제한을 해결할 수 있습니다.

- 클라이언트가 시도했던 연결을 잠시 후에 다시 시도하도록 프로그래밍합니다(이 문제가 본래 일시적이기 때문에 이렇게 하면 대개 제대로 작동됨).
- imq.portmapper.backlog 값을 늘립니다.
- 클라이언트가 너무 자주 연결을 닫은 다음 열고 있지는 않은지 확인합니다.

**운영 체제가 동시 연결 수를 제한합니다.**

Windows 운영 체제 사용권에서는 지원되는 동시 원격 연결 수를 제한합니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

연결에 사용할 수 있는 스레드가 충분한지 확인하고(imqcmd query svc 사용) Windows 사용권 계약 조항을 확인합니다. 로컬 클라이언트에서는 연결할 수 있지만 원격 클라이언트에서는 연결할 수 없는 경우 운영 체제 제한이 문제의 원인일 수 있습니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

- 더 많은 연결을 허용하도록 Windows 사용권을 업그레이드합니다.
- 다중 브로커 클러스터를 설정하여 연결을 여러 브로커 인스턴스에 분산합니다.

**사용자의 인증 또는 권한 부여가 실패합니다.**

사용자 저장소에 사용자에 대한 항목이 없거나 사용자가 연결 서비스에 대한 액세스 권한이 없기 때문에 잘못된 비밀번호로 인해 인증이 실패할 수 있습니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

브로커 로그의 항목에 Forbidden 오류 메시지가 있는지 확인합니다. 이 메시지는 인증 오류를 나타낼 뿐 그 원인은 나타내지 않습니다.

- 파일 기반 사용자 저장소를 사용하는 경우 다음 명령을 입력합니다.  
`imqusermgr list -i instanceName -u userName`
- 출력에 사용자가 표시되는 경우 잘못된 비밀번호가 제출된 것일 수 있습니다. 출력에 다음과 같은 오류가 표시되는 경우 사용자 저장소에 항목이 없는 것입니다.  
`Error [B3048]: User does not exist in the password file,`
- LDAP 서버 사용자 저장소를 사용 중인 경우 적절한 도구를 사용하여 사용자 항목이 있는지 확인합니다.
- 액세스 제어 등록 정보에 연결 서비스 액세스에 대한 제한이 있는지 확인합니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

- 사용자 저장소에 사용자 항목이 없는 경우 사용자 저장소에 사용자를 추가합니다 (147페이지의 "사용자 저장소 채우기 및 관리" 참조).
- 잘못된 비밀번호가 사용된 경우 올바른 비밀번호를 제공합니다.
- 액세스 제어 등록 정보가 잘못 설정된 경우 액세스 제어 등록 정보 파일을 편집하여 연결 서비스 권한을 부여합니다(156페이지의 "연결 서비스에 대한 액세스 제어" 참조).

## 연결 처리 속도가 너무 느림

이 문제의 증상은 다음과 같습니다.

- 메시지 처리량이 기대에 미치지 못합니다.
- 브로커에 대해 지원되는 연결 수가 [248페이지의 "클라이언트가 연결을 설정할 수 없음"](#)에 설명되어 있는 대로 제한되는 것이 아니라 메시지 입출력 속도에 의해 제한됩니다.

이 절에서는 다음과 같은 가능한 원인에 대해 설명합니다.

- [네트워크 연결 또는 WAN이 너무 느립니다.](#)
- [연결 서비스 프로토콜이 기본적으로 TCP에 비해 느립니다.](#)
- [연결 서비스 프로토콜이 최적으로 조정되어 있지 않습니다.](#)
- 메시지가 너무 커서 너무 많은 대역폭을 사용합니다.
- 느린 연결 처리 능력으로 보이는 것이 실제로는 메시지 전달 프로세스의 어떤 단계에 병목 현상이 있는 것입니다.

### *네트워크 연결 또는 WAN이 너무 느립니다.*

문제의 원인을 확인하려면 다음 작업을 수행합니다.

네트워크를 핑하여 핑이 반환되는 데 걸리는 시간을 확인한 다음 네트워크 관리자에게 문의합니다. 또는 로컬 클라이언트를 사용하여 메시지를 보내고 받은 다음 이 전달 시간을 네트워크 링크를 사용하는 원격 클라이언트의 전달 시간과 비교할 수 있습니다.

문제를 해결하려면 다음 작업을 수행합니다.

연결이 너무 느린 경우 네트워크 링크를 업그레이드합니다.

### *연결 서비스 프로토콜이 기본적으로 TCP에 비해 느립니다.*

예를 들어 SSL 기반 또는 HTTP 기반 프로토콜이 TCP보다 느립니다([234페이지의 그림 11-5](#) 참조).

문제의 원인을 확인하려면 다음 작업을 수행합니다.

SSL 기반 또는 HTTP 기반 프로토콜을 사용하는 경우 TCP를 사용해보고 전달 시간을 비교합니다.

문제를 해결하려면 다음 작업을 수행합니다.

일반적으로 응용 프로그램 요구 사항에 따라 사용되는 프로토콜이 지정되므로 [238페이지의 "전송 프로토콜 조정"](#)에 설명되어 있는 대로 프로토콜을 조정해보는 것 이외에 할 수 있는 작업이 거의 없습니다.

**연결 서비스 프로토콜이 최적으로 조정되어 있지 않습니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

프로토콜을 조정해보고 차이가 있는지 확인합니다.

문제를 해결하려면 다음 작업을 수행합니다.

(238페이지의 "전송 프로토콜 조정")에 설명되어 있는 대로 프로토콜을 조정해봅니다.

**메시지가 너무 커서 너무 많은 대역폭을 사용합니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

작은 크기의 메시지를 사용하여 벤치마크를 실행해 봅니다.

문제를 해결하려면 다음 작업을 수행합니다.

- 응용 프로그램 개발자에게 응용 프로그램을 수정하여 메시지 압축 기능을 사용하게 합니다. 이 방법에 대해서는 *Message Queue Developer's Guide for Java Clients*에 설명되어 있습니다.
- 메시지를 보낼 데이터의 알림으로 사용하고 데이터는 다른 프로토콜을 사용하여 이동합니다.

**느린 연결 처리 능력으로 보이는 것이 실제로는 메시지 전달 프로세스의 어떤 단계에 병목 현상이 있는 것입니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

위의 항목 중 어떤 것도 느린 연결 처리량으로 보이는 문제의 원인이 아닌 것 같으면 [223 페이지의 그림 11-1](#)을 참조하여 다른 병목 현상이 있을 수 있는지 확인하고 다음 문제와 관련된 증상이 있는지 확인합니다.

- [256페이지의 "메시지 생성이 지연되거나 느림"](#)
- [259페이지의 "메시지가 백로그됨"](#)
- [264페이지의 "메시지 서버 처리 능력이 일정하지 않음"](#)

문제를 해결하려면 다음 작업을 수행합니다.

위의 문제 해결 절에 제공되어 있는 문제 해결 지침을 따릅니다.



# 클라이언트가 메시지 생성자를 만들 수 없음

이 문제의 증상은 다음과 같습니다.

- 물리적 대상에 대해 메시지 생성자를 만들 수 없기 때문에 클라이언트에서 예외가 발생합니다.

이 절에서는 다음과 같은 가능한 원인에 대해 설명합니다.

- 물리적 대상이 제한된 생성자 수만 허용하도록 구성되었습니다.
- 액세스 제어 등록 정보 파일의 설정으로 인해 사용자가 메시지 생성자를 만들 수 있는 권한이 없습니다.

## **물리적 대상이 제한된 생성자 수만 허용하도록 구성되었습니다.**

물리적 대상에 메시지가 누적되는 것을 방지하는 방법 중 하나는 물리적 대상에서 지원할 수 있는 생성자의 수(maxNumProducers)를 제한하는 것입니다.

문제의 원인을 확인하려면 다음 작업을 수행합니다.

물리적 대상을 확인합니다(131페이지의 "물리적 대상 정보 표시" 참조).

```
imqcmd query dst
```

출력에 현재 생성자 수와 maxNumProducers 값이 표시됩니다. 두 값이 같은 경우 생성자의 수가 구성된 제한에 도달한 것입니다. 새 생성자가 브로커에서 거부되는 경우 브로커는 ResourceAllocationException [C4088]: A JMS destination limit was reached를 반환하고 브로커 로그에 [B4183]: Producer can not be added to destination이라는 항목을 만듭니다.

문제를 해결하려면 다음 작업을 수행합니다.

maxNumProducers 속성 값을 늘립니다(133페이지의 "물리적 대상 등록 정보 업데이트" 참조).

## **액세스 제어 등록 정보 파일의 설정으로 인해 사용자가 메시지 생성자를 만들 수 있는 권한이 없습니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

브로커가 새 생성자를 거부할 때 다음과 같은 메시지를 반환합니다.

```
JMSSecurityException [C4076]: Client does not have permission to create producer on destination
```

또한 브로커 로그에 다음 항목을 생성합니다.

```
[B2041]: Producer on destination denied 및 [B4051]: Forbidden guest.
```

문제를 해결하려면 다음 작업을 수행합니다.

사용자가 메시지를 생성할 수 있도록 액세스 제어 등록 정보를 변경합니다([157페이지](#)의 "물리적 대상에 대한 액세스 제어" 참조).

## 메시지 생성이 지연되거나 느림

이 문제의 증상은 다음과 같습니다.

- 지속성 메시지를 보낼 때 `send()` 메소드가 반환되지 않고 클라이언트가 차단됩니다.
- 지속성 메시지를 보낼 때 클라이언트에 예외가 발생합니다.
- 생성자 클라이언트가 느려집니다.

이 절에서는 다음과 같은 가능한 원인에 대해 설명합니다.

- 메시지 서버가 백로그되고 느린 메시지 생성자로 응답했습니다.
- 브로커가 지속성 메시지를 데이터 저장소에 저장할 수 없습니다.
- 브로커 확인 시간 제한이 너무 짧습니다.
- 생성자 클라이언트에서 JVM 제한이 발생했습니다.

**메시지 서버가 백로그되고 느린 메시지 생성자로 응답했습니다.**

백로그된 서버가 브로커 메모리에 메시지를 누적시킵니다.

물리적 대상 메모리의 메시지 수와 메시지 바이트 수가 구성된 제한에 도달하면 브로커가 지정된 제한 동작에 따라 메모리 자원을 절약하려고 합니다. 다음 제한 동작은 메시지 생성자를 느리게 만듭니다.

- `FLOW_CONTROL`: 브로커가 지속성 메시지의 수신을 바로 확인하지 않습니다. 따라서 생성자 클라이언트가 차단됩니다.
- `REJECT_NEWEST`: 브로커가 새 지속성 메시지를 거부합니다.

마찬가지로 모든 물리적 대상의 브로커 전체 메모리에서 메시지 수나 메시지 바이트 수가 구성된 제한에 도달하면 브로커가 최신 메시지를 거부하여 메모리 자원을 절약하려고 합니다.

또한 물리적 대상이나 브로커 전체 제한이 제대로 설정되어 있지 않아 시스템 메모리 제한에 도달하면 브로커는 메모리 오버로드를 막기 위해 점점 더 중대한 조치를 취합니다. 이러한 조치로는 메시지 생성자 억제가 있습니다.

### 문제의 원인을 확인하려면 다음 작업을 수행합니다.

구성된 메시지 제한으로 인해 브로커가 메시지를 거부할 경우 브로커는 다음과 같은 메시지를 반환합니다.

```
JMSEException [C4036]: A server error occurred
```

또한 브로커 로그에 다음 항목을 생성합니다.

```
WARNING [B2011]: Storing of JMS message from IMQconn failed
```

이 메시지 다음에는 제한에 도달했음을 나타내는 메시지가 표시됩니다. 물리적 대상에 메시지 제한이 있는 경우 브로커는 다음과 같은 항목을 생성합니다.

```
[B4120]: Can not store message on destination destName because capacity of maxNumMsgs would be exceeded.
```

브로커 전체에 메시지 제한이 있는 경우 브로커는 다음과 같은 항목을 생성합니다.

```
[B4024]: The Maximum Number of messages currently in the system has been exceeded, rejecting message.
```

좀 더 일반적으로 다음과 같은 방법으로 거부가 발생하기 전에 메시지 제한 조건을 확인할 수 있습니다.

- 물리적 대상과 브로커를 쿼리하여 각각에 구성된 메시지 제한 설정을 검사
- 적절한 `imqcmd` 명령을 사용하여 물리적 대상이나 브로커 전체에서 현재 메시지 수나 메시지 바이트 수를 모니터링. 모니터링할 수 있는 메트릭과 메트릭을 가져올 때 사용하는 명령에 대한 자세한 내용은 [18장](#), "[메트릭 참조](#)"를 참조하십시오.

**문제를 해결하려면 다음 작업을 수행합니다.**

메시지가 백로그되어 생성자가 느려지는 것을 해결하는 데에는 다음과 같은 몇 가지 방법이 있습니다.

- 메모리 자원을 초과하지 않도록 주의하면서 물리적 대상 또는 브로커 전체에 대한 메시지 제한을 수정합니다.

일반적으로는 브로커 전체 메시지 제한에 도달하는 일이 없도록 대상별로 메모리를 관리해야 합니다. 자세한 내용은 [242페이지의 "브로커 조정"](#)을 참조하십시오.

- 메시지 제한에 도달하면 메시지 생성을 느리게 하기보다 메모리에서 메시지를 버리도록 대상의 제한 동작을 변경합니다.

예를 들어, 메모리에 누적되는 메시지를 삭제하는 REMOVE\_OLDEST 및 REMOVE\_LOW\_PRIORITY 제한 동작을 지정할 수 있습니다([329페이지의 표 15-1](#) 참조).

***브로커가 지속성 메시지를 데이터 저장소에 저장할 수 없습니다.***

브로커가 데이터 저장소에 액세스할 수 없거나 지속성 메시지를 데이터 저장소에 기록할 수 없는 경우 생성자 클라이언트가 차단됩니다. 위에 설명되어 있는 대상 또는 브로커 전체 메시지 제한에 도달한 경우에도 이러한 상태가 발생할 수 있습니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

브로커가 데이터 저장소에 기록할 수 없는 경우 브로커는 브로커 로그에 [B2011]: Storing of JMS message from connectionID failed... 또는 [B4004]: Failed to persist message messageID... 항목 중 하나를 만듭니다

**문제를 해결하려면 다음 작업을 수행합니다.**

- 기본 제공 지속성의 경우 파일 기반 데이터 저장소의 디스크 공간을 늘립니다.
- JDBC 호환 데이터 저장소의 경우 플러그인 지속성이 제대로 구성되어 있는지 확인합니다([4장, "브로커 구성"](#) 참조). 제대로 구성되어 있는 경우 데이터베이스 관리자에게 문의하여 다른 데이터베이스 문제를 해결합니다.

***브로커 확인 시간 제한이 너무 짧습니다.***

느린 연결 또는 높은 CPU 사용률이나 부족한 메모리 자원 때문에 무기력해진 메시지 서버로 인해 브로커가 지속성 메시지의 수신을 확인하는 데 연결 팩토리의 imqAckTimeout 속성 값에서 허용하는 것보다 더 많은 시간이 필요할 수 있습니다.

문제의 원인을 확인하려면 다음 작업을 수행합니다.

`imqAckTimeout` 값이 초과되면 브로커는 다음 메시지를 반환합니다.

```
JMSEException [C4000]: Packet acknowledge failed
```

문제를 해결하려면 다음 작업을 수행합니다.

`imqAckTimeout` 연결 팩토리 속성 값을 변경합니다([177페이지의 "연결 팩토리 속성" 참조](#)).

**생성자 클라이언트에서 JVM 제한이 발생했습니다.**  
문제의 원인을 확인하려면 다음 작업을 수행합니다.

- 클라이언트 응용 프로그램이 메모리 부족 오류를 수신하는지 확인합니다.
- `freeMemory()`, `MaxMemory()`, `totalMemory()` 같은 런타임 메소드를 사용하여 JVM 힙에서 사용 가능한 메모리를 확인합니다.

문제를 해결하려면 다음 작업을 수행합니다.

JVM을 조정합니다([238페이지의 "Java 가상 머신 조정" 참조](#)).

## 메시지가 백로그됨

이 문제의 증상은 다음과 같습니다.

- 브로커 또는 특정 대상의 메시지나 메시지 바이트 수가 시간에 따라 꾸준하게 증가합니다.

메시지가 누적되고 있는지 확인하려면 브로커의 메시지나 메시지 바이트 수가 시간에 따라 어떻게 변하는지 확인하고 구성된 제한과 비교합니다. 먼저 구성된 제한을 확인합니다.

```
imqcmd query bkr
```

(**주:** `imqcmd metrics bkr` 하위 명령은 이 정보를 표시하지 않습니다.)

그런 다음 각 대상에서 메시지 누적을 확인합니다.

```
imqcmd list dst
```

메시지가 대상이나 브로커 전체에 구성된 제한을 초과했는지 확인하려면 브로커 로그에서 `WARNING [B2011]: Storing of JMS message from...failed`라는 항목이 있는지 확인합니다. 이 항목 다음에는 초과된 제한에 대해 설명하는 다른 항목이 표시됩니다.

- 메시지 생성이 지연되거나 생성된 메시지가 브로커에서 거부됩니다.
- 메시지가 사용자에게 도달하기까지 비정상적으로 오래 걸립니다.

이 절에서는 다음과 같은 가능한 원인에 대해 설명합니다.

- 주제 대상에 비활성 영구 가입이 있습니다.
- 대기열의 메시지를 사용할 수 있는 사용자 수가 너무 적습니다.
- 메시지 사용자가 너무 느리게 처리하여 메시지 생성자를 따라가지 못합니다.
- 클라이언트 확인 처리가 메시지 사용을 느리게 합니다.
- 브로커가 생성된 메시지를 따라갈 수 없습니다.
- 클라이언트 코드에 결함이 있습니다. 사용자가 메시지를 확인하지 않습니다.

#### **주제 대상에 비활성 영구 가입이 있습니다.**

영구 가입이 비활성 상태인 경우 해당 사용자가 활성화되어 메시지를 사용할 수 있을 때까지 대상에 메시지가 저장됩니다.

#### **문제의 원인을 확인하려면 다음 작업을 수행합니다.**

각 주제 대상에서 영구 가입 상태를 확인합니다.

```
imqcmd list dur -d destName
```

#### **문제를 해결하려면 다음 작업을 수행합니다.**

다음 작업 중 하나를 수행할 수 있습니다.

- 문제가 있는 영구 가입의 모든 메시지를 제거합니다(122페이지의 "영구 가입 관리" 참조).
- 주제에 대해 메시지 제한 및 제한 동작 속성을 지정합니다(329페이지의 표 15-1 참조). 예를 들어, 메모리에 누적되는 메시지를 삭제하는 REMOVE\_OLDEST 및 REMOVE\_LOW\_PRIORITY 제한 동작을 지정할 수 있습니다.
- 해당 대상에서 모든 메시지를 제거합니다(134페이지의 "물리적 대상 제거" 참조).
- 메시지가 메모리에 남아 있을 수 있는 시간을 제한합니다. 생성자 클라이언트를 다시 작성하여 각 메시지에 활성 기간 값을 설정할 수 있습니다. imqOverrideJMSEExpiration 및 imqJMSEExpiration 연결 팩토리 속성을 설정하여 연결을 공유하는 모든 생성자에 대해 이러한 설정을 대체할 수 있습니다(338페이지의 "메시지 헤더 무시" 참조).

**대기열의 메시지를 사용할 수 있는 사용자 수가 너무 적습니다.**

메시지를 전달할 수 있는 활성 사용자의 수가 너무 적은 경우 메시지가 누적될 때 대기열 대상이 백로그됨 수 있습니다. 이런 상태는 다음과 같은 이유 중 하나 때문에 발생할 수 있습니다.

- 대상에 활성 사용자가 수가 너무 적습니다.
- 사용자 클라이언트가 연결을 설정하지 못했습니다.
- 대기열의 메시지에 일치하는 선택기를 사용하는 활성 사용자가 없습니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

사용자가 대기열의 메시지를 사용할 수 없는 원인을 확인하려면 대상에서 활성 사용자의 수를 확인합니다.

```
imqcmd metrics dst -n destName -t q -m con
```

**문제를 해결하려면 다음 작업을 수행합니다.**

사용자가 대기열의 메시지를 사용할 수 없는 원인에 따라 다음 작업 중 하나를 수행할 수 있습니다.

- 추가 사용자 클라이언트를 시작하여 대기열에 더 많은 활성 사용자를 만듭니다.
- `imq.consumerFlowLimit` 브로커 등록 정보를 조정하여 여러 사용자에게 대한 대기열 전달을 최적화합니다(243페이지의 "다중 사용자 대기열 성능" 참조).
- 대기열에 대해 메시지 제한 및 제한 동작 속성을 지정합니다(329페이지의 표 15-1 참조). 예를 들어, 메모리에 누적되는 메시지를 삭제하는 `REMOVE_OLDEST` 및 `REMOVE_LOW_PRIORITY` 제한 동작을 지정할 수 있습니다.
- 해당 대상에서 모든 메시지를 제거합니다(134페이지의 "물리적 대상 제거" 참조).
- 메시지가 메모리에 남아 있을 수 있는 시간을 제한합니다. 생성자 클라이언트를 다시 작성하여 각 메시지의 수명 값을 설정할 수 있습니다. `imqOverrideJMSEExpiration` 및 `imqJMSEExpiration` 연결 팩토리 속성을 설정하여 연결을 공유하는 모든 생성자에 대해 이러한 설정을 대체할 수 있습니다(338페이지의 "메시지 헤더 무시" 참조).

**메시지 사용자가 너무 느리게 처리하여 메시지 생성자를 따라가지 못합니다.**

이 경우 주제 가입자나 대기열 수신자가 메시지를 사용하는 것이 생성자가 메시지를 보내는 것보다 느립니다. 이러한 불균형으로 인해 하나 이상의 대상이 메시지로 백로그됩니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

메시지가 브로커에 유입 및 유출되는 속도를 확인합니다.

```
imqcmd metrics bkr -m rts
```

그런 다음 개별 대상 각각에 대한 흐름 속도를 확인합니다.

```
imqcmd metrics bkr -t destType -n destName -m rts
```

**문제를 해결하려면 다음 작업을 수행합니다.**

- 사용자 클라이언트 코드를 최적화합니다.
- 대기열 대상의 경우 활성 사용자의 수를 늘립니다(243페이지의 "다중 사용자 대기열 성능" 참조).

**클라이언트 확인 처리가 메시지 사용을 느리게 합니다.**

두 가지 요소가 클라이언트의 확인 처리에 영향을 미칩니다.

- 상당한 브로커 자원이 클라이언트 확인 처리에 사용될 수 있습니다. 따라서 브로커가 클라이언트 확인을 확인할 때까지 사용자 클라이언트가 차단되는 확인 모드에서는 메시지 사용이 느려질 수 있습니다.
- JMS 페이로드 메시지와 Message Queue 제어 메시지(예: 클라이언트 확인)는 같은 연결을 공유합니다. 따라서 JMS 페이로드 메시지에 의해 제어 메시지가 일시적으로 중단되어 메시지 사용이 느려질 수 있습니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

- 패킷 흐름을 기준으로 메시지 흐름을 확인합니다. 초당 패킷 수와 메시지 수의 비율이 맞지 않으면 클라이언트 확인에 문제가 있을 수 있습니다.
- 클라이언트가 다음과 같은 메시지를 받았는지 확인합니다.

```
JMSException [C4000]: Packet acknowledge failed
```

**문제를 해결하려면 다음 작업을 수행합니다.**

- 클라이언트가 사용하는 확인 모드를 수정합니다. 예를 들어 DUPS\_OK\_ACKNOWLEDGE나 CLIENT\_ACKNOWLEDGE로 전환합니다.
- CLIENT\_ACKNOWLEDGE 또는 트랜잭션된 세션을 사용 중인 경우 많은 수의 메시지를 하나의 확인으로 그룹화합니다.
- 사용자 및 연결 흐름 제어 매개 변수를 조정합니다(244페이지의 "클라이언트 런타임 메시지 흐름 조정" 참조).



**브로커가 생성된 메시지를 따라갈 수 없습니다.**

이 경우 메시지가 브로커로 유입되는 것이 브로커가 메시지를 라우팅하여 사용자에게 전달하는 것보다 빠릅니다. 브로커의 지체는 CPU, 네트워크 소켓 읽기/쓰기 작업, 디스크 읽기/쓰기 작업, 메모리 페이징, 영구 저장소, JVM 메모리 제한 중 하나나 모두에 대한 제한 때문일 수 있습니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

이 문제를 일으키는 다른 원인은 없는지 확인합니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

- 컴퓨터나 데이터 저장소의 속도를 업그레이드합니다.
- 브로커 클러스터를 사용하여 로드를 여러 브로커 인스턴스에 분산합니다.

**클라이언트 코드에 결함이 있습니다. 사용자가 메시지를 확인하지 않습니다.**

메시지는 메시지를 받은 모든 사용자가 확인할 때까지 대상에 보관됩니다. 따라서 클라이언트가 사용된 메시지를 확인하지 않는 경우 메시지는 삭제되지 않고 대상에 누적됩니다.

예를 들어, 클라이언트 코드에 다음 결함이 있을 수 있습니다.

- `CLIENT_ACKNOWLEDGE` acknowledgment 또는 트랜잭션된 세션을 사용하는 사용자가 `Session.acknowledge()` 또는 `Session.commit()`을 정기적으로 호출하고 있지 않을 수 있습니다.
- `AUTO_ACKNOWLEDGE` 세션을 사용하는 사용자가 몇 가지 이유로 중지되어 있을 수 있습니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

먼저 이 절에 나열된 모든 다른 가능한 원인을 확인하고 다음 명령을 사용하여 대상을 나열합니다.

```
imqcmd list dst
```

UnAcked 헤더에 나열된 메시지 수가 대상의 메시지 수와 동일한지를 확인합니다.

UnAcked 헤더에 있는 메시지가 사용자에게 전송되었지만 확인되지 않았습니. 이 수가 총 메시지 수와 같은 경우 브로커가 모든 메시지를 전송하고 확인을 대기 중인 것입니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

이 문제를 디버깅할 때 응용 프로그램 개발자에게 도움을 요청하십시오.

## 메시지 서버 처리 능력이 일정하지 않음

이 문제의 증상은 다음과 같습니다.

- 메시지 처리 능력이 산발적으로 떨어지다가 정상 성능을 회복합니다.
- 이 절에서는 다음과 같은 가능한 원인에 대해 설명합니다.
- **브로커의 메모리 자원이 매우 적습니다.**
  - **JVM 메모리 재생 이용(가비지 컬렉션)이 발생합니다.**
  - **JVM이 성능을 높이기 위해 JIT(Just-In-Time) 컴파일러를 사용하고 있습니다.**

### *브로커의 메모리 자원이 매우 적습니다.*

대상 및 브로커 제한이 제대로 설정되어 있지 않기 때문에 브로커가 메모리 오버로드를 막기 위해 점점 더 중대한 조치를 취하게 되어 메시지 백로그를 지울 때까지 브로커가 아주 느려질 수 있습니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

브로커 로그에서 부족한 메모리 상태([B1089]: In low memory condition, broker is attempting to free up resources)와 새 메모리 상태 및 사용 중인 전체 메모리 양에 대해 설명하는 항목이 있는지 확인합니다.

또한 JVM 힙에서 사용 가능한 메모리를 확인합니다.

```
imqcmd metrics bkr -m cxn
```

전체 JVM 메모리의 값이 최대 JVM 메모리 값에 근접한 경우 사용 가능한 메모리가 적습니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

- JVM을 조정합니다([238페이지의 "Java 가상 머신 조정"](#) 참조).
- 시스템 스왑 공간을 늘립니다.

**JVM 메모리 재생 이용(가비지 컬렉션)이 발생합니다.**

메모리를 확보하기 위해 메모리 재생 이용이 주기적으로 시스템을 정리합니다. 이럴 경우 모든 스레드가 차단됩니다. 확보할 메모리 양이 크고 JVM 힙 크기가 클수록 메모리 재생 이용으로 인한 지체가 길어집니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

컴퓨터의 CPU 사용을 모니터링합니다. 메모리 재생 이용이 발생하면 CPU 사용률이 떨어집니다.

또는 다음 명령줄 옵션을 사용하여 브로커를 시작합니다.

```
-vmargs -verbose:gc
```

표준 출력에 메모리 재생 이용이 발생하는 시간이 표시됩니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

다중 CPU 컴퓨터인 경우 메모리 재생 이용이 병렬로 발생되도록 설정합니다.

```
-XX:+UseParallelGC=true
```

**JVM이 성능을 높이기 위해 JIT(Just-In-Time) 컴파일러를 사용하고 있습니다.****문제의 원인을 확인하려면 다음 작업을 수행합니다.**

이 문제를 일으키는 다른 원인은 없는지 확인합니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

잠시 동안 시스템이 실행되도록 놓아두면 성능이 향상됩니다.

## 메시지가 사용자에게 도달하지 않음

이 문제의 증상은 다음과 같습니다.

- 생성자가 보낸 메시지를 사용자가 받지 못합니다.
- 이 절에서는 다음과 같은 가능한 원인에 대해 설명합니다.
- 제한 동작으로 인해 메시지가 브로커에서 삭제됩니다.
  - 메시지 시간 초과 값이 만료됩니다.
  - 클럭이 동기화되지 않습니다.
  - 사용자 클라이언트가 연결에서 메시지 전달을 시작하지 못했습니다.

### **제한 동작으로 인해 메시지가 브로커에서 삭제됩니다.**

대상 메모리의 메시지 수나 메시지 바이트 수가 구성된 제한에 도달하면 브로커는 메모리 자원을 절약하려고 합니다. 이러한 제한에 도달하면 브로커가 수행하는 구성 가능한 동작 중 다음 세 가지 동작으로 인해 메시지가 손실됩니다.

- REMOVE\_OLDEST: 가장 오래된 메시지를 삭제
- REMOVE\_LOW\_PRIORITY: 메시지의 보존 기간을 기준으로 우선 순위가 가장 낮은 메시지 삭제
- REJECT\_NEWEST: 새 지속성 메시지 거부

브로커 메모리의 메시지 수나 메시지 바이트 수가 구성된 제한에 도달하면 브로커는 최신 메시지를 거부하여 메모리 자원을 절약하려고 합니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

269페이지의 "사용 불능 메시지 대기열에 메시지가 포함되어 있음"에서 설명한 대로 사용 불능 메시지 대기열을 확인합니다. 특히, 270페이지의 "메시지 수 또는 메시지 크기가 대상 제한을 초과합니다."의 지침을 사용합니다. REMOVE\_OLDEST 또는 REMOVE\_LOW\_PRIORITY 이유를 찾습니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

대상 제한을 늘립니다. 예를 들면 다음과 같습니다.

```
imqcmd update dst -n MyDest -o maxNumMsgs=1000
```

### **메시지 시간 초과 값이 만료됩니다.**

브로커가 시간 초과 값이 만료된 메시지를 삭제합니다. 대상이 충분히 메시지로 백로그된 경우 수명 값이 너무 짧은 메시지는 삭제될 수 있습니다.

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

사용 불능 메시지 대기열에서 메시지 시간이 초과되었는지 확인합니다.

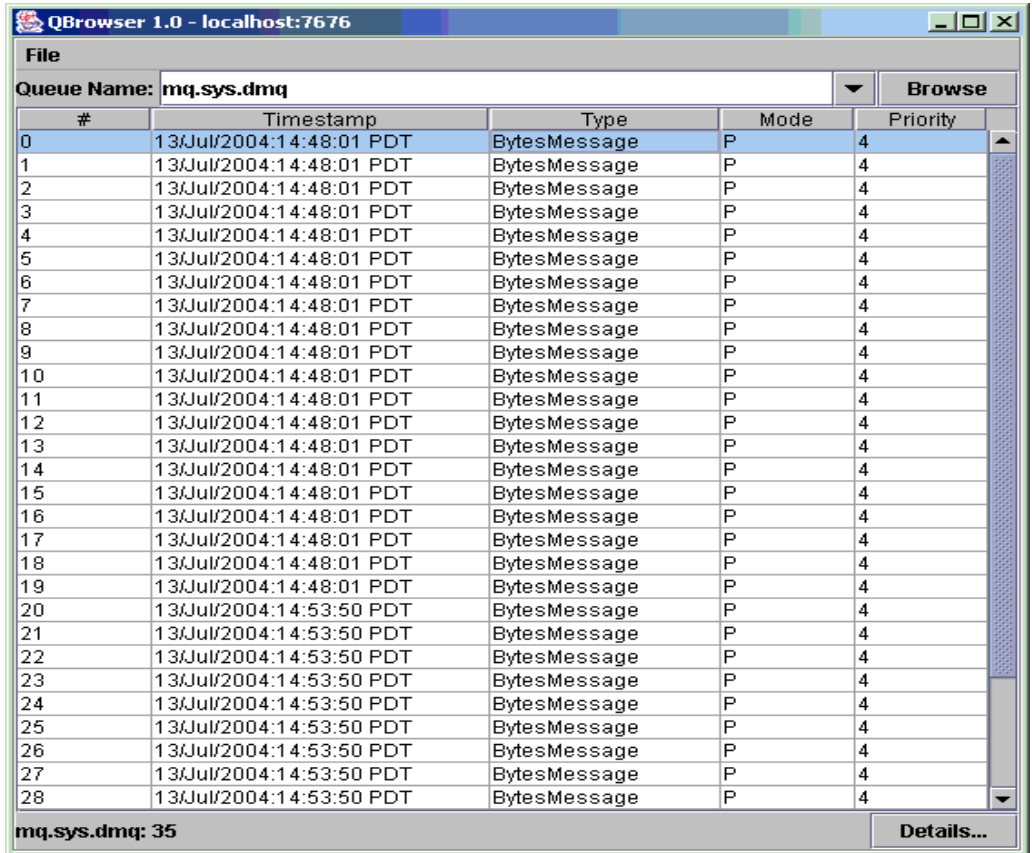
QBrower 데모 응용 프로그램을 사용하여 DMQ 내용을 살펴봅니다. QBrower 데모 위치는 운영 체제별로 다릅니다. 위치는 **부록 A, "운영 체제별 Message Queue 데이터의 위치"**의 "예제 응용 프로그램 및 위치"를 참조하십시오.

다음은 Windows에서의 호출 예입니다.

```
cd \MessageQueue3\demo\applications\qbrowser java QBrower
```

QBrower 기본 창이 나타나면 대기열 이름 mq.sys.dmq를 선택한 다음 **Browse**를 누릅니다. 다음과 같은 목록이 나타납니다.

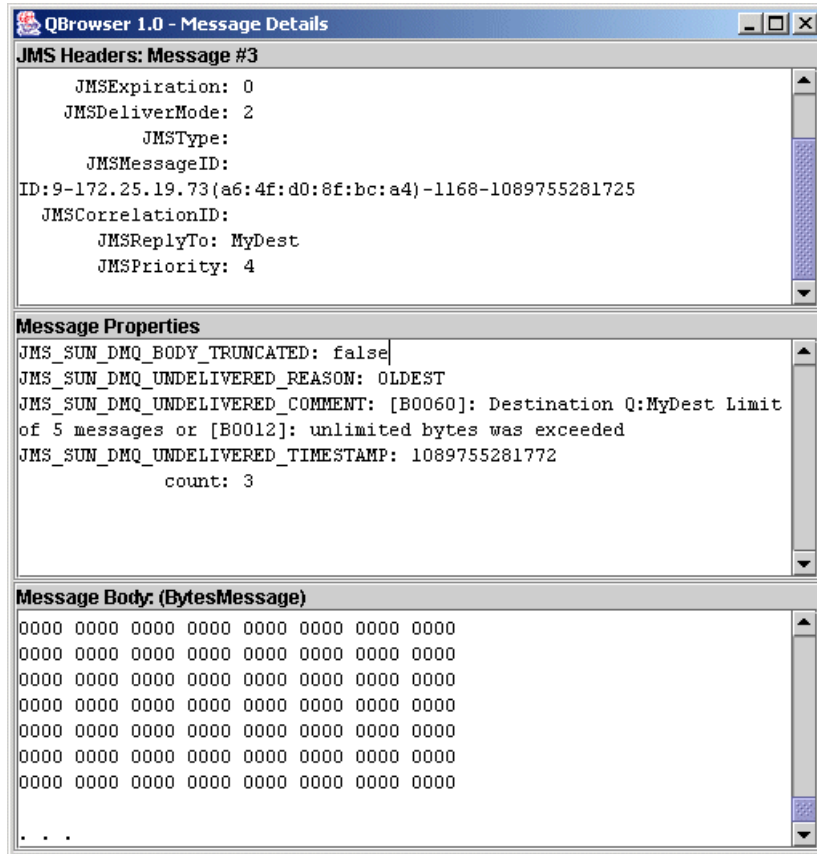
그림 12-1 QBrowser 창



메시지를 두 번 누르면 해당 메시지에 대한 세부 정보가 표시됩니다.

메시지가 사용자에게 도달하지 않음

그림 12-2 QBrowser 메시지 세부 정보



메시지에 대한 JMS\_SUN\_DMQ\_UNDELIVERED\_REASON 등록 정보 값이 EXPIRED인지 확인합니다.

문제를 해결하려면 다음 작업을 수행합니다.

응용 프로그램 개발자에게 문의하여 수명 값을 늘립니다.

**클럭이 동기화되지 않습니다.**

클럭이 동기화되어 있지 않은 경우 브로커의 메시지 수명 계산 오류로 인해 메시지가 만료 시간을 초과하여 삭제될 수 있습니다.

문제의 원인을 확인하려면 다음 작업을 수행합니다.

브로커 로그 파일에서 B2102, B2103, B2104 메시지 중 하나가 있는지 확인합니다. 이러한 모든 메시지는 가능한 클럭 스큐가 감지되었음을 보고합니다.

문제를 해결하려면 다음 작업을 수행합니다.

66페이지의 "시스템 자원 준비"에서 설명한 대로 시간 동기화 프로그램이 실행 중인지 확인합니다.

*사용자 클라이언트가 연결에서 메시지 전달을 시작하지 못했습니다.*

클라이언트 코드가 연결을 설정하고 해당 연결에서 메시지 전달을 시작할 때까지 메시지를 전달할 수 없습니다.

문제의 원인을 확인하려면 다음 작업을 수행합니다.

클라이언트 코드가 연결을 설정하고 메시지 전달을 시작하는지 확인합니다.

문제를 해결하려면 다음 작업을 수행합니다.

연결을 설정하고 메시지 전달을 시작하도록 클라이언트 코드를 다시 작성합니다.

## 사용 불능 메시지 대기열에 메시지가 포함되어 있음

이 문제의 증상은 다음과 같습니다.

- 대상을 나열하면 사용 불능 메시지 대기열에 메시지가 포함되어 있습니다. 예를 들어, 다음과 같은 명령을 실행합니다.

```
imqcmd list dst
```

사용자 아이디와 비밀번호를 입력하면 다음과 같은 메시지가 출력됩니다.

```
Listing all the destinations on the broker specified by:
-----
Host          Primary Port
-----
localhost     7676
-----
Name          Type      State   Producers  Consumers Msgs
              Total Count UnAck   Avg Size
-----
MyDest        Queue    RUNNING 0           0         5         0         1177.0
mq.sys.dmqs  Queue    RUNNING 0           0        35         0         1422.0
Successfully listed destinations.
```

이 예의 사용 불능 메시지 대기열 `mq.sys.dmq`에는 35개의 메시지가 포함되어 있습니다.

이 절에서는 다음과 같은 가능한 원인에 대해 설명합니다.

- 메시지 수 또는 메시지 크기가 대상 제한을 초과합니다.
- 브로커 클럭과 생성자 클럭이 동기화되어 있지 않습니다.
- 메시지 시간이 초과되기 전에 사용자가 메시지를 받지 못합니다.
- 사용자 수에 비해 생성자가 너무 많습니다.
- 생성자가 사용자보다 더 빠릅니다.
- 사용자가 너무 느립니다.
- 클라이언트가 메시지를 완결하지 못합니다.
- 영구 사용자가 비활성 상태입니다.
- 예기치 않은 브로커 오류가 발생했습니다.

***메시지 수 또는 메시지 크기가 대상 제한을 초과합니다.***

문제의 원인을 확인하려면 다음 작업을 수행합니다.

QBrowser 데모 응용 프로그램을 사용하여 사용 불능 메시지 대기열의 내용을 살펴봅니다. QBrowser 데모 위치는 운영 체제별로 다릅니다. 위치는 **부록 A, "운영 체제별 Message Queue 데이터의 위치"**의 "예제 응용 프로그램 및 위치"를 참조하십시오.

다음은 Windows에서의 호출 예입니다.

```
cd \MessageQueue3\demo\applications\qbrowser java QBrowser
```

QBrowser 기본 창이 나타나면 대기열 이름 `mq.sys.dmq`를 선택한 다음 **Browse**를 누릅니다. **267페이지의 그림 12-1**에 표시된 것과 유사한 목록이 나타납니다.

아무 메시지도 두 번 누르면 해당 메시지에 대한 세부 정보가 표시됩니다. **268페이지의 그림 12-2**에 표시된 창이 나타납니다.

다음 메시지 등록 정보 값을 확인합니다.

- JMS\_SUN\_DMQ\_UNDELIVERED\_REASON
- JMS\_SUN\_DMQ\_UNDELIVERED\_COMMENT
- JMS\_SUN\_DMQ\_UNDELIVERED\_TIMESTAMP

JMS 헤더에서 `JMSDestination` 값을 검토하여 메시지가 사용 불능 상태인 대상을 확인합니다.



**문제를 해결하려면 다음 작업을 수행합니다.**

대상 제한을 늘립니다. 예를 들면 다음과 같습니다.

```
imqcmd update dst -n MyDest -o maxNumMsgs=1000
```

**브로커 클럭과 생성자 클럭이 동기화되어 있지 않습니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

QBrowser 응용 프로그램을 사용하여 사용 불능 메시지 대기열에 있는 메시지의 세부 정보를 살펴봅니다. JMS\_SUN\_DMQ\_UNDELIVERED\_REASON 값을 확인하여 원인이 EXPIRED인 메시지를 찾습니다.

브로커 로그 파일에서 B2102, B2103, B2104 메시지 중 하나가 있는지 확인합니다. 이러한 모든 메시지는 가능한 클럭 스큐가 감지되었음을 보고합니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

66페이지의 "시스템 자원 준비"에 설명된 대로 시간 동기화 프로그램이 실행 중인지 확인합니다.

**메시지 시간이 초과되기 전에 사용자가 메시지를 받지 못합니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

QBrowser 응용 프로그램을 사용하여 사용 불능 메시지 대기열에 있는 메시지의 세부 정보를 살펴봅니다. JMS\_SUN\_DMQ\_UNDELIVERED\_REASON 값을 확인하여 원인이 EXPIRED인 메시지를 찾습니다.

대상에 사용자가 있는지 확인합니다. 예를 들면 다음과 같습니다.

```
imqcmd query dst -t q -n MyDest
```

현재 활성 사용자 수에 나열된 값을 확인합니다. 활성 사용자가 있는 경우 다음 중 하나에 해당됩니다.

- 사용자의 연결이 일시 중단되었습니다.
- 메시지 시간 초과가 사용자의 실행 속도에 비해 너무 짧습니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

응용 프로그램 개발자에게 메시지 수명 값을 늘리도록 요청합니다.

**사용자 수에 비해 생성자가 너무 많습니다.**

문제의 원인을 확인하려면 다음 작업을 수행합니다.

QBrowser 응용 프로그램을 사용하여 사용 불능 메시지 대기열에 있는 메시지의 세부 정보를 살펴봅니다. JMS\_SUN\_DMQ\_UNDELIVERED\_REASON의 값을 확인합니다.

이유가 REMOVE\_OLDEST 또는 REMOVE\_LOW\_PRIORITY인 경우 `imqcmd query dst` 명령을 사용하여 대상의 생성자 수와 사용자 수를 확인합니다. 생성자 수가 사용자 수를 초과하는 경우 생성 속도가 사용 속도보다 훨씬 빠를 수 있습니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

사용자 클라이언트를 추가하거나 `FLOW_CONTROL` 제한 동작을 사용할 대상을 설정합니다. `FLOW_CONTROL` 제한 동작은 사용 속도를 사용하여 생산 속도를 제어합니다.

다음 예와 같은 명령을 사용하여 흐름 제어 동작을 시작합니다.

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

**생성자가 사용자보다 더 빠릅니다.**

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

느린 사용자로 인해 생성자가 느려지는지 확인하려면 대상 제한 동작을 `FLOW_CONTROL`로 설정합니다. `FLOW_CONTROL` 제한 동작은 사용 속도를 사용하여 생산 속도를 제어합니다.

다음 예와 같은 명령을 사용하여 흐름 제어 동작을 시작합니다.

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

다음 예와 같은 명령을 실행하여 메트릭을 통해 대상 입력과 출력을 검토합니다.

```
imqcmd metrics dst -n myDst -t q -m rts
```

메트릭 출력에서 다음 값을 검토합니다.

- `Msgs/sec Out`

이 값은 브로커가 제거하는 초 당 메시지 수를 나타냅니다. 브로커는 모든 사용자가 메시지 수신을 확인하면 메시지를 제거하기 때문에 메트릭이 사용 속도에 영향을 줍니다.

- `Msgs/sec In`

이 값은 브로커가 생성자로부터 수신하는 초 당 메시지 수를 나타냅니다. 따라서 메트릭이 생성 속도에 영향을 미칩니다.

흐름 제어는 생성을 소비에 맞추므로 생성이 느려지거나 중단되었는지 확인합니다. 속도가 느려지거나 중단되는 경우 생성자와 사용자의 처리 속도가 일치하지 않는 것입니다.

`imqcmd list dst` 명령을 사용하여 전송되었지만 확인되지 않은(`UnAcked`) 메시지 수를 확인할 수도 있습니다. 확인되지 않은 메시지 수가 대상 크기보다 작은 경우 대상에 추가 용량이 있기 때문에 클라이언트 흐름 제어에 의해 대상이 다시 보관됩니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

생성 속도가 사용 속도보다 일관되게 더 빠를 경우 흐름 제어를 정기적으로 사용하여 시스템을 정렬된 상태로 유지하십시오.

또는 뒤에 나오는 절들을 참조하여 다음과 같은 가능한 요소 각각을 해결해 보십시오.

- [사용자가 너무 느립니다.](#)
- [클라이언트가 메시지를 완결하지 못합니다.](#)
- [사용자가 메시지를 확인할 수 없습니다.](#)
- [영구 사용자가 비활성 상태입니다.](#)
- [예기치 않은 브로커 오류가 발생했습니다.](#)

***사용자가 너무 느립니다.***

**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

[272페이지의 "생성자가 사용자보다 더 빠릅니다."](#)에서 설명한 대로 메트릭을 사용하여 생성 속도와 사용 속도를 확인합니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

다음 중 하나 이상을 시도해 보십시오.

- `FLOW_CONTROL` 제한 동작을 사용할 대상을 설정합니다. 다음과 같은 명령을 사용합니다.

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

흐름 제어의 사용으로 인해 생산 속도가 사용 속도만큼 느려지므로 브로커에 메시지가 누적되지 않습니다. 대상에서 메시지를 적절한 시기에 처리할 수 있을 때까지 생성자 응용 프로그램이 메시지를 보관하므로 만료될 위험이 적습니다.

- 응용 프로그램 개발자에게 문의하여 생성자가 메시지를 일정한 속도로 보내는지 주기적으로 많은 메시지를 보내는지 확인합니다.

응용 프로그램이 한 번에 많은 메시지를 보내는 경우 다음 항목의 지침에 따라 대상 제한을 늘립니다.

- 메시지 수 및/또는 바이트 수를 기반으로 대상 제한을 늘립니다.  
대상의 메시지 수를 변경하려면 다음과 같은 형식으로 명령을 입력합니다.  

```
imqcmd update dst -n destName -t {q/t} -o maxNumMsgs=number
```

  
대상의 크기를 변경하려면 다음과 같은 형식으로 명령을 입력합니다.  

```
imqcmd update dst -n destName -t {q/t} -o maxTotalMsgBytes=number
```

  
제한을 늘리면 브로커가 사용하는 메모리의 양이 증가합니다. 따라서 제한이 너무 높을 경우 메모리가 부족하여 브로커가 메시지를 처리할 수 없게 됩니다.
- 생산 로드가 높은 수준에서 메시지 손실을 허용할 수 있는지 여부를 고려합니다.

**클라이언트가 메시지를 완결하지 못합니다.**  
문제의 원인을 확인하려면 다음 작업을 수행합니다.

응용 프로그램 개발자에게 문의하여 응용 프로그램이 트랜잭션을 사용하는지 여부를 확인합니다. 응용 프로그램이 트랜잭션을 사용하는 경우 다음과 같이 활성 트랜잭션을 나열합니다.

```
imqcmd list txn
```

다음은 명령 출력의 예입니다.

Transaction ID	State	User name	# Msgs/# Acks	Creation time
6800151593984248832	STARTED	guest	3/2	7/19/04 11:03:08 AM

메시지 수와 확인 수를 확인합니다.

메시지 수가 더 많은 경우 생성자가 개별 메시지를 보내는 중일 수 있지만 트랜잭션을 완결하지 못한 것입니다. 브로커는 완결을 수신할 때까지 해당 트랜잭션에 대한 메시지를 경로 지정 및 전달할 수 없습니다.

확인 수가 더 많은 경우 사용자가 개별 메시지에 대한 확인을 보내는 중일 수 있지만 트랜잭션을 완결하지 못한 것입니다. 브로커는 완결을 수신할 때까지 해당 트랜잭션에 대한 확인을 제거할 수 없습니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

코딩 오류를 수정하려면 응용 프로그램 개발자에게 문의하십시오.

**사용자가 메시지를 확인할 수 없습니다.**  
**문제의 원인을 확인하려면 다음 작업을 수행합니다.**

응용 프로그램 개발자에게 문의하여 응용 프로그램이 시스템 기반 확인을 사용하는지 클라이언트 기반 확인을 사용하는지 확인합니다. 응용 프로그램이 시스템 기반 확인을 사용하는 경우 이 절을 건너뛸니다.

응용 프로그램이 클라이언트 기반 확인을 사용하는 경우(CLIENT\_ACKNOWLEDGE 유형) 먼저 클라이언트에 저장된 메시지 수를 줄입니다. 다음과 같은 명령을 사용합니다.

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=1
```

그런 다음 사용자가 느리기 때문에 브로커가 메시지를 버퍼링하는지 아니면 사용자가 메시지를 빠르게 처리하지만 메시지를 확인하지 않았는지를 확인합니다.

다음 명령을 사용하여 대상을 나열합니다.

```
imqcmd list dst
```

사용자 아이디와 비밀번호를 입력하면 다음과 같은 메시지가 출력됩니다.

```
Listing all the destinations on the broker specified by:
-----
Host          Primary Port
-----
localhost     7676
-----

```

Name	Type	State	Producers	Consumers	Msgs Total Count	UnAck	Avg Size
MyDest	Queue	RUNNING	0	0	5	200	1177.0
mq.sys.dmq	Queue	RUNNING	0	0	35	0	1422.0

```
-----
Successfully listed destinations.
```

UnAck 수는 브로커가 보낸 다음 확인을 대기 중인 메시지 수를 나타냅니다. UnAck 수가 높거나 증가하는 경우 브로커가 메시지를 보내고 있는 중이므로 느린 사용자를 기다리지 않습니다. 또한 사용자가 메시지를 확인하지 않고 있음을 알 수 있습니다.

**문제를 해결하려면 다음 작업을 수행합니다.**

코딩 오류를 수정하려면 응용 프로그램 개발자에게 문의하십시오.

***영구 사용자가 비활성 상태입니다.***

문제의 원인을 확인하려면 다음 작업을 수행합니다.

다음 명령 형식을 사용하여 주제의 영구 가입자를 확인합니다.

```
imqcmd list dur -d topicName
```

문제를 해결하려면 다음 작업을 수행합니다.

- `imqcmd purge dur` 명령을 사용하여 영구 사용자를 제거합니다.
- 사용자 응용 프로그램을 다시 시작합니다.

***예기치 않은 브로커 오류가 발생했습니다.***

문제의 원인을 확인하려면 다음 작업을 수행합니다.

272페이지의 "**생성자가 사용자보다 더 빠릅니다.**"에서 설명한 대로 QBrowser를 사용하여 메시지를 확인합니다.

JMS\_SUN\_DMQ\_UNDELIVERED\_REASON 값이 ERROR이면 브로커 오류가 발생한 것입니다.

문제를 해결하려면 다음 작업을 수행합니다.

- 브로커 로그 파일을 검토하여 관련 오류를 확인합니다.
- Sun 기술 지원부에 연락하여 브로커 문제를 보고합니다.

## 참조

- 13장, "명령 참조"
- 14장, "브로커 등록 정보 참조"
- 15장, "물리적 대상 등록 정보 참조"
- 16장, "관리 대상 객체 속성 참조"
- 17장, "JMS 자원 어댑터 속성 참조"
- 18장, "메트릭 참조"





## 명령 참조

이 장에서는 공통 명령줄 구문에 대한 설명과 각 Message Queue 명령에 대한 참조 정보를 제공합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 280페이지의 "명령줄 구문"
- 282페이지의 "imqbrokerd"
- 287페이지의 "imqcmd"
- 297페이지의 "imqobjmgr"
- 300페이지의 "imqdbmgr"
- 302페이지의 "imqusermgr"
- 304페이지의 "imqsvcadmin"
- 306페이지의 "imqkeytool"

## 명령줄 구문

Message Queue 명령줄 유틸리티는 셸 명령입니다. 유틸리티 이름이 명령이고 하위 명령 또는 옵션은 해당 명령에 전달되는 인수입니다. 따라서 유틸리티를 시작하거나 종료하는 명령은 없으며 필요하지도 않습니다.

모든 명령줄 유틸리티는 다음과 같은 명령 구문을 공유합니다.

```
Utility_Name [subcommand] [argument] [[-option_name [-option_argument]]...]
```

*Utility\_Name*은 Message Queue 유틸리티의 이름(예: `mqcmd`, `mqobjmgr`, `mqusermgr` 등)을 지정합니다.

## 명령 입력 규칙

다음은 명령 입력을 위한 일반 규칙입니다.

- 하위 명령(유틸리티가 두 가지 피연산자 유형을 모두 받아들이는 경우 인수도 포함) *다음에* 옵션을 지정합니다.
- 옵션 값에 공백이 포함되어 있는 경우 전체 값을 따옴표로 묶습니다. 일반적으로 속성-값 쌍을 따옴표로 묶는 것이 가장 안전합니다.
- 명령줄에 `-v`(버전)나 `-h/-H`(도움말) 옵션을 지정하면 해당 명령줄의 다른 부분은 실행되지 않습니다.
- 하위 명령, 인수, 옵션 및 옵션 인수는 공백으로 구분합니다.

## 명령줄 예

다음은 하위 명령절이 없는 명령줄의 예입니다. 이 명령은 기본 브로커를 시작합니다.

```
mqbroker
```

다음 명령은 좀 더 복잡합니다. 이 명령은 `myQueue`라는 이름의 `queue` 유형의 대상을 완전 삭제합니다. 인증은 사용자 `admin`을 기반으로 수행되고 명령에서는 사용자의 비밀번호를 묻는 메시지를 표시합니다. `-f` 옵션은 확인이 없음을 지정하고 `-s` 옵션은 명령이 자동 모드로 실행됨을 지정합니다.

```
mqcmd destroy dst -t q -n myQueue -u admin -f -s
```

## 공통 명령 옵션

표 13-1은 모든 Message Queue 관리 유틸리티에서 공통적으로 사용하는 옵션을 설명합니다. 이러한 옵션은 명령줄에서 하위 명령의 *뒤에* 지정해야 합니다. 옵션을 입력하는 순서는 상관이 없습니다.

**표 13-1**      공통 Message Queue 명령줄 옵션

옵션	설명
-h	지정한 유틸리티의 사용 도움말을 표시합니다.
-H	속성 목록 및 예를 포함하여 확장 사용 도움말을 표시합니다(imqcmd및 imqobjmgr만 지원).
-s	자동 모드를 시작합니다. 표시되는 출력이 없습니다. imqbrokerd의 경우 -silent로 지정합니다.
-v	버전 정보를 표시합니다.
-f	사용자 확인 프롬프트 없이 해당 작업을 수행합니다.
-pre	(imqobjmgr에서만 사용) 미리 보기 모드를 시작하며, 사용자는 실제로 명령을 수행하지 않은 상태에서 명령줄 나머지 부분의 실행 결과를 확인할 수 있습니다. 기본 속성 값을 확인할 때 유용합니다.
-javahome <i>path</i>	Java 2와 호환할 수 있는 대체 런타임을 지정하여 사용합니다(기본값은 시스템의 런타임 또는 Message Queue와 함께 제공되는 런타임 사용).

# imgbrokerd

imgbrokerd 명령은 브로커를 시작합니다. 명령줄 옵션은 현재 브로커 세션에 대해서만 브로커 구성 파일의 값을 무시합니다.

## 구문

```
imgbrokerd [[ -Dproperty=value]...]
  [ -backup fileName]
  [ -cluster "broker1" [[, broker2]...] ]
  [ -dbuser userName]
  [ -force]
  [ -h|-help]
  [ -javahome path]
  [ -license licenseName]
  [ -loglevel level]
  [ -metrics interval]
  [ -name instanceName]
  [ -passfile fileName]
  [ -port number]
  [ -remove instance]
  [ -reset data]
  [ -restore fileName]
  [ -shared]
  [ -silent|-s] [ -tty]
  [ -upgrade-store-nobackup]
  [ -version]
  [ -vmargs arg1 [[arg2]...]
```

## 명령 옵션

표 13-2에는 imgbrokerd 명령의 옵션과 각 옵션에 영향을 받는 구성 등록 정보(있는 경우)에 대한 설명이 나와 있습니다.

**표 13-2** imgbrokerd 옵션

옵션	영향을 받는 등록 정보	설명
-backup <i>fileName</i>	없음	브로커 클러스터에만 적용됩니다. 마스터 브로커의 구성 변경 기록을 지정된 파일에 백업합니다. 201페이지의 "구성 변경 기록 관리"를 참조하십시오.

표 13-2 imqbrokerd 옵션(계속)

옵션	영향을 받는 등록 정보	설명
-cluster " <i>broker1</i> [ <i>,broker2</i> ] ..."	imq.cluster.brokerlist를 연결할 브로커의 목록으로 대체합니다.	브로커 클러스터에만 적용됩니다. 지정된 호스트 및 포트의 모든 브로커에 연결합니다. 이 목록은 imq.cluster.brokerlist 등록 정보의 목록과 병합됩니다. <i>host</i> 에 값을 지정하지 않으면 localhost가 사용됩니다. <i>port</i> 에 값을 지정하지 않으면 7676이 사용됩니다. 이 옵션을 사용해서 여러 브로커에 연결하는 방법에 대한 자세한 내용은 195페이지의 " <a href="#">브로커 클러스터 작업</a> "을 참조하십시오.
여기서 <i>broker</i> 는 다음 중 하나입니다. <ul style="list-style-type: none"> <li>• <i>host</i></li> <li>• <i>:port</i></li> <li>• <i>host:port</i></li> </ul>		
-dbpassword <i>password</i>	imq.persist.jdbc.password를 지정한 비밀번호로 대체합니다.	플러그인 JDBC 호환 데이터 저장소의 비밀번호를 지정합니다. 이 옵션은 더 이상 사용되지 않으며 향후 버전에서는 제거됩니다. 다음 중 하나를 대신 사용합니다. <ul style="list-style-type: none"> <li>• 비밀번호를 묻는 프롬프트가 표시되도록 명령줄에서 비밀번호를 생략합니다.</li> <li>• -passfile 옵션을 사용하여 데이터베이스 비밀번호가 들어 있는 파일을 지정합니다.</li> </ul>
-dbuser <i>userName</i>	imq.persist.jdbc.user를 지정한 사용자 아이디로 대체합니다.	플러그인 JDBC 호환 데이터베이스의 사용자 아이디를 지정합니다. 99페이지의 " <a href="#">영구 저장소 설정</a> "을 참조하십시오.
-Dproperty= <i>value</i>	시스템 등록 정보를 설정합니다. 인스턴스 구성 파일의 해당 등록 정보를 값을 대체합니다.	지정한 등록 정보를 지정한 값으로 설정합니다. 브로커 구성 등록 정보에 대한 자세한 내용은 14장, " <a href="#">브로커 등록 정보 참조</a> "를 참조하십시오. <b>주의:</b> -D 옵션으로 설정한 등록 정보의 맞춤법과 형식에 주의하십시오. 잘못된 값을 전달한 경우 시스템에서는 경고를 하지 않으며 Message Queue에서는 해당 값을 설정할 수 없습니다.
-force	없음	사용자의 확인 없이 작업을 수행합니다. 이 옵션은 일반적으로 확인이 필요한 -remove instance 및 -upgrade-store-nobackup 옵션에만 적용됩니다.
-h -help	없음	도움말을 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.
-javahome <i>path</i>	없음	대체 Java 2 호환 JDK의 경로를 지정합니다. 기본값은 번들형 런타임을 사용하는 것입니다.

표 13-2 imqbrokerd 옵션(계속)

옵션	영향을 받는 등록 정보	설명
<code>-ldappassword password</code>	<code>imq.user_repository.ldap.password</code> 를 지정한 비밀번호로 대체합니다.	LDAP 사용자 저장소에 액세스할 때 필요한 비밀번호를 지정합니다. 이 옵션은 더 이상 사용되지 않으며 향후 버전에서는 제거됩니다. 다음 중 하나를 대신 사용합니다. <ul style="list-style-type: none"> <li>비밀번호를 묻는 프롬프트가 표시되도록 명령줄에서 비밀번호를 생략합니다.</li> <li><code>-passfile</code> 옵션을 사용하여 LDAP 비밀번호가 들어 있는 파일을 지정합니다.</li> </ul>
<code>-license [licenseName]</code>	없음	로드할 사용권이 Message Queue 제품의 기본값과 다른 경우 해당 사용권을 지정합니다. 사용권 이름을 지정하지 않으면 시스템에 설치된 모든 사용권이 표시됩니다. <code>licenseName</code> 값은 설치된 Message Queue 버전에 따라 <code>pe</code> (플랫폼판-기본 기능), <code>try</code> (플랫폼판-90일 시험 엔터프라이즈 기능), <code>unl</code> (엔터프라이즈 판)입니다.
<code>-loglevel level</code>	<code>imq.broker.log.level</code> 을 지정한 수준으로 대체합니다.	로깅 수준을 NONE, ERROR, WARNING 또는 INFO 중 하나로 지정합니다. 기본값은 INFO입니다.
<code>-metrics interval</code>	<code>imq.metrics.interval</code> 을 지정한 시간(초)으로 대체합니다.	지정된 간격(초)으로 브로커 메트릭이 로거에 기록되도록 지정합니다.
<code>-name instanceName</code>	<code>imq.instancename</code> 을 지정한 이름으로 설정합니다.	이 브로커의 인스턴스 이름을 지정하고 해당 인스턴스 구성 파일을 사용합니다. 브로커 이름을 지정하지 않으면 인스턴스 이름은 <code>imqbroker</code> 로 설정됩니다. <b>주:</b> 같은 호스트에서 두 개 이상의 브로커 인스턴스를 실행하는 경우에는 각각에 고유한 이름이 있어야 합니다.
<code>-passfile fileName</code>	<code>imq.passfile.enabled</code> 를 대체하여 <code>true</code> 로 설정합니다. <code>imq.passfile.dirpath</code> 를 파일이 포함된 경로로 대체합니다. <code>imq.passfile.name</code> 을 파일 이름으로 대체합니다.	<code>imqcmd</code> 명령 유틸리티, SSL 키 저장소, LDAP 사용자 저장소, JDBC 호환 데이터베이스 또는 이들의 조합에 대한 비밀번호를 읽을 파일의 이름을 지정합니다. 자세한 내용은 169페이지의 "Passfile 사용"을 참조하십시오.

표 13-2 imqbrokerd 옵션(계속)

옵션	영향을 받는 등록 정보	설명
-password <i>keypassword</i>	img.keystore.password를 지정한 비밀번호로 대체합니다.	SSL 인증서 키 저장소에 대한 비밀번호를 지정합니다. 이 옵션은 더 이상 사용되지 않으며 향후 버전에서는 제거됩니다. 다음 중 하나를 대신 사용합니다. <ul style="list-style-type: none"> <li>비밀번호를 묻는 프롬프트가 표시되도록 명령줄에서 비밀번호를 생략합니다.</li> <li>-passfile 옵션을 사용하여 SSL 인증서 키 저장소 비밀번호가 들어 있는 파일을 지정합니다.</li> </ul>
-port <i>number</i>	img.portmapper.port를 지정한 번호로 대체합니다.	브로커의 포트 매퍼 포트 번호를 지정합니다. 이 값은 기본적으로 7676으로 설정됩니다. 같은 서버에서 두 개의 브로커 인스턴스를 실행하려면 각 브로커의 포트 매퍼에 서로 다른 포트 번호가 있어야 합니다. Message Queue 클라이언트는 이 포트 번호를 사용하여 브로커 인스턴스에 연결합니다.
-remove instance	없음	브로커 인스턴스를 제거합니다. 인스턴스 구성 파일, 로그 파일, 영구 저장소, 인스턴스와 관련된 기타 파일 및 디렉토리를 삭제합니다. -force 옵션을 함께 지정한 경우가 아니면 사용자의 확인이 필요합니다.
-reset store  messages  durables  props	없음	주어진 인수에 따라 데이터 저장소(또는 데이터 저장소의 일부) 또는 브로커 인스턴스의 등록 정보를 재설정합니다.  데이터 저장소를 재설정하면 지속성 메시지, 영구 가입, 트랜잭션 정보 등의 모든 지속성 데이터가 지워집니다. 그러면 브로커 인스턴스를 초기 상태로 시작할 수 있습니다. 지속성 메시지만 지우거나 영구 가입만 지울 수도 있습니다. 이후에 다시 시작할 때 영구 저장소가 재설정되지 않게 하려면 -reset 옵션을 사용하지 않고 브로커 인스턴스를 다시 시작합니다.  브로커의 등록 정보를 재설정하면 기존 인스턴스 구성 파일(config.properties)이 빈 파일로 교체됩니다. 모든 등록 정보에 기본값이 설정됩니다.
-restore <i>fileName</i>	없음	브로커 클러스터에만 적용됩니다. 마스터 브로커의 구성 변경 레코드를 지정한 백업 파일로 교체합니다. 이 파일은 -backup 옵션을 사용해서 미리 작성해 두어야 합니다. 201페이지의 "구성 변경 기록 관리"를 참조하십시오.
-shared	img.jms.threadpool_model을 대체하여 shared로 설정합니다.	연결 간에 스레드를 공유하여 브로커 인스턴스가 지원하는 연결 수가 늘어나도록 jms 연결 서비스가 공유 스레드 풀 모델을 사용해서 구현되게 지정합니다.

**표 13-2** imqbrokerd 옵션(계속)

옵션	영향을 받는 등록 정보	설명
-silent -s	imq.log.console.output을 대체하여 NONE으로 설정합니다.	콘솔에 대한 로깅을 끕니다.
-tty	imq.log.console.output을 대체하여 ALL로 설정합니다.	모든 메시지가 콘솔에 표시되도록 설정합니다. 기본적으로 WARNING 및 ERROR 수준 메시지만 표시됩니다.
-upgrade-store-nobackup	없음	호환되지 않는 버전에서 Message Queue 3.5 또는 Message Queue 3.5 SPx로 업그레이드할 때 이전 데이터 저장소를 자동으로 제거하도록 지정합니다. 자세한 내용은 <i>Message Queue 설치 설명서</i> 를 참조하십시오.
-version	없음	설치된 제품의 버전 번호를 표시합니다.
-vmargs arg1 [[arg2]...]	없음	Java VM으로 전달할 인수를 지정합니다. 인수는 공백으로 구분합니다. 인수를 두 개 이상 전달하거나 인수에 공백이 포함된 경우에는 따옴표를 사용합니다. 예를 들면 다음과 같습니다. imqbrokerd -tty -vmargs "-Xmx128m -Xincgc" 이러한 인수는 명령줄에서만 전달할 수 있습니다. config.props 파일에는 연결된 구성 등록 정보가 없습니다.

## 참조

imqbrokerd 사용 및 명령 예에 대한 자세한 내용은 [67페이지의 "대화식으로 브로커 시작"](#)을 참조하십시오.



# imqcmd

imqcmd 명령 유틸리티를 사용하여 브로커와 해당 서비스를 관리할 수 있습니다.

## 구문

```
imqcmd subcommand argument [options]
imqcmd -h|H
imqcmd -v
```

## 하위 명령

도움말이나 제품 버전을 표시하려는 경우가 아니라면 항상 imqcmd와 함께 하위 명령을 사용하게 됩니다. 표 13-3에서는 imqcmd 하위 명령을 나열하고 해당 하위 명령에 대한 참조 정보가 있는 위치를 지정합니다.

**표 13-3** imqcmd 하위 명령

하위 명령 및 인수	설명	참조
commit txn	트랜잭션을 완결합니다.	293페이지의 "트랜잭션 관리 하위 명령"
destroy dur	영구 가입을 완전 삭제합니다.	293페이지의 "영구 가입 하위 명령"
list cxn	브로커에 대한 연결을 나열합니다.	293페이지의 "연결 하위 명령"
list dur	주체의 영구 가입을 나열합니다.	293페이지의 "영구 가입 하위 명령"
list svc	브로커의 서비스를 나열합니다.	292페이지의 "연결 서비스 관리 하위 명령"
list txn	브로커의 트랜잭션을 나열합니다.	293페이지의 "트랜잭션 관리 하위 명령"
metrics bkr	브로커 메트릭을 표시합니다.	289페이지의 "브로커 관리 하위 명령"
metrics svc	서비스 메트릭을 표시합니다.	292페이지의 "연결 서비스 관리 하위 명령"
pause bkr	브로커의 모든 서비스를 일시 중지합니다.	289페이지의 "브로커 관리 하위 명령"
pause svc	브로커에서 단일 서비스를 일시 중지합니다.	292페이지의 "연결 서비스 관리 하위 명령"

**표 13-3** imqcmd 하위 명령(계속)

하위 명령 및 인수	설명	참조
purge dur	영구 가입을 완전 삭제하지 않고 영구 가입의 모든 메시지를 제거합니다.	293페이지의 "영구 가입 하위 명령"
query bkr	브로커의 정보를 쿼리 및 표시합니다.	289페이지의 "브로커 관리 하위 명령"
query cxn	연결의 정보를 쿼리 및 표시합니다.	293페이지의 "연결 하위 명령"
query svc	서비스의 정보를 쿼리 및 표시합니다.	292페이지의 "연결 서비스 관리 하위 명령"
query txn	트랜잭션의 정보를 쿼리 및 표시합니다.	293페이지의 "트랜잭션 관리 하위 명령"
reload cls	브로커 클러스터 구성을 다시 로드합니다.	289페이지의 "브로커 관리 하위 명령"
restart bkr	현재 실행 중인 브로커 인스턴스를 다시 시작합니다.	289페이지의 "브로커 관리 하위 명령"
resume bkr	브로커의 모든 서비스를 다시 시작합니다.	289페이지의 "브로커 관리 하위 명령"
resume svc	한 서비스를 다시 시작합니다.	292페이지의 "연결 서비스 관리 하위 명령"
rollback txn	트랜잭션을 롤백합니다.	293페이지의 "트랜잭션 관리 하위 명령"
shutdown bkr	브로커 인스턴스를 종료합니다.	289페이지의 "브로커 관리 하위 명령"
update bkr	브로커의 속성을 업데이트합니다.	289페이지의 "브로커 관리 하위 명령"
update svc	서비스의 속성을 업데이트합니다.	292페이지의 "연결 서비스 관리 하위 명령"

또한, imqcmd 명령 유틸리티에는 브로커의 물리적 대상에 사용할 하위 명령이 있습니다. 대상 하위 명령은 6장, "물리적 대상 관리"에 설명되어 있습니다.

다음 절에서는 imqcmd 하위 명령을 기능별로 나열합니다.

## 브로커 관리 하위 명령

표 13-4에서는 브로커 관리에 사용되는 imqcmd 하위 명령을 나열합니다. 호스트 이름 또는 포트가 지정되어 있지 않으면 기본값(localhost:7676)인 것으로 가정합니다.

표 13-4 브로커 관리에 사용되는 imqcmd 하위 명령

하위 명령 구문	설명
<pre>metrics bkr [-b <i>hostName:port</i>]             [-m <i>metricType</i>]             [-int <i>interval</i>]             [-msp <i>numSamples</i>]</pre>	<p>기본 브로커 또는 지정한 호스트 및 포트의 브로커 메트릭을 표시합니다.</p> <p>-m 옵션을 사용하여 표시할 메트릭 유형을 지정합니다.</p> <p><b>tt1</b> 브로커에 유입 및 유출되는 메시지와 패킷의 메트릭을 표시합니다(기본 메트릭 유형).</p> <p><b>rts</b> 브로커에 유입 및 유출되는 메시지와 패킷의 메트릭을 초당 속도로 표시합니다.</p> <p><b>cxm</b> 연결, 가상 메모리 힙 및 스레드를 표시합니다.</p> <p>-int 옵션을 사용하여 메트릭 표시 간격(초)을 지정합니다. 기본값은 5초입니다.</p> <p>-msp 옵션을 사용하여 출력에 표시되는 샘플 수를 지정합니다. 기본값은 무제한 수입니다(무한).</p>
<pre>pause bkr [-b <i>hostName:port</i>]</pre>	<p>기본 브로커 또는 지정한 호스트 및 포트의 브로커를 일시 중지합니다. 113페이지의 "브로커 일시 중지 및 다시 시작"을 참조하십시오.</p>
<pre>query bkr -b <i>hostName:port</i></pre>	<p>기본 브로커 또는 지정한 호스트 및 포트의 브로커에 해당하는 현재 등록 정보 설정을 나열합니다. 지정한 브로커에 연결된 실행 중인 브로커(브로커가 여러 개인 클러스터에서)의 목록도 표시합니다.</p>
<pre>reload cls</pre>	<p>브로커 클러스터에만 적용됩니다. 클러스터에 있는 모든 브로커에서 imq.cluster.brokerlist 등록 정보를 다시 로드하고 클러스터 정보를 업데이트하도록 합니다. 자세한 내용은 199페이지의 "클러스터에 브로커 추가"를 참조하십시오.</p>
<pre>restart bkr [-b <i>hostName:port</i>]</pre>	<p>브로커를 시작했을 때 지정한 옵션을 사용하여 기본 브로커 또는 지정한 호스트 및 포트의 브로커를 종료하고 다시 시작합니다.</p>
<pre>resume bkr [-b <i>hostName:port</i>]</pre>	<p>기본 브로커 또는 지정한 호스트 및 포트의 브로커를 다시 시작합니다.</p>
<pre>shutdown bkr [-b <i>hostName:port</i>]</pre>	<p>기본 브로커 또는 지정한 호스트 및 포트의 브로커를 종료합니다.</p>
<pre>update bkr [-b <i>hostName:port</i>]            -o <i>attribute=value</i>            [-o <i>attribute=value1</i>]...</pre>	<p>기본 브로커 또는 지정한 호스트 및 포트의 브로커에 해당하는 지정된 속성을 변경합니다.</p>

## 물리적 대상 관리 하위 명령

표 13-5에서는 물리적 대상 관리에 사용되는 `imgcmd` 하위 명령을 나열합니다. 호스트 이름 또는 포트가 지정되어 있지 않으면 기본값(`localhost:7676`)인 것으로 가정합니다.

**표 13-5** 대상 관리에 사용되는 `imgcmd` 하위 명령

하위 명령 구문	설명
<code>compact dst [-t destType -n destName]</code>	기본 제공 파일 기반 데이터 저장소에서 지정된 유형과 이름의 대상을 압축합니다. 대상 유형과 이름을 지정하지 않으면 모든 대상이 압축됩니다. 대상을 압축하려면 먼저 일시 중지해야 합니다.
<code>create dst -t destType -n destName [-o attribute=value] [-o attribute=value1]...</code>	지정한 유형의 대상을 지정한 이름과 지정한 속성을 사용하여 만듭니다. 대상 이름은 영숫자(공백 없음)만 포함하고 영문자나 "_" 및 "\$" 문자로 시작할 수 있습니다. "mq" 문자 열로는 시작할 수 없습니다.  마스터 브로커를 일시적으로 사용할 수 없는 클러스터에서는 이 작업을 수행할 수 없습니다.
<code>destroy dst -t destType -n destName</code>	지정한 유형과 이름의 대상을 완전 삭제합니다. 사용 불가능 메시지 대기열처럼 시스템에서 생성된 대상은 완전 삭제할 수 없습니다.  마스터 브로커를 일시적으로 사용할 수 없는 클러스터에서는 이 작업을 수행할 수 없습니다.
<code>list dst [-t destType] [-tmp]</code>	지정된 유형의 대상을 모두 나열합니다. 임시 대상을 함께 나열할 수도 있습니다.  유형 인수는 다음과 같은 두 가지 값을 가질 수 있습니다. <i>destType</i> = q (대기열) <i>destType</i> = t (주제)  유형을 지정하지 않으면 모든 유형의 대상이 모두 나열됩니다.

**표 13-5** 대상 관리에 사용되는 imqcmd 하위 명령(계속)

하위 명령 구문	설명
<pre>metrics dst -t destType -n destName [-m metricType] [-int interval] [-msp numSamples]</pre>	<p>지정한 유형과 이름의 대상에 대한 메트릭 정보를 표시합니다.</p> <p><b>-m</b> 옵션을 사용하여 표시할 메트릭 유형을 지정합니다.</p> <p><b>ttl</b> 대상에 유입 및 유출되고 메모리에 있는 메시지와 패킷의 메트릭을 표시합니다(기본 메트릭 유형).</p> <p><b>rts</b> 대상에 유입 및 유출되는 메시지와 패킷의 메트릭을 초당 속도로 나타내고 기타 속도 정보를 표시합니다.</p> <p><b>con</b> 사용자 관련 메트릭을 표시합니다.</p> <p><b>dsk</b> 디스크 사용 메트릭을 표시합니다.</p> <p><b>-int</b> 옵션을 사용하여 메트릭 표시 간격(초)을 지정합니다. 기본값은 5초입니다.</p> <p><b>-msp</b> 옵션을 사용하여 출력에 표시되는 샘플 수를 지정합니다. 기본값은 무제한 수입니다(무한).</p>
<pre>pause dst [-t destType -n destName] [-pst pauseType]</pre>	<p>지정된 유형과 이름의 대상에서 사용자에게 메시지 전달(-pst CONSUMERS), 생성자로부터 메시지 전달(-pst PRODUCERS) 또는 두 가지 모두(-pst ALL)를 일시 중지합니다. 대상 유형과 이름을 지정하지 않으면 모든 대상이 일시 중지됩니다. 기본값은 ALL입니다.</p>
<pre>purge dst -t destType -n destName</pre>	<p>지정한 유형과 이름의 대상에서 메시지를 제거합니다.</p>
<pre>query dst -t destType -n destName</pre>	<p>지정한 유형과 이름의 대상에 대한 정보를 나열합니다.</p>
<pre>resume dst [-t destType -n destName]</pre>	<p>지정된 유형과 이름의 일시 중지된 대상에서 메시지 전달을 다시 시작합니다. 대상 유형과 이름을 지정하지 않으면 모든 대상이 다시 시작됩니다.</p>
<pre>update dst -t destType -n destName -o attribute=value [-o attribute=value1]...</pre>	<p>지정한 대상에서 지정한 속성의 값을 업데이트합니다.</p> <p>대상이 사용 불가능 메시지 대기열 mq.sys.dmq가 아닌 경우 <a href="#">표 15-1</a>에 설명된 속성 중 하나를 속성 이름으로 사용할 수 있습니다.</p>

## 연결 서비스 관리 하위 명령

표 13-6에 연결 서비스 관리에 사용되는 imqcmd 하위 명령이 나열되어 있습니다. 호스트 이름 또는 포트가 지정되어 있지 않으면 기본값(localhost:7676)인 것으로 가정합니다.

**표 13-6** 연결 서비스 관리에 사용되는 imqcmd 하위 명령

하위 명령 구문	설명
<code>list svc [-b <i>hostName:port</i>]</code>	기본 브로커 또는 지정한 호스트 및 포트의 브로커에 있는 모든 연결 서비스를 나열합니다.
<code>metrics svc -n <i>serviceName</i> [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</code>	기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 지정된 서비스의 메트릭을 표시합니다. -m 옵션을 사용하여 표시할 메트릭 유형을 지정합니다. <b>ttl</b> 브로커에 유입 및 유출되는 메시지와 패킷의 메트릭을 지정된 서비스 방법으로 표시합니다(기본 메트릭 유형). <b>rts</b> 브로커에 메시지와 패킷이 유입 및 유출되는 초당 속도에 대한 메트릭을 지정된 연결 서비스를 통해 표시합니다. <b>cxm</b> 연결, 가상 메모리 힙 및 스레드를 표시합니다. -int 옵션을 사용하여 메트릭 표시 간격(초)을 지정합니다. 기본값은 5초입니다. -msp 옵션을 사용하여 출력에 표시되는 샘플 수를 지정합니다. 기본값은 무제한 수입입니다(무한).
<code>pause svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</code>	기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 실행 중인, 지정된 서비스를 일시 중지합니다. 관리 서비스는 일시 중지할 수 없습니다.
<code>query svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</code>	기본 브로커 또는 지정한 호스트 및 포트의 서비스에서 실행 중인, 지정된 서비스에 대한 정보를 표시합니다.
<code>resume svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</code>	기본 브로커 또는 지정한 호스트 및 포트에서 실행 중인, 지정된 서비스를 다시 시작합니다.
<code>update svc -n <i>serviceName</i> [-b <i>hostName:port</i>] -o <i>attribute=value</i> [-o <i>attribute=value1</i>]...</code>	기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 실행 중인, 지정된 서비스의 지정된 속성을 업데이트합니다. 서비스 속성에 대한 설명은 311페이지의 "연결 서비스 등록 정보"를 참조하십시오.

## 연결 하위 명령

표 13-7에 연결에 적용되는 imqcmd 하위 명령이 나열되어 있습니다. 호스트 이름 또는 포트가 지정되어 있지 않으면 localhost, 7676인 것으로 가정합니다.

**표 13-7** 연결 서비스 관리에 사용되는 imqcmd 하위 명령

하위 명령 구분	설명
list cxn [-svn <i>serviceName</i> ] [-b <i>hostName:port</i> ]	기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 지정된 서비스 이름의 연결을 모두 나열합니다. 서비스 이름을 지정하지 않는 경우 모든 연결이 나열됩니다.
query cxn -n <i>connectionID</i> [-b <i>hostName:port</i> ]	기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 지정된 연결에 대한 정보를 표시합니다.

## 영구 가입 하위 명령

표 13-8에서는 imqcmd 영구 가입 하위 명령을 요약합니다. 호스트 이름 또는 포트가 지정되어 있지 않으면 기본값(localhost:7676)인 것으로 가정합니다.

**표 13-8** 영구 가입 관리에 사용되는 imqcmd 하위 명령

하위 명령	설명
list dur -d <i>destName</i>	지정한 대상의 모든 영구 가입을 나열합니다.
destroy dur -n <i>subscrName</i> -c <i>client_id</i>	지정한 클라이언트 식별자에 해당하는 지정된 영구 가입을 완전 삭제합니다. 마스터 브로커를 일시적으로 사용할 수 없는 클러스터에서는 이 작업을 수행할 수 없습니다.
purge dur -n <i>subscrName</i> -c <i>client_id</i>	지정한 클라이언트 식별자에 해당하는 지정된 영구 가입에 대한 모든 메시지를 제거합니다.

## 트랜잭션 관리 하위 명령

표 13-9에서는 imqcmd 트랜잭션 하위 명령을 요약합니다. 호스트 이름 또는 포트가 지정되어 있지 않으면 기본값(localhost:7676)인 것으로 가정합니다.

**표 13-9** 트랜잭션 관리에 사용되는 imqcmd 하위 명령

하위 명령	설명
list txn	브로커에서 추적하는 모든 트랜잭션을 나열합니다.
query txn -n <i>transaction_id</i>	지정한 트랜잭션에 대한 정보를 나열합니다.
commit txn -n <i>transaction_id</i>	지정한 트랜잭션을 완결합니다.
rollback txn -n <i>transaction_id</i>	지정한 트랜잭션을 롤백합니다.

## 명령 옵션

표 13-10에 imqcmd 명령의 옵션이 나열되어 있습니다.

**표 13-10** imqcmd 옵션

옵션	설명
-b <i>hostName:port</i>	브로커의 호스트 이름과 해당 포트 번호를 지정합니다. 기본값은 localhost:7676입니다. 포트만 지정하려면: -b :7878 이름만 지정하려면: -b somehost
-c <i>clientID</i>	주제에 영구 가입의 아이디를 지정합니다. <a href="#">122페이지의 "영구 가입 관리"</a> 를 참조하십시오.
-d <i>destinationName</i>	주제의 이름을 지정합니다. list dur 및 destroy dur 하위 명령에 사용됩니다. <a href="#">122페이지의 "영구 가입 관리"</a> 를 참조하십시오.
-f	사용자의 확인 없이 작업을 수행합니다.
-h	사용 도움말을 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다. 이 옵션에는 사용자 아이디와 비밀번호가 필요하지 않습니다.
-H	사용 도움말, 속성 목록 및 예를 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다. 이 옵션에는 사용자 아이디와 비밀번호가 필요하지 않습니다.
-int <i>interval</i>	metrics bkr, metrics dst 및 metrics svc 하위 명령이 메트릭 출력을 표시하는 간격(초)을 지정합니다.
-javahome <i>path</i>	Java 2와 호환할 수 있는 대체 런타임을 지정하여 사용합니다(기본값은 시스템의 런타임 또는 Message Queue와 함께 제공되는 런타임 사용).



표 13-10 imqcmd 옵션 (계속)

옵션	설명
-m <i>metricType</i>	표시할 메트릭 정보의 유형을 지정합니다. 이 옵션은 <code>metrics dst</code> , <code>metrics svc</code> 또는 <code>metrics bkr</code> 하위 명령에 사용됩니다. <i>metricType</i> 값은 메트릭이 대상, 서비스 또는 브로커 중 어느 것에 대해 생성되었는지에 따라 다릅니다.
-msp <i>numSamples</i>	<code>metrics bkr</code> , <code>metrics dst</code> 및 <code>metrics svc</code> 하위 명령이 메트릭 출력에 표시하는 메트릭 샘플 수를 지정합니다.
-n <i>argumentName</i>	하위 명령 인수의 이름을 지정합니다. 하위 명령에 따라 서비스나 물리적 대상, 영구 가입의 이름일 수도 있고 연결 아이디나 트랜잭션 아이디일 수도 있습니다.
-o <i>attribute=value</i>	속성의 값을 지정합니다. 하위 명령 인수에 따라 브로커(108페이지의 "imqcmd 명령 유틸리티 사용" 참조), 서비스(116페이지의 "연결 서비스 관리" 참조) 또는 대상(122페이지의 "영구 가입 관리" 참조)의 속성이 될 수 있습니다.
-p <i>password</i>	자신(관리자)의 비밀번호를 지정합니다. 이 옵션은 더 이상 사용되지 않으며 향후 릴리스에서는 지원되지 않습니다. 다음 중 하나를 대신 사용합니다. <ul style="list-style-type: none"> <li>비밀번호를 묻는 프롬프트가 표시되도록 명령줄에서 비밀번호를 생략합니다.</li> <li>-passfile 옵션을 사용하여 관리자 비밀번호가 들어 있는 파일을 지정합니다.</li> </ul>
-passfile <i>path</i>	명령을 실행하는 사용자의 비밀번호가 들어 있는 파일의 경로를 지정합니다. 자세한 내용은 169페이지의 "Passfile 사용"을 참조하십시오.
-pst <i>pauseType</i>	대상을 일시 중지할 때 생성자나 사용자 또는 둘 다를 일시 중지할지 여부를 지정합니다. 122페이지의 "영구 가입 관리"를 참조하십시오.
-rtm <i>timeout</i>	imqcmd 하위 명령의 초기(다시 시도) 시간 초과 기간(초)을 지정합니다. 시간 초과는 imqcmd 하위 명령이 브로커에 요청한 후 기다리는 시간입니다. 이후 하위 명령이 다시 시도할 때마다 초기 시간 초과 기간의 배수인 시간 초과 값을 사용합니다. 기본값: 10
-rtr <i>numRetries</i>	imqcmd 하위 명령이 처음으로 시간 초과된 후에 다시 시도하는 횟수를 지정합니다. 기본값: 5
-s	비대화형 모드입니다. 출력이 표시되지 않습니다.
-secure	ssladmin 연결 서비스를 사용해서 브로커에 대한 보안 관리 연결을 지정합니다(164페이지의 "4단계. SSL기반 클라이언트 구성 및 실행" 참조). 이 옵션을 생략하면 연결이 보안되지 않습니다.

**표 13-10** imqcmd 옵션(계속)

옵션	설명
-svn <i>serviceName</i>	연결을 나열할 서비스를 지정합니다. <a href="#">121페이지의 "연결 정보 얻기"</a> 를 참조하십시오.
-t <i>destType</i>	대상의 유형을 지정합니다. t (주제) 또는 q (대기열). <a href="#">122페이지의 "영구 가입 관리"</a> 를 참조하십시오.
-tmp	임시 대상을 표시합니다. <a href="#">290페이지의 표 13-5</a> 를 참조하십시오.
-u <i>userName</i>	자신(관리자)의 이름을 지정합니다. 이 값을 생략하면 값을 묻는 메시지가 표시됩니다.
-v	버전 정보를 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다. 이 옵션에는 사용자 아이디와 비밀번호가 필요하지 않습니다.

## 참조

imqbrokerd 사용 및 명령 예에 대한 자세한 내용은 [5장, "브로커 관리"](#) 및 [6장, "물리적 대상 관리"](#)를 참조하십시오.

# imqobjmgr

객체 관리자 유틸리티 `imqobjmgr`은 Message Queue 관리 대상 객체를 만들고 관리합니다.

## 구문

```
imqobjmgr subcommand [options]
imqobjmgr -h|H
imqobjmgr -v
```

## 하위 명령

객체 관리자 유틸리티(`imqobjmgr`)에는 표 13-3에 나열된 하위 명령이 포함되어 있습니다.

**표 13-11** imqobjmgr 하위 명령

하위 명령	설명
add	객체 저장소에 관리 대상 객체를 추가합니다.
delete	객체 저장소에서 관리 대상 객체를 삭제합니다.
list	객체 저장소에 있는 관리 대상 객체를 나열합니다.
query	지정한 관리 대상 객체에 대한 정보를 표시합니다.
update	객체 저장소에 있는 기존 관리 대상 객체를 수정합니다.

## 명령 옵션

표 13-12에는 imqobjmgr 명령의 옵션이 나열되어 있습니다. 사용 설명은 다음에서 해당 작업 기반 절을 참조하십시오.

**표 13-12** imqobjmgr 옵션

옵션	설명
-f	사용자의 확인 없이 작업을 수행합니다.
-h	사용 도움말을 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.
-H	사용 도움말, 속성 목록 및 예를 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.
-i <i>fileName</i>	하위 명령 절의 일부 또는 전부를 포함하는 명령 파일 이름을 지정하며, 객체 유형, 조회 이름, 객체 속성, 객체 저장소 속성 또는 기타 옵션을 지정합니다. 보통은 객체 저장소 속성과 같은 반복 정보에 사용됩니다.
-j <i>attribute=value</i>	JNDI 객체 저장소를 확인하고 액세스할 때 필요한 속성을 지정합니다. <a href="#">174페이지의 "객체 저장소 정보"</a> 를 참조하십시오.
-jvashome <i>path</i>	Java 2와 호환할 수 있는 대체 런타임을 지정하여 사용합니다(기본값은 시스템의 런타임 또는 Message Queue와 함께 제공되는 런타임 사용).
-l <i>lookupName</i>	관리 대상 객체의 JNDI 조회 이름을 지정합니다. 이 이름은 관리 대상 객체 저장소의 컨텍스트에서 고유해야 합니다.
-o <i>attribute=value</i>	관리 대상 객체의 속성을 지정합니다. <a href="#">333페이지의 16장, "관리 대상 객체 속성 참조"</a> 를 참조하십시오.
-pre	미리 보기 모드입니다. 명령을 수행하지 않고 수행 결과를 확인합니다.
-r <i>read-only_state</i>	관리 대상 객체가 읽기 전용인지 여부를 지정합니다. true 값은 관리 대상 객체가 읽기 전용임을 나타냅니다. 클라이언트는 읽기 전용 관리 대상 객체의 속성을 수정할 수 없습니다. 읽기 전용 상태는 기본적으로 false로 설정됩니다.
-s	비대화형 모드입니다. 출력이 표시되지 않습니다.

표 13-12 imqobjmgr 옵션(계속)

옵션	설명
-t <i>objectType</i>	Message Queue 관리 대상 객체의 유형을 지정합니다. q = 대기열 t = 주제 cf = 연결 팩토리 qf = 대기열 연결 팩토리 tf = 주제 연결 팩토리 xcf = XA 연결 팩토리(분산 트랜잭션) xqf = XA 대기열 연결 팩토리(분산 트랜잭션) xtf = XA 주제 연결 팩토리(분산 트랜잭션) e = SOAP endpoint( <i>Message Queue Developer's Guide for Java Clients</i> 에서 설명한 것처럼 이 관리 대상 객체 유형은 SOAP 메시지를 지원하는 데 사용됨)
-v	버전 정보를 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.

## 참조

imqobjmgr 및 명령 예에 대한 자세한 내용은 8장, "관리 대상 객체 관리"를 참조하십시오.

# imqdbmgr

데이터베이스 관리자 유틸리티(imqdbmgr)는 지속성에 필요한 스키마를 설정합니다. imqdbmgr 명령을 사용하여 손상된 Message Queue 데이터베이스 테이블을 삭제하거나 데이터 저장소를 변경할 수도 있습니다.

## 구문

```
imqdbmgr subcommand argument [options]
imqdbmgr -h|-help
imqdbmgr -v|-version
```

## 하위 명령

데이터베이스 관리자 유틸리티(imqdbmgr)에는 표 13-13에 나열된 하위 명령이 포함되어 있습니다.

**표 13-13** imqdbmgr 하위 명령

하위 명령 및 인수	설명
create all	새 데이터베이스와 Message Queue 영구 저장소 스키마를 만듭니다. 이 명령은 내장 데이터베이스 시스템에 사용되며, 사용하는 경우에는 <code>img.persist.jdbc.createdburl</code> 등록 정보를 지정해야 합니다.
create tbl	기존 데이터베이스 시스템에 Message Queue 영구 저장소 스키마를 만듭니다. 이 명령은 외부 데이터베이스 시스템에 사용됩니다.
delete tbl	현재 영구 저장소 데이터베이스에서 기존 Message Queue 데이터베이스 테이블을 삭제합니다.
delete oldtbl	이전 버전의 영구 저장소 데이터베이스에서 Message Queue 데이터베이스 테이블을 모두 삭제합니다. 영구 저장소가 자동으로 현재 버전의 Message Queue로 이전된 이후에 사용됩니다.
recreate tbl	현재 영구 저장소 데이터베이스에서 기존 Message Queue 데이터베이스 테이블을 삭제한 후 Message Queue 영구 저장소 스키마를 다시 만듭니다.
reset lck	영구 저장소 데이터베이스를 다른 프로세스에서 사용할 수 있도록 잠금을 재설정합니다.

## 명령 옵션

표 13-14에는 imqdbmgr 명령의 옵션이 나열되어 있습니다.

표 13-14 imqdbmgr 옵션

옵션	설명
-Dproperty=value	지정한 등록 정보를 지정한 값으로 설정합니다.
-b instanceName	브로커 인스턴스 이름을 지정하고 해당 인스턴스 구성 파일을 사용합니다.
-h	사용 도움말을 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.
-p password	데이터베이스 비밀번호를 지정합니다. 이 옵션은 더 이상 사용되지 않으며 향후 릴리스에서는 지원되지 않습니다. 다음 중 하나를 대신 사용합니다. <ul style="list-style-type: none"> <li>비밀번호를 묻는 프롬프트가 표시되도록 명령줄에서 비밀번호를 생략합니다.</li> <li>-passfile 옵션을 사용하여 데이터베이스 비밀번호가 들어 있는 파일을 지정합니다.</li> </ul>
-passfile path	데이터베이스 비밀번호가 들어 있는 파일의 경로를 지정합니다. 자세한 내용은 <a href="#">169페이지의 "Passfile 사용"</a> 을 참조하십시오.
-u name	데이터베이스 사용자 아이디를 지정합니다.
-v	버전 정보를 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.

## 참조

영구 저장소 설정에 대한 자세한 내용은 [99페이지의 "영구 저장소 설정"](#)을 참조하십시오.

## imqusermgr

사용자 관리자 유틸리티(imqusermgr)를 사용하면 플랫폼 파일 사용자 저장소를 편집하거나 채울 수 있습니다. imqusermgr을 사용하기 전에 주의해야 할 사항은 다음과 같습니다.

- 브로커별 사용자 저장소가 없는 경우 해당 브로커 인스턴스를 시작하여 브로커별 사용자 저장소를 만들어야 합니다.
- imqusermgr 명령은 브로커가 설치되는 호스트에서 실행해야 합니다.
- 이 경우 저장소에 쓸 수 있는 적절한 쓰기 권한이 있어야 합니다. 즉, Solaris 및 Linux의 경우 해당 브로커 인스턴스를 처음으로 만든 사용자나 루트 사용자여야 합니다.

## 구문

```
imqusermgr subcommand [options]
imqusermgr -h
imqusermgr -v
```

## 하위 명령

표 13-15에는 이 장에서 설명하는 imqusermgr 하위 명령이 나열되어 있습니다.

**표 13-15** imqusermgr 하위 명령

하위 명령	설명
add [-i <i>instanceName</i> ] -u <i>userName</i> -p <i>passwd</i> [-g <i>group</i> ] [-s]	지정한(또는 기본) 브로커 인스턴스 저장소에 사용자와 관련 비밀번호를 추가하고 선택적으로 사용자 그룹을 지정합니다.
delete [-i <i>instanceName</i> ] -u <i>userName</i> [-s] [-f]	지정한(또는 기본) 브로커 인스턴스 저장소에서 지정한 사용자를 삭제합니다.
list [-i <i>instanceName</i> ] [-u <i>userName</i> ]	지정한(또는 기본) 브로커 인스턴스 저장소의 지정한 사용자 또는 모든 사용자에 대한 정보를 표시합니다.
update [-i <i>instanceName</i> ] -u <i>userName</i> -p <i>passwd</i> [-a <i>state</i> ] [-s] [-f]	지정한(또는 기본) 브로커 인스턴스 저장소에 있는 지정한 사용자의 비밀번호 및/또는 상태를 업데이트합니다.
update [-i <i>instanceName</i> ] -u <i>userName</i> -a <i>state</i> [-p <i>passwd</i> ] [-s] [-f]	



## 명령 옵션

표 13-16에는 imqusermgr 명령의 옵션이 나열되어 있습니다.

표 13-16 imqusermgr 옵션

옵션	설명
-a <i>active_state</i>	사용자 상태의 활성화 여부를 지정합니다(true/false). true 값은 활성화 상태를 나타내며 기본값입니다.
-f	사용자의 확인 없이 작업을 수행합니다.
-h	사용 도움말을 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.
-i <i>instanceName</i>	명령이 적용될 브로커 인스턴스 사용자 저장소를 지정합니다. 지정하지 않으면 기본 인스턴스 이름 imqbroker인 것으로 가정합니다.
-p <i>passwd</i>	사용자의 비밀번호를 지정합니다.
-g <i>group</i>	사용자 그룹을 지정합니다. 유효한 값에는 admin, user, anonymous가 있습니다.
-s	자동 모드를 설정합니다.
-u <i>userName</i>	사용자 아이디를 지정합니다.
-v	버전 정보를 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.

## 참조

플랫 파일 사용자 저장소 설정 및 관리에 대한 자세한 내용과 imqusermgr 명령 예는 [142페이지의 "플랫 파일 사용자 저장소 사용"](#)을 참조하십시오.

# imqsvcadmin

서비스 관리(imqsvcadmin) 유틸리티는 브로커를 Windows 서비스로 설치합니다.

## 구문

```
imqsvcadmin subcommand [options]
```

```
imqsvcadmin -h
```

## 하위 명령

Message Queue 서비스 관리자 유틸리티(imqsvcadmin)에는 표 13-17에 나열된 하위 명령이 포함됩니다.

**표 13-17** imqsvcadmin 하위 명령

하위 명령	설명
install	서비스를 설치하고 시작 옵션을 지정합니다.
query	imqsvcadmin 명령의 시작 옵션을 표시합니다. 여기에는 서비스의 수동 또는 자동 시작, 위치, Java 런타임 위치, 시작할 때 브로커에 전달되는 인수 값이 포함됩니다.
remove	서비스를 제거합니다.

## 명령 옵션

표 13-18에는 imqsvcadmin 명령의 옵션이 나열되어 있습니다.

**표 13-18** imqsvcadmin 옵션

옵션	설명
-h	사용 도움말을 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.
-javahome <i>path</i>	Java 2와 호환할 수 있는 대체 런타임의 경로로 지정하여 사용합니다(기본값은 시스템의 런타임이나 Message Queue와 함께 제공되는 런타임 사용). 예: imqsvcadmin -install -javahome d:\jdk1.4
-jrehome <i>path</i>	Java 2 호환 JRE의 경로를 지정합니다. 예: imqsvcadmin -install -jrehome d:\jre\1.4

**표 13-18** imqsvcadmin 옵션(계속)

옵션	설명
-vmargs <i>arg</i> [[ <i>arg</i> ]...]	브로커 서비스를 실행하는 Java VM에 전달할 추가 인수를 지정합니다. (이러한 인수는 Windows 서비스 제어판 시작 매개 변수 필드에서 지정할 수도 있습니다.)  예: -vmargs "-Xms16m -Xmx128m"
-args <i>arg</i> [[ <i>arg</i> ]...]	브로커 서비스에 전달할 추가 명령줄 인수를 지정합니다. imqbrokerd 옵션에 대한 설명은 <a href="#">282페이지의 "imqbrokerd"</a> 를 참조하십시오.  (이러한 인수는 Windows 서비스 제어판 시작 매개 변수 필드에서 지정할 수도 있습니다.) 예를 들면 다음과 같습니다.  imqsvcadmin -install -args "passfile d:\imqpassfile"

-javahome, -vmargs, -args 옵션을 사용해서 지정하는 정보는 Windows 레지스트리의 다음 경로에 있는 JREHome, JVMArgs, ServiceArgs 키 아래에 저장됩니다.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
  \Services\IMQ_Broker\Parameters
```

## 참조

Message Queue를 Windows 서비스로 실행에 관한 자세한 내용은 [69페이지의 "자동 시작\(Windows\)"](#)을 참조하십시오.

# imqkeytool

imqusermgr 명령은 브로커에 대해 자체 서명된 인증서를 생성합니다. ssljms, ssladmin 또는 cluster 연결 서비스에 동일한 인증서를 사용할 수 있습니다. UNIX 시스템의 경우 superuser(root) 계정에서 imqkeytool을 실행해야 할 수도 있습니다.

## 구문

```
imqkeytool -broker
```

## 참조

보안 연결 설정에 대한 자세한 내용은 [159페이지](#)의 "SSL 기반 서비스 작업"을 참조하십시오.

# 브로커 등록 정보 참조

이 장에서는 브로커 구성 등록 정보를 나열하고 설명합니다. 첫 번째 절에서는 전체 설명이 나오는 절에 대한 참조와 함께 모든 브로커 등록 정보를 알파벳 순으로 나열합니다. 모든 다른 절에서는 브로커 등록 정보 집합을 기능별로 그룹화하여 자세히 설명합니다.

이 장은 다음 내용으로 구성되어 있습니다.

- 307페이지의 "알파벳 순 등록 정보 목록"
- 311페이지의 "연결 서비스 등록 정보"
- 313페이지의 "메시지 라우터 등록 정보"
- 316페이지의 "지속성 관리자 등록 정보"
- 320페이지의 "보안 관리자 등록 정보"
- 324페이지의 "모니터링 및 로깅 등록 정보"
- 326페이지의 "클러스터 구성 등록 정보"

설명 테이블에서 등록 정보는 `imgcmd update bkr` 명령을 사용하여 설정 가능한 경우에 표시됩니다.

## 알파벳 순 등록 정보 목록

**표 14-1**은 브로커 인스턴스 등록 정보의 알파벳 순 목록입니다. 이 목록에서 등록 정보의 범주를 확인한 다음 범주 설명을 사용하여 이 장의 다른 부분에 있는 전체 등록 정보 설명을 찾을 수 있습니다.

표의 왼쪽 열에는 각 등록 정보가 알파벳 순으로 나열됩니다. 오른쪽 열에는 등록 정보가 속하는 범주와 해당 절에 대한 상호 참조가 표시됩니다.

표 14-1 브로커 인스턴스 구성 등록 정보

등록 정보 이름	참조
<code>imq.accesscontrol.enabled</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.accesscontrol.file.filename</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.audit.enabled</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.authentication.basic.user_repository</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.authentication.client.response.timeout</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.authentication.type</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.autocreate.destination.isLocalOnly</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.destination.limitBehavior</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.destination.maxBytesPerMsg</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.destination.maxNumMsgs</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.destination.maxNumProducers</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.destination.maxTotalMsgBytes</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.destination.useDMQ</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.queue</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.queue.consumerFlowLimit</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.queue.localDeliveryPreferred</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.queue.maxNumActiveConsumers</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.queue.maxNumBackupConsumers</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.topic</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.autocreate.topic.consumerFlowLimit</code>	314페이지의 "자동 작성 구성 등록 정보"
<code>imq.cluster.<i>property_name</i></code>	326페이지의 "클러스터 구성 등록 정보"
<code>imq.destination.DMQ.truncateBody</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.destination.logDeadMsgs</code>	324페이지의 "모니터링 및 로깅 등록 정보"
<code>imq.hostname</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.httpjms.http.<i>property_name</i></code>	385페이지의 표 C-3
<code>imq.httpsjms.https.<i>property_name</i></code>	385페이지의 표 C-3
<code>imq.imqcmd.password</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.keystore.<i>property_name</i></code>	320페이지의 "보안 관리자 등록 정보"

**표 14-1**      브로커 인스턴스 구성 등록 정보(계속)

등록 정보 이름	참조
<code>imq.log.console.output</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.console.stream</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.file.dirpath</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.file.filename</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.file.output</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.file.rolloverbytes</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.file.rolloversecs</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.level</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.syslog.facility</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.syslog.identity</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.syslog.logconsole</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.syslog.logpid</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.syslog.output</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.log.timezone</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.message.expiration.interval</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.message.max_size</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.metrics.enabled</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.metrics.interval</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.metrics.topic.enabled</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.metrics.topic.interval</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.metrics.topic.persist</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.metrics.topic.timetolive</code>	324페이지의 "모니터링 서비스 등록 정보"
<code>imq.passfile.dirpath</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.passfile.enabled</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.passfile.name</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.persist.file.destination.message.filepool.limit</code>	317페이지의 "파일 기반 지속성 등록 정보"
<code>imq.persist.file.message.cleanup</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.persist.file.message.filepool.cleanratio</code>	313페이지의 "메시지 라우터 등록 정보"

**표 14-1**      브로커 인스턴스 구성 등록 정보(계속)

등록 정보 이름	참조
<code>imq.persist.file.message.max_record_size</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.persist.file.sync.enabled</code>	317페이지의 "파일 기반 지속성 등록 정보"
<code>imq.persist.jdbc.property_name</code>	316페이지의 "지속성 관리자 등록 정보"
<code>imq.persist.store</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.ping.interval</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.portmapper.backlog</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.portmapper.hostname</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.portmapper.port</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.resource_state.count</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.resource_state.threshold</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.service.activelist</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.service_name.accesscontrol.enabled</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.service_name.accesscontrol.file.filename</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.service_name.authentication.type</code>	320페이지의 "보안 관리자 등록 정보"
<code>imq.service_name.max_threads</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.service_name.min_threads</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.service_name.protocol_type.hostname</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.service_name.protocol_type.port</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.service_name.threadpool_model</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.shared.connectionMonitor_limit</code>	311페이지의 "연결 서비스 등록 정보"
<code>imq.system.max_count</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.system.max_size</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.transaction.autorollback</code>	313페이지의 "메시지 라우터 등록 정보"
<code>imq.user_repository.ldap.property_name</code>	320페이지의 "보안 관리자 등록 정보"



## 연결 서비스 등록 정보

표 14-2에서는 연결 서비스 등록 정보를 나열합니다. 첫 번째 열에는 등록 정보 이름이 나와 있습니다. 각 등록 정보 이름에 대해 두 번째 열에서는 등록 정보를 설명하고 세 번째 열에서는 등록 정보의 유형을 지정하며 네 번째 열에서는 기본값을 제공합니다.

표 14-2 연결 서비스 등록 정보

등록 정보 이름	설명	유형	기본값
imq.service.activelist	브로커 시작 시 활성화되는 쉘표로 구분된 이름별 연결 서비스 목록입니다. 지원되는 서비스: jms, ssljms, httpjms, httpsjms, admin, ssladmin.	목록	jms, admin
imq.ping.interval	브로커가 연결을 통해 Message Queue 클라이언트 런타임을 연속적으로 ping하는 시도 사이의 간격(초)입니다.	정수	120
imq.hostname	사용 가능한 호스트가 두 개 이상 있는 경우(예: 컴퓨터에 네트워크 인터페이스 카드가 두 개 이상인 경우) 모든 연결 서비스를 바인드할 호스트(호스트 이름 또는 IP 주소)입니다.	문자열	모든 사용 가능한 IP 주소
imq.portmapper.port <sup>1</sup>	브로커의 기본 포트, 즉 포트 매퍼가 위치한 포트입니다. 한 호스트에서 둘 이상의 브로커 인스턴스를 실행하는 경우 각 인스턴스는 고유한 포트 매퍼 포트를 지정 받아야 합니다.	정수	7676
imq.portmapper.hostname	사용 가능한 호스트가 두 개 이상 있는 경우(예: 컴퓨터에 네트워크 인터페이스 카드가 두 개 이상인 경우) 포트 매퍼를 바인드할 호스트(호스트 이름 또는 IP 주소)입니다.	문자열	상속 위치: imq.hostname
imq.portmapper.backlog	포트 매퍼가 요청 거부 전까지 처리할 수 있는 최대 동시 요청 수입니다. 이 등록 정보는 운영 체제 백로그에 저장되어 포트 매퍼의 처리를 대기할 수 있는 요청 수를 설정합니다.	정수	50
imq.service_name.protocol_type <sup>2</sup> .port	jms, ssljms, admin 및 ssladmin 서비스에 한해 명명된 연결 서비스의 포트 번호입니다.  httpjms 및 httpsjms 연결 서비스를 구성하려면 <b>부록 C, "HTTP/HTTPS 지원"</b> 을 참조하십시오.	정수	0  포트는 포트 매퍼에 의해 동적으로 할당됩니다.
imq.service_name.protocol_type <sup>2</sup> .hostname	jms, ssljms, admin 및 ssladmin 서비스에 한해, 사용 가능한 호스트가 두 개 이상 있는 경우(예: 컴퓨터에 네트워크 인터페이스 카드가 두 개 이상인 경우) 명명된 연결 서비스를 바인드할 호스트(호스트 이름 또는 IP 주소)입니다.	문자열	상속 위치: imq.hostname

**표 14-2**      연결 서비스 등록 정보(계속)

등록 정보 이름	설명	유형	기본값
imq.service_name. min_threads	이 스레드 수에 도달하면 명명된 연결 서비스가 사용할 수 있도록 스레드 풀에서 스레드가 유지 관리됩니다. 기본값은 연결 서비스에 따라 다릅니다.	정수	10 (jms) 10 (ssljms) 10 (httpjms) 10 (httpsjms) 4 (admin) 4 (ssladmin)
imq.service_name. max_threads	이 스레드 수를 초과하면 명명된 연결 서비스가 사용할 수 있는 신규 스레드가 더 이상 스레드 풀에 추가되지 않습니다. 이 수는 0보다 크고 min_threads 값보다 커야 합니다. 기본값은 연결 서비스에 따라 다릅니다.	정수	1000 (jms) 500 (ssljms) 500 (httpjms) 500 (httpsjms) 10 (admin) 10 (ssladmin)
imq.service_name. threadpool_model	명명된 연결 서비스에 대해 스레드가 연결 전용인지 (dedicated) 또는 필요에 따라 여러 연결에 의해 공유되는 지(shared)를 지정하는 문자열입니다. 공유 모델(스레드 풀 관리)은 브로커가 지원하는 연결 수를 늘리지만, jms 및 admin 연결 서비스에 대해서만 구현됩니다. 기본값은 연결 서비스에 따라 다릅니다.	문자열	dedicated (jms) dedicated (ssljms) dedicated (httpjms) dedicated (httpsjms) dedicated (admin) dedicated (ssladmin)
imq.shared. connectionMonitor_limit	공유 스레드 풀 모델에 한해 분산자 스레드가 모니터링할 수 있는 최대 연결 수입니다. (시스템은 모든 연결을 모니터링하기에 충분한 수의 분산자 스레드를 할당). 이 값이 작을수록 시스템은 더 신속하게 스레드에 활성 연결을 지정할 수 있습니다. 값 -1은 제한이 없음을 의미합니다. 기본값은 운영 체제에 따라 다릅니다.	정수	512 (Solaris & Linux) 64 (Windows)

- 이 등록 정보는 imqcmd update bkr 명령과 함께 사용할 수 있습니다.
- protocol\_type은 표 4-2에 지정되어 있습니다.

## 메시지 라우터 등록 정보

표 14-3에서는 메시지 라우터 등록 정보를 나열합니다. 첫 번째 열에는 등록 정보 이름이 나와 있습니다. 각 등록 정보 이름에 대해 두 번째 열에서는 등록 정보를 설명하고 세 번째 열에서는 등록 정보의 유형을 지정하며 네 번째 열에서는 기본값을 제공합니다.

대상을 자동으로 작성하도록 메시지 서버 기능을 구성하는 자동 작성 등록 정보는 314페이지의 표 14-4에 나열되어 있습니다.

**표 14-3** 메시지 라우터 등록 정보

등록 정보 이름	설명	유형	기본값
imq.destination. DMQ.truncateBody <sup>1</sup>	브로커가 메시지를 사용 불능 메시지 대기열에 저장하기 전에 메시지 본문을 제거할지 여부를 지정하는 부울 값입니다. 값이 true인 경우 브로커는 메시지 헤더와 등록 정보 데이터만 저장합니다. 값이 false인 경우 브로커는 헤더와 본문을 저장합니다.	부울	false
imq.message. expiration.interval	만료된 메시지 재생 이용이 발생하는 간격(초)입니다.	정수	60
imq.system.max_count <sup>1</sup>	브로커가 보관하는 최대 메시지 수입니다. 추가 메시지가 거부됩니다. 값 -1은 제한이 없음을 의미합니다.	정수	-1
imq.system.max_size <sup>1</sup>	브로커가 보관하는 최대 전체 메시지 최대 크기(바이트, KB, MB)입니다. 추가 메시지가 거부됩니다. 값 -1은 제한이 없음을 의미합니다.	바이트 문자 열 <sup>2</sup>	-1
imq.message.max_size <sup>1</sup>	메시지 본문의 최대 허용 크기(바이트, KB, MB 단위)입니다. 이 크기를 초과하는 메시지는 거부됩니다. 값 -1은 제한이 없음을 의미합니다.	바이트 문자 열 <sup>2</sup>	70m
imq.resource_state. threshold	각 메모리 자원 상태가 트리거되는 메모리 사용률입니다. 자원 상태는 초록, 노랑, 주황 및 빨강으로 표시됩니다.	정수 (퍼센트)	0 (green) 80 (yellow) 90 (orange) 98 (red)
imq.resource_state.count	시스템 메모리에서 새 메모리 임계값에 도달했는지 여부를 확인하기 전에 일괄 처리에서 허용되는, 받는 메시지의 최대 수입니다. 이 제한은 시스템 메모리가 점점 부족해지면 메시지 생성자를 억제합니다.	정수 (퍼센트)	5000 (green) 500 (yellow) 50 (orange) 0 (red)

**표 14-3** 메시지 라우터 등록 정보(계속)

등록 정보 이름	설명	유형	기본값
imq.transaction. autorollback	PREPARED 상태에 있는 분산 트랜잭션이 브로커 시작 시 자동으로 롤백되는지 여부를 지정하는 부울 값입니다. false라면 imqcmd를 사용하여 수동으로 트랜잭션을 완결하거나 롤백해야 합니다(123페이지의 "트랜잭션 관리" 참조).	부울	false

- 이 등록 정보는 imqcmd update bkr 명령과 함께 사용할 수 있습니다.
- 바이트 문자열로 입력하는 값은 바이트, KB, MB로 나타낼 수 있습니다. 예를 들어, 1000은 1000바이트를, 7500b는 7500바이트를, 77k는 77KB(77 x 1024 = 78848바이트)를, 17m는 17MB(17 x 1024 x 1024 = 17825792바이트)를 나타냅니다.

표 14-4에서는 대상을 자동으로 작성할 때 브로커가 사용하는 등록 정보를 나열합니다.

**표 14-4** 자동 작성 구성 등록 정보

등록 정보 이름	설명	유형	기본값
imq.autocreate.destination. isLocalOnly	(브로커 클러스터에만 적용됨). 대상을 다른 브로커에 복제하지 않도록 지정하는 부울 값입니다. 따라서 메시지 전달이 로컬 사용자(대상이 생성되는 브로커에 연결된 사용자)에게만 제한됩니다. 대상이 생성된 이후에는 이 속성을 업데이트할 수 없습니다.	부울	false
imq.autocreate.destination. limitBehavior	메모리 제한 임계값에 도달할 경우에 브로커가 응답하는 방법을 지정하는 문자열입니다. 값은 다음과 같습니다. <ul style="list-style-type: none"> <li>FLOW_CONTROL — 생성자 수를 조금씩 줄입니다.</li> <li>REMOVE_OLDEST — 가장 오래된 메시지를 삭제합니다.</li> <li>REMOVE_LOW_PRIORITY — 메시지 보존 기간을 기준으로 우선 순위가 가장 낮은 메시지를 삭제합니다.</li> <li>REJECT_NEWEST — 최신 메시지를 거부합니다. 생성자 클라이언트는 지속성 메시지가 거부되는 경우에만 예외 메시지를 받습니다. 이 제한 동작을 비지속성 메시지와 함께 사용하려면 imqAckOnProduce 연결 팩토리 속성을 설정합니다.</li> </ul> 이 등록 정보를 REMOVE_OLDEST 또는 REMOVE_LOW_PRIORITY로 설정하고 imq.autocreate.destination.useDMQ를 true로 설정하는 경우 브로커는 초과하는 메시지를 사용 불능 메시지 대기열로 이동합니다.	문자열	REJECT NEWEST

표 14-4 자동 작성 구성 등록 정보(계속)

등록 정보 이름	설명	유형	기본값
imq.autocreate.destination.maxBytesPerMsg	자동 작성된 대상에 허용되는 단일 메시지의 최대 크기(바이트)입니다. 값이 -1이면 메시지 크기가 제한되지 않습니다.	바이트 문자열 <sup>2</sup>	10k
imq.autocreate.destination.maxNumMsgs	자동 작성된 대상에 허용되는 사용되지 않은 최대 메시지 수입니다. 값이 -1이면 수에 제한이 없습니다.	정수	100,000
imq.autocreate.destination.maxNumProducers	대상에 허용되는 최대 생성자 수입니다. 이 제한에 도달하면 새로운 생성자가 생성되지 않습니다. 값이 -1이면 생성자 수에 제한이 없습니다.	정수	100
imq.autocreate.destination.maxTotalMsgBytes	대상의 사용되지 않은 메시지에 허용되는 최대 메모리 합계(바이트)입니다. 값이 -1이면 메모리에 제한이 없습니다.	바이트 문자열 <sup>2</sup>	10m
imq.autocreate.destination.useDMQ	브로커가 자동 작성된 대상에 대한 사용 불가능 메시지를 사용 불가능 메시지 대기열로 이동시킬지 여부를 지정하는 부울 값입니다.	부울	true
imq.autocreate.queue <sup>1</sup>	브로커에 대기열 대상 자동 작성이 허용되는지 여부를 지정하는 부울 값입니다.	부울	true
imq.autocreate.queue.consumerFlowLimit	일괄적으로 사용자에게 전달되는 최대 메시지 수를 지정합니다. 로드 균형 조정된 대기열 전달에서 이 수는 로드 균형 조정을 시작하기 전에 활성 사용자에게 경로 지정되는 초기 대기 메시지 수입니다. 이 제한은 해당 연결에서 대상 사용자에게 지정된 낮은 값으로 대체될 수 있습니다. 값 -1은 무제한을 의미합니다.	정수	1000
imq.autocreate.queue.localDeliveryPreferred	(브로커 클러스터의 로드 균형 조정된 대기열 전달에만 적용됨). 로컬 브로커에 사용자가 없는 경우에만 원격 사용자에게 메시지를 전달하도록 지정하는 부울 값입니다. 자동 작성된 대상을 로컬에만 전달로 제한해서는 안 됩니다(isLocalOnly = false).	부울	false
imq.autocreate.queue.maxNumActiveConsumers	자동 작성된 대기열 대상으로부터의 로드 균형 조정 전달에서 활성 상태가 될 수 있는 최대 사용자 수입니다. 값 -1은 무제한을 의미합니다.	정수	1
imq.autocreate.queue.maxNumBackupConsumers	자동 작성된 대기열 대상으로부터의 로드 균형 조정 전달 중에 오류가 발생할 경우 활성 사용자를 대신할 수 있는 최대 백업 사용자 수입니다. 값 -1은 무제한을 의미합니다.	정수	0
imq.autocreate.topic	브로커에 주제 대상의 자동 작성이 허용되는지 여부를 지정하는 부울 값입니다.	부울	true
imq.autocreate.topic.consumerFlowLimit	일괄적으로 사용자에게 전달되는 최대 메시지 수를 지정합니다. 값 -1은 무제한을 의미합니다.	정수	1000

1. 이 등록 정보는 imqcmd update bkr과 함께 사용할 수 있습니다.

2. 바이트 문자열로 입력하는 값은 바이트, KB, MB로 나타낼 수 있습니다. 예를 들어, 1000은 1000바이트를, 7500b는 7500바이트를, 77k는 77KB(77 x 1024 = 78848바이트)를, 17m은 17MB(17 x 1024 x 1024 = 17825792바이트)를 나타냅니다.

## 지속성 관리자 등록 정보

브로커의 지속성 기능을 구성하려면 `img.persist.store` 값을 지정하거나 기본값을 적용해야 합니다.

**표 14-5** 필수 지속성 관리자 등록 정보

등록 정보 이름	설명	유형	기본값
<code>img.persist.store</code>	브로커가 기본 제공 파일 기반 지속성을 사용하는지 아니면 플러그인 JDBC 호환 지속성을 사용하는지를 지정하는 문자열입니다. 값은 <code>file</code> 또는 <code>jdbc</code> 여야 합니다.	문자열	<code>file</code>

파일 기반 지속성 및 JDBC 기반 지속성을 지원하는 등록 정보는 다음 절에서 설명합니다.

### 파일 기반 지속성

[표 14-6](#)에서는 파일 기반 지속성을 지원하는 등록 정보를 나열합니다. 첫 번째 열에는 등록 정보 이름이 나와 있습니다. 각 등록 정보 이름에 대해 두 번째 열에서는 등록 정보를 설명하고 세 번째 열에서는 등록 정보의 유형을 지정하며 네 번째 열에서는 기본값을 제공합니다.

표 14-6 파일 기반 지속성 등록 정보

등록 정보 이름	설명	유형	기본값
imq.persist.file.sync.enabled	지속성 작업이 메모리 상태에서 물리적 저장 장치와 동기화될 것인지 여부를 지정하는 부울 값입니다. 이 등록 정보가 true로 설정되어 있으면 시스템 충돌로 인한 데이터 손실은 줄어들지만 지속성 작업의 성능이 저하됩니다. Sun Cluster 및 Sun Cluster Data Service for Message Queue를 실행하는 경우 모든 클러스터 노드의 브로커에 대해 이 등록 정보를 true로 설정합니다.	부울	false
imq.persist.file.message.max_record_size	기본 제공 파일 기반 지속성에 대해 개별 파일에 저장되지 않고 메시지 저장소 파일에 추가될 메시지의 최대 크기입니다.	바이트 문자열 <sup>1</sup>	1m
imq.persist.file.destination.message.filepool.limit	기본 제공 파일 기반 지속성에 대해 대상 파일 풀에서 재사용 가능한 최대 파일 수입니다. 이 수가 클수록 브로커는 지속성 데이터를 더 빠르게 처리할 수 있습니다. 이 값을 초과하는 사용 가능한 파일은 삭제됩니다. 이 한도를 초과할 경우 브로커는 필요에 따라 추가 파일을 작성하고 삭제합니다.	정수	100
imq.persist.file.message.filepool.cleanratio	기본 제공 파일 기반 지속성에 대해 대상 파일 풀에서 <i>clean</i> 상태(0으로 잘라낸 상태)로 유지되는 사용 가능한 파일의 비율입니다. 이 값이 클수록 작업 중 파일을 지우는 데 필요한 오버헤드가 늘어나지만, 파일 풀에서 필요한 디스크 공간은 줄어듭니다.	정수	0
imq.persist.file.message.cleanup	기본 제공 파일 기반 지속성에 대해 브로커가 종료 시 대상 파일 저장소에서 사용 가능한 파일을 지울 것인지 여부를 지정하는 부울 값입니다. false 값은 브로커 종료 속도를 향상시키지만, 파일 저장소에 더 많은 디스크 공간이 필요합니다.	부울	false

1. *바이트 문자열*로 입력하는 값은 바이트, KB, MB로 나타낼 수 있습니다. 예를 들어, 1000은 1000바이트를, 7500b는 7500바이트를, 77k는 77KB(77 x 1024 = 78848바이트)를, 17m은 17MB(17 x 1024 x 1024 = 17825792바이트)를 나타냅니다.

## JDBC 기반 지속성

표 14-7에는 JDBC 기반 지속성을 지원하는 등록 정보가 포함되어 있습니다. 이 표에서는 등록 정보를 나열 및 설명하고 PointBase 제품에서 구성하여 사용하는 방법에 대한 예를 제공합니다.

표 14-7 JDBC 기반 지속성 등록 정보

등록 정보 이름	설명	예
imq.persist.store	파일 기반 또는 JDBC 기반 데이터 저장소를 지정하는 문자열입니다.	jdbc

표 14-7 JDBC 기반 지속성 등록 정보(계속)

등록 정보 이름	설명	예
imq.persist.jdbc.brokerid	<p>(선택 사항) 같은 데이터베이스를 영구 데이터 저장소로 사용하고 있는 브로커 인스턴스가 두 개 이상인 경우 데이터베이스 테이블 이름에 추가하여 고유한 이름으로 만들어 주는 브로커 인스턴스 식별자입니다.</p> <p>보통 한 브로커 인스턴스에 대해서만 데이터를 저장하는 내장 데이터베이스에는 이 속성이 필요 없습니다.</p> <p>식별자는 데이터베이스에 허용되는 최대 테이블 이름 길이에서 12를 뺀 길이를 넘지 않는 영문자 및 숫자 문자열이어야 합니다.</p>	(PointBase 내장 버전에는 필요하지 않음)
imq.persist.jdbc.driver	데이터베이스에 연결할 JDBC 드라이버의 Java 클래스 이름입니다.	com.pointbase.jdbc.jdbcUniversalDriver
imq.persist.jdbc.opendburl	기존 데이터베이스에 대한 연결을 여는 데이터베이스 URL입니다.	jdbc:pointbase:embedded:dbName;database.home=../instances/instanceName/dbstore
imq.persist.jdbc.createdburl	<p>(선택 사항) 데이터베이스를 만들기 위해 연결을 여는 데이터베이스 URL입니다.</p> <p>imqdbmgr를 사용하여 데이터베이스를 만드는 경우에만 이 속성을 지정합니다.</p>	jdbc:pointbase:embedded:dbName;new,database.home=/instances/instanceName/dbstore
imq.persist.jdbc.closedburl	(선택 사항) 브로커를 종료할 때 현재 데이터베이스 연결을 종료하는 데이터베이스 URL입니다.	PointBase에는 필요하지 않음
imq.persist.jdbc.user	<p>(선택 사항) 필요한 경우 데이터베이스 연결을 열 때 사용되는 사용자 아이디입니다. 보안상의 이유로 명령줄 옵션을 사용해서 값을 지정할 수도 있습니다.</p> <p>imqbrokerd -dbuser 및 imqdbmgr -u</p>	
imq.persist.jdbc.needpassword	<p>(선택 사항) 브로커에 액세스할 때 데이터베이스에 비밀번호가 필요한지 여부를 지정하는 부울 값입니다. 값이 true이면 비밀번호가 필요합니다.</p> <p>이 옵션을 설정한 경우 -passfile 옵션을 사용하여 비밀번호가 들어 있는 파일을 지정하지 않으면 imqbrokerd 및 imqdbmgr 명령에서 비밀번호를 묻는 프롬프트를 표시합니다.</p>	



표 14-7 JDBC 기반 지속성 등록 정보(계속)

등록 정보 이름	설명	예
imq.persist.jdbc.password	(선택 사항) 필요한 경우 데이터베이스 연결을 열 때 사용하는 비밀번호입니다. passfile에만 이 등록 정보를 지정합니다.	
imq.persist.jdbc.table.IMQSV35	버전 테이블을 만들 때 사용하는 SQL 명령입니다.	CREATE TABLE \${name} (STOREVERSION INTEGER NOT NULL, BROKERID VARCHAR(100))
imq.persist.jdbc.table.IMQCCREC35	구성 변경 레코드 테이블을 만들 때 사용하는 SQL 명령입니다.	CREATE TABLE \${name} (RECORDTIME BIGINT NOT NULL, RECORD BLOB(10k))
imq.persist.jdbc.table.IMQDEST35	대상 테이블을 만들 때 사용하는 SQL 명령입니다.	CREATE TABLE \${name} (DID VARCHAR(100) NOT NULL, DEST BLOB(10k), primary key(DID))
imq.persist.jdbc.table.IMQINT3	인터레스트 테이블을 만들 때 사용하는 SQL 명령입니다.	CREATE TABLE \${name} (CUID BIGINT NOT NULL, INTEREST BLOB(10k), primary key(CUID))
imq.persist.jdbc.table.IMQMSG35	메시지 테이블을 만들 때 사용하는 SQL 명령입니다. MSG 열의 기본 최대 길이는 1MB(1m)입니다. 이보다 큰 메시지가 있을 것으로 예상 되면 이 길이를 알맞게 설정합니다. 이미 만든 테이블이 있는 경우 변경 사항을 적용하려면 테이블을 다시 만들어야 합니다.	CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, DID VARCHAR(100), MSGSIZE BIGINT, MSG BLOB(1m), primary key(MID))
imq.persist.jdbc.table.IMQPROPS35	등록 정보 테이블을 만들 때 사용하는 SQL 명령입니다.	CREATE TABLE \${name} (PROPNAME VARCHAR(100) NOT NULL, PROPVALUE BLOB(10k), primary key(PROPNAME))
imq.persist.jdbc.table.IMQILIST35	인터레스트 상태 테이블을 만들 때 사용하는 SQL 명령입니다.	CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, CUID BIGINT, DID VARCHAR(100), STATE INTEGER, primary key(MID, CUID))
imq.persist.jdbc.table.IMQTXN35	트랜잭션 테이블을 만들 때 사용하는 SQL 명령입니다.	CREATE TABLE \${name} (TUID BIGINT NOT NULL, STATE INTEGER, TSTATEOBJ BLOB(10K), primary key(TUID))

**표 14-7** JDBC 기반 지속성 등록 정보(계속)

등록 정보 이름	설명	예
imq.persist.jdbc.table. IMQTACK35	트랜잭션 확인 테이블을 만들 때 사용하는 SQL 명령입니다.	CREATE TABLE \${name} (TUID BIGINT NOT NULL, TXNACK BLOB(10k))

## 보안 관리자 등록 정보

표 14-8에서는 보안 관리자 등록 정보를 나열합니다. 첫 번째 열에는 등록 정보 이름이 나와 있습니다. 각 등록 정보 이름에 대해 두 번째 열에서는 등록 정보를 설명하고 세 번째 열에서는 등록 정보의 유형을 지정하며 네 번째 열에서는 기본값을 제공합니다.

SSL을 사용하는 경우 다음과 같이 표 14-9에 나열된 키 저장소 구성 등록 정보를 참조하십시오.

**표 14-8** 보안 관리자 등록 정보

등록 정보 이름	설명	유형	기본값
imq.accesscontrol.file.enabled	브로커가 지원하는 모든 연결 서비스에 대해 액세스 제어를 설정할지 여부를 지정하는 부울 값입니다. 액세스 제어 등록 정보 파일에 지정된 대로 인증된 사용자가 연결 서비스를 사용하거나 특정 대상에 대해 특정 Message Queue 작업을 수행할 권한이 있는가를 시스템이 확인할 것인지 여부를 표시합니다.	부울	true
imq.accesscontrol.file.filename	브로커 인스턴스가 지원하는 모든 연결 서비스에 대한 액세스 제어 등록 정보 파일의 이름입니다. 파일 이름은 액세스 제어 디렉토리에 대한 상대 파일 경로를 지정합니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).	문자열	accesscontrol.properties
imq.audit.enabled	브로커 로그 파일에 대한 감사 로깅(엔터프라이즈판에만 적용)을 시작할지 여부를 지정하는 부울 값입니다.	부울	false
imq.authentication.basic.user_repository	(기본 64 코딩에 대해) 인증에 사용할 사용자 저장소 유형을 파일 기반(file) 또는 LDAP(ldap) 중 하나로 지정하는 문자열입니다.	문자열	file
imq.authentication.client.response.timeout	클라이언트가 브로커로부터의 인증 요청에 응답할 때까지 시스템이 대기할 시간(초)입니다.	정수	180
imq.authentication.type	비밀번호를 기본 64 코딩(basic)으로 전달할지 MD5 다이제스트(digest)로 전달할지 여부를 지정하는 문자열입니다. 브로커가 지원하는 모든 연결 서비스에 대한 인코딩을 설정합니다.	문자열	digest

표 14-8 보안 관리자 등록 정보(계속)

등록 정보 이름	설명	유형	기본값
imq.imqcmd.password	관리 사용자의 비밀번호입니다. imqcmd 명령 유틸리티는 작업을 수행하기 전에 이 비밀번호를 사용하여 명령 사용자를 인증합니다.	문자열	없음
imq.keystore.property_name	SSL 기반 서비스에 대해 SSL 키 저장소와 관련된 보안 등록 정보를 지정하는 문자열입니다. <a href="#">323페이지의 표 14-9</a> 를 참조하십시오.	문자열	없음
imq.passfile.dirpath	passfile이 포함된 디렉토리의 경로입니다(운영 체제에 따라 다름).	문자열	부록 A 참조
imq.passfile.enabled	안전한 통신을 위해 사용자 비밀번호(SSL, LDAP, JDBC™)가 passfile에 설정되는지 여부를 지정하는 부울 값입니다.	부울	false
imq.passfile.name	passfile의 이름입니다.	문자열	passfile
imq.service_name.accesscontrol.enabled	명명된 연결 서비스에 대해 브로커 전체에 대한 설정을 무시하는, 액세스 제어를 설정할지 여부를 지정하는 부울 값입니다. 액세스 제어 등록 정보 파일에 지정된 대로 인증된 사용자가 명명된 연결 서비스를 사용하거나 특정 대상에 대해 특정 Message Queue 작업을 수행할 권한이 있는가를 시스템이 확인할 것인지 여부를 표시합니다.	부울	시스템 전체 등록 정보 imq.accesscontrol.enabled에서 상속
imq.service_name.accesscontrol.file.filename	브로커 인스턴스의 명명된 연결 서비스에 대한 액세스 제어 등록 정보 파일의 이름입니다. 파일 이름은 액세스 제어 디렉토리에 대한 상대 파일 경로를 지정합니다 (부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조). 기본값은 시스템 전체 등록 정보 imq.accesscontrol.file.filename에서 상속됩니다.	문자열	설명 참조
imq.service_name.authentication.type	비밀번호를 기본 64 코딩(basic)으로 전달할지 MD5 다이제스트(digest)로 전달할지 여부를 지정하는 문자열입니다. 명명된 연결 서비스의 인코딩을 설정하며, 이 값은 브로커 전체에 대한 설정을 무시합니다. 기본값은 시스템 전체 등록 정보 imq.authentication.type에서 상속됩니다.	문자열	설명 참조
imq.user_repository.ldap.base	사용자 항목에 사용되는 디렉토리 기반입니다.	문자열	없음
imq.user_repository.ldap.gidattr	그룹 이름을 값으로 가진 공급자별 속성 식별자입니다.	문자열	없음
imq.user_repository.ldap.grpbase	그룹 항목에 사용되는 디렉토리 기반입니다.	문자열	없음

**표 14-8** 보안 관리자 등록 정보(계속)

등록 정보 이름	설명	유형	기본값
imq.user_repository. ldap.grpfiltler	JNDI 검색 필터(논리식으로 표현된 검색 쿼리)입니다. 그룹에 대해 검색 필터를 지정하면 브로커에서 검색 범위를 좁혀 효율을 높일 수 있습니다. 자세한 내용은 다음 위치의 JNDI 자습서를 참조하십시오. <a href="http://java.sun.com/products/jndi/tutorial">http://java.sun.com/products/jndi/tutorial</a> 이 등록 정보는 설정하지 않아도 됩니다.	문자열	없음
imq.user_repository. ldap.grpsearch	그룹 검색을 사용할지 여부를 지정하는 부울 값입니다. 사용자를 그룹에 연결할 수 있는지 확인하려면 LDAP 공급자가 제공하는 설명서를 참조하십시오. Message Queue에서는 중첩 그룹이 지원되지 않습니다.	부울	false
imq.user_repository. ldap.memattr	그룹 구성원의 고유 이름을 값으로 가진 그룹 항목 내의 속성 식별자입니다.	문자열	없음
imq.user_repository. ldap.password	브로커에서 사용하는 고유 이름에 연결된 비밀번호입니다. passfile에서만 이 등록 정보를 지정합니다. 디렉토리 서버가 익명 검색을 허용하는 경우에는 비밀번호가 필요 없습니다.	문자열	없음
imq.user_repository. ldap.principal	브로커가 검색할 디렉토리 서버에 바인드할 때 사용하는 고유 이름입니다. 디렉토리 서버에서 익명 검색을 허용하는 경우에는 이 등록 정보에 값을 할당할 필요가 없습니다.	문자열	없음
imq.user_repository. ldap.property_name	제공 예정	제공 예정	제공 예정
imq.user_repository. ldap.server	LDAP 서버의 <i>host:port</i> 입니다. <i>host</i> 는 디렉토리 서버를 실행 중인 호스트의 정규화된 DNS 이름을 지정하고 <i>port</i> 는 디렉토리 서버에서 통신에 사용하는 포트 번호를 지정합니다. 파일오버 서버 목록을 지정하려면 다음 구문을 사용합니다. <i>host1:port1 ldap://host2:port2 ldap://host3:port3...</i> 목록의 항목을 공백으로 구분합니다. 각 파일오버 서버 주소는 <i>ldap://</i> 로 시작합니다. SSL을 사용하고 imq.user_repository.ldap.ssl.enabled 등록 정보를 true로 설정한 경우에도 이 형식을 사용합니다. 따라서 주소에 "ldaps"를 지정할 필요가 없습니다.	문자열	없음

**표 14-8** 보안 관리자 등록 정보(계속)

등록 정보 이름	설명	유형	기본값
imq.user_repository. ldap.ssl.enabled	브로커가 LDAP 서버와 통신할 때 SSL 프로토콜을 사용해야 하는지 여부를 지정하는 부울 값입니다.	부울	false
imq.user_repository. ldap.timeout	검색 시간 제한(초)입니다.	정수	280
imq.user_repository. ldap.uidattr	사용자를 고유하게 식별하는 값을 가진 공급자별 속성 식별자입니다. uid, cn 등이 있습니다.	문자열	없음
imq.user_repository. ldap.usrfilter	JNDI 검색 필터(논리식으로 표현된 검색 쿼리)입니다. 사용자에 대해 검색 필터를 지정하면 브로커에서 검색 범위를 좁혀 효율을 높일 수 있습니다. 자세한 내용은 <a href="http://java.sun.com/products/jndi/tutorial">http://java.sun.com/products/jndi/tutorial</a> 에 있는 JNDI 자습서를 참조하십시오. 이 등록 정보는 설정하지 않아도 됩니다.	문자열	없음

Message Queue 키 저장소의 구성 가능한 등록 정보는 표 14-9에 나와 있습니다. 이러한 등록 정보는 SSL에서 사용하십시오.

**표 14-9** 키 저장소 등록 정보

등록 정보 이름	설명	유형	기본값
imq.keystore.file.dirpath	SSL 기반 서비스에서 키 저장소 파일이 있는 디렉토리의 경로입니다. 기본값: <a href="#">부록 A, "운영 체제별 Message Queue 데이터의 위치"</a> 참조	문자열	없음
imq.keystore.file.name	SSL 기반 서비스에서 키 저장소 파일의 이름입니다.	문자열	키 저장소
imq.keystore.password	SSL 기반 서비스에서 키 저장소 비밀번호입니다. passfile에만 이 등록 정보를 지정하십시오.	문자열	없음

## 모니터링 및 로깅 등록 정보

표 14-10에는 모니터링 및 로깅과 관련된 등록 정보가 나열되어 있습니다. 첫 번째 열에는 등록 정보 이름이 나와 있습니다. 각 등록 정보 이름에 대해 두 번째 열에서는 등록 정보를 설명하고 세 번째 열에서는 등록 정보의 유형을 지정하며 네 번째 열에서는 기본값을 제공합니다.

표 14-10 모니터링 서비스 등록 정보

등록 정보 이름	설명	유형	기본값
imq.destination.logDeadMsgs <sup>1</sup>	브로커가 다음과 같은 이벤트 유형을 기록하는지 여부를 지정하는 부울 값입니다. <ul style="list-style-type: none"> <li>대상이 최대 크기 또는 최대 메시지 수에 도달하여 꽉 찼습니다.</li> <li>브로커가 관리 명령 또는 전달 확인 이외의 메시지를 제거합니다.</li> <li>브로커가 메시지를 사용 불가능 메시지 대기열로 이동합니다.</li> </ul>	부울	false
imq.log.console.output	콘솔에 기록하는 로깅 정보의 범주를 지정하는 문자열입니다. 값은 다음 중 하나가 될 수 있습니다. <ul style="list-style-type: none"> <li>ALL</li> <li>NONE</li> <li>ERROR, WARNING, INFO 값 중 하나 이상이며 세로 막대( )로 구분합니다. 로그 메시지의 각 범주를 개별적으로 지정합니다. 메시지 범주에는 다른 범주가 포함되지 않습니다.</li> </ul>	문자열	ERROR WARNING
imq.log.console.stream	콘솔 출력을 stdout(OUT) 또는 stderr(ERR)에 기록할 것인지를 지정하는 문자열입니다.	문자열	ERR
imq.log.file.dirpath	로그 파일이 포함된 디렉토리의 경로입니다(운영 체제에 따라 다름).	문자열	부록 A 참조
imq.log.file.filename	로그 파일의 이름입니다.	문자열	log.txt
imq.log.file.output	콘솔에 기록할 로깅 정보의 범주입니다. 값은 다음 중 하나가 될 수 있습니다. <ul style="list-style-type: none"> <li>ALL</li> <li>NONE</li> <li>ERROR, WARNING, INFO 값 중 하나 이상이며 세로 막대( )로 구분합니다. 로그 메시지의 각 범주를 개별적으로 지정합니다. 메시지 범주에는 다른 범주가 포함되지 않습니다.</li> </ul>	문자열	ALL
imq.log.file.rolloverbytes <sup>1</sup>	새 로그 파일로 출력을 롤오버할 로그 파일 크기(바이트)입니다. 값이 -1이면 파일 크기를 기준으로 롤오버를 사용할 수 없습니다.	정수	-1

**표 14-10** 모니터링 서비스 등록 정보(계속)

등록 정보 이름	설명	유형	기본값
<code>imq.log.file.rolloversecs<sup>1</sup></code>	새 로그 파일로 출력을 롤오버할 로그 파일의 보존 기간(초)입니다. 값이 -1이면 파일 보존 기간을 기준으로 롤오버를 사용할 수 없습니다.	정수	604800 (1주)
<code>imq.log.level<sup>1</sup></code>	로거 수준, 즉 출력 채널에 기록 가능한 출력 범주를 지정하는 문자열입니다. 지정된 범주 및 모든 상위 범주가 함께 포함됩니다. 값은 내림차순으로 ERROR, WARNING, INFO입니다.	문자열	INFO
<code>imq.log.syslog.facility</code>	(Solaris에만 적용) Message Queue 브로커가 어떤 <code>syslog</code> 기능으로 로깅할 것인지를 지정하는 문자열입니다. 값은 <code>syslog(3C)</code> 설명서 페이지에 있는 값을 미러링합니다. Message Queue에서 사용 가능한 값은 LOG_USER, LOG_DAEMON 그리고 LOG_LOCAL0부터 LOG_LOCAL7까지입니다.	문자열	LOG_DAEMON
<code>imq.log.syslog.identity</code>	(Solaris에만 적용) <code>syslog</code> 에 기록되는 모든 메시지의 앞에 추가될 아이디 문자열입니다. 기본값은 <code>imqbrokerd_\${imq.instanceName}</code> 입니다.	문자열	설명 참조
<code>imq.log.syslog.logconsole</code>	(Solaris에만 적용) 메시지를 <code>syslog</code> 에 보낼 수 없을 경우 시스템 콘솔에 메시지를 기록할 것인지 여부를 지정하는 부울 값입니다.	부울	false
<code>imq.log.syslog.logpid</code>	(Solaris에만 적용) 브로커 프로세스 아이디를 메시지와 함께 로깅할 것인지 여부를 지정하는 부울 값입니다(true/false).	부울	true
<code>imq.log.syslog.output</code>	(Solaris에만 적용) <code>syslogd(1M)</code> 에 기록하는 로깅 정보의 범주를 지정하는 문자열입니다. 값은 다음 중 하나가 될 수 있습니다. <ul style="list-style-type: none"> <li>ALL</li> <li>NONE</li> <li>ERROR, WARNING, INFO 값 중 하나 이상이며 세로 막대( )로 구분합니다. 로그 메시지의 각 범주를 개별적으로 지정합니다. 메시지 범주에는 다른 범주가 포함되지 않습니다.</li> </ul>	문자열	ERROR
<code>imq.log.timezone</code>	로그 타임스탬프의 표준 시간대를 표시하는 문자열입니다. 식별자는 <code>java.util.TimeZone.getTimeZone()</code> 에 사용되는 식별자와 동일합니다. 예: GMT, Europe/Rome, Asia/Tokyo.	문자열	지역 표준 시간대
<code>imq.metrics.enabled</code>	메트릭 정보를 로거에 기록할지 여부를 지정하는 부울 값입니다. 메트릭 메시지 생성에는 영향을 미치지 않습니다( <code>imq.metrics.topic.enabled</code> 참조).	부울	true

표 14-10 모니터링 서비스 등록 정보(계속)

등록 정보 이름	설명	유형	기본값
imq.metrics.interval	메트릭 로깅이 사용 가능한 경우 (imq.metrics.enabled=true) 메트릭 정보가 로거에 기록되는 간격(초)입니다. 메트릭 메시지 생성 간격에는 영향을 미치지 않습니다 (imq.metrics.topic.interval 참조). 값이 -1이면 기록하지 않습니다.	정수	-1
imq.metrics.topic.enabled	메트릭 메시지 생성을 사용할지 여부를 지정하는 부울 값입니다. false인 경우 메트릭 주제 대상에 가입하려고 하면 클라이언트측 예외가 발생합니다.	부울	true
imq.metrics.topic.interval	메트릭 메시지를 생성하여 메트릭 주제 대상에 보내는 간격(초)입니다.	정수	60
imq.metrics.topic.persist	메트릭 메시지가 지속되는지 여부를 지정하는 부울 값입니다.	부울	false
imq.metrics.topic.timetolive	메트릭 주제 대상에 보낸 메트릭 메시지의 수명(초)입니다.	정수	300

1. 이 등록 정보는 imqcmd update bkr과 함께 사용할 수 있습니다.

## 클러스터 구성 등록 정보

표 14-11에서는 브로커 클러스터와 관련된 구성 등록 정보를 요약합니다.

표 14-11 클러스터 구성 등록 정보

등록 정보 이름	설명	유형	기본값
imq.cluster.brokerlist	클러스터의 모든 브로커를 식별하는 쉼표로 구분된 <i>host:port</i> 항목 목록이며, 여기서 <i>host</i> 는 각 브로커의 호스트 이름, <i>port</i> 는 포트 매핑의 포트 번호입니다. 예: host1:3000,host2:8000,ctrlhost 클러스터의 모든 브로커가 동일한 값을 가져야 합니다.	문자열	없음



표 14-11 클러스터 구성 등록 정보(계속)

등록 정보 이름	설명	유형	기본값
<code>imq.cluster.masterbroker</code>	클러스터 마스터 브로커(있는 경우)의 호스트 이름과 포트 번호입니다.  값의 형식은 <code>host:port</code> 입니다. 여기서 <code>host</code> 는 마스터 브로커의 호스트 이름이고 <code>port</code> 는 포트 매퍼의 포트 번호입니다.  예: <code>ctrlhost:7676</code>  클러스터의 모든 브로커가 동일한 값을 가져야 합니다.	문자열	없음
<code>imq.cluster.url<sup>1</sup></code>	클러스터 구성 파일(있는 경우)의 URL입니다.  예: <code>http://webserver/imq/cluster.properties</code> (웹 서버의 파일인 경우) <code>file:/net/mfsserver/imq/cluster.properties</code> (공유 드라이브의 파일인 경우)  클러스터의 모든 브로커가 동일한 값을 가져야 합니다.	문자열	없음
<code>imq.cluster.hostname</code>	사용 가능한 호스트가 두 개 이상인 경우(예: 컴퓨터에 네트워크 인터페이스 카드가 두 개 이상 있는 경우) 클러스터 연결 서비스(클러스터의 브로커 간 내부 통신에 사용됨)가 바인드될 호스트 이름 또는 IP 주소입니다.  클러스터의 각 브로커에 대해 독립적으로 지정할 수 있습니다.	문자열	<code>imq.hostname</code> 값에서 상속됨(311페이지의 표 14-2 참조)
<code>imq.cluster.port</code>	클러스터 연결 서비스의 포트 번호입니다.  클러스터의 각 브로커에 대해 독립적으로 지정할 수 있습니다.	정수	0(동적으로 할당됨)
<code>imq.cluster.transport</code>	클러스터 연결 서비스에 사용되는 네트워크 전송 프로토콜입니다. 브로커 간의 암호화된 보안 메시지를 전달하려면 이 등록 정보를 <code>ssl</code> 로 설정합니다.  클러스터의 모든 브로커가 동일한 값을 가져야 합니다.	문자열	<code>tcp</code>

1. 이 등록 정보는 `imqcmd update bkr`과 함께 사용할 수 있습니다.



## 물리적 대상 등록 정보 참조

이 장에서는 각 물리적 대상 유형에 대해 설정 가능한 등록 정보를 설명합니다. 물리적 대상을 만들거나 업데이트할 때 등록 정보 값을 설정할 수 있습니다.

자동 작성 대상의 경우 브로커의 인스턴스 구성 파일에 기본값을 설정합니다([314페이지](#)의 [표 14-4](#) 참조).

**표 15-1** 물리적 대상 등록 정보

등록 정보	대상 유형	기본값	설명
maxNumMsgs <sup>1</sup>	대기열 주제	-1 (제한 없음)	대상에 허용되는 사용되지 않은 최대 메시지 수를 지정합니다. 사용 불능 메시지 대기열의 경우 기본값이 1000입니다.
maxTotalMsgBytes <sup>1</sup>	대기열 주제	-1 (제한 없음)	대상의 사용되지 않은 메시지에 허용되는 최대 메모리 합계(바이트)를 지정합니다. 사용 불능 메시지 대기열의 기본값은 10MB입니다.

**표 15-1** 물리적 대상 등록 정보(계속)

등록 정보	대상 유형	기본값	설명
limitBehavior	대기열 주제	REJECT_ NEWEST	<p>메모리 제한 임계값에 도달할 경우에 브로커가 응답하는 방법을 지정하는 문자열입니다. 값은 다음과 같습니다.</p> <p>FLOW_CONTROL — 생성자 수를 조금씩 줄입니다.</p> <p>REMOVE_OLDEST — 가장 오래된 메시지를 삭제합니다.</p> <p>REMOVE_LOW_PRIORITY — 메시지의 보존 기간을 기준으로 우선 순위가 가장 낮은 메시지를 삭제합니다(생성자 클라이언트는 메시지 삭제에 대한 알림 메시지를 받지 않음).</p> <p>REJECT_NEWEST — 최신 메시지를 거부합니다. 생성자 클라이언트는 지속성 메시지가 거부되는 경우에만 예외 메시지를 받습니다. 이 제한 동작을 비지속성 메시지와 함께 사용하려면 <code>imgAckOnProduce</code> 연결 팩토리 속성을 설정합니다.</p> <p>이 등록 정보를 REMOVE_OLDEST 또는 REMOVE_LOW_PRIORITY로 설정하고 대상 등록 정보 <code>useDMQ</code>를 true로 설정하는 경우 브로커는 초과하는 메시지를 사용 불가능 메시지 대기열로 이동합니다.</p> <p>다른 대상과는 다르게 사용 불가능 메시지 대기열의 기본 제한 동작은 REMOVE_OLDEST로 자동으로 설정되며 FLOW_CONTROL 동작으로 설정할 수 없습니다.</p>
maxBytesPerMsg	대기열 주제	-1 (제한 없음)	<p>대상에 허용되는 단일 메시지의 최대 크기(바이트)를 지정합니다. 생성자 클라이언트는 <code>ackOnProduce</code> 등록 정보가 설정되어 있지 않는 한 지속성 메시지가 거부될 경우에는 예외 메시지를 받지만 비지속성 메시지가 거부될 때는 알림 메시지를 받지 않습니다.</p>
maxNumProducers	대기열 주제	-1 (제한 없음)	<p>대상에 허용되는 최대 생성자 수를 지정합니다. 이 제한에 도달하면 새로운 생성자가 생성되지 않습니다.</p> <p>사용 불가능 메시지 대기열에 대해서는 이 등록 정보를 설정할 수 없습니다.</p>
maxNumActiveConsumers	대기열에만 해당	1	<p>대기열 대상으로부터의 로드 균형 조정 전달에서 활성 상태가 될 수 있는 최대 사용자 수를 지정합니다. 값 -1은 제한이 없음을 의미합니다.</p> <p>플랫폼판은 이 값을 2로 제한합니다.</p>

**표 15-1** 물리적 대상 등록 정보(계속)

등록 정보	대상 유형	기본값	설명
maxNumBackupConsumers	대기열에만 해당	0	대기열 대상으로부터의 로드 균형 조정 전달 중에 오류가 발생할 경우 활성 사용자를 대신할 수 있는 백업 사용자의 최대 수를 지정합니다. 값 -1은 무제한을 의미합니다. 플랫폼판은 이 값을 1로 제한합니다.
consumerFlowLimit	대기열 주제	주제: 1000 대기열: 1000	일괄적으로 사용자에게 전달되는 최대 메시지 수를 지정합니다. 로드 균형 조정된 대기열 전달에서 이 수는 로드 균형 조정을 시작하기 전에 활성 사용자에게 경로 지정되는 초기 대기 메시지 수입니다. 대상 사용자는 연결에 대해 낮은 값을 지정하여 이 제한을 무시할 수 있습니다. 값 -1은 무제한을 의미합니다.
localDeliveryPreferred	대기열에만 해당	false	브로커 클러스터의 로드 균형 조정된 대기열 전달에서 로컬 브로커에 사용자가 없는 경우에만 원격 사용자에게 메시지를 전달할지 여부를 지정하는 부울 값입니다. 대상을 로컬에만 전달로 제한해서는 안 됩니다(isLocalOnly = false). 이 등록 정보는 사용 불능 메시지 대기열에는 적용되지 않습니다.
isLocalOnly	대기열 주제	false	브로커 클러스터 대상의 경우 대상을 로컬 전용 전달로 제한할지 여부를 지정하는 부울 값입니다. 이 값이 true이면 대상을 다른 브로커에 복사하지 않으므로 메시지 전달이 로컬 사용자(대상이 생성된 브로커에 연결된 사용자)에게만 제한됩니다. 대상이 생성된 이후에는 이 속성을 변경할 수 없습니다. 이 등록 정보는 사용 불능 메시지 대기열에는 적용되지 않습니다.
useDMQ	대기열 주제	true	사용 불능 메시지를 삭제하는 대신 사용 불능 메시지 대기열로 보낼지 여부를 지정하는 부울 값입니다. 이 등록 정보는 사용 불능 메시지 대기열에는 적용되지 않습니다.

1. 클러스터 환경에서 이 등록 정보는 클러스터의 모든 인스턴스에 한꺼번에 적용되는 것이 아니라 클러스터에 있는 대상의 각 인스턴스에 적용됩니다.



# 관리 대상 객체 속성 참조

이 장에서는 관리 대상 객체의 속성에 대한 참조 정보를 제공합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 333페이지의 "대상 등록 정보"
- 334페이지의 "연결 팩토리 속성"
- 342페이지의 "SOAP 종점 속성"

## 대상 등록 정보

표 16-1에는 대상 관리 대상 객체를 구성하는 속성이 나열되어 있습니다.

**표 16-1** 대상 관리 대상 객체 속성

속성 이름	설명	유형	기본값
imqDestinationDescription	대상 객체의 설명	문자열	없음
imqDestinationName	물리적 대상의 이름	문자열 <sup>1</sup>	Untitled_Destination_Object

1. 대상 이름은 영문자와 숫자만 포함할 수 있으며(공백 없음) 영문자 또는 "\_" 문자나 "\$" 문자로 시작해야 합니다.

## 연결 팩토리 속성

이 절에는 연결 팩토리 관리 대상 객체를 구성하는 속성에 대한 참조 정보가 포함되어 있습니다. 속성은 다음과 같은 내용으로 분류됩니다.

- 334페이지의 "연결 처리"
- 338페이지의 "클라이언트 아이디"
- 338페이지의 "메시지 헤더 무시"
- 339페이지의 "안정성 및 흐름 제어"
- 340페이지의 "대기열 브라우저 동작 및 서버 세션"
- 341페이지의 "JMS 정의 등록 정보 지원"

표 16-2는 연결 팩토리 관리 대상 객체 속성에 대한 색인입니다. 첫 번째 열에는 각 속성이 알파벳순으로 나열되고 두 번째 열에는 해당 범주가 나열되며 세 번째 열은 속성을 설명하는 표에 대한 상호 참조입니다.

## 연결 처리

표 16-2에는 연결 처리를 위한 연결 팩토리 속성이 나열되어 있습니다.

**표 16-2** 연결 팩토리 속성: 연결 처리

속성 이름	설명	유형	기본값
imqAddressList	하나 이상의 메시지 서버 주소를 쉼표로 구분한 목록입니다. 사용할 연결 서비스와 포트 지정 방법과 관련된 몇 가지 주소 지정 체계가 있습니다.  주소 목록 지정 방법과 목록 항목을 설명하는 예에 대한 자세한 내용은 335페이지의 "imqAddressList 속성 값 구문"을 참조하십시오.	문자열	기존 Message Queue 3.0 주소 (있는 경우) 또는 336페이지의 표 16-3의 첫 번째 항목(없는 경우)
imqAddressListBehavior	연결을 imqAddressList 속성에 있는 주소 순서 (PRIORITY)로 시도할지 임의의 순서(RANDOM)로 시도할지 여부를 지정하는 문자열입니다. 많은 클라이언트가 동일한 연결 팩토리를 사용하여 연결을 시도할 경우 임의의 순서를 사용하면 모든 클라이언트가 같은 주소에 연결되는 것을 막을 수 있습니다.	문자열	PRIORITY
imqAddressListIterations	클라이언트 런타임이 연결을 설정 또는 재설정하기 위해 imqAddressList를 반복하는 횟수입니다. 값이 -1이면 시도 횟수에 제한이 없습니다.	정수	5



표 16-2 연결 팩토리 속성: 연결 처리(계속)

속성 이름	설명	유형	기본값
imqPingInterval	클라이언트 런타임이 응용 프로그램과 브로커 간의 연결을 테스트하는 빈도(초)입니다. 값이 -1 또는 0(영)이면 클라이언트 런타임이 연결을 주기적으로 테스트하지 않습니다.	정수	30
imqReconnectEnabled	연결이 끊겼을 때 클라이언트 런타임이 메시지 서버 또는 imqAddressList의 주소 목록에 재연결을 시도해야 하는지 여부를 지정하는 부울 값입니다.	부울	false
imqReconnectAttempts	클라이언트 런타임이 목록의 다음 주소를 연결하려고 시도하기 전에 imqAddressList의 각 주소에 대해 연결 또는 재연결을 시도하는 횟수입니다. 값이 -1이면 재연결 시도 횟수가 제한되지 않으므로 클라이언트 런타임이 연결에 성공할 때까지 첫 번째 주소에 연결하려고 시도합니다.	정수	0(영)
imqReconnectInterval	재연결 시도 간의 간격(밀리초)입니다. 이 값은 imqAddressList의 각 주소에 대한 시도와 목록의 연속적인 주소에 대해 적용됩니다. 값이 너무 작은 경우 브로커의 복구 시간이 부족합니다. 값이 너무 큰 경우 재연결 시 너무 오래 지연될 수 있습니다.	long	3000(밀리초)
imqSSLIsHostTrusted	클라이언트가 브로커의 자체 서명된 인증서를 승인할 수 있는지 여부를 지정하는 부울 값입니다. 인증 기관의 서명된 인증서를 사용하려면 이 값을 false로 설정합니다.	부울	true

### imqAddressList 속성 값 구문

imqbrokerlist 값의 각 주소는 클라이언트 런타임이 연결할 수 있는 브로커 인스턴스에 해당합니다.

연결 서비스마다 다른 브로커 주소를 지정합니다. 구문은 일반적으로 다음과 같습니다.

```
scheme://address_syntax
```

목록에 주소를 추가하려면 쉼표와 다른 주소를 추가합니다. 목록에는 다음과 같은 형식으로 여러 항목이 표시될 수 있습니다.

```
scheme://address_syntax,scheme://address_syntax...
```

*scheme* 변수는 표 16-3에서 설명한 대로 *mq*, *mqtcp*, *mqssl*, *http*, *https* 중에서 사용 중인 주소 유형을 지정합니다. *address\_syntax* 변수는 체계별 브로커 주소를 나타냅니다. 주소 지정 체계는 표 16-3에 나와 있습니다. 첫 번째 열에는 주소 지정 체계의 이름이 표시되고 두 번째 열에는 해당 이름과 연결되는 연결 서비스가 표시되며 세 번째 열에는 설명이 표시되고 네 번째 열에는 사용할 구문이 제공됩니다.

**표 16-3** imqAddressList 속성의 주소 지정 체계

방식	Connection 서비스	설명	구문
mq	ms ssljms	<p>jms 또는 ssljms 서비스와 함께 사용할 포트를 동적으로 할당합니다.</p> <p>포트 매퍼 호스트와 포트를 지정합니다. 포트 매퍼는 연결에 사용되는 포트를 동적으로 지정합니다.</p>	<p><i>[hostName][:port]/[serviceName]</i></p> <p>jms 연결 서비스에는 다음과 같은 기본값이 적용됩니다.</p> <p><i>hostName</i> = localhost <i>port</i> = 7676 <i>serviceName</i> = jms</p> <p>ssljms 연결 서비스에는 기본값이 없습니다. 모든 변수를 지정해야 합니다.</p>
mqtcp	jms	<p>포트 번호를 지정하고 jms 연결 서비스를 사용합니다.</p> <p>Message Queue 클라이언트 런타임은 지정된 호스트와 포트에 tcp 연결을 설정하여 연결합니다.</p>	<i>hostName:port/jms</i>
mqssl	ssljms	<p>포트 번호를 지정하고 ssljms 연결 서비스를 사용합니다.</p> <p>Message Queue 클라이언트 런타임은 지정된 호스트와 포트에 보안 ssl 연결을 설정하여 연결합니다.</p>	<i>port/ssljms</i>
http	httpjms	<p>httpjms 연결 서비스를 사용합니다.</p> <p>클라이언트 런타임은 지정된 URL의 Message Queue 터널 서블릿에 HTTP 연결을 설정합니다. 브로커가 HTTP 터널 서블릿에 액세스하도록 구성해야 합니다.</p>	<p><i>http://hostName:port/contextRoot/tunnel</i></p> <p>여러 브로커 인스턴스가 동일한 터널 서블릿을 사용하는 경우 다음 구문을 사용하여 임의로 선택된 인스턴스가 아니라 특정 브로커 인스턴스에 연결합니다.</p> <p><i>http://hostName:port/contextRoot/tunnel?ServerName=hostName:instanceName</i></p>
https	httpsjms	<p>httpsjms 연결 서비스를 사용합니다.</p> <p>Message Queue 클라이언트 런타임은 지정된 Message Queue 터널 서블릿 URL에 보안 HTTPS 연결을 설정합니다. 브로커가 HTTPS 터널 서블릿에 액세스하도록 구성해야 합니다.</p>	<p><i>https://hostName:port/contextRoot/tunnel</i></p> <p>여러 브로커 인스턴스가 동일한 터널 서블릿을 사용하는 경우 다음 구문을 사용하여 임의로 선택된 인스턴스가 아니라 특정 브로커 인스턴스에 연결합니다.</p> <p><i>https://hostName:port/contextRoot/tunnel?ServerName=hostName:instanceName</i></p>

표 16-4에는 주소 지정 형식의 예가 포함되어 있습니다. 첫 번째 열은 연결 서비스의 이름입니다. 두 번째 열은 예에 나오는 호스트가 로컬 호스트인지, 지정되지 않은 호스트인지, 지정된 호스트인지 또는 적용할 없는 호스트인지를 지정합니다. 세 번째 열은 예에 나오는 포트가 지정되는지, 지정되지 않는지 또는 적용할 수 없는지 여부를 지정하고, 네 번째 열은 예입니다.

**표 16-4** 메시지 서버 주소 예

연결 서비스	브로커 호스트	포트	주소 예
지정하지 않음	로컬 호스트	지정하지 않음	기본값 (mq://localhost:7676/jms)
지정하지 않음	지정된 호스트	지정하지 않음	myBkrHost (mq://myBkrHost:7676/jms)
지정하지 않음	지정하지 않음	지정된 포트 매핑 포트	1012 (mq://localhost:1012/jms)
ssljms	로컬 호스트	지정되지 않은 포트 매핑 포트	mq://localhost:7676/ssljms
ssljms	지정된 호스트	포트 매핑 포트	mq://myBkrHost:7676/ssljms
ssljms	지정된 호스트	지정된 포트 매핑 포트	mq://myBkrHost:1012/ssljms
jms	로컬 호스트	지정된 서비스 포트	mqtcp://localhost:1032/jms
ssljms	지정된 호스트	지정된 서비스 포트	mqssl://myBkrHost:1034/ssljms
httpjms	적용할 수 없음	적용할 수 없음	http://webservr1:8085/imq/tunnel
httpsjms	적용할 수 없음	적용할 수 없음	https://webservr2:8090/imq/tunnel

## 클라이언트 아이디

표 16-5에는 클라이언트 아이디에 대한 연결 팩토리 속성이 나열되어 있습니다.

**표 16-5** 연결 팩토리 속성: 클라이언트 아이디

속성 이름	설명	유형	기본값
imqDefaultUsername	브로커 인증을 위한 기본 아이디입니다.	문자열	guest
imqDefaultPassword	브로커 인증을 위한 기본 비밀번호입니다.	문자열	guest
imqConfiguredClientID	관리상 구성된 클라이언트 아이디입니다.	문자열	null
imqDisableSetClientID	JMS API의 setClientID() 메소드를 사용하여 클라이언트가 클라이언트 아이디를 변경하지 못하게 할지 여부를 지정하는 부울 값입니다.	부울	false

## 메시지 헤더 무시

표 16-6에는 JMS 메시지 헤더 필드 무시를 위한 연결 팩토리 속성이 나열되어 있습니다.

**표 16-6** 연결 팩토리 속성: 메시지 헤더 무시

속성 이름	설명	유형	기본값
imqOverrideJMSDeliveryMode	클라이언트에서 설정한 JMSDeliveryMode 필드를 무시할 수 있는지 여부를 지정하는 부울 값입니다.	부울	false
imqJMSDeliveryMode	JMSDeliveryMode의 무시하는 값입니다. 값은 1(비지속성) 또는 2(지속성)입니다.	정수	2
imqOverrideJMSEExpiration	클라이언트에서 설정한 JMSEExpiration 필드를 무시할 수 있는지 여부를 지정하는 부울 값입니다.	부울	false
imqJMSEExpiration	JMSEExpiration의 무시하는 값(밀리초)입니다.	long	0 (만료 없음)
imqOverrideJMSPriority	클라이언트에서 설정한 JMSPriority 필드를 무시할 수 있는지 여부를 지정하는 부울 값입니다.	부울	false
imqJMSPriority	JMSPriority의 무시하는 값(0에서 9까지의 정수)입니다.	정수	4 (일반)
imqOverrideJMSHeadersToTemporaryDestinations	무시가 임시 대상에 적용되는지 여부를 지정하는 부울 값입니다.	부울	false

## 안정성 및 흐름 제어

표 16-7에는 안정성 및 흐름 제어를 구성하는 연결 팩토리 속성이 나열되어 있습니다.

표 16-7 연결 팩토리 속성: 안정성 및 흐름 제어

속성 이름	설명	유형	기본값
imqAckTimeout	클라이언트 런타임이 브로커 응답을 기다리는 최대 시간(밀리초)입니다. 이 시간이 초과되면 예외가 발생합니다. 값이 0이면 시간 초과가 발생하지 않고 클라이언트 런타임이 계속해서 대기합니다.  이 값이 너무 낮아 클라이언트 런타임이 시간 초과하는 경우도 있습니다. 예를 들어, 브로커가 LDAP 사용자 저장소와 보안(SSL) 연결을 사용하여 인증하는 각 사용자에게 대해 첫 번째 인증이 30초 이상 걸릴 수 있습니다.	문자열	0
imqAckOnProduce	브로커가 생성자 클라이언트의 메시지에 응답하는 방법을 지정하는 문자열입니다.  이 속성을 true로 설정하면 브로커는 생성자 클라이언트로부터 수신하는 모든 JMS 메시지(지속성 및 비지속성)에 응답합니다. 이러한 응답을 대기하는 동안에는 생성자 클라이언트 스레드가 차단됩니다.  이 속성을 지정하지 않은 경우 브로커는 지속성 메시지에만 응답합니다. 이러한 응답을 대기하는 동안에는 생성자 클라이언트 스레드가 차단됩니다.	문자열	지정 안 됨
imqConnectionFlowCount	측정된 일괄 처리의 JMS 메시지 수입니다. 이 JMS 메시지 수가 클라이언트 런타임에 전달되면 전달이 일시적으로 중단되어 보류된 제어 메시지를 모두 전달할 수 있도록 합니다. 클라이언트 런타임의 알림 메시지가 도착하면 페이로드 메시지 전달이 다시 시작되어 지정된 수에 이를 때까지 계속됩니다.  지정된 수가 0이면 측정된 일괄 처리에서 JMS 메시지의 수에 제한이 없습니다. 0이 아닌 값을 설정하면 클라이언트 런타임에서 메시지 흐름을 측정하여 JMS 메시지 전달 폭주로 인해 Message Queue 제어 메시지가 차단되지 않도록 합니다.	정수	100
imqConnectionFlowLimitEnabled	imqConnectionFlowLimit 값을 사용하여 연결 수준에서 메시지 흐름을 제한할지 여부를 지정하는 부울 값입니다.	부울	false
imqConnectionFlowLimit	연결을 통해 전달되고 클라이언트 런타임에 버퍼링되어 사용 대기 중일 수 있는 최대 메시지 수입니다. 하지만 imqConnectionFlowIsLimited가 사용 가능한 경우에만 이 제한을 선택할 수 있습니다.  imqConnectionFlowCount에서 관리하는 흐름 측정에 따라 클라이언트 런타임으로 전달되는 JMS 메시지의 수가 이 제한을 초과하면 메시지 전달이 중지됩니다. 사용되지 않는 메시지의 수가 이 속성의 설정 값보다 낮아지는 경우에만 메시지 전달이 다시 시작됩니다.  이 제한은 메시지 처리 시간이 오래 걸리는 사용자 클라이언트에 대기 중인 메시지가 너무 많아 메모리 부족이 발생하는 것을 방지합니다.	정수	1000

**표 16-7** 연결 팩토리 속성: 안정성 및 흐름 제어(계속)

속성 이름	설명	유형	기본값
imqConsumerFlowLimit	<p>연결을 통해 전달되고 클라이언트 런타임에 버퍼링되어 사용 대기 중일 수 있는 사용자별 최대 메시지 수입입니다. 이 제한은 다중 사용자 대기열 전달의 경우 사용자 간의 로드 균형 조정을 향상시키는 데 사용됩니다(한 사용자에게 보내는 메시지 수의 불균형 방지). 브로커 쪽에서 대기열의 consumerFlowLimit 속성 값을 더 낮게 설정하여 이 제한을 무시할 수 있습니다(Message Queue 관리 설명서의 대상 속성 정보 참조).</p> <p>또한 이 제한을 사용하면 연결된 한 사용자가 연결된 다른 사용자에 비해 메시지가 폭주하는 것을 막을 수 있습니다.</p> <p>어떤 사용자에 대해서도 클라이언트 런타임에 전달되는 JMS 메시지의 수가 이 제한을 초과하면 해당 사용자에 대한 메시지 전달이 중단됩니다. 해당 사용자의 사용되지 않은 메시지 수가 imqConsumerFlowThreshold에 설정된 값 이하로 떨어져야만 메시지 전달이 재개됩니다.</p> <p>특정 연결에서 모든 사용자에 대해 버퍼링된 전체 메시지 수가 imqConnectionFlowLimit을 초과하면 전체 메시지 수가 연결 제한 이하로 떨어질 때까지 해당 연결을 통한 메시지 전달이 중단됩니다.</p>	정수	100
imqConsumerFlowThreshold	<p>클라이언트 런타임에 버퍼링된 사용자당 메시지의 수가 imqConsumerFlowLimit에 대한 백분율로 지정하는데 이 값보다 낮으면 해당 사용자에 대한 메시지 전달이 다시 시작됩니다.</p>	정수	50

## 대기열 브라우저 동작 및 서버 세션

표 16-8에는 대기열에서 클라이언트 찾아보기에 영향을 미치는 속성이 설명되어 있습니다.

**표 16-8** 연결 팩토리 속성: 대기열 브라우저 동작

속성 이름	설명	유형	기본값
imqQueueBrowserMaxMessagesPerRetrieve	대기열 대상의 내용을 찾아볼 때 클라이언트 런타임이 한 번에 검색하는 메시지의 최대 수입입니다.	정수	1000
imqQueueBrowserRetrieveTimeout	대기열 대상의 내용을 찾아볼 때 클라이언트 런타임이 예외를 발생하기 전에 메시지 검색을 대기하는 최대 시간(밀리초)입니다.	long	60000
imqLoadMaxToServerSession	JMS 응용 프로그램 서버 기능의 경우 Message Queue 연결 사용자가 ServerSession의 세션에서 maxMessages 메시지 수까지 로드해야 하는지 여부를 지정하는 부울 값입니다. 값이 false이면 클라이언트는 한 번에 하나의 메시지만 로드합니다.		true

## JMS 정의 등록 정보 지원

JMS 정의 등록 정보는 JMS에 의해 예약되고 클라이언트 프로그래밍 기능 향상 지원을 위해 JMS 공급자가 선택할 수 있는 이름입니다. Message Queue에서 지원되는 JMS 정의 등록 정보에 대한 설명은 표 16-9에 나와 있습니다.

**표 16-9** 연결 팩토리 속성: JMS 정의 등록 정보 지원

등록 정보 이름	설명	유형	기본값
imqSetJMSXUserID	생성된 메시지에 대해 JMS 정의 등록 정보 JMSXUserID(메시지를 보내는 사용자의 아이디)를 설정할지 여부를 지정하는 부울 값입니다.	부울	false
imqSetJMSXAppID	생성된 메시지에 대해 JMS 정의 등록 정보 JMSXAppID(메시지를 보내는 응용 프로그램의 아이디)를 설정할지 여부를 지정하는 부울 값입니다.	부울	false
imqSetJMSXProducerTXID	생성된 메시지에 대해 JMS 정의 등록 정보 JMSXProducerTXID(이 메시지를 생성한 트랜잭션의 트랜잭션 아이디)를 설정할지 여부를 지정하는 부울 값입니다.	부울	false
imqSetJMSXConsumerTXID	사용된 메시지에 대해 JMS 정의 등록 정보 JMSXConsumerTXID(이 메시지를 사용한 트랜잭션의 트랜잭션 아이디)를 설정할지 여부를 지정하는 부울 값입니다.	부울	false
imqSetJMSXRcvTimestamp	사용된 메시지에 대해 JMS 정의 등록 정보 JMSXRcvTimestamp(메시지가 사용자에게 전달되는 시간)를 설정할지 여부를 지정하는 부울 값입니다.	부울	false

## SOAP 종점 속성

표 16-10에는 SOAP를 사용하는 응용 프로그램의 종점 URL을 구성하는 속성이 나열되어 있습니다. SOAP를 사용하는 응용 프로그램에 대한 자세한 내용은 *Message Queue Developer's Guide for Java Clients*를 참조하십시오.

**표 16-10** SOAP 종점 속성

속성 이름	설명	유형	기본값
imqSOAPEndpointList	메시지를 보낼 SOAP 종점을 나타내는 많은 공백으로 구분된 URL 중 하나의 목록입니다.  URL을 두 개 이상 지정하면 메시지가 목록에 있는 모든 URL로 브로드캐스트됩니다. 각 URL은 서블릿에 연결되어 있어야 SOAP 메시지를 받아서 처리할 수 있습니다.  예: http://www.myServlet/ http://www.myServlet2/	문자열	
imqEndpointName	SOAP 종점의 이름입니다.  예: MyTopicEndpoint	문자열	Untitled_Endpoint_Object
imqEndpointDescription	SOAP 종점에 대한 설명입니다.  예: "imqEndpointDescription=my endpoints for broadcast"	문자열	종점 객체에 대한 설명



## JMS 자원 어댑터 속성 참조

Message Queue JMS 자원 어댑터(JMS RA)를 사용하면 표준 JCA(J2EE Connector Architecture)를 통해 Sun Java System Message Queue를 J2EE 1.4 응용 프로그램 서버와 통합할 수 있습니다. Message Queue JMS 자원 어댑터를 응용 프로그램 서버에 플러그인 하면 해당 응용 프로그램 서버에 배포된 응용 프로그램에서 Message Queue를 사용하여 JMS 메시지를 보내고 받을 수 있습니다.

Message Queue JMS 자원 어댑터는 세 가지 JavaBean 세 구성 요소를 통해서 구성 속성을 제공합니다.

- ResourceAdapter 구성은 자원 어댑터의 동작에 전체적으로 영향을 미칩니다.
- ManagedConnectionFactory 구성은 MDB(Message-Driven Bean)에서 사용하기 위해 자원 어댑터에서 생성한 연결에 영향을 미칩니다.
- ActivationSpec 구성은 메시징 시스템과의 상호 작용에서 MDB(Message Driven Bean)를 나타내는 메시지 종점에 영향을 미칩니다.

이러한 항목에 대한 속성 값을 설정하려면 응용 프로그램 서버에서 자원 어댑터의 구성과 배포 및 MDB의 배포를 위해 제공하는 도구를 사용하십시오.

이 장에서는 Message Queue JMS 자원 어댑터의 구성 속성을 나열하고 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 344페이지의 "ResourceAdapter JavaBean"
- 345페이지의 "ManagedConnectionFactory JavaBean"
- 346페이지의 "ActivationSpec JavaBean"

# ResourceAdapter JavaBean

ResourceAdapter 구성은 기본 JMS 자원 어댑터 동작을 구성합니다. 표 17-1에서는 이 JavaBean을 구성할 때 사용할 수 있는 속성을 나열하고 설명합니다. 각주는 필수 등록 정보를 각각 표시합니다.

**표 17-1** 자원 어댑터 속성

이름	설명	기본값
addressList <sup>1</sup>	<p>자원 어댑터가 Message Queue 서비스에 설정하는 연결로 메시지 서비스 주소 형식을 사용하여 지정됩니다.</p> <p>자원 어댑터가 기본값을 제공합니다.</p> <p>이 속성 이름 addressList는 Sun Java System Message Queue와 관련되어 있지만 표준 속성 connectionURL과 같은 의미를 갖습니다. Sun Java System Message Queue에서는 두 속성 이름을 모두 제공합니다. connectionURL 또는 addressList를 설정해야 합니다. 두 속성은 동일합니다.</p>	mq://localhost:7676/jms
addressListBehavior	<p>자원 어댑터가 Message Queue 서비스에 연결하는 방법을 지정하는 문자열입니다. 값은 PRIORITY 또는 RANDOM입니다.</p> <p>PRIORITY 연결은 주소 목록(addressList)에 지정된 첫 번째 주소를 선택하여 Message Queue 브로커를 선택합니다.</p> <p>RANDOM 연결은 주소 목록에서 Message Queue 브로커를 임의로 선택합니다.</p> <p>연결 실패 후의 재연결은 PRIORITY와 RANDOM이 동일합니다. 재연결 시도는 연결이 실패한 브로커에서 시작됩니다. 재연결 시도도 실패할 경우 자원 어댑터는 활성 주소 목록을 통해 순차적으로 연결을 시도합니다.</p>	PRIORITY
addressListIterations	주소 목록을 통해 반복하는 횟수입니다. 이 값은 초기 연결과 후속 재연결 시도에 적용됩니다.	1
connectionURL	<p>자원 어댑터가 Message Queue 서비스에 설정하는 연결로 메시지 서비스 주소 형식을 사용하여 지정됩니다.</p> <p>addressList 속성과 동일합니다. 자세한 내용은 위 설명을 참조하십시오.</p>	mq://localhost:7676/jms
userName <sup>1</sup>	<p>자원 어댑터가 Message Queue 서비스에 연결할 때 사용하는 기본 사용자 아이디입니다.</p> <p>자원 어댑터가 기본값을 제공합니다.</p>	guest
password <sup>1</sup>	<p>자원 어댑터가 Message Queue 서비스에 연결할 때 사용하는 기본 비밀번호입니다.</p> <p>자원 어댑터가 기본값을 제공합니다.</p>	guest

**표 17-1** 자원 어댑터 속성(계속)

이름	설명	기본값
reconnectAttempts	주소 목록의 단일 항목에 재연결을 시도하는 횟수입니다. 이 속성은 reconnectEnabled를 true로 설정할 때 사용됩니다.	6
reconnectEnabled	연결 실패 후 재연결을 시도할지 여부를 지정하는 부울 값입니다. 재연결 시도 동작은 reconnectInterval 및 reconnectAttempts의 값에 따라 결정됩니다.	false
reconnectInterval	재연결 시도 간격(밀리초)입니다. 이 속성은 reconnectEnabled를 true로 설정할 때 사용됩니다.	30000

1. 이 등록 정보는 필수입니다.

## ManagedConnectionFactory JavaBean

관리 대상 연결 팩토리는 자원 어댑터가 Message-Driven Bean에 제공하는 연결을 설정하고 정의합니다. ResourceAdapter JavaBean에 유사한 속성이 있는 속성을 설정하면 해당 설정이 ResourceAdapter Bean에 지정된 유사 값보다 우선합니다.

표 17-2에서는 Message Queue 자원 어댑터가 제공하는 관리 대상 연결 팩토리의 구성 가능한 속성을 나열하고 설명합니다.

**표 17-2** 관리 대상 연결 팩토리 속성

이름	설명	기본값
addressList	이 관리 대상 연결 팩토리에서 파생된 연결 목록입니다. 이 등록 정보 형식은 344페이지의 표 17-1에서 설명한 대로 메시지 서비스 addressList를 따릅니다. 이 값을 설정하지 않을 경우 ResourceAdapter JavaBean에 지정되고 해당 표에 설명된 addressList 값이 연결에 사용됩니다.	없음
addressListBehavior	자원 어댑터가 Message Queue 서비스에 연결하는 방법을 지정하는 문자열입니다. 값은 PRIORITY 또는 RANDOM입니다. PRIORITY 연결은 주소 목록(addressList)에 지정된 첫 번째 주소를 선택하여 Message Queue 브로커를 선택합니다. RANDOM 연결은 주소 목록에서 Message Queue 브로커를 임의로 선택합니다. 연결 실패 후의 재연결은 PRIORITY와 RANDOM이 동일합니다. 재연결 시도는 연결이 실패한 브로커에서 시작됩니다. 재연결 시도도 실패할 경우 연결 시도는 활성 주소 목록을 통해 순차적으로 처리됩니다.	PRIORITY

**표 17-2** 관리 대상 연결 팩토리 속성(계속)

이름	설명	기본값
addressListIterations	주소 목록을 통해 반복하는 횟수입니다. 이 값은 초기 연결과 후속 재연결 시도에 적용됩니다.	1
clientID	이 관리 대상 연결 팩토리에서 파생된 연결에 사용할 클라이언트 식별자입니다.	없음
password	(선택 사항) 연결에 사용할 비밀번호입니다. 이 값을 설정하지 않으면 344페이지의 표 17-1에 설명한 대로 ResourceAdapter JavaBean에 지정된 비밀번호가 연결에 사용됩니다.	guest
reconnectAttempts	주소 목록의 단일 항목에 재연결을 시도하는 횟수입니다.	6
reconnectEnabled	연결 실패 후 재연결을 시도할지 새 연결을 시도할지 여부를 지정하는 부울 값입니다. 재연결 시도는 reconnectInterval 및 reconnectAttempts 등록 정보에 따라 결정됩니다.	false
reconnectInterval	Message Queue 서비스에 재연결을 시도하기 위해 대기하는 최소 간격(밀리초)입니다.	30000
userName	(선택 사항) 연결에 사용할 사용자 아이디입니다. 이 값을 설정하지 않으면 344페이지의 표 17-1에 설명한 대로 ResourceAdapter JavaBean에 지정된 사용자 아이디가 연결에 사용됩니다.	guest

## ActivationSpec JavaBean

ActivationSpec JavaBean 등록 정보는 응용 프로그램 서버가 메시지 중점을 활성화하여 Message-Driven Bean에 연결하도록 자원 어댑터에 지시할 때 사용됩니다.

표 17-3에서는 메시지 종점 활성화 사양에 대해 구성 가능한 속성을 나열하고 설명합니다. 이 표는 Message Queue 자원 어댑터 관련 등록 정보와 Enterprise JavaBean 2.1 표준 또는 J2EE CA(J2EE Connector Architecture) 1.5 표준 관련 등록 정보를 나타냅니다.

**표 17-3**      활성화 사양 속성

이름	설명	기본값
acknowledgeMode	(선택 사항) 사용자에게 사용할 JMS 세션 확인 모드입니다. 표준 EJB 2.1 및 J2EE CA 1.5 등록 정보입니다. 값은 Auto-acknowledge 또는 Dups-ok-acknowledge일 수 있습니다.	Auto-acknowledge
addressList	(선택 사항) 메시지 종점 대신 자원 어댑터에서 만든 연결 사양입니다. 이 속성은 Message Queue JMS 자원 어댑터와 관련됩니다. 유효한 값은 메시지 서비스 연결 주소 구문을 준수해야 합니다.	ResourceAdapter JavaBean 구성의 addressList에서 상속됨
clientId	이 사용자에게 대해 만든 JMS 연결에 사용할 JMS 클라이언트 아이디입니다. subscriptionDurability 속성을 Durable로 설정한 경우 이 속성을 설정해야 합니다. 표준 EJB 2.1 및 J2EE CA 1.5 등록 정보입니다.	없음
customAcknowledgeMode	MDB 메시지 사용을 위한 모드를 지정하는 문자열입니다. 이 속성에 유효한 값은 No_acknowledge 또는 null입니다. 비트랜잭션, 비영구 주제 가입에 대해서만 No_acknowledge 모드를 사용할 수 있습니다. 트랜잭션 가입 또는 영구 가입에 이 설정을 사용하면 가입이 활성화되지 않습니다.	없음
destination	이 MDB가 메시지를 사용하는 대상의 이름입니다. 이 속성은 필수 속성입니다. 표준 EJB 2.1 및 J2EE CA 1.5 등록 정보입니다. 이 값을 Message Queue 대상 관리 대상 객체의 destinationName 등록 정보 값으로 설정해야 합니다.	없음
destinationType	destination 속성에 지정된 대상의 유형입니다. 유효한 값은 javax.jms.Queue 또는 javax.jms.Topic입니다. 이 속성은 필수 속성입니다. 표준 EJB 2.1 및 J2EE CA 1.5 등록 정보입니다.	없음
endpointExceptionRedeliveryAttempts	MDB가 메시지 전달 중에 예외를 발생할 때 MDB에 메시지를 재전송하는 횟수입니다.	6

**표 17-3** 활성화 사양 속성(계속)

이름	설명	기본값
messageSelector	(선택 사항) 사용자에게 전달된 메시지를 필터링할 때 사용할 JMS 메시지 선택기입니다. 값 유형은 String입니다. 표준 EJB 2.1 및 J2EE CA 1.5 등록 정보입니다.	없음
sendUndeliverableMsgsToDMQ	MDB가 런타임 예외를 발생하고 재전송 시도 횟수가 endpointExceptionRedeliveryAttempts 값을 초과할 때 메시지를 사용 불능 메시지 대기열에 넣을지 여부를 지정하는 부울 값입니다.  값이 false이면 Message Queue 브로커는 동일한 MDB를 포함하여 유효한 사용자에게 메시지를 재전송하려고 시도합니다.	true
subscriptionDurability	주제 대상의 사용자가 영구 사용자인지 비영구 사용자인지를 지정하는 문자열입니다. 값은 NonDurable 또는 Durable일 수 있습니다.  이 속성은 비영구 가입의 경우 선택 속성이고 영구 가입의 경우 필수 속성입니다. 이 값을 Durable로 설정하는 경우 clientID 및 subscriptionName 속성도 설정해야 합니다.  이 속성은 표준 EJB 2.1 및 J2EE CA1.5 등록 정보이며 destinationType 속성을 avax.jms.Topic으로 설정한 경우에만 유효합니다.	NonDurable
subscriptionName	영구 가입의 이름을 지정할 때 사용할 문자열입니다.  subscriptionDurability 속성을 Durable로 설정한 경우 이 속성을 설정해야 합니다.  표준 EJB 2.1 및 J2EE CA 1.5 등록 정보입니다.	없음

# 메트릭 참조

이 장에서는 Message Queue 제품에서 생성되는 메트릭을 나열하고 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- [349페이지의 "JVM 메트릭"](#)
- [350페이지의 "브로커 전체 메트릭"](#)
- [352페이지의 "연결 서비스 메트릭"](#)
- [354페이지의 "대상 메트릭"](#)

## JVM 메트릭

[표 18-1](#)에서는 브로커가 브로커 프로세스 JVM 힙에 대해 생성하는 메트릭을 나열하고 설명합니다. 이 표에서는 각 메트릭을 제공하는 메트릭 모니터링 도구를 보여줍니다.

**표 18-1** JVM 메트릭

메트릭 개수	설명	imqcmd		
		metrics bkr (metricType)	로그 파일	메트릭 메시지 (metrics topic) <sup>2</sup>
JVM 힙: 사용 가능한 메모리	JVM 힙에서 사용할 수 있는 사용 가능한 메모리 양	예 (cxn)	예	예 (...jvm)
JVM 힙: 전체 메모리	현재 JVM 힙 크기	예 (cxn)	예	예 (...jvm)
JVM 힙: 최대 메모리	JVM 힙 크기가 늘어날 수 있는 최대값	아니요	예 <sup>1</sup>	예 (...jvm)

1. 브로커 시작 시에만 표시됩니다.
2. 메트릭 주제 대상 이름에 대해서는 [216페이지의 표 10-7](#)을 참조하십시오.

# 브로커 전체 메트릭

표 18-2에서는 브로커 전체 메트릭 정보에 대해 브로커가 보고하는 데이터를 나열하고 설명합니다. 또한 메트릭 모니터링 도구에 따라 어떤 데이터를 얻을 수 있는지 보여 줍니다.

**표 18-2** 브로커 전체 메트릭

메트릭 개수	설명	imqcmd metrics bkr (metricType)	로그 파일	메트릭 메시지 (metrics topic) <sup>1</sup>
<b>연결 데이터</b>				
Num connections	브로커에 대해 현재 열려 있는 연결 수	예 (cxn)	예	예 (...broker)
Num threads	모든 연결 서비스에 현재 사용 중인 전체 스레드 수	예 (cxn)	예	아니요
Min threads	이 스레드 수에 도달하면 스레드 풀에서 연결 서비스용으로 유지 관리되는 스레드 수	예 (cxn)	예	아니요
Max threads	이 스레드 수를 초과하면 스레드 풀에 연결 서비스용으로 새 스레드가 더 이상 추가되지 않는 스레드 수	예 (cxn)	예	아니요
<b>저장된 메시지 데이터</b>				
Num messages	브로커 메모리와 영구 저장소에 현재 저장되어 있는 JMS 메시지 수	아니요 query bkr 사용	아니요	예 (...broker)
Total message bytes	브로커 메모리와 영구 저장소에 현재 저장되어 있는 JMS 메시지 바이트 수	아니요 query bkr 사용	아니요	예 (...broker)
<b>메시지 흐름 데이터</b>				
Num messages in	브로커가 마지막으로 시작된 이후 브로커에 유입된 JMS 메시지 수	예 (ttl)	예	예 (...broker)
Message bytes in	브로커가 마지막으로 시작된 이후 브로커에 유입된 JMS 메시지 바이트 수	예 (ttl)	예	예 (...broker)
Num packets in	브로커가 마지막으로 시작된 이후 브로커에 유입된 패킷 수(JMS 메시지와 제어 메시지 모두 포함)	예 (ttl)	예	예 (...broker)
Packet bytes in	브로커가 마지막으로 시작된 이후 브로커에 유입된 패킷 바이트 수(JMS 메시지와 제어 메시지 모두 포함)	예 (ttl)	예	예 (...broker)



**표 18-2**      브로커 전체 메트릭(계속)

메트릭 개수	설명	imqcmd metrics bkr (metricType)	로그 파일	메트릭 메시지 (metrics topic) <sup>1</sup>
Num messages out	브로커가 마지막으로 시작된 이후 브로커에서 유출된 JMS 메시지 수	예 (ttl)	예	예 (...broker)
Message bytes out	브로커가 마지막으로 시작된 이후 브로커에서 유출된 JMS 메시지 바이트 수	예 (ttl)	예	예 (...broker)
Num packets out	브로커가 마지막으로 시작된 이후 브로커에서 유출된 패킷 수(JMS 메시지와 제어 메시지 모두 포함)	예 (ttl)	예	예 (...broker)
Packet bytes out	브로커가 마지막으로 시작된 이후 브로커에서 유출된 패킷 바이트 수(JMS 메시지와 제어 메시지 모두 포함)	예 (ttl)	예	예 (...broker)
Rate messages in	브로커로 JMS 메시지가 유입되는 현재 속도	예 (rts)	예	아니요
Rate message bytes in	브로커로 JMS 메시지 바이트가 유입되는 현재 속도	예 (rts)	예	아니요
Rate packets in	브로커로 패킷이 유입되는 현재 속도(JMS 메시지와 제어 메시지 모두 포함)	예 (rts)	예	아니요
Rate packet bytes in	브로커로 패킷 바이트가 유입되는 현재 속도(JMS 메시지와 제어 메시지 모두 포함)	예 (rts)	예	아니요
Rate messages out	브로커에서 JMS 메시지가 유출되는 현재 속도	예 (rts)	예	아니요
Rate message bytes out	브로커에서 JMS 메시지 바이트가 유출되는 현재 속도	예 (rts)	예	아니요
Rate packets out	브로커에서 패키지가 유출되는 현재 속도(JMS 메시지와 제어 메시지 모두 포함)	예 (rts)	예	아니요
Rate packet bytes out	브로커에서 패키지 바이트가 유출되는 현재 속도(JMS 메시지와 제어 메시지 모두 포함)	예 (rts)	예	아니요
<b>대상 데이터</b>				
Num destinations	브로커에 있는 실제 대상 수	아니요	아니요	예 (...broker)

1. 메트릭 주제 대상 이름에 대해서는 [216페이지의 표 10-7](#)을 참조하십시오.

# 연결 서비스 메트릭

표 18-3에서는 개별 연결 서비스에 대해 브로커가 보고하는 메트릭 데이터를 나열하고 설명합니다. 또한 메트릭 모니터링 도구에 따라 어떤 데이터를 얻을 수 있는지 보여 줍니다.

**표 18-3** 연결 서비스 메트릭

메트릭 개수	설명	imqcmd metrics svc (metricType)	로그 파일	메트릭 메시지 (metrics topic)
<b>연결 데이터</b>				
Num connections	현재 열려 있는 연결의 수	예 (cxn) 또는 query svc	아니요	아니요
Num threads	현재 사용 중인 스레드 수	예 (cxn) 또는 query svc	아니요	아니요
Min threads	이 스레드 수에 도달하면 스레드 풀에서 연결 서비스용으로 유지 관리되는 스레드 수(전체 연결 서비스에서 합산)	예 (cxn)	아니요	아니요
Max threads	이 스레드 수를 초과하면 스레드 풀에 연결 서비스용으로 새 스레드가 더 이상 추가되지 않는 스레드 수(전체 연결 서비스에서 합산)	예 (cxn)	아니요	아니요
<b>메시지 흐름 데이터</b>				
Num messages in	브로커가 마지막으로 시작된 이후 연결 서비스에 유입된 JMS 메시지의 수	예 (ttl)	아니요	아니요
Message bytes in	브로커가 마지막으로 시작된 이후 연결 서비스에 유입된 JMS 메시지 바이트 수	예 (ttl)	아니요	아니요
Num packets in	브로커가 마지막으로 시작된 이후 연결 서비스에 유입된 패킷 수(JMS 메시지와 제어 메시지 모두 포함)	예 (ttl)	아니요	아니요
Packet bytes in	브로커가 마지막으로 시작된 이후 연결 서비스에 유입된 패킷 바이트 수(JMS 메시지와 제어 메시지 모두 포함)	예 (ttl)	아니요	아니요
Num messages out	브로커가 마지막으로 시작된 이후 연결 서비스에서 유출된 JMS 메시지 수	예 (ttl)	아니요	아니요
Message bytes out	브로커가 마지막으로 시작된 이후 연결 서비스에서 유출된 JMS 메시지 바이트 수	예 (ttl)	아니요	아니요

표 18-3 연결 서비스 메트릭(계속)

메트릭 개수	설명	imqcmd metrics svc (metricType)	로그 파일	메트릭 메시지 (metrics topic)
Num packets out	브로커가 마지막으로 시작된 이후 연결 서비스에서 유출된 패킷 수(JMS 메시지와 제어 메시지 모두 포함)	예 (ttl)	아니요	아니요
Packet bytes out	브로커가 마지막으로 시작된 이후 연결 서비스에서 유출된 패킷 바이트 수(JMS 메시지와 제어 메시지 모두 포함)	예 (ttl)	아니요	아니요
Rate messages in	연결 서비스를 통해 JMS 메시지가 브로커로 유입되는 현재 속도	예 (rts)	아니요	아니요
Rate message bytes in	연결 서비스로 JMS 메시지 바이트가 유입되는 현재 속도	예 (rts)	아니요	아니요
Rate packets in	연결 서비스로 패킷이 유입되는 현재 속도(JMS 메시지와 제어 메시지 모두 포함)	예 (rts)	아니요	아니요
Rate packet bytes in	연결 서비스로 패킷 바이트가 유입되는 현재 속도(JMS 메시지와 제어 메시지 모두 포함)	예 (rts)	아니요	아니요
Rate messages out	연결 서비스에서 JMS 메시지가 유출되는 현재 속도	예 (rts)	아니요	아니요
Rate message bytes out	연결 서비스에서 JMS 메시지 바이트가 유출되는 현재 속도	예 (rts)	아니요	아니요
Rate packets out	연결 서비스에서 패킷이 유출되는 현재 속도(JMS 메시지와 제어 메시지 모두 포함)	예 (rts)	아니요	아니요
Rate packet bytes out	연결 서비스에서 패킷 바이트가 유출되는 현재 속도(JMS 메시지와 제어 메시지 모두 포함)	예 (rts)	아니요	아니요

# 대상 메트릭

표 18-4에서는 개별 대상에 대해 브로커가 보고하는 메트릭 데이터를 나열하고 설명합니다. 또한 메트릭 모니터링 도구에 따라 어떤 데이터를 얻을 수 있는지 보여 줍니다.

표 18-4 대상 메트릭

메트릭 개수	설명	imqcmd metrics dst (metricType)	로그 파일	메트릭 메시지 (metrics topic) <sup>1</sup>
<b>사용자 데이터</b>				
Num consumers	현재 사용자 수. 주제에 대해 이 값은 비영구 가입, 활성 영구 가입 및 비활성 영구 가입을 포함합니다. 대기열에 대해 이 값은 활성 사용자와 백업 사용자를 포함합니다.	예 (con)	아니요	예 (...destName)
Avg num consumers	브로커가 마지막으로 시작된 이후 평균 사용자 수	예 (con)	아니요	예 (...destName)
Peak num consumers	브로커가 마지막으로 시작된 이후 최대 사용자 수	예 (con)	아니요	예 (...destName)
Num active consumers	현재 활성 사용자의 수	예 (con)	아니요	예 (...destName)
Avg num active consumers	브로커가 마지막으로 시작된 이후 평균 활성 사용자 수	예 (con)	아니요	예 (...destName)
Peak num active consumers	브로커가 마지막으로 시작된 이후 최대 활성 사용자 수	예 (con)	아니요	예 (...destName)
Num backup consumers	현재 백업 사용자 수(대기열에만 적용)	예 (con)	아니요	예 (...destName)
Avg num backup consumers	브로커가 마지막으로 시작된 이후 평균 백업 사용자 수(대기열에만 적용)	예 (con)	아니요	예 (...destName)
Peak num backup consumers	브로커가 마지막으로 시작된 이후 최대 백업 사용자 수(대기열에만 적용)	예 (con)	아니요	예 (...destName)
<b>저장된 메시지 데이터</b>				
Num messages	대상 메모리와 영구 저장소에 현재 저장되어 있는 JMS 메시지 수	예 (con) (ttl) (rts) 또는 query dst	아니요	예 (...destName)
Avg num messages	브로커가 마지막으로 시작된 이후 대상 메모리와 영구 저장소에 저장되어 있는 평균 JMS 메시지 수	예 (con) (ttl) (rts)	아니요	예 (...destName)

**표 18-4** 대상 메트릭(계속)

메트릭 개수	설명	imqcmd metrics dst (metricType)	로그 파일	메트릭 메시지 (metrics topic) <sup>1</sup>
Peak num messages	브로커가 마지막으로 시작된 이후 대상 메모리와 영구 저장소에 저장되어 있는 최대 JMS 메시지 수	예 (con) (ttl) (rts)	아니요	예 (...destName)
Total message bytes	대상 메모리와 영구 저장소에 현재 저장되어 있는 JMS 메시지 바이트 수	예 (ttl) (rts) 또는 query dst	아니요	예 (...destName)
Avg total message bytes	브로커가 마지막으로 시작된 이후 대상 메모리와 영구 저장소에 저장되어 있는 평균 JMS 메시지 바이트 수	예 (ttl) (rts)	아니요	예 (...destName)
Peak total message bytes	브로커가 마지막으로 시작된 이후 대상 메모리와 영구 저장소에 저장되어 있는 최대 JMS 메시지 바이트 수	예 (ttl) (rts)	아니요	예 (...destName)
Peak message bytes	브로커가 마지막으로 시작된 이후 대상에서 수신한 단일 메시지에 있는 최대 JMS 메시지 바이트 수	예 (ttl) (rts)	아니요	예 (...destName)
<b>메시지 흐름 데이터</b>				
Num messages in	브로커가 마지막으로 시작된 이후 이 대상에 유입된 JMS 메시지 수	예 (ttl)	아니요	예 (...destName)
Msg bytes in	브로커가 마지막으로 시작된 이후 이 대상에 유입된 JMS 메시지 바이트 수	예 (ttl)	아니요	예 (...destName)
Num messages out	브로커가 마지막으로 시작된 이후 이 대상에서 유출된 JMS 메시지 수	예 (ttl)	아니요	예 (...destName)
Msg bytes out	브로커가 마지막으로 시작된 이후 이 대상에서 유출된 JMS 메시지 바이트 수	예 (ttl)	아니요	예 (...destName)
Rate num messages in	JMS 메시지가 대상에 유입되는 현재 속도	예 (rts)	아니요	아니요
Rate num messages out	JMS 메시지가 대상에서 유출되는 현재 속도	예 (rts)	아니요	아니요
Rate msg bytes in	JMS 메시지 바이트가 대상에 유입되는 현재 속도	예 (rts)	아니요	아니요
Rate Msg bytes out	JMS 메시지가 대상에서 유출되는 현재 속도	예 (rts)	아니요	아니요

**표 18-4** 대상 메트릭(계속)

메트릭 개수	설명	imqcmd metrics dst (metricType)	로그 파일	메트릭 메시지 (metrics topic) <sup>1</sup>
<b>디스크 사용률 데이터</b>				
Disk reserved	대상 파일 기반 저장소에서 모든 메시지 레코드(활성 및 사용 가능)가 사용하는 디스크 공간(바이트)	예 (disk)	아니요	예 (...destName)
Disk used	대상 파일 기반 저장소에서 활성 메시지 레코드가 사용하는 디스크 공간(바이트)	예 (disk)	아니요	예 (...destName)
Disk utilization ratio	예약된 디스크 공간에 대해 사용 중인 디스크 공간의 비율. 비율이 높을수록 더 많은 디스크 공간이 활성 메시지를 보관하는 데 사용되고 있는 것입니다.	예 (disk)	아니요	예 (...destName)

1. 메트릭 주제 대상 이름에 대해서는 [216페이지의 표 10-7](#)을 참조하십시오.

## 부록

- 부록 A, "운영 체제별 Message Queue 데이터의 위치"
- 부록 B, "Message Queue 인터페이스의 안정성"
- 부록 C, "HTTP/HTTPS 지원"





# 운영 체제별 Message Queue 데이터의 위치

Sun Java System 다음 절에서 알 수 있듯이 Message Queue 데이터는 운영 체제별로 다른 위치에 저장됩니다.

이 부록에서는 다음과 같은 운영 체제에서 다양한 유형의 Message Queue 데이터 위치를 설명합니다.

- [359페이지의 "Solaris"](#)
- [361페이지의 "Linux"](#)
- [362페이지의 "Windows"](#)

다음에 나오는 표에서 *instanceName*은 데이터와 관련된 브로커 인스턴스의 이름을 식별합니다.

## Solaris

[표 A-1](#)에서는 Solaris 운영 체제에서 Message Queue 데이터의 위치를 보여 줍니다.

Message Queue를 Solaris에서 표준 버전 Sun Java System Application Server와 함께 사용하는 경우 디렉토리 구조는 [362페이지의 "Windows"](#)에서 설명한 구조와 같습니다.

**표 A-1** Solaris에서 Message Queue 데이터의 위치

데이터 범주	Solaris에서 위치
브로커 인스턴스 구성 등록 정보	<code>/var/imq/instances/instanceName/props/config.properties</code>
브로커 구성 파일 템플릿	<code>/usr/share/lib/imq/props/broker/</code>
영구 저장소(메시지, 대상, 영구 가입, 트랜잭션)	<code>/var/imq/instances/instanceName/fs350/</code> 또는 JDBC 액세스 가능 데이터 저장소
브로커 인스턴스 로그 파일 디렉토리(기본 위치)	<code>/var/imq/instances/instanceName/log/</code>
관리 대상 객체 (객체 저장소)	선택한 로컬 디렉토리 또는 LDAP 서버
보안: 사용자 저장소	<code>/var/imq/instances/instanceName/etc/passwd</code> 또는 LDAP 서버
보안: 액세스 제어 파일(기본 위치)	<code>/var/imq/instances/instanceName/etc/accesscontrol.properties</code>
보안: passfile 디렉토리 (기본 위치)	<code>/var/imq/instances/instanceName/etc/</code>
보안: passfile 예	<code>/etc/imq/passfile.sample</code>
보안: 브로커 키 저장소 파일 위치	<code>/etc/imq/</code>
JavaDoc API 설명서	<code>/usr/share/javadoc/imq/index.html</code>
응용 프로그램 및 구성 예	<code>/usr/demo/imq/</code>
Java 아카이브(.jar), 웹 아카이브(.war) 및 자원 어댑터 아카이브(.rar) 파일	<code>/usr/share/lib/</code>

# Linux

표 A-2에서는 Linux 운영 체제에서 Message Queue 데이터의 위치를 보여줍니다.

**표 A-2** Linux에서 Message Queue 데이터의 위치

데이터 범주	Windows에서 위치
브로커 인스턴스 구성 등록 정보	<code>/var/opt/sun/mq/instances/instanceName/props/config.properties</code>
브로커 구성 파일 템플릿	<code>/opt/sun/mq/private/share/lib/props/</code>
영구 저장소(메시지, 대상, 영구 가입, 트랜잭션)	<code>/var/opt/sun/mq/instances/instanceName/fs350/</code> 또는 JDBC 액세스 가능 데이터 저장소
브로커 인스턴스 로그 파일 디렉토리(기본 위치)	<code>/var/opt/sun/mq/instances/instanceName/log/</code>
관리 대상 객체 (객체 저장소)	선택한 로컬 디렉토리 또는 LDAP 서버
보안: 사용자 저장소	<code>/var/opt/sun/mq/instances/instanceName/etc/passwd</code> 또는 LDAP 서버
보안: 액세스 제어 파일 (기본 위치)	<code>/var/opt/sun/mq/instances/instanceName/etc/accesscontrol.properties</code>
보안: passfile 디렉토리 (기본 위치)	<code>/var/opt/sun/mq/instances/instanceName/etc/</code>
보안: passfile 예	<code>/etc/opt/sun/mq/passfile.sample</code>
보안: 브로커 키 저장소 파일 위치	<code>/etc/opt/sun/mq/</code>
JavaDoc API 설명서	<code>/opt/sun/mq/javadoc/index.html</code>
응용 프로그램 및 구성 예	<code>/opt/sun/mq/examples/</code>
Java 아카이브(.jar), 웹 아카이브(.war) 및 자원 어댑터 아카이브(.rar) 파일	<code>/opt/sun/mq/share/lib/</code>
공유 라이브러리(.so) 파일	<code>/opt/sun/mq/lib/</code>

# Windows

표 A-3에서는 Windows 운영 체제에서 Message Queue 데이터의 위치를 보여 줍니다.

또한 Message Queue가 독립 실행형 버전 Sun Java System Application Server와 함께 제공될 때 Solaris에서 Message Queue 데이터의 위치를 보여 줍니다. 독립 실행형 버전 Application Server는 Solaris나 Sun Java Enterprise System에는 제공되지 않습니다. 표 A-3의 경로 이름을 사용하십시오. 이 때 슬래시 방향을 Windows 백슬래시(\)에서 Solaris 슬래시(/)로 변경하지 마십시오. 자세한 내용은 25페이지의 표 3의 IMQ\_HOME 및 IMQ\_VARHOME에 대한 정의를 참조하십시오.

**표 A-3** Windows에서 Message Queue 데이터의 위치

데이터 범주	Windows에서 위치
브로커 인스턴스 구성 등록 정보	IMQ_VARHOME\instances\ <i>instanceName</i> \props\ config.properties
브로커 구성 파일 템플릿	IMQ_HOME\lib\props\broker\
영구 저장소(메시지, 대상, 영구 가입, 트랜잭션)	IMQ_VARHOME\instances\ <i>instanceName</i> \fs350\ 또는 JDBC 액세스 가능 데이터 저장소
브로커 인스턴스 로그 파일 디렉토리(기본 위치)	IMQ_VARHOME\instances\ <i>instanceName</i> \log\
관리 대상 객체 (객체 저장소)	선택한 로컬 디렉토리 또는 LDAP 서버
보안: 사용자 저장소	IMQ_VARHOME\instances\ <i>instanceName</i> \etc\ passwd 또는 LDAP 서버
보안: 액세스 제어 파일(기본)	IMQ_VARHOME\instances\ <i>instanceName</i> \ etc\accesscontrol.properties
보안: passfile 디렉토리 (기본 위치)	IMQ_HOME\etc\
보안: passfile 예	IMQ_HOME\etc\passfile.sample
보안: 브로커 키 저장소 파일 위치	IMQ_HOME\etc\
JavaDoc API 설명서	IMQ_HOME\javadoc\index.html

**표 A-3** Windows에서 Message Queue 데이터의 위치(계속)

데이터 범주	Windows에서 위치
응용 프로그램 및 구성 예	IMQ_HOME\demo\
Java 아카이브(.jar), 웹 아카이브(.war) 및 자원 어댑터 아카이브(.rar) 파일	IMQ_HOME\lib\

Windows

# Message Queue 인터페이스의 안정성

Sun Java System Message Queue에서는 관리자가 작업을 자동화할 때 도움이 되는 여러 인터페이스를 사용합니다. 이 부록에서는 안정성을 기준으로 이러한 인터페이스를 분류합니다. 안정적인 인터페이스일수록 향후 제품 버전에서 변경될 확률이 낮습니다.

이 부록에 나열되지 않은 인터페이스는 고객용이 아니라 개인 인터페이스입니다.

표 B-1에서는 안정성 분류 체계에 대해 설명합니다.

**표 B-1** 인터페이스 안정성 분류 체계

분류	설명
개인	고객이 직접 사용하지 않음. 임의의 릴리스에서 변경 또는 삭제될 수 있습니다.
변화	고객용. 주 릴리스(예: 3.0, 4.0) 또는 부 릴리스(예: 3.1, 3.2)에 호환되지 않는 변경이 있을 수 있습니다. 신중하고 천천히 변경됩니다. 모든 변경 사항이 호환성을 갖도록 타당한 조치가 이행되지만 보장되지는 않습니다.
안정	고객용. 주 릴리스(예: 3.0, 4.0)에 한해 호환되지 않는 변경이 있을 수 있습니다.
표준	고객용. 이 인터페이스는 공식 표준에서 정의하며 표준 기관이 관리합니다. 이 인터페이스와 호환되지 않는 변경은 거의 없습니다.
불안정	고객용. 주 릴리스(예: 3.0, 4.0) 또는 부 릴리스(예: 3.1, 3.2)에 호환되지 않는 변경이 있을 수 있습니다. 이 인터페이스는 향후 릴리스에서 사실상 그리고 호환되지 않는 방식으로 제거 또는 변경될 수 있음을 고객에게 알립니다. 고객은 불안정한 인터페이스에 대한 명시적인 종속성을 생성하지 않는 것이 바람직합니다.

표 B-2에 인터페이스와 해당 분류가 나열되어 있습니다.

**표 B-2** Message Queue 인터페이스의 안정성

인터페이스	분류
<b>명령줄 인터페이스</b>	
imqbrokerd 명령줄 인터페이스	변화
imqadmin 명령줄 인터페이스	불안정
imqcmd 명령줄 인터페이스	변화
imqdbmgr 명령줄 인터페이스	불안정
imqkeytool 명령줄 인터페이스	변화
imqobjmgr 명령줄 인터페이스	변화
imqusermgr 명령줄 인터페이스	불안정
imqbrokerd, imqadmin, imqcmd, imqdbmgr, imqkeytool, imqobjmgr, imqusermgr의 출력	불안정
<b>명령</b>	
imqobjmgr 명령 파일	변화
imqbrokerd 명령	안정
imqadmin 명령	불안정
imqcmd 명령	안정
imqdbmgr 명령	불안정
imqkeytool 명령	안정
imqobjmgr 명령	안정
imqusermgr 명령	불안정
<b>API</b>	
JMS API (javax.jms)	표준
JAXM API (javax.xml)	표준
C-API	변화
C-API 환경 변수	불안정
메시지 기반 모니터링 API	변화
관리 대상 객체 API (com.sun.messaging)	변화
<b>JAR 파일 및 WAR 파일</b>	
imq.jar 위치 및 이름	안정
jms.jar 위치 및 이름	변화



**표 B-2** Message Queue 인터페이스의 안정성(계속)

인터페이스	분류
imqbroker.jar 위치 및 이름	개인
imqutil.jar 위치 및 이름	개인
imqadmin.jar 위치 및 이름	개인
imqservlet.jar 위치 및 이름	변화
imqhttp.war 위치 및 이름	변화
imqhttps.war 위치 및 이름	변화
imqjmsra.rar 위치 및 이름	변화
imqxm.jar 위치 및 이름	변화
jaxm-api.jar 위치 및 이름	변화
saaj-api.jar 위치 및 이름	변화
saaj-impl.jar 위치 및 이름	변화
activation.jar 위치 및 이름	변화
mail.jar 위치 및 이름	변화
dom4j.jar 위치 및 이름	개인
fscontext.jar 위치 및 이름	불안정
<b>파일</b>	
브로커 로그 파일 위치 및 내용 형식	불안정
비밀번호 파일	불안정
accesscontrol.properties 파일	불안정
<b>시스템 대상</b>	
mq.sys.dmq 대상	안정
mq.metrics.* 대상	변화
<b>구성 등록 정보</b>	
Message Queue JMS 자원 어댑터 구성 등록 정보	변화
Message Queue JMS 자원 어댑터 JavaBean 및 ActivationSpec 구성 등록 정보	변화

**표 B-2** Message Queue 인터페이스의 안정성(계속)

인터페이스	분류
<b>메시지 등록 정보 및 형식</b>	
사용 불능 메시지 대기열 메시지 등록 정보, JMSXDeliveryCount	표준
사용 불능 메시지 대기열 메시지 등록 정보, JMS_SUN_*	변화
Message Queue 클라이언트 메시지 등록 정보: JMS_SUN_*	변화
메트릭 또는 모니터링 메시지의 JMS 메시지 형식	변화
<b>기타</b>	
Message Queue JMS 자원 어댑터 패키지, com.sun.messaging.jms.ra	변화
지속성 메시지 저장소의 JDBC 스키마	변화

# HTTP/HTTPS 지원

Message Queue 엔터프라이즈판은 직접 TCP 연결이 아닌 HTTP 또는 보안 HTTP (HTTPS) 전송을 통한 Java 클라이언트와 브로커 간의 통신을 지원합니다. HTTP/HTTPS 지원은 C 클라이언트에는 사용할 수 없습니다.

이 부록에서는 이 지원을 활성화하는 데 사용되는 구조를 설명하고 클라이언트에서 Message Queue 메시징에 HTTP 기반 연결을 사용할 수 있도록 하는 데 필요한 설정 작업에 대해 설명합니다. 이 부록은 다음 내용으로 구성되어 있습니다.

- [370페이지의 "HTTP/HTTPS 지원 구조"](#)
- [371페이지의 "HTTP 지원 활성화"](#)
- [382페이지의 "HTTPS 지원 활성화"](#)
- [396페이지의 "문제 해결"](#)

# HTTP/HTTPS 지원 구조

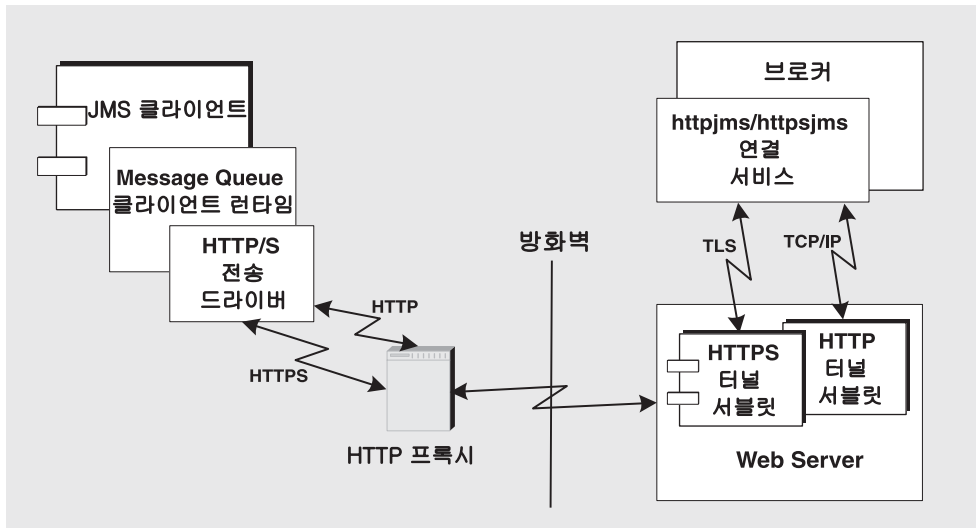
Message Queue 메시징은 HTTP/HTTPS 연결 위에서 실행할 수 있습니다.

HTTP/HTTPS 연결은 보통 방화벽을 통해서도 허용되기 때문에 이렇게 하면 방화벽으로 클라이언트 응용 프로그램과 브로커를 분리할 수 있습니다.

370페이지의 그림 C-1에서는 HTTP/HTTPS 지원과 관련된 주요 구성 요소를 보여줍니다.

- 클라이언트쪽에서는 HTTP 또는 HTTPS 전송 드라이버가 Message Queue 메시지를 HTTP 요청으로 캡슐화하고 요청이 정확한 순서로 웹 서버에 전송되는지 확인합니다.
- 클라이언트는 필요한 경우 HTTP 프록시 서버를 사용해서 브로커와 통신할 수 있습니다. 프록시의 주소는 클라이언트를 시작할 때 명령줄 옵션을 사용해서 지정합니다. 자세한 내용은 376페이지의 "HTTP 프록시 사용"을 참조하십시오.
- HTTP 또는 HTTPS 터널 서블릿(모두 Message Queue와 함께 제공)이 웹 서버에 로드되고 JMS 메시지를 브로커로 전달하기 전에 클라이언트 HTTP 요청으로부터 추출하는 데 사용됩니다. HTTP/HTTPS 터널 서블릿은 클라이언트에서 이루어진 HTTP 요청에 대한 응답으로 브로커 메시지를 클라이언트에 다시 보내는 작업도 수행합니다. 한 HTTP/HTTPS 터널 서블릿을 사용해서 여러 브로커에 액세스할 수 있습니다.

그림 C-1 HTTP/HTTPS 지원 구조



- 브로커쪽에서는 `httpjms` 또는 `httpsjms` 연결 서비스가 해당 터널 서블릿에서 들어오는 메시지를 분해하고 다중화를 해제합니다.
- 웹 서버가 오류로 다시 시작되면 모든 연결이 복원되며 클라이언트에는 아무 영향도 주지 않습니다. 브로커가 오류로 다시 시작되면 예외가 발생하며 클라이언트는 연결을 다시 설정해야 합니다. 드물기는 하지만, 웹 서버와 브로커에 모두 오류가 발생하고 브로커가 다시 시작되지 않은 경우에는 웹 서버가 클라이언트 연결을 복원한 후 클라이언트에 알리지 않고 브로커 연결을 계속 기다릴 수 있습니다. 이런 상황을 방지하려면 항상 브로커를 다시 시작해야 합니다.

그림 C-1에서 알 수 있듯이, HTTP와 HTTPS가 지원하는 구조는 서로 매우 비슷합니다. 가장 큰 차이는 HTTPS(`httpsjms` 연결 서비스)의 경우 터널 서블릿이 클라이언트 응용 프로그램과 브로커 모두에 대해 보안 연결을 갖는다는 점입니다.

브로커에 대한 보안 연결은 Message Queue의 HTTPS 터널 서블릿에 해당하는 SSL 사용 터널 서블릿을 통해 제공되며 연결을 요청하는 모든 브로커에 자체 서명된 인증서를 전달합니다. 인증서는 브로커가 HTTPS 터널 서블릿에 대해 암호화된 연결을 설정할 때 사용됩니다. 이 연결이 설정되고 나면 클라이언트 응용 프로그램과 웹 서버에서 클라이언트 응용 프로그램과 터널 서블릿 사이의 보안 연결을 처리할 수 있습니다.

## HTTP 지원 활성화

다음 절에서는 HTTP 지원을 활성화하기 위해 수행하는 단계를 설명합니다.

### ▶ HTTP 지원을 활성화하는 방법

1. HTTP 터널 서블릿을 웹 서버에 배포합니다.
2. 브로커의 `httpjms` 연결 서비스를 구성하고 브로커를 시작합니다.
3. HTTP 연결을 구성합니다.

## 1단계. 웹 서버에 HTTP 터널 서블릿 배포

웹 서버에 HTTP 터널 서블릿을 배포하는 일반적인 방법에는 두 가지가 있습니다.

- jar 파일로 배포 — Servlet 2.1 이전 버전을 지원하는 웹 서버용
- 웹 아카이브(WAR) 파일로 배포 — Servlet 2.2 이상 버전을 지원하는 웹 서버용

### Jar 파일로 배포

Message Queue 터널 서블릿 배포는 호스트 웹 서버에 액세스할 수 있는 적절한 jar 파일을 생성하고 웹 서버 시작 시 서블릿을 로드하도록 구성하며 서블릿 URL의 컨텍스트 루트 부분을 지정하는 작업으로 구성됩니다.

터널 서블릿 jar 파일(`imgServlet.jar`)은 HTTP 터널 서블릿에서 필요한 모든 클래스를 포함하며, 운영 체제에 따라 다른 디렉토리에 있습니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

서블릿 2.x를 지원하는 모든 웹 서버를 사용하여 이 서블릿을 로드할 수 있습니다. 서블릿 클래스 이름은 다음과 같습니다.

```
com.sun.messaging.jmq.transport.  
httpunnel.servlet.HttpTunnelServlet
```

웹 서버에서 `imgServlet.jar` 파일을 볼 수 있어야 합니다. 웹 서버와 브로커를 서로 다른 호스트에서 실행하려면 `imgServlet.jar` 파일의 복사본을 웹 서버에서 액세스할 수 있는 위치에 저장해야 합니다.

또한 웹 서버 시작 시 이 서블릿을 로드하도록 구성해야 하며, 서블릿 URL의 컨텍스트 루트 부분을 지정해야 하는 경우도 있습니다(376페이지의 "예 1: Sun Java System Web Server에 HTTP 터널 서블릿 배포" 참조).

또한 성능 향상을 위해 웹 서버의 액세스 로깅 기능은 사용하지 않는 것이 좋습니다.

### 웹 아카이브 파일로 배포

HTTP 터널 서블릿을 WAR 파일로 배포하는 작업은 웹 서버에서 제공하는 메커니즘을 사용하여 수행합니다. HTTP 터널 서블릿 WAR 파일(`imghttp.war`)은 `.jar`, `.war` 및 `.rar` 파일이 포함된 디렉토리에 있으며, 이 디렉토리는 운영 체제에 따라 다릅니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

WAR 파일에는 웹 서버에서 서블릿을 로드하여 실행하는 데 필요한 기본 구성 정보가 들어 있는 배포 설명자가 포함되어 있습니다. 웹 서버에 따라 서블릿 URL의 컨텍스트 루트 부분을 지정해야 하는 경우도 있습니다(380페이지의 "예 2: Sun Java System Application Server 7.0에 HTTP 터널 서블릿 배포" 참조).

## 2단계. httpjms 연결 서비스 구성

기본적으로 브로커에 대해 HTTP 지원이 활성화되어 있지 않으므로 httpjms 연결 서비스가 활성화되도록 브로커를 재구성해야 합니다. 재구성한 경우 67페이지의 "대화식으로 브로커 시작"에 설명된 대로 브로커를 시작할 수 있습니다.

### ▶ httpjms 연결 서비스를 활성화하는 방법

1. 브로커의 인스턴스 구성 파일을 엽니다.

인스턴스 구성 파일은 구성 파일이 연결되어 있는 브로커 인스턴스의 이름 (*instanceName*)으로 식별되는 디렉토리에 저장됩니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

```
.../instances/instanceName/props/config.properties
```

2. `imq.service.activelist` 등록 정보에 httpjms 값을 추가합니다.

```
imq.service.activelist=jms,admin,httpjms
```

브로커는 시작할 때 호스트 시스템에서 실행 중인 웹 서버와 HTTP 터널 서블릿을 찾습니다. 그러나 원격 터널 서블릿에 액세스하기 위해 `servletHost` 및 `servletPort` 연결 서비스 등록 정보를 재구성할 수 있습니다.

또한 성능 향상을 위해 `pullPeriod` 등록 정보를 재구성할 수도 있습니다. httpjms 연결 서비스 구성 등록 정보에 대한 자세한 설명은 373페이지의 표 C-1을 참조하십시오.

**표 C-1** httpjms 연결 서비스 등록 정보

등록 정보 이름	설명
<code>imq.httpjms.http.servletHost</code>	필요한 경우 이 값을 변경하여 HTTP 터널 서블릿을 실행하는 호스트의 이름(호스트 이름 또는 IP 주소)을 지정합니다(원격 호스트이거나 로컬 호스트의 특정 호스트 이름일 수 있음). 기본값: localhost
<code>imq.httpjms.http.servletPort</code>	이 값을 변경하여 브로커가 HTTP 터널 서블릿에 액세스하는 데 사용하는 포트 번호를 지정합니다(웹 서버에서 기본 포트를 변경한 경우 이 등록 정보를 적절히 변경해야 함). 기본값: 7675

**표 C-1** httpjms 연결 서비스 등록 정보(계속)

등록 정보 이름	설명
img.httpjms.http.pullPeriod	클라이언트 런타임이 브로커에서 메시지를 가져오기 위해 만든 HTTP 요청 사이의 간격(초)을 지정합니다(이 등록 정보는 브로커에서 설정되고 클라이언트 런타임에 전파됨). 값이 0 또는 음수인 경우 클라이언트는 하나의 HTTP 요청을 항상 보류 상태로 두고 가능한 빨리 메시지를 가져오도록 준비합니다. 클라이언트 수가 많은 경우 웹 서버 자원을 많이 사용하여 서버가 응답하지 않을 수 있습니다. 그런 경우 pullPeriod 등록 정보를 양수(초)로 설정해야 합니다. 이 등록 정보는 후속 가져오기 요청을 만들기 전에 클라이언트의 HTTP 전송 드라이버가 대기하는 시간을 설정합니다. 값을 양수로 설정하면 클라이언트가 응답 시간 동안 대기하는 대신 웹 서버 자원이 절약됩니다. 기본값: -1
img.httpjms.http.connectionTimeout	클라이언트 런타임이 HTTP 터널 서블릿의 응답을 기다리는 시간(초)을 지정합니다. 이 시간이 초과되면 예외가 발생합니다(이 등록 정보는 브로커에서 설정되고 클라이언트 런타임에 전파됨). 이 등록 정보는 브로커가 HTTP 터널 서블릿과 통신한 후 연결을 해제할 때까지 기다리는 시간도 지정합니다. 이 경우에는 브로커와 터널 서블릿이 HTTP 서블릿에 액세스 중인 클라이언트가 비정상적으로 종료했는지 여부를 알 수 없으므로 시간 초과가 필요합니다. 기본값: 60

### 3단계. HTTP 연결 구성

클라이언트 응용 프로그램에서는 제대로 구성된 연결 팩토리 관리 대상 객체를 사용해서 브로커에 대한 HTTP 연결을 설정해야 합니다. 이 절에서는 HTTP 연결 구성 문제에 대해 논의합니다.

#### 연결 팩토리 구성

HTTP 지원을 활성화하려면 연결 팩토리의 `imgAddressList` 속성을 HTTP 터널 서블릿 URL로 설정해야 합니다. HTTP 터널 서블릿 URL의 일반 구문은 다음과 같습니다.

```
http://hostName:port/contextRoot/tunnel
```

여기서 `hostName:port`는 HTTP 터널 서블릿을 호스트하는 웹 서버의 이름과 포트이며, `contextRoot`는 웹 서버에서 터널 서블릿을 배포할 때 설정된 경로입니다.

일반적인 연결 팩토리 속성, 특히 `imgAddressList` 속성에 대한 자세한 내용은 *Message Queue Developer's Guide for Java Clients*를 참조하십시오.



다음 방법 중 하나를 사용하여 연결 팩토리 속성을 설정할 수 있습니다.

- 연결 팩토리 관리 대상 객체를 만드는 `imqobjmgr` 명령에 `-o` 옵션을 사용하거나(189 페이지의 "연결 팩토리 추가" 참조) 관리 콘솔(`imqadmin`)을 사용하여 연결 팩토리 관리 대상 객체를 만들 때 속성을 설정합니다.
- 클라이언트를 시작하는 명령에 `-D` 옵션을 사용합니다(*Message Queue Developer's Guide for Java Clients* 참조).
- 클라이언트 코드에서 프로그래밍 방식으로 연결 팩토리를 만든 후 API 호출을 사용하여 해당 연결 팩토리의 속성을 설정합니다(*Message Queue Developer's Guide for Java Clients* 참조).

### 단일 서블릿을 사용하여 다중 브로커에 액세스

다중 브로커를 실행할 경우 다중 웹 서버 및 서블릿 인스턴스를 구성하지 않아도 됩니다. 동시에 실행 중인 브로커 간에 단일 웹 서버와 HTTP 터널 서블릿을 공유할 수 있습니다. 다중 브로커 인스턴스가 단일 터널 서블릿을 공유하는 경우 `imqAddressList` 연결 팩토리 속성을 다음과 같이 구성해야 합니다.

```
http://hostName:port/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

여기서 `bkrHostName`은 브로커 인스턴스 호스트 이름이고 `instanceName`은 클라이언트가 액세스할 특정 브로커 인스턴스의 이름입니다.

`bkrHostName`과 `instanceName`에 정확한 문자열을 입력했는지 확인하려면 브라우저에서 서블릿 URL에 액세스하여 HTTP 터널 서블릿의 상태 보고서를 생성합니다. 보고서에는 서블릿이 액세스하는 모든 브로커가 나열됩니다.

```
HTTP tunnel servlet ready.
Servlet Start Time : Thu May 30 01:08:18 PDT 2002
Accepting TCP connections from brokers on port : 7675
Total available brokers = 2
Broker List :
  jpgserv:broker2
  cochin:broker1
```

## HTTP 프록시 사용

HTTP 프록시를 사용해서 HTTP 터널 서블릿에 액세스하는 경우:

- `http.proxyHost` 시스템 등록 정보를 프록시 서버 호스트 이름으로 설정합니다.
- `http.proxyPort` 시스템 등록 정보를 프록시 서버 포트 번호로 설정합니다.

클라이언트 응용 프로그램을 시작하는 명령에 `-D` 옵션을 사용하여 이러한 등록 정보를 설정할 수 있습니다.

## 예 1: Sun Java System Web Server에 HTTP 터널 서블릿 배포

이 절에서는 Sun Java System Web Server에서 HTTP 터널 서블릿을 jar 파일과 WAR 파일로 배포하는 방법을 모두 설명합니다. 사용 방법은 Sun Java System Web Server 버전에 따라 다르지만, Servlet 2.2 이상을 지원하지 않는다면 WAR 파일 배포를 처리할 수 없습니다.

### Jar 파일로 배포

다음은 브라우저 기반 관리 GUI를 사용하여 Sun Java System Web Server 6.1에 배포하는 경우에 대한 지침입니다. 이 절차는 다음과 같은 일반적인 단계로 구성됩니다.

1. 서블릿 추가
2. 서블릿 가상 경로 구성
3. 서블릿 로드
4. 서블릿 액세스 로그 비활성화

이러한 단계는 다음 하위 절에 설명되어 있습니다. 웹 브라우저를 사용하여 서블릿 URL에 액세스함으로써 HTTP 터널 서블릿 배포 성공 여부를 확인할 수 있습니다. 상태를 표시해야 합니다.

### 서블릿 추가

#### ▶ 터널 서블릿을 추가하는 방법

1. 서블릿 탭을 선택합니다.
2. 서블릿 속성 구성을 선택합니다.
3. 서블릿 이름 필드에 터널 서블릿의 이름을 지정합니다.

4. 서블릿 코드(클래스 이름) 필드를 다음 값으로 설정합니다.  
`com.sun.messaging.jmq.transport.httptunnel.servlet.HttpTunnelServlet`
5. 서블릿 클래스 경로 필드에 `imqservlet.jar`에 대한 전체 경로를 입력합니다. 예를 들면 다음과 같습니다.  
`/usr/share/lib/imq/imqservlet.jar(Solaris)`  
`/opt/sun/mq/share/lib/imqservlet.jar(Linux)`  
`IMQ_HOME/lib/imqservlet.jar(Windows)`
6. 서블릿 인수 필드에 표 C-2와 같은 선택 인수를 입력합니다.

**표 C-2** HTTP 터널 서블릿 Jar 파일 배포에 사용되는 서블릿 인수

인수	기본값	참조
<code>servletHost</code>	모든 호스트	373페이지의 표 C-1 참조
<code>servletPort</code>	7675	373페이지의 표 C-1 참조

두 인수를 모두 사용하는 경우에는 인수 사이를 쉼표로 구분합니다.

`servletPort=portNumber, servletHost=...`

`servletHost` 및 `servletPort` 인수는 웹 서버와 브로커간 통신에만 적용되며 기본값에 문제가 있는 경우에만 설정됩니다. 그런 경우에도 브로커 구성 등록 정보를 적절하게 설정해야 합니다(373페이지의 표 C-1 참조). 예를 들면 다음과 같습니다.

`imq.httpjms.http.servletPort`

### 서블릿 가상 경로(서블릿 URL) 구성

▶ 터널 서블릿에 대한 가상 경로(서블릿 URL)를 구성하는 방법

1. 서블릿 탭을 선택합니다.
2. 서블릿 가상 경로 변환 구성을 선택합니다.
3. 가상 경로 필드를 설정합니다.

가상 경로는 터널 서블릿 URL의 `/contextRoot/tunnel` 부분입니다.

`http://hostName:port/contextRoot/tunnel`

예를 들어, `contextRoot`를 `imq`로 설정한 경우 가상 경로 필드는 다음과 같습니다.

`/imq/tunnel`

4. 서블릿 이름 필드를 376페이지의 "서블릿 추가"의 단계 3과 동일한 값으로 설정합니다.

### 서블릿 로드

▶ 웹 서버 시작 시 터널 서블릿을 로드하는 방법

1. 서블릿 탭을 선택합니다.
2. 전역 속성 구성을 선택합니다.
3. 시작 서블릿 필드에 376페이지의 "서블릿 추가"의 단계 3과 동일한 서블릿 이름 값을 입력합니다.

### 서버 액세스 로그 비활성화

서버 액세스 로그를 비활성화할 필요는 없지만 그렇게 하면 성능이 향상됩니다.

▶ 서버 액세스 로그를 비활성화하는 방법

1. 상태 탭을 선택합니다.
2. 로그 기본 설정 페이지를 선택합니다.
3. 로그 클라이언트 액세스 제어를 사용하여 로깅을 비활성화합니다.

### WAR 파일로 배포

다음은 Sun Java System Web Server 6.0 서비스 팩 2에서의 배포 지침입니다. 웹 브라우저에서 서블릿 URL에 액세스하여 성공적인 HTTP 터널 서블릿 배포를 확인할 수 있습니다. 상태 정보를 표시해야 합니다.

▶ http 터널 서블릿을 WAR 파일로 배포하는 방법

1. 브라우저 기반 관리 GUI에서 가상 서버 클래스 탭을 선택한 후 클래스 관리를 선택합니다.
2. 해당 가상 서버 클래스 이름(예: defaultClass)을 선택하고 관리 버튼을 누릅니다.
3. 가상 서버 관리를 선택합니다.
4. 해당 가상 서버 이름을 선택하고 관리 버튼을 누릅니다.
5. 웹 응용 프로그램 탭을 선택합니다.
6. 웹 응용 프로그램 배포를 누릅니다.

7. WAR 파일 위치 및 WAR 파일 경로 필드에서 `imghttp.war` 파일을 가리키는 적절한 값을 선택합니다. 이 파일이 있는 디렉토리는 운영 체제에 따라 다릅니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

8. 응용 프로그램 URI 필드에 경로를 입력합니다.

응용 프로그램 URI 필드 값은 터널 서블릿 URL의 `/contextRoot` 부분입니다.

```
http://hostName:port/contextRoot/tunnel
```

예를 들어, `contextRoot`를 `img`로 설정한 경우 응용 프로그램 URI 필드는 다음과 같습니다.

```
/img
```

9. 서블릿을 배포할 설치 디렉토리 경로(일반적으로 Sun Java System Web Server 설치 루트 아래)를 입력합니다.

10. 확인을 누릅니다.

11. 웹 서버 인스턴스를 다시 시작합니다.

이제 서블릿을 다음 주소에서 사용할 수 있습니다.

```
http://hostName:port/contextRoot/tunnel
```

이제 클라이언트에서 이 URL을 통해 HTTP 연결을 사용하는 메시지 서비스에 연결할 수 있습니다.

## 예 2: Sun Java System Application Server 7.0에 HTTP 터널 서블릿 배포

이 절에서는 Sun Java System Application Server 7.0에서 HTTP 터널 서블릿을 WAR 파일로 배포하는 방법을 설명합니다.

다음 두 단계를 거쳐야 합니다.

- Application Server 7.0 배포 도구를 사용하여 HTTP 터널 서블릿 배포
- 응용 프로그램 서버 인스턴스의 `server.policy` 파일 수정

### 배포 도구 사용

#### ▶ Application Server 7.0 환경에 HTTP 터널 서블릿을 배포하는 방법

1. 웹 기반 관리 GUI에서 다음을 선택합니다.

Application Server > 인스턴스 > server1 > 응용 프로그램 > 웹 응용 프로그램

2. 배포 버튼을 누릅니다.

3. 파일 경로: 텍스트 필드에 HTTP 터널 서블릿 WAR 파일(`imqhttp.war`)의 위치를 입력합니다.

`imqhttp.war` 파일의 위치는 운영 체제에 따라 다릅니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

4. 확인을 누릅니다.

5. 다음 화면에서 컨텍스트 루트 텍스트 필드의 값을 설정합니다.

컨텍스트 루트 필드 값은 터널 서블릿 URL의 `/contextRoot` 부분입니다.

```
http://hostName:port/contextRoot/tunnel
```

예를 들어, 컨텍스트 루트 필드를 `/imq`로 설정할 수 있습니다.

6. 확인을 누릅니다.

다음 화면에 터널 서블릿이 성공적으로 배포되었고 기본적으로 사용되며 다음 위치(이 예의 경우)에 있다고 표시됩니다.

```
/var/opt/SUNWappserver7/domains/domain1/server1/applications/
j2ee-modules/imqhttp_1
```

이제 서블릿을 다음 주소에서 사용할 수 있습니다.

```
http://hostName:port/contextRoot/tunnel
```

이제 클라이언트에서 이 URL을 통해 HTTP 연결을 사용하는 메시지 서비스에 연결할 수 있습니다.

## server.policy 파일 수정

Application Server 7.0이 시행하는 일련의 기본 보안 정책은 수정하지 않으면 HTTP 터널 서블릿이 Message Queue 브로커에서 연결을 수신할 수 없도록 합니다.

응용 프로그램 서버 인스턴스마다 해당 보안 정책이나 규칙이 포함된 파일이 있습니다. 예를 들어, Solaris의 server1 인스턴스의 경우 이 파일의 위치는 다음과 같습니다.

```
/var/opt/SUNWappserver7/domains/domain1/server1/config/
server.policy
```

터널 서블릿이 Message Queue 브로커에서 연결을 수신하도록 하려면 이 파일에 항목을 추가해야 합니다.

### ▶ 응용 프로그램 서버의 server.policy 파일을 수정하는 방법

1. server.policy 파일을 엽니다.
2. 다음 항목을 추가합니다.

```
grant codeBase
"file:/var/opt/SUNWappserver7/domains/domain1/server1/
  applications/j2ee-modules/imqhttp_1/-"
{
  permission java.net.SocketPermission "*",
    "connect,accept,resolve";
};
```

## HTTPS 지원 활성화

다음 절에서는 HTTPS 지원을 활성화하기 위해 수행하는 단계에 대해 설명합니다. [371페이지의 "HTTP 지원 활성화"](#)와 비슷한 내용에 SSL 인증서 생성 및 액세스에 필요한 단계가 추가되어 있습니다.

### ▶ HTTPS 지원을 활성화하는 방법

1. HTTPS 터널 서블릿에 대해 자체 서명된 인증서를 생성합니다.
2. HTTPS 터널 서블릿을 웹 서버에 배포합니다.
3. 브로커의 `httpsjms` 연결 서비스를 구성하고 브로커를 시작합니다.
4. HTTPS 연결을 구성합니다.

각 단계에 대해서는 뒤에서 자세히 설명합니다.

## 1단계. HTTPS 터널 서블릿에 대해 자체 서명된 인증서 생성

Message Queue의 SSL 지원은 클라이언트가 알려지고 신뢰할 수 있는 서버와 통신한다는 가정 하에 전송 데이터를 보호하기 위한 것입니다. 따라서 자체 서명된 서버 인증서만 사용하여 SSL을 구현합니다. `httpsjms` 연결 서비스 구조에서 HTTPS 터널 서블릿은 브로커와 응용 프로그램 클라이언트에 대해 서버 역할을 담당합니다.

`imqkeytool` 유틸리티를 실행하여 터널 서블릿에 대해 자체 서명된 인증서를 생성합니다. 명령 프롬프트에서 다음을 입력합니다.

```
imqkeytool -servlet keystore_location
```

유틸리티는 필요한 정보를 묻는 메시지를 표시합니다(Unix 시스템에서 키 저장소를 만들 권한을 가지려면 `imqkeytool`을 슈퍼유저(루트)로 실행해야 할 수도 있음).

`imqkeytool`은 키 저장소 비밀번호, 일부 조직 정보 및 사용자의 확인을 묻는 메시지를 차례로 표시합니다. 확인이 끝나면 키 쌍을 생성하는 동안 일시 중지됩니다. 그런 다음 특정 키 쌍을 잠금 비밀번호(키 비밀번호)를 묻습니다. 이 프롬프트에 대한 응답으로 **Return** 키를 눌러야 합니다. 그러면 키 비밀번호가 키 저장소 비밀번호와 동일하게 지정됩니다.

---

**주** 입력한 비밀번호를 기억해야 합니다. 나중에 이 비밀번호를 제공해야 터널 서블릿이 키 저장소를 열 수 있습니다.

---



imqkeytool을 실행하면 JDK keytool 유틸리티를 실행하여 자체 서명된 인증서를 생성하고 `keystore_location` 인수에서 지정한 위치의 Message Queue 키 저장소 파일에 넣을 수 있습니다(키 저장소의 형식은 JDK1.2 keytool에서 지원하는 것과 같음).

---

**주** HTTPS 터널 서블릿에서 키 저장소를 볼 수 있어야 합니다. `keystore_location`에 생성된 키 저장소를 HTTPS 터널 서블릿에서 액세스할 수 있는 위치로 이동/복사합니다(383페이지의 "2단계. 웹 서버에 HTTPS 터널 서블릿 배포" 참조).

---

## 2단계. 웹 서버에 HTTPS 터널 서블릿 배포

웹 서버에 HTTPS 터널 서블릿을 배포하는 일반적인 방법에는 두 가지가 있습니다.

- jar 파일로 배포 — Servlet 2.1 이전 버전을 지원하는 웹 서버용
- 웹 아카이브(WAR) 파일로 배포 — Servlet 2.2 이상 버전을 지원하는 웹 서버용

두 경우 모두 클라이언트와 브로커 사이에서 종단간 보안 통신이 이루어질 수 있도록 웹 서버에 암호화가 활성화되었는지 확인해야 합니다.

### Jar 파일로 배포

Message Queue 터널 서블릿 배포는 호스트 웹 서버에 액세스할 수 있는 적절한 jar 파일을 생성하고 웹 서버 시작 시 서블릿을 로드하도록 구성하며 서블릿 URL의 컨텍스트 루트 부분을 지정하는 작업으로 구성됩니다.

터널 서블릿 jar 파일(imqservlet.jar)은 HTTPS 터널 서블릿에서 필요한 모든 클래스를 포함하며, 운영 체제에 따라 다른 디렉토리에 있습니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

서블릿 2.x를 지원하는 모든 웹 서버를 사용하여 이 서블릿을 로드할 수 있습니다. 서블릿 클래스 이름은 다음과 같습니다.

```
com.sun.messaging.jmq.transport.  
httpunnel.servlet.HttpsTunnelServlet
```

웹 서버에서 imqservlet.jar 파일을 볼 수 있어야 합니다. 웹 서버와 브로커를 서로 다른 호스트에서 실행하려면 imqservlet.jar 파일의 복사본을 웹 서버에서 액세스할 수 있는 위치에 저장해야 합니다.

또한 웹 서버 시작 시 이 서블릿을 로드하도록 구성해야 하며, 서블릿 URL의 컨텍스트 루트 부분을 지정해야 하는 경우도 있습니다(389페이지의 "예 3: Sun Java System Web Server에 HTTPS 터널 서블릿 배포" 참조).

JSSE jar 파일이 웹 서버에서 서블릿을 실행하는 클래스 경로에 있는지 확인합니다. 수행 방법은 웹 서버 설명서를 확인하십시오.

웹 서버 구성에서 중요한 부분 중 하나는 HTTPS 터널 서블릿에서 브로커와 보안 연결을 설정할 때 사용할 자체 서명된 인증서의 위치와 비밀번호를 지정하는 것입니다. 382페이지의 "1단계. HTTPS 터널 서블릿에 대해 자체 서명된 인증서 생성"에서 만든 키 저장소를 HTTPS 터널 서블릿에서 액세스할 수 있는 위치에 두어야 합니다.

또한 성능 향상을 위해 웹 서버의 액세스 로깅 기능은 사용하지 않는 것이 좋습니다.

### 웹 아카이브 파일로 배포

HTTPS 터널 서블릿을 WAR 파일로 배포하는 작업은 웹 서버에서 제공하는 배포 메커니즘을 사용하여 수행합니다. HTTPS 터널 서블릿 WAR 파일(imqhttps.war)은 운영 체제에 따라 다른 디렉토리에 있습니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

WAR 파일에는 웹 서버에서 서블릿을 로드하여 실행하는 데 필요한 기본 구성 정보가 들어 있는 배포 설명자가 포함되어 있습니다. 웹 서버에 따라 서블릿 URL의 컨텍스트 루트 부분을 지정해야 하는 경우도 있습니다(394페이지의 "예 4: Sun Java System Application Server 7.0에 HTTPS 터널 서블릿 배포" 참조).

하지만 imqhttps.war 파일의 배포 설명자에서는 터널 서블릿에 필요한 키 저장소 파일이 있는 위치를 알 수 없습니다(382페이지의 "1단계. HTTPS 터널 서블릿에 대해 자체 서명된 인증서 생성" 참조). 따라서 imqhttps.war 파일을 배포하기 전에 터널 서블릿의 배포 설명자(XML 파일)를 편집하여 키 저장소 위치를 지정해야 합니다.

## 3단계. httpsjms 연결 서비스 구성

기본적으로 브로커에 대해 HTTPS 지원이 활성화되어 있지 않으므로 httpsjms 연결 서비스가 활성화되도록 브로커를 재구성해야 합니다. 재구성한 경우 67페이지의 "대화식으로 브로커 시작"에 설명된 대로 브로커를 시작할 수 있습니다.

### ▶ httpsjms 연결 서비스를 활성화하는 방법

1. 브로커의 인스턴스 구성 파일을 엽니다.

인스턴스 구성 파일은 구성 파일이 연결되어 있는 브로커 인스턴스의 이름 (*instanceName*)으로 식별되는 디렉토리에 저장됩니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

```
.../instances/instanceName/props/config.properties
```

2. `imq.service.activelist` 등록 정보에 `httpsjms` 값을 추가합니다.

```
imq.service.activelist=jms,admin,httpsjms
```

브로커는 시작할 때 호스트 시스템에서 실행 중인 웹 서버와 HTTPS 터널 서블릿을 찾습니다. 그러나 원격 터널 서블릿에 액세스하기 위해 `servletHost` 및 `servletPort` 연결 서비스 등록 정보를 재구성할 수 있습니다.

또한 성능 향상을 위해 `pullPeriod` 등록 정보를 재구성할 수도 있습니다. `httpsjms` 연결 서비스 구성 등록 정보에 대한 자세한 설명은 표 C-3을 참조하십시오.

**표 C-3** httpsjms 연결 서비스 등록 정보

등록 정보 이름	설명
<code>imq.httpsjms.https servletHost</code>	필요한 경우 이 값을 변경하여 HTTPS 터널 서블릿을 실행하는 호스트의 이름(호스트 이름 또는 IP 주소)을 지정합니다(원격 호스트이거나 로컬 호스트의 특정 호스트 이름일 수 있음). 기본값: localhost
<code>imq.httpsjms.https servletPort</code>	이 값을 변경하여 브로커가 HTTPS 터널 서블릿에 액세스하는 데 사용하는 포트 번호를 지정합니다(웹 서버에서 기본 포트를 변경한 경우 이 등록 정보를 적절히 변경해야 함). 기본값: 7674

**표 C-3** httpsjms 연결 서비스 등록 정보(계속)

등록 정보 이름	설명
imq.httpsjms.https pullPeriod	브로커에서 메시지를 가져오기 위해 각 클라이언트가 만든 HTTP 요청 사이의 간격(초)을 지정합니다(이 등록 정보는 브로커에서 설정되고 클라이언트 런타임에 전파됨). 값이 0 또는 음수인 경우 클라이언트는 하나의 HTTP 요청을 항상 보류 상태로 두고 가능한 빨리 메시지를 가져오도록 준비합니다. 클라이언트 수가 많은 경우 웹 서버 자원을 많이 사용하여 서버가 응답하지 않을 수 있습니다. 그런 경우 pullPeriod 등록 정보를 양수(초)로 설정해야 합니다. 이 등록 정보는 후속 가져오기 요청을 만들기 전에 클라이언트의 HTTP 전송 드라이버가 대기하는 시간을 설정합니다. 값을 양수로 설정하면 클라이언트가 응답 시간 동안 대기하는 대신 웹 서버 자원이 절약됩니다. 기본값: -1
imq.httpsjms.https connectionTimeout	클라이언트 런타임이 HTTPS 터널 서블릿의 응답을 기다리는 시간(초)을 지정합니다. 이 시간이 초과되면 예외가 발생합니다(이 등록 정보는 브로커에서 설정되고 클라이언트 런타임에 전파됨). 이 등록 정보는 브로커가 HTTPS 터널 서블릿과 통신한 후 연결을 해제할 때까지 기다리는 시간도 지정합니다. 이 경우에는 브로커와 터널 서블릿이 HTTPS 서블릿에 액세스 중인 클라이언트가 비정상적으로 종료했는지 여부를 알 수 없으므로 시간 초과가 필요합니다. 기본값: 60

## 4단계. HTTPS 연결 구성

클라이언트 응용 프로그램에서는 제대로 구성된 연결 팩토리 관리 대상 객체를 사용해서 브로커에 대한 HTTPS 연결을 설정해야 합니다.

하지만 클라이언트는 Java Secure Socket Extension(JSSE)에서 제공하는 SSL 라이브러리에도 액세스해야 하며 루트 인증서도 있어야 합니다. SSL 라이브러리는 JDK 1.4와 함께 제공됩니다. 이전 버전의 JDK가 있는 경우에는 ["JSSE 구성"](#)을 참조하십시오. 그렇지 않으면 ["루트 인증서 가져오기"](#)로 이동하십시오.

이 문제를 해결하고 나면 계속해서 HTTPS 연결을 구성할 수 있습니다.

### JSSE 구성

#### ▶ JSSE를 구성하는 방법

1. JSSE jar 파일을 JRE\_HOME/lib/ext 디렉토리로 복사합니다.

jsse.jar, jnet.jar, jcert.jar

2. JSSE 보안 공급자를 정적으로 추가합니다. 이 작업은

```
security.provider.n=com.sun.net.ssl.internal.ssl.Provider
```

를 `JRE_HOME/lib/security/java.security` 파일에 추가하여 수행할 수 있습니다. (여기서 `n`은 보안 공급자 패키지에서 다음으로 사용할 수 있는 우선 순위 번호입니다.)

3. JDK1.4를 사용하지 않는 경우에는 클라이언트 응용 프로그램을 시작하는 명령에 `-D` 옵션을 사용해서 다음 JSSE 등록 정보를 설정해야 합니다.

```
java.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol
```

## 루트 인증서 가져오기

웹 서버 인증서에 서명한 CA의 루트 인증서가 기본적으로 트러스트 데이터베이스에 있지 않거나 독자적인 웹 서버 인증서를 사용하는 경우에는 인증서를 트러스트 데이터베이스에 추가해야 합니다. 여기에 해당되면 다음 지침에 따르고, 그렇지 않으면 "[연결 팩토리 구성](#)"으로 이동합니다.

인증서가 `cert_file`에 저장되어 있고 `trust_store_file`이 키 저장소라면 다음 명령을 실행합니다.

```
JRE_HOME/bin/keytool -import -trustcacerts
-alias alias_for_certificate -file cert_file
-keystore trust_store_file
```

다음 질문에 YES로 응답합니다. Trust this certificate?

클라이언트 응용 프로그램을 시작하는 명령에 `-D` 옵션을 사용해서 다음과 같은 JSSE 등록 정보도 지정해야 합니다.

```
javax.net.ssl.trustStore=trust_store_file
javax.net.ssl.trustStorePassword=trust_store_passwd
```

## 연결 팩토리 구성

HTTPS 지원을 활성화하려면 연결 팩토리의 `imgAddressList` 속성을 HTTPS 터널 서블릿 URL로 설정해야 합니다. HTTPS 터널 서블릿 URL의 일반 구문은 다음과 같습니다.

```
https://hostName:port/contextRoot/tunnel
```

여기서 `hostName:port`는 HTTPS 터널 서블릿을 호스트하는 웹 서버의 이름과 포트이며, `contextRoot`는 웹 서버에 터널 서블릿을 배포할 때 설정된 경로입니다.

일반적인 연결 팩토리 속성, 특히 `imgAddressList` 속성에 대한 자세한 내용은 *Message Queue Developer's Guide for Java Clients*를 참조하십시오.

다음 방법 중 하나를 사용하여 연결 팩토리 속성을 설정할 수 있습니다.

- 연결 팩토리 관리 대상 객체를 만드는 `imqobjmgr` 명령에 `-o` 옵션을 사용하거나(189 페이지의 "연결 팩토리 추가" 참조) 관리 콘솔(`imqadmin`)을 사용하여 연결 팩토리 관리 대상 객체를 만들 때 속성을 설정합니다.
- 클라이언트 응용 프로그램을 시작하는 명령에 `-D` 옵션을 사용합니다(*Message Queue Developer's Guide for Java Clients* 참조).
- 클라이언트 응용 프로그램 코드에서 프로그래밍 방식으로 연결 팩토리를 만든 후 API 호출을 사용하여 해당 연결 팩토리의 속성을 설정합니다(*Message Queue Developer's Guide for Java Clients* 참조).

### 단일 서블릿을 사용하여 다중 브로커에 액세스

다중 브로커를 실행할 경우 다중 웹 서버 및 서블릿 인스턴스를 구성하지 않아도 됩니다. 동시에 실행 중인 브로커 간에 단일 웹 서버와 HTTPS 터널 서블릿을 공유할 수 있습니다. 다중 브로커 인스턴스가 단일 터널 서블릿을 공유하는 경우 `imqAddressList` 연결 팩토리 속성을 다음과 같이 구성해야 합니다.

```
https://hostName:port/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

여기서 `bkrHostName`은 브로커 인스턴스 호스트 이름이고 `instanceName`은 클라이언트가 액세스할 특정 브로커 인스턴스의 이름입니다.

`bkrhostName`과 `instanceName`에 정확한 문자열을 입력했는지 확인하려면 브라우저에서 서블릿 URL에 액세스하여 HTTPS 터널 서블릿의 상태 보고서를 생성합니다. 보고서에는 서블릿이 액세스하는 모든 브로커가 나열됩니다.

```
HTTPS tunnel servlet ready.
Servlet Start Time : Thu May 30 01:08:18 PDT 2002
Accepting TCP connections from brokers on port : 7674
Total available brokers = 2
Broker List :
  jpgserv:broker2
  cochin:broker1
```

## HTTP 프록시 사용

HTTP 프록시를 사용해서 HTTPS 터널 서블릿에 액세스하는 경우:

- `http.proxyHost` 시스템 등록 정보를 프록시 서버 호스트 이름으로 설정합니다.
- `http.proxyPort` 시스템 등록 정보를 프록시 서버 포트 번호로 설정합니다.

클라이언트 응용 프로그램을 시작하는 명령에 `-D` 옵션을 사용하여 이러한 등록 정보를 설정할 수 있습니다.

## 예 3: Sun Java System Web Server에 HTTPS 터널 서블릿 배포

이 절에서는 Sun Java System Web Server에서 HTTPS 터널 서블릿을 jar 파일과 WAR 파일로 배포하는 방법을 모두 설명합니다. 사용 방법은 Sun Java System Web Server 버전에 따라 다르지만, Servlet 2.2 이상을 지원하지 않는다면 WAR 파일 배포를 처리할 수 없습니다.

### Jar 파일로 배포

다음은 브라우저 기반 관리 GUI를 사용하여 Sun Java System Web Server 6.1에 배포하는 경우에 대한 지침입니다. 이 절차는 다음과 같은 일반적인 단계로 구성됩니다.

1. 서블릿 추가
2. 서블릿 가상 경로 구성
3. 서블릿 로드
4. 서블릿 액세스 로그 비활성화

이러한 단계는 다음 하위 절에 설명되어 있습니다. 웹 브라우저를 사용하여 서블릿 URL에 액세스함으로써 HTTPS 터널 서블릿 배포 성공 여부를 확인할 수 있습니다. 상태 정보를 표시해야 합니다.

### 서블릿 추가

#### ▶ 터널 서블릿을 추가하는 방법

1. 서블릿 탭을 선택합니다.
2. 서블릿 속성 구성을 선택합니다.
3. 서블릿 이름 필드에 터널 서블릿의 이름을 지정합니다.

4. 서블릿 코드(클래스 이름) 필드를 다음 값으로 설정합니다.  
`com.sun.messaging.jmq.transport.  
 httptunnel.servlet.HttpsTunnelServlet`
5. 서블릿 클래스 경로 필드에 `imqservlet.jar`에 대한 전체 경로를 입력합니다. 예를 들면 다음과 같습니다.  
`/usr/share/lib/imq/imqservlet.jar(Solaris)  
 /opt/sun/mq/share/lib/imqservlet.jar(Linux)  
 IMQ_HOME/lib/imqservlet.jar(Windows)`
6. 서블릿 인수 필드에 표 C-4와 같은 필수 및 선택 인수를 입력합니다.

**표 C-4** HTTPS 터널 서블릿 Jar 파일 배포에 사용되는 서블릿 인수

인수	기본값	필수?
<code>keystoreLocation</code>	없음	예
<code>keystorePassword</code>	없음	예
<code>servletHost</code>	모든 호스트	아니요
<code>servletPort</code>	7674	아니요

인수를 쉼표로 분리합니다. 예를 들면 다음과 같습니다.

`keystoreLocation=keystore_location, keystorePassword=keystore_password,  
 servletPort=portnumber`

`servletHost` 및 `servletPort` 인수는 웹 서버와 브로커간 통신에만 적용되며 기본값에 문제가 있는 경우에만 설정됩니다. 그런 경우에도 브로커 구성 등록 정보를 적절하게 설정해야 합니다(385페이지의 표 C-3 참조). 예를 들면 다음과 같습니다.

`imq.httpsjms.https.servletPort`



### 서블릿 가상 경로(서블릿 URL) 구성

#### ▶ 터널 서블릿에 대한 가상 경로(서블릿 URL)를 구성하는 방법

1. 서블릿 탭을 선택합니다.
2. 서블릿 가상 경로 변환 구성을 선택합니다.
3. 가상 경로 필드를 설정합니다.

가상 경로는 터널 서블릿 URL의 `/contextRoot/tunnel` 부분입니다.

`https://hostName:port/contextRoot/tunnel`

예를 들어, `contextRoot`를 `img`로 설정한 경우 가상 경로 필드는 다음과 같습니다.

`/img/tunnel`

4. 서블릿 이름 필드를 389페이지의 "서블릿 추가"의 단계 3과 동일한 값으로 설정합니다.

### 서블릿 로드

#### ▶ 웹 서버 시작 시 터널 서블릿을 로드하는 방법

1. 서블릿 탭을 선택합니다.
2. 전역 속성 구성을 선택합니다.
3. 시작 서블릿 필드에 389페이지의 "서블릿 추가"의 단계 3과 동일한 서블릿 이름 값을 입력합니다.

### 서버 액세스 로그 비활성화

서버 액세스 로그를 비활성화할 필요는 없지만 그렇게 하면 성능이 향상됩니다.

#### ▶ 서버 액세스 로그를 비활성화하는 방법

1. 상태 탭을 선택합니다.
2. 로그 기본 설정 페이지를 선택합니다.
3. 로그 클라이언트 액세스 제어를 사용하여 로깅을 비활성화합니다.

## WAR 파일로 배포

다음은 Sun Java System Web Server 6.0 서비스 팩 2에 배포하는 경우에 대한 지침입니다. 웹 브라우저를 사용하여 서블릿 URL에 액세스하면 HTTPS 터널 서블릿의 배포 성공 여부를 확인할 수 있습니다. 상태 정보를 표시해야 합니다.

HTTPS 터널 서블릿을 배포하기 전에 웹 서버의 클래스 경로에 JSSE jar 파일이 포함되어 있는지 확인합니다. 이 작업을 수행하는 가장 간단한 방법은 jsse.jar, jnet.jar 및 jcert.jar 파일을 IWS60\_TOPDIR/bin/https/jre/lib/ext로 복사하는 것입니다.

또한, HTTPS 터널 서블릿을 배포하기 전에 키 저장소 파일의 위치를 가리키고 키 저장소 비밀번호를 지정하도록 배포 설명자를 수정해야 합니다.

### ▶ HTTPS 터널 서블릿 WAR 파일을 수정하는 방법

1. WAR 파일을 임시 디렉토리로 복사합니다.

```
cp /usr/share/lib/imq/imqhttps.war /tmp(Solaris)
```

```
cp /opt/sun/mq/share/lib/imqhttps.war /tmp(Linux)
```

```
cp IMQ_HOME/lib/imqhttps.war /tmp(Windows)
```

2. 임시 디렉토리를 현재 디렉토리로 만듭니다.

```
$ cd /tmp
```

3. WAR 파일의 내용을 추출합니다.

```
$ jar xvf imqhttps.war
```

4. WAR 파일의 배포 설명자를 나열합니다.

```
$ ls -l WEB-INF/web.xml
```

5. keystoreLocation과 keystorePassword 인수(필요한 경우 servletPort 및 servletHost 인수 포함)에 정확한 값을 제공하도록 web.xml 파일을 편집합니다.

6. WAR 파일의 내용을 다시 어셈블합니다.

```
$ jar uvf imqhttps.war WEB-INF/web.xml
```

이제 수정된 imqhttps.war 파일을 사용하여 HTTPS 터널 서블릿을 배포할 수 있습니다. (키 저장소 비밀번호 노출이 우려되는 경우에는 파일 시스템 권한을 사용하여 imqhttps.war 파일에 대한 액세스를 제한할 수 있습니다.)

▶ **https 터널 서블릿을 WAR 파일로 배포하는 방법**

1. 브라우저 기반 관리 GUI에서 가상 서버 클래스 탭을 선택합니다. 클래스 관리를 누릅니다.
2. 해당 가상 서버 클래스 이름(예: defaultClass)을 선택하고 관리 버튼을 누릅니다.
3. 가상 서버 관리를 선택합니다.
4. 해당 가상 서버 이름을 선택하고 관리 버튼을 누릅니다.
5. 웹 응용 프로그램 탭을 선택합니다.
6. 웹 응용 프로그램 배포를 누릅니다.
7. WAR 파일 위치와 WAR 파일 경로 필드에서 수정된 `imqhttps.war` 파일(392페이지의 "HTTPS 터널 서블릿 WAR 파일을 수정하는 방법" 참조)을 가리키는 적절한 값을 선택합니다.
8. 응용 프로그램 URI 필드에 경로를 입력합니다.

응용 프로그램 URI 필드 값은 터널 서블릿 URL의 `/contextRoot` 부분입니다.

```
https://hostName:port/contextRoot/tunnel
```

예를 들어, `contextRoot`를 `imq`로 설정한 경우 응용 프로그램 URI 필드는 다음과 같습니다.

```
/imq
```

9. 서블릿을 배포할 설치 디렉토리 경로(일반적으로 Sun Java System Web Server 설치 루트 아래)를 입력합니다.
10. 확인을 누릅니다.
11. 웹 서버 인스턴스를 다시 시작합니다.

이제 서블릿을 다음 주소에서 사용할 수 있습니다.

```
https://hostName:port/imq/tunnel
```

이제 클라이언트에서 이 URL을 통해 보안 HTTPS 연결을 사용하는 메시지 서비스에 연결할 수 있습니다.

## 예 4: Sun Java System Application Server 7.0에 HTTPS 터널 서블릿 배포

이 절에서는 Sun Java System Application Server 7.0에서 HTTPS 터널 서블릿을 WAR 파일로 배포하는 방법을 설명합니다.

다음 두 단계를 거쳐야 합니다.

- Application Server 7.0 배포 도구를 사용하여 HTTPS 터널 서블릿 배포
- 응용 프로그램 서버 인스턴스의 `server.policy` 파일 수정

### 배포 도구 사용

#### ▶ Application Server 7.0 환경에 HTTPS 터널 서블릿을 배포하는 방법

1. 웹 기반 관리 GUI에서 다음을 선택합니다.

Application Server > 인스턴스 > server1 > 응용 프로그램 > 웹 응용 프로그램

2. 배포 버튼을 누릅니다.

3. 파일 경로 텍스트 필드에 HTTPS 터널 서블릿 WAR 파일(`imghttps.war`)의 위치를 입력합니다.

`imghttps.war` 파일의 위치는 운영 체제에 따라 다릅니다(부록 A, "운영 체제별 Message Queue 데이터의 위치" 참조).

4. 확인을 누릅니다.

5. 다음 화면에서 컨텍스트 루트 텍스트 필드의 값을 설정합니다.

컨텍스트 루트 필드 값은 터널 서블릿 URL의 `/contextRoot` 부분입니다.

```
https://hostName:port/contextRoot/tunnel
```

예를 들어, 컨텍스트 루트 필드를 다음과 같이 설정할 수 있습니다.

```
/img
```

6. 확인을 누릅니다.

다음 화면에 터널 서블릿이 성공적으로 배포되었고 기본적으로 사용되며 다음 위치(이 예의 경우)에 있다고 표시됩니다.

```
/var/opt/SUNWappserver7/domains/domain1/server1/applications/
j2ee-modules/imghttps_1
```

이제 서블릿을 다음 주소에서 사용할 수 있습니다.

```
https://hostName:port/contextRoot/tunnel
```

이제 클라이언트에서 이 URL을 통해 HTTPS 연결을 사용하는 메시지 서비스에 연결할 수 있습니다.

### server.policy 파일 수정

Application Server 7.0이 시행하는 일련의 기본 보안 정책은 수정하지 않으면 HTTPS 터널 서블릿이 Message Queue 브로커에서 연결을 수신할 수 없도록 합니다.

응용 프로그램 서버 인스턴스마다 해당 보안 정책이나 규칙이 포함된 파일이 있습니다. 예를 들어, Solaris의 server1 인스턴스의 경우 이 파일의 위치는 다음과 같습니다.

```
/var/opt/SUNWappserver7/domains/domain1/server1/config/  
server.policy
```

터널 서블릿이 Message Queue 브로커에서 연결을 수신하도록 하려면 이 파일에 항목을 추가해야 합니다.

#### ▶ 응용 프로그램 서버의 server.policy 파일을 수정하는 방법

1. server.policy 파일을 엽니다.
2. 다음 항목을 추가합니다.

```
grant codeBase  
"file:/var/opt/SUNWappserver7/domains/domain1/server1/  
    applications/j2ee-modules/imqhttps_1/-"  
{  
    permission java.net.SocketPermission "*",  
        "connect,accept,resolve";  
};
```

## 문제 해결

이 절에서는 HTTP 또는 HTTPS 연결에서 발생할 수 있는 문제와 문제 해결 방법에 대해 설명합니다.

### 서버 또는 브로커 오류

웹 서버가 오류로 다시 시작되면 모든 연결이 복원되며 클라이언트에는 아무 영향도 주지 않습니다. 그러나, 브로커가 오류로 다시 시작되면 예외가 발생하며 클라이언트는 연결을 다시 설정해야 합니다.

웹 서버와 브로커 모두에 오류가 발생하여 브로커가 다시 시작되지 않는 경우 웹 서버는 클라이언트에게 알리지 않고 클라이언트 연결을 복원한 후 브로커가 연결될 때까지 기다립니다. 이런 상황을 방지하려면 브로커가 다시 시작되었는지 항상 확인하십시오.

### 터널 서블릿을 통한 클라이언트 연결 오류

HTTPS 클라이언트가 터널 서블릿을 통해 브로커에 연결되지 않는 경우 다음을 수행합니다.

1. 서블릿과 브로커를 시작합니다.
2. 브라우저에서 HTTPS 터널 서블릿 URL을 사용하여 서블릿에 수동으로 액세스합니다.
3. 다음 관리 명령을 사용하여 연결을 일시 중지하였다가 다시 시작합니다.

```
imqcmd pause svc -n httpsjms -u admin  
imqcmd resume svc -n httpsjms -u admin
```

서비스가 다시 시작되면 HTTPS 클라이언트가 터널 서블릿을 통해 브로커에 연결할 수 있어야 합니다.

# 용어집

Message Queue 용어에 대한 자세한 내용은 *Message Queue 기술 개요*의 용어집을 참조하십시오. Sun Java System 제품군에 사용되는 전체 용어 목록은 Java Enterprise System 용어집(<http://docs.sun.com/doc/819-1934>)을 참조하십시오.





## 가

- 감사 로깅 171
- 개발 환경 관리 작업 34
- 객체 저장소
  - 위치 360, 361, 362
  - 정보 174
  - 파일 시스템 저장소 175
  - 파일 시스템 저장소 속성 176
  - LDAP 서버 174
  - LDAP 서버 속성 174
- 객체 저장소 위치 174, 176
- 관리 대상 객체
  - 객체 저장소, 객체 저장소 참조
  - 나열 193
  - 대기열, 대기열 참조
  - 삭제 192
  - 속성 185
  - 속성(참조) 333
  - 업데이트 194
  - 조회 이름 298
  - 쿼리 193
  - 필요한 정보 185
  - 항목, 항목 참조
  - XA 연결 팩토리, 연결 팩토리 관리 대상 객체 참조
- 관리 도구 37
  - 관리 콘솔 39
  - 명령줄 유틸리티 37
- 관리 작업
  - 개발 환경 34
  - 작업 환경 34
  - 관리 콘솔
    - 빠른 시작 41
    - 시작 42
  - 관리자 비밀번호 148
  - 구성 등록 정보의 우선 순위 97
  - 구성 변경 기록 201
    - 백업 201
    - 복원 202
  - 구성 파일 96
    - 기본값 96
    - 브로커(그림) 97
    - 설치 96
    - 위치 360, 361, 362
  - 인스턴스 96, 197, 360, 361, 362
  - 클러스터 196, 197, 198, 327
  - 템플리트 360, 361, 362
  - 템플리트 위치 360, 361, 362
  - 편집 98
- 권한
  - 계산 155
  - 내장 데이터베이스 102
  - 데이터 저장소 86
  - 사용자 저장소 144, 302
  - 액세스 제어 등록 정보 파일 88, 153
  - 키 저장소 382
  - admin 서비스 89
  - Message Queue 작업 88
  - passfile 170

## 다

권한 부여

관리 [152](#)

사용자 그룹 [89](#)

액세스 제어 파일 [참조](#)

정보 [88](#)

기본 제공 지속성 [85](#)

## 다

다시 시작

물리적 대상 [134](#)

브로커 [113](#), [114](#), [289](#)

연결 서비스 [120](#), [121](#), [292](#)

다시 연결, 자동 재연결 [참조](#)

대기열

관리 대상 객체 추가 [191](#)

자동 작성 [308](#), [315](#)

대기열 로드 균형 조정 전달

속성 [130](#), [330](#)

대상 메트릭

메시지 기반 모니터링 사용 [216](#)

메트릭 개수 [354](#)

imqcmd metrics 사용 [211](#), [214](#), [291](#)

imqcmd query 사용 [215](#)

대상 삭제 [135](#)

데이터 저장소

구성 [99](#)

내용 [99](#)

디스크에 동기화 [100](#)

성능 영향 [236](#)

압축 [85](#)

위치 [360](#), [361](#), [362](#)

재설정 [285](#)

정보 [83](#)

플랫 파일 [85](#)

JDBC 액세스 가능 [86](#)

도구, 관리, 관리 도구 [참조](#)

도움말(명령줄) [281](#)

동기화

메모리를 디스크에 [85](#), [100](#)

클릭 [66](#)

등록 정보

구문 [98](#)

로거 [324](#)

메모리 관리 [129](#), [313](#)

메시지 라우터 [313](#)

물리적 대상, 물리적 대상 [참조](#), 등록 정보

보안 [320](#)

브로커 모니터링 서비스 [324](#)

브로커 인스턴스 구성 [98](#)

연결 서비스 [311](#)

자동 작성 [314](#)

지속성 [317](#)

클러스터 구성 [326](#)

키 저장소 [323](#)

httpjms 연결 서비스 [373](#)

httpsjms 연결 서비스 [385](#)

JDBC 관련 [102](#), [317](#)

디렉토리 변수

IMQ\_HOME [25](#)

IMQ\_JAVAHOME [26](#)

IMQ\_VARHOME [25](#)

디스크 공간

물리적 대상 사용률 [136](#)

확보 [137](#)

## 라

라우팅, 메시지 라우터 [참조](#)

로거

구성 변경 [207](#)

등록 정보 설정 [207](#)

로그 메시지 리디렉션 [208](#)

롤오버 기준 [208](#)

메시지 형식 [206](#)

메트릭 정보 [94](#), [325](#)

범주 [205](#)

브로커 구성 요소 [75](#)

수준 [94](#), [205](#), [284](#), [325](#)

정보 [92](#)

출력 채널 [92](#), [205](#), [207](#)

- 콘솔에 쓰기 94, 286, 324
- 로그 파일
  - 기본 위치 360, 361, 362
  - 롤오버 기준 94, 324
- 로깅, 로거 참조
- 로드 균형 조정 대기열 전달
  - 성능 조정 243
  - 속성 315

## 마

- 마스터 브로커
  - 구성 변경 기록 201
  - 사용 불가능 202
  - 지정 196, 197
- 메모리 관리
  - 물리적 대상 등록 정보 사용 129
  - 브로커용 81
  - 성능 조정 242
  - 임계값 82
- 메시지
  - 단편화 85
  - 대기 시간 220
  - 대상 제한 329
  - 라우팅 및 전달 79
  - 만료 재생 이용 83, 313
  - 메트릭 93
  - 메트릭 메시지, 메트릭 메시지 참조
  - 물리적 대상 제한 129
  - 물리적 대상에서 제거 134, 291
  - 본문 유형 및 성능 231
  - 브로커 제한 83, 112, 310, 313
  - 안정적인 전달 182
  - 재전송 81
  - 지속성 81, 83
  - 처리량 220
  - 크기, 성능 230
  - 확인 80
  - 흐름 일시 중지 133
  - 흐름 제어, 메시지 흐름 제어 참조

- 메시지 단편화 85
- 메시지 라우터
  - 등록 정보 313
  - 브로커 구성 요소 75
  - 정보 79
- 메시지 서버 구조 235
- 메시지 서비스 성능 232
- 메시지 헤더 무시 184
- 메시지 흐름 제어
  - 브로커 81, 129
  - 성능 영향 237
  - 성능 조정 244
  - 속성 182
  - 제한 244
  - 측정 244
- 메트릭
  - 데이터, 메트릭 데이터 참조
  - 메시지, 메트릭 메시지 참조
  - 정보 92
  - 주제 대상 93, 216
- 메트릭 데이터
  - 메시지 기반 모니터링 API 사용 216
  - 물리적 대상, 물리적 대상 메트릭 참조
  - 브로커 로그 파일 사용 209
  - 브로커, 브로커 메트릭 참조
  - 연결 서비스, 연결 서비스 메트릭 참조
  - imqcmd metrics 사용 212
- 메트릭 메시지
  - 내용 93
  - 유형 93, 216
  - 정보 93, 215
- 메트릭 모니터링 도구
  - 메시지 기반 모니터링 API 215
  - 비교 203
  - Message Queue 로그 파일 209
  - Message Queue 명령 유틸리티(imqcmd) 210
- 명령 옵션 281
  - 구성 무시 71
- 명령 파일 186
- 명령줄 구문 280

## 바

### 명령줄 유틸리티

공통 옵션 281

기본 구문 280

도움말 281

버전 표시 281

정보 37

imqbrokerd, imqbrokerd 명령 참조

imqcmd, imqcmd 명령 참조

imqdbmgr, imqdbmgr 명령 참조

imqkeytool, imqkeytool 명령 참조

imqobjmgr, imqobjmgr 명령 참조

imqsvcadm, imqsvcadm 명령 참조

imqusermgr, imqusermgr 명령 참조

모니터링, 성능 모니터링 참조

모든 명령 구문 280

### 무시

메시지 헤더 184

명령줄에서 71

문제 해결 247

Windows 서비스 시작 70

### 물리적 대상

관리 127

나열 131, 290

다시 시작 134, 291

등록 정보 329

등록 정보 값 131

등록 정보 값 표시 131

등록 정보 업데이트 133

디스크 공간 확보 137

디스크 사용률 136

만들기 129

메시지 전달 일괄 처리 129, 315, 331

메시지 제거 134, 291

메트릭, 물리적 대상 메트릭 참조

사용 불능 메시지 대기열 138

사용 불능 메시지 대기열 사용 138

속성 업데이트 291

압축 136

완전 삭제 135, 290

유형 131, 290

일시 중지 133, 134, 291

임시 131

자동 작성 158

정보 131

정보 얻기 131, 291

제한 동작 82, 129, 330

클러스터의 제한된 범위 130, 314, 331

파일 기반 데이터 저장소 압축 137, 290

물리적 대상 완전 삭제 135

물리적 대상의 속성 329

물리적 대상 자동 작성

구성 89

등록 정보(표) 314

비활성화 36

액세스 제어 158

## 바

방화벽 370

버전 281

벤치마크, 성능 221

병목 현상, 성능 224

### 보안

객체 저장소 174

관리자, 보안 관리자 참조

권한 부여, 권한 부여 참조

암호화, 암호화 참조

인증, 인증 참조

### 보안 관리자

등록 정보 320

브로커 구성 요소 75

정보 88

### 분산 트랜잭션

XA 자원 관리자 123

### 브로커

관리 107

구성 요소 및 기능(표) 75

구성 파일, 구성 파일 참조

다시 시작 84, 113, 114, 289

등록 정보 업데이트 112

등록 정보(참조) 307

등록 정보 표시 111

- 로깅, 로거 [참조](#)
- 메모리 관리 [81, 129, 236](#)
- 메시지 라우팅, 메시지 라우터 [참조](#)
- 메시지 용량 [83, 112, 310, 313](#)
- 메시지 흐름 제어, 메시지 흐름 제어 [참조](#)
- 메트릭, 브로커 메트릭 [참조](#)
- 모니터링, 브로커 모니터링 서비스 [참조](#)
- 물리적 대상 등록 정보 자동 작성 [314](#)
- 보안 관리자, 보안 관리자 [참조](#)
- 사용 불능 메시지 대기열 [138](#)
- 상호 연결, 브로커 클러스터 [참조](#)
- 서비스(그림) [74](#)
- 시작하는 데 필요한 사용 권한 [67](#)
- 액세스 제어, 권한 부여 [참조](#)
- 연결 [198](#)
- 연결 서비스 나열 [117](#)
- 연결 서비스, 연결 서비스 [참조](#)
- 오류 복구 [84](#)
- 인스턴스 구성 등록 정보 [98](#)
- 인스턴스 이름 [284](#)
- 일시 중지 [113, 289](#)
- 자동으로 다시 시작 [68](#)
- 제거 [72](#)
- 제한 동작 [82, 236](#)
- 종료 [114](#)
- 지속성 관리자, 지속성 관리자 [참조](#)
- 쿼리 [111](#)
- 클러스터, 브로커 클러스터 [참조](#)
- 클럭 동기화 [66](#)
- HTTP 지원 [371](#)
- httpjms 연결 서비스 등록 정보 [373](#)
- HTTPS 지원 [382](#)
- httpsjms 연결 서비스 등록 정보 [385](#)
- SSL을 사용하여 시작 [163](#)
- Windows 서비스로 실행 [69](#)
- 브로커 다시 시작 [114, 289](#)
- 브로커 메트릭
  - 로거 등록 정보 [94, 209, 325](#)
  - 메시지 기반 모니터링 사용 [216](#)
  - 메트릭 개수(표) [350](#)
  - 메트릭 메시지 [94](#)
  - 보고 간격, 로거 [284](#)
  - 브로커 로그 파일 사용 [209](#)
  - imqcmd 사용 [115, 213, 215](#)
- 브로커 모니터링 서비스
  - 등록 정보 [324](#)
  - 정보 [91](#)
- 브로커 연결 [198](#)
- 브로커 오류 및 보안 연결 [396](#)
- 브로커 응답
  - 생성 시 [339](#)
  - 클라이언트를 위한 대기 시간 [182, 339](#)
- 브로커 종료 [114, 289](#)
- Windows 서비스로 [70](#)
- 브로커 클러스터
  - 구성 등록 정보 [196, 326](#)
  - 구성 변경 기록 [201](#)
  - 구성 파일 [196, 197, 198, 327](#)
  - 구조 [235](#)
  - 물리적 대상 복제 [130](#)
  - 물리적 대상 일시 중지 [134](#)
  - 브로커 간 연결 보안 [199](#)
  - 브로커 연결 [198](#)
  - 브로커 추가 [199](#)
  - 사용 이유 [235](#)
  - 성능 영향 [235](#)
  - 지정할 옵션 [283](#)
- 브로커 Windows 서비스의 시작 매개 변수 [70](#)
- 비밀번호
  - 관리자 [148](#)
  - 기본값 [181, 338](#)
  - 이름 지정 규약 [146](#)
  - 인코딩 [320](#)
  - JDBC [170](#)
  - LDAP [170](#)
  - passfile, passfile [참조](#)
  - SSL 키 저장소 [162, 170, 285](#)
- 비밀번호 파일, passfile [참조](#)

## 사

- 사용 도움말 281
- 사용 불능 메시지
  - 로깅 95
  - 사용 불능 메시지 대기열 참조
- 사용 불능 메시지 대기열
  - 구성 138
  - 로깅 95, 140
  - 메시지 자르기 83
  - 제한 동작 139
  - maxNumMsgs 값 139
  - maxTotalMsgBytes 값 139
- 사용 불능 메시지 대기열의 자르기 83
- 사용권
  - 시작 옵션 284
- 사용률 137
- 사용자 그룹 145
  - 기본값 89
  - 사전 정의됨 145
  - 정보 88
  - 할당 삭제 146
- 사용자 아이디 181, 338
  - 기본값 143
  - 형식 146
- 사용자 저장소
  - 관리 147
  - 등록 정보 90
  - 사용자 그룹 146
  - 사용자 상태 146
  - 위치 360, 361, 362
  - 정보 88
  - 채우기 147
  - 초기 항목 143
  - 플랫 파일 142
  - 플랫폼 의존성 144, 302
  - LDAP 149
  - LDAP 서버 150
- 삭제
  - 브로커 인스턴스 72
- 생성자
  - 대상 제한 315, 330
  - 물리적 대상 제한 129
- 서버 오류 및 보안 연결 396
- 서비스(Windows)
  - 브로커 제거 70
  - 브로커를 다음으로 실행 69
  - 시작 매개 변수 70
  - 시작 문제 해결 70
  - 재구성 69
  - Java 런타임 70
- 서비스 유형
  - ADMIN 75
  - NORMAL 75
- 선택기
  - 성능 영향 230
  - 정보 230
- 성능
  - 기본 패턴 222
  - 모니터링, 성능 모니터링 참조
  - 문제 해결 247
  - 벤치마크 221
  - 병목 현상 224
  - 안정성 균형 225
  - 영향을 미치는 요소, 성능 요소 참조
  - 정보 219
  - 조정, 성능 조정 참조
  - 최적화, 성능 조정 참조
  - 측정 220
  - 표시기 220
- 성능 모니터링
  - 도구, 메트릭 모니터링 도구 참조 203
  - 메트릭 데이터, 메트릭 데이터 참조
- 성능 요소
  - 데이터 저장소 236
  - 메시지 본문 유형 231
  - 메시지 서버 구조 235
  - 메시지 크기 230
  - 메시지 흐름 제어 237
  - 브로커 제한 동작 236
  - 선택기 230
  - 연결 233
  - 영구 가입 228
  - 운영 체제 233

- 전달 모드 226
- 전송 프로토콜 234
- 트랜잭션 227
- 파일 동기화 317
- 하드웨어 232
- 확인 모드 228
- JVM 233
- 성능 조정
  - 브로커 조정 242
  - 시스템 조정 237
  - 클라이언트 런타임 조정 244
  - 프로세스 개요 219
- 스레드 풀 관리자
  - 공유 스레드 78
  - 전용 스레드 77
  - 정보 77
- 시간 동기화 서비스 66
- 시스템 클럭 동기화 66
- 시작
  - 클라이언트 71
  - SSL 기반 연결 서비스 163

## 아

- 안정적인 전달 182
- 성능 균형 225
- 암호화
  - 정보 90
  - 키 도구 및 90
  - SSL 기반 서비스 159
- 압축
  - 물리적 대상 136
  - 파일 기반 데이터 저장소 85
- 액세스 규칙 155
- 액세스 제어 파일
  - 버전 153
  - 액세스 규칙 155
  - 용도 153
  - 위치 360, 361, 362
  - 형식 154

- 업데이트
  - 브로커 112
  - 연결 서비스 118, 119, 122, 292
- 역순환 주소 198
- 연결
  - 나열 121, 293
  - 서버 또는 브로커 오류 396
  - 성능 영향 233
  - 자동 재연결, 자동 재연결 참조
  - 쿼리 122, 293
  - 파일 설명자 제한에 의해 제한됨 66
  - 페일오버, 자동 재연결 참조
- 연결 서비스
  - 다시 시작 120, 121, 292
  - 등록 정보 118, 311
  - 등록 정보 표시 118
  - 메트릭 데이터, 연결 서비스 메트릭 참조
  - 서비스 유형 75
  - 스레드 풀 관리자 77
  - 스레드 할당 118
  - 시작 시 활성화 311
  - 액세스 제어 90, 320
  - 업데이트 118, 119, 122, 292
  - 연결 유형 75
  - 영향을 주는 명령 292
  - 일시 중지 120, 292
  - 정보 75
  - 쿼리 118, 122, 292
  - 클러스터 160, 199
  - 포트 매핑, 포트 매핑 참조
  - admin 76, 116
  - HTTP, HTTP 연결 참조
  - httpjms 76, 116
  - HTTPS, HTTPS 연결 참조
  - httpsjms 76, 116
  - jms 76, 116
  - SSL 기반 162
  - ssladmin, ssladmin 연결 서비스 참조
  - ssljms, ssljms 연결 서비스 참조
- 연결 서비스 메트릭
  - 메트릭 개수 352
  - imqcmd metrics 사용 119, 213
  - imqcmd query 사용 215

## 자

- 연결 팩토리 관리 대상 객체
  - 대기열 브라우저 동작 속성 183, 340
  - 메시지 헤더 필드 무시 184
  - 속성 177
  - 안정성 및 흐름 제어 속성 182
  - 연결 처리 속성 178
  - 추가 189
  - 클라이언트 아이디 속성 180
  - Application Server 지원 속성 183, 340
  - JMS 등록 정보 지원 속성 183, 341
- 영구 가입
  - 관리 122
  - 나열 122, 293
  - 메시지 제거 293
  - 성능 영향 228
  - 완전 삭제 123, 293
  - id 294
- 운영 체제
  - 성능 영향 233
  - Solaris 성능 조정 237
- 응용 프로그램 예 28, 360, 361, 363
- 응용 프로그램, 클라이언트 응용 프로그램 참조
- 인스턴스 구성 파일, 구성 파일 참조
- 인스턴스 디렉토리
  - 및 인스턴스 구성 파일 149
  - 및 파일 기반 데이터 저장소 100
  - 제거 72
- 인증
  - 관리 142
  - 정보 88
- 인증서 160, 382
- 일시 중지
  - 물리적 대상 133, 134, 291
  - 브로커 113, 289
  - 연결 서비스 120, 292
- 임계값
  - 메모리 82
- 임시 물리적 대상 131

## 자

- 자동 재연결 기능
  - 속성 179
- 자습서 41
- 자원 어댑터 343
  - 재연결 344, 345, 346
- 자체 서명된 인증서 160, 382
- 작업 환경
  - 관리 작업 34
  - 설정 35
  - 유지 관리 36
- 재전송 플래그 81
- 전달 모드
  - 성능 영향 226
- 전송 프로토콜
  - 상대 속도 234
  - 성능 영향 234
  - 성능 조정 238
  - 프로토콜 유형, 프로토콜 유형 참조
- 제거
  - 물리적 대상 135
  - 브로커 72
- 제거, 물리적 대상에서 메시지 134
- 제어 메시지 79
- 제품 버전 표시 281
- 제한 동작
  - 물리적 대상 82, 129, 330
  - 브로커 82
- 주제
  - 관리 대상 객체 추가 190
  - 자동 작성 308, 315
- 지속성
  - 기본 제공 85
  - 데이터 저장소 데이터 저장소 참조
  - 보안 104
  - 옵션(그림) 84
  - 지속성 관리자, 지속성 관리자 참조
  - 플러그 인, 플러그 인 지속성 참조
  - JDBC, JDBC 지속성 참조
- 지속성 관리자



데이터 저장소, 데이터 저장소 참조  
 등록 정보 317  
 브로커 구성 요소 75  
 정보 83  
 플러그인 지속성 99

## 카

### 쿼리

브로커 111  
 연결 서비스 118, 122, 292

### 클라이언트

시작 71  
 클럭 동기화 66

### 클라이언트 런타임

구성 237  
 메시지 흐름 조정 244

### 클라이언트 식별자(ClientID) 180

영구 가입 완전 삭제 123

### 클라이언트 응용 프로그램

성능에 영향을 미치는 요소 224  
 예 28, 360, 361, 363

### 클러스터 구성 등록 정보 196, 326

### 클러스터 구성 파일 196, 197, 198, 327

### 클러스터, 브로커 클러스터 참조

### 클러스터 연결 서비스 160, 199

네트워크 전송 196, 197, 327  
 포트 번호 196, 327

호스트 이름 또는 IP 주소 196, 327

### 클러스터용 디렉토리 조회(Linux) 198

### 클럭 동기화 66

### 키 도구 90

### 키 쌍

다시 생성 162  
 생성 161

### 키 저장소

등록 정보 323  
 file 162, 323, 383

## 타

### 터널 서블릿 연결 396

### 트랜잭션

관리 123  
 롤백 124, 294  
 성능 영향 227  
 완결 125, 294  
 정보 294  
 확인 및 81

## 파

### 파일 기반 저장소에 대한 쓰기 작업 100

### 파일 기반 지속성 85

지속성 관리자 파일 기반 지속성 참조

### 파일 동기화

imq.persist.file.sync.enabled 옵션 317  
 Sun Cluster와 317

### 파일 설명자 제한 66

연결 제한 및 66

### 포트 매핑

정보 77  
 포트 지정 285

### 프로토콜 유형

HTTP 76, 116  
 TCP 76, 116  
 TLS 76, 116

### 프로토콜, 전송 프로토콜 참조

### 플러그인 지속성

설정 100  
 성능 조정 242  
 정보 86

## 하

### 하드웨어, 성능 영향 232

### 확인

전달 80

## A

클라이언트 80  
트랜잭션 및 81  
환경 변수, 디렉토리 변수 참조  
흐름 제어, 메시지 흐름 제어 참조

## A

acknowledgeMode 활성화 사양 속성 347  
ActivationSpec JavaBean 346  
addressList 관리 대상 연결 팩토리 속성 345  
addressList 자원 어댑터 속성 344, 345  
addressList 활성화 사양 속성 347  
addressListBehavior 관리 대상 연결 팩토리 속성 345  
addressListBehavior 자원 어댑터 속성 344  
addressListIterations 관리 대상 연결 팩토리 속성 346  
addressListIterations 자원 어댑터 속성 344  
admin 그룹 145  
admin 사용자 143, 148, 151  
ADMIN 서비스 유형 75  
admin 연결 서비스 76, 116  
anonymous 그룹 146  
API 설명서 360, 361, 362  
AUTOSTART 등록 정보 68

## C

clientID 관리 대상 연결 팩토리 속성 346  
clientId 활성화 사양 속성 347, 348  
config.properties 파일 98, 199, 201  
connectionURL 자원 어댑터 속성 344  
customAcknowledgeMode 활성화 사양 속성 347

## D

default.properties 파일 96  
destination 활성화 사양 속성 347  
destinationType 활성화 사양 속성 347, 348

## E

endpointExceptionRedeliveryAttempts 활성화 사양 속성 347, 348  
/etc/hosts 파일(Linux) 198

## G

guest 사용자 143

## H

hosts 파일(Linux) 198  
HTTP  
연결 서비스, httpjms 연결 서비스 참조  
전송 드라이버 370  
지원 구조 370  
프록시 370  
HTTP 연결  
다중 브로커, 375  
요청 간격 374  
지원 370  
터널 서블릿, HTTP 터널 서블릿 참조  
HTTP 터널 서블릿  
배포 372  
정보 370  
httpjms 연결 서비스  
구성 373  
설정 371  
정보 76, 116

## HTTPS

연결 서비스, [httpsjms 연결 서비스 참조](#)  
지원 구조 370

### HTTPS 연결

다중 브로커, 388  
요청 간격 386  
지원 370  
터널 서블릿, [HTTPS 터널 서블릿 참조](#)

### HTTPS 터널 서블릿

배포 383  
정보 370

### httpsjms 연결 서비스

구성 385  
설정 382  
정보 76, 116

## I

imq.accesscontrol.enabled 등록 정보 90, 308, 320

imq.accesscontrol.file.filename 등록 정보 91, 308, 320

imq.audit.enabled 등록 정보 308, 320

imq.authentication.basic.user\_repository  
등록 정보 90, 308, 320

imq.authentication.client.response.timeout  
등록 정보 90, 308, 320

imq.authentication.type 등록 정보 90, 308, 320

imq.autocreate.destination.isLocalOnly  
등록 정보 308, 314

imq.autocreate.destination.limitBehavior  
등록 정보 308, 314

imq.autocreate.destination.maxBytesPerMsg  
등록 정보 308, 315

imq.autocreate.destination.maxCount  
등록 정보 308, 315

imq.autocreate.destination.maxNumMsgs  
등록 정보 315

imq.autocreate.destination.maxNumProducers  
등록 정보 308, 315

imq.autocreate.destination.maxTotalMsgBytes  
등록 정보 308, 315

imq.autocreate.destination.useDMQ 등록 정보 138, 308

imq.autocreate.queue 등록 정보 112, 308, 315

imq.autocreate.queue.consumerFlowLimit  
등록 정보 308, 315

imq.autocreate.queue.localDeliveryPreferred  
등록 정보 308, 315

imq.autocreate.queue.maxNumActiveConsumers  
등록 정보 112, 308, 315

imq.autocreate.queue.maxNumBackupConsumers  
등록 정보 112, 308, 315

imq.autocreate.topic 등록 정보 112, 308, 315

imq.cluster.brokerlist 등록 정보 196, 198, 199, 200, 326

imq.cluster.masterbroker 등록 정보 196, 199, 201, 327

imq.cluster.port 등록 정보 196, 327

imq.cluster.property\_name 등록 정보 308

imq.cluster.transport 등록 정보 196, 199, 327

imq.cluster.url 등록 정보 112, 196, 197, 198, 199, 201, 327

imq.destination.DMQ.truncateBody 등록 정보 83, 112, 308, 313

imq.destination.logDeadMsgs 등록 정보 95, 112, 308, 324

imq.hostname 등록 정보 78, 308, 311

imq.httpjms.http.connectionTimeout 등록 정보 374

imq.httpjms.http.property\_name 등록 정보 308

imq.httpjms.http.pullPeriod 등록 정보 374

imq.httpjms.http.servletHost 등록 정보 373

imq.httpjms.http.servletPort 등록 정보 373

imq.httpsjms.https.connectionTimeout  
등록 정보 386

imq.httpsjms.https.property\_name 등록 정보 308

imq.httpsjms.https.pullPeriod 등록 정보 386

imq.httpsjms.https.servletHost 등록 정보 385

imq.httpsjms.https.servletPort 등록 정보 385

imq.imqcmd.password 등록 정보 308, 321

imq.keystore.dirpath 등록 정보 162, 323  
 imq.keystore.file.name 등록 정보 162, 323  
 imq.keystore.password 등록 정보 162, 170, 323  
 imq.keystore.property\_name 등록 정보 91  
 imq.keystore.*property\_name* 등록 정보 308, 321  
 imq.log.console.output 등록 정보 94, 309, 324  
 imq.log.console.stream 등록 정보 94, 309, 324  
 imq.log.file.dirpath 등록 정보 94, 309, 324  
 imq.log.file.filename 등록 정보 94, 324  
 imq.log.file.name 등록 정보 309  
 imq.log.file.output 등록 정보 94, 309, 324  
 imq.log.file.rolloverbytes 등록 정보 94, 112, 309, 324  
 imq.log.file.rolloversecs 등록 정보 94, 112, 309, 325  
 imq.log.level 등록 정보 94, 112, 309, 325  
 imq.log.syslog.facility 등록 정보 94, 309, 325  
 imq.log.syslog.identity 등록 정보 95, 309, 325  
 imq.log.syslog.logconsole 등록 정보 95, 309, 325  
 imq.log.syslog.logpid 등록 정보 95, 309, 325  
 imq.log.syslog.output 등록 정보 95, 309, 325  
 imq.log.timezone 등록 정보 95, 309, 325  
 imq.message.expiration.interval 등록 정보 83, 309, 313  
 imq.message.max\_size 등록 정보 83, 112, 309, 313  
 imq.metrics.enabled 등록 정보 94, 309, 325  
 imq.metrics.interval 등록 정보 94, 309, 326  
 imq.metrics.topic.enabled 등록 정보 95, 309, 326  
 imq.metrics.topic.interval 등록 정보 95, 309, 326  
 imq.metrics.topic.persist 등록 정보 95, 309, 326  
 imq.metrics.topic.timetolive 등록 정보 95, 309, 326  
 imq.passfile.dirpath 등록 정보 91, 309, 321  
 imq.passfile.enabled 등록 정보 91, 309, 321  
 imq.passfile.name 등록 정보 91, 309, 321  
 imq.persist.file.destination.message.filepool.limit 등록 정보 85, 86, 309, 317  
 imq.persist.file.message.cleanup 등록 정보 85, 86, 309, 317  
 imq.persist.file.message.filepool.cleanratio 등록 정보 86, 309, 317  
 imq.persist.file.message.max\_record\_size 등록 정보 86, 310, 317  
 imq.persist.file.message.vrfile.max\_record\_size 등록 정보 85  
 imq.persist.file.sync 등록 정보 100  
 imq.persist.file.sync.enabled 등록 정보 85, 86, 310, 317  
 Sun Cluster 요구 사항 317  
 imq.persist.jdbc.brokerid 등록 정보 86, 103, 318  
 imq.persist.jdbc.closedburl 등록 정보 87, 103, 318  
 imq.persist.jdbc.createdburl 등록 정보 87, 103, 318  
 imq.persist.jdbc.driver 등록 정보 86, 103, 318  
 imq.persist.jdbc.needpassword 등록 정보 87, 103, 318  
 imq.persist.jdbc.opendburl 등록 정보 86, 103, 318  
 imq.persist.jdbc.password 등록 정보 87, 103, 170, 319  
 imq.persist.jdbc.*property\_name* 등록 정보 310  
 imq.persist.jdbc.table.IMQCCREC35 등록 정보 87, 103, 319  
 imq.persist.jdbc.table.IMQDEST35 등록 정보 87, 103, 319  
 imq.persist.jdbc.table.IMQILIST35 등록 정보 87, 104, 319  
 imq.persist.jdbc.table.IMQINT3 등록 정보 87, 104, 319  
 imq.persist.jdbc.table.IMQMSG35 등록 정보 87, 104, 319  
 imq.persist.jdbc.table.IMQPROPS35 등록 정보 87, 104, 319  
 imq.persist.jdbc.table.IMQSV35 등록 정보 87, 103, 319  
 imq.persist.jdbc.table.IMQTACK35 등록 정보 87, 104, 320  
 imq.persist.jdbc.table.IMQTXN35 등록 정보 87, 104, 319  
 imq.persist.jdbc.user 등록 정보 87, 103, 318  
 imq.persist.store 등록 정보 86, 103, 310, 316, 317  
 imq.ping.interval 등록 정보 310, 311  
 imq.portmapper.backlog 등록 정보 78, 310, 311  
 imq.portmapper.hostname 등록 정보 78, 310, 311

imq.portmapper.port 등록 정보 78, 112, 310, 311  
 imq.protocol protocol\_type inbufsz 239  
 imq.protocol protocol\_type nodelay 239  
 imq.protocol protocol\_type outbufsz 239  
 imq.resource\_state.count 등록 정보 83, 310, 313  
 imq.resource\_state.threshold 등록 정보 83, 310, 313  
 imq.service.activelist 등록 정보 78, 310, 311  
 imq.service\_name.accesscontrol.enabled  
 등록 정보 90, 310, 321  
 imq.service\_name.accesscontrol.file.filename  
 등록 정보 91, 310, 321  
 imq.service\_name.authentication.type 등록 정보 90,  
 310, 321  
 imq.service\_name.max\_threads 등록 정보 79, 310, 312  
 imq.service\_name.min\_threads 등록 정보 78, 310, 312  
 imq.service\_name.protocol\_type.hostname  
 등록 정보 78, 196, 310, 311, 327  
 imq.service\_name.protocol\_type.port 등록 정보 78,  
 310, 311  
 imq.service\_name.threadpool\_model 등록 정보 79,  
 310, 312  
 imq.shared.connectionMonitor\_limit 등록 정보 79,  
 310, 312  
 imq.system.max\_count 등록 정보 83, 112, 310, 313  
 imq.system.max\_size 등록 정보 83, 112, 310, 313  
 imq.transaction.autorollback 등록 정보 81, 83, 125,  
 310, 314  
 imq.user\_repository.ldap.base 등록 정보 150, 321  
 imq.user\_repository.ldap.gidattr 등록 정보 151, 321  
 imq.user\_repository.ldap.grpbase 등록 정보  
 151, 321  
 imq.user\_repository.ldap.grpfilter 등록 정보  
 151, 322  
 imq.user\_repository.ldap.grpsearch 등록 정보  
 151, 322  
 imq.user\_repository.ldap.memattr 등록 정보  
 151, 322  
 imq.user\_repository.ldap.password 등록 정보  
 150, 170, 322  
 imq.user\_repository.ldap.principal 등록 정보  
 150, 322

imq.user\_repository.ldap.property\_name  
 등록 정보 310, 322  
 imq.user\_repository.ldap.server 등록 정보 150, 322  
 imq.user\_repository.ldap.ssl.enabled 등록 정보  
 151, 323  
 imq.user\_repository.ldap.timeout 등록 정보 151, 323  
 imq.user\_repository.ldap.uidattr 등록 정보 150, 323  
 imq.user\_repository.ldap.usrfilter 등록 정보 151,  
 323  
 IMQ\_HOME 디렉토리 변수 25  
 IMQ\_JAVAHOME 디렉토리 변수 26  
 IMQ\_VARHOME 디렉토리 변수 25  
 imqAckOnProduce 속성 339  
 imqAckTimeout 속성 182, 339  
 imqAddressList 속성 334  
 imqAddressListBehavior 속성 334  
 imqAddressListIterations 속성 334  
 imqbrokerd 명령 67  
 구문 282  
 구성 변경 기록 백업 201  
 구성 변경 기록 복원 202  
 구성 파일(Solaris, Linux) 68, 72  
 데이터 저장소 지우기 100, 135  
 로깅 등록 정보 설정 207  
 브로커 제거 72  
 브로커 연결 198  
 옵션 282  
 인수 전달 위치 99  
 정보 38  
 참조 282  
 클러스터에 브로커 추가 199  
 클러스터에서 브로커 제거 200  
 passfile에서 169  
 imqbrokerd.conf 파일 68, 72  
 imqcmd 명령  
 구문 287  
 마스터 브로커에 종속 202  
 메트릭 모니터링 210  
 물리적 대상 관리 127  
 물리적 대상 하위 명령(표) 128  
 브로커에 보안 연결 165, 295

## J

- 영구 가입 하위 명령 122
- 옵션 294
- 정보 38
- 참조 287
- 트랜잭션 관리 123
- passfile에서 169
- imqConfiguredClientID 속성 181, 338
- imqConnectionFlowCount 속성 182, 339
- imqConnectionFlowLimit 속성 182, 339
- imqConnectionFlowLimitEnabled 속성 182, 339
- imqConsumerFlowLimit 속성 182, 340
- imqConsumerFlowThreshold 속성 182, 340
- imqdbmgr 명령
  - 구문 300
  - 옵션 301
  - 정보 39
  - 참조 300
  - 하위 명령 300
  - passfile에서 169
- imqDefaultPassword 속성 181, 338
- imqDefaultUsername 속성 181, 338
- imqDestinationDescription 속성 333
- imqDestinationName 속성 333
- imqDisableSetClientID 속성 181, 338
- imqFlowControlLimit 속성 182, 340
- imqJMSDeliveryMode 속성 184, 338
- imqJMSExpiration 속성 184, 338
- imqJMSPriority 속성 184, 338
- imqkeytool 명령
  - 명령 구문 161, 382
  - 사용 160, 382
  - 정보 38
  - 참조 306
- imqLoadMaxToServerSession 속성 183, 340
- imqobjmgr 명령
  - 구문 297
  - 옵션 298
  - 정보 38
  - 참조 297
  - 하위 명령 297
- imqOverrideJMSDeliveryMode 속성 184, 338
- imqOverrideJMSExpiration 속성 184, 338
- imqOverrideJMSHeadersToTemporaryDestinations 속성 184, 338
- imqOverrideJMSPriority 속성 184, 338
- imqQueueBrowserMax MessagesPerRetrieve 속성 183, 340
- imqQueueBrowserRetrieveTimeout 속성 183, 340
- imqReconnectAttempts 속성 335
- imqReconnectEnabled 속성 335
- imqReconnectInterval 속성 335
- imqSetJMSXAppID 속성 183, 341
- imqSetJMSXConsumerTXID 속성 183, 341
- imqSetJMSXProducerTXID 속성 183, 341
- imqSetJMSXRcvTimestamp 속성 183, 341
- imqSetJMSXUserID 속성 183, 341
- imqSSLIsHostTrusted 속성 335
- imqsvcadm 명령
  - 구문 304
  - 옵션 304
  - 정보 39
  - 참조 304
  - 하위 명령 304
- imqusermgr 명령
  - 구문 302
  - 비밀번호 146
  - 사용자 아이디 146
  - 옵션 302, 303
  - 용도 143
  - 정보 38
  - 참조 302
  - 하위 명령 302
- install.properties 파일 96

## J

- Java 가상 머신, JVM 참조
- Java runtime 281
- Windows 서비스용 70

java.naming.factory.initial 속성 174, 176  
 java.naming.provider.url 속성 174, 176  
 java.naming.security.authentication 속성 175  
 java.naming.security.credentials 속성 175  
 java.naming.security.principal 속성 175  
 javahome 옵션 70, 281  
 JCA (J2EE Connector Architecture) 343, 347  
 JDBC 지원  
   구성 99  
   드라이버 86, 99, 103, 318  
   설정 100  
   정보 86  
 JDK  
   경로 지정 283, 294, 298, 304  
 JMS 사양 29  
 jms 연결 서비스 76, 116  
 JMSDeliveryMode 메시지 헤더 필드 184  
 JMSEExpiration 메시지 헤더 필드 184  
 JMSPriority 메시지 헤더 필드 184  
 JNDI  
   객체 저장소 38, 174  
   객체 저장소 속성 174, 186  
   위치(공급자 URL) 174, 176  
   조회 54, 185  
   조회 이름 185, 190  
   초기 컨텍스트 174, 176  
 jrehome 옵션 70  
 JVM  
   메트릭, JVM 메트릭 참조  
   성능 영향 233  
   성능 조정 238  
 JVM 메트릭  
   메시지 기반 모니터링 사용 216  
   메트릭 개수 349  
   브로커 로그 파일 사용 209  
   imqcmd metrics 사용 212

## L

LDAP 서버  
   객체 저장소 속성 174  
   사용자 저장소 액세스 150  
   사용자 저장소로 149  
   인증 페일오버 150

## M

ManagedConnectionFactory JavaBean 345  
 MDB, Message-Driven Bean 참조  
 Message-Driven Bean  
   자원 어댑터 구성 343, 346  
 messageSelector 활성화 사양 속성 348

## N

NORMAL 서비스 유형 75  
 nsswitch.conf 파일(Linux) 198

## O

Oracle 100, 105

## P

passfile  
   명령줄 옵션 284  
   브로커 구성 등록 정보 91, 321  
   사용 169  
   위치 170, 360, 361, 362  
 password 관리 대상 연결 팩토리 속성 346  
 password 자원 어댑터 속성 344  
 PointBase 100

## R

## R

reconnectAttempts 관리 대상 연결 팩토리 속성 346  
reconnectAttempts 자원 어댑터 속성 345  
reconnectEnabled 관리 대상 연결 팩토리 속성 346  
reconnectEnabled 자원 어댑터 속성 345  
reconnectInterval 관리 대상 연결 팩토리 속성 346  
reconnectInterval 자원 어댑터 속성 345  
reset messages 옵션 135  
ResourceAdapter JavaBean 344  
RESTART 등록 정보 68

## S

Secure Socket Layer 표준, SSL 참조  
sendUndeliverableMsgsToDMQ 활성화  
사양 속성 348  
Simple Network Time Protocol 66  
SNTP 66  
SSL  
암호화 159  
연결 서비스, SSL 기반 연결 서비스 참조  
정보 90  
활성화 163  
TCP/IP에서 160  
SSL 기반 연결 서비스  
설정 159, 160  
시작 163  
ssladmin 연결 서비스  
설정 160  
정보 76, 116  
ssljms 연결 서비스  
설정 160  
정보 76, 116  
subscriptionDurability 활성화 사양 속성 347, 348  
subscriptionName 활성화 사양 속성 348  
Sun Cluster  
구성 317  
동기화 속성 85

syslog 93, 208

## T

TCP 76, 116  
TimeToLive 기능  
클릭 동기화 66  
TLS 76, 116

## U

ulimit 명령 66  
update dst 하위 명령  
제한 133  
userName 관리 대상 연결 팩토리 속성 346  
userName 자원 어댑터 속성 344

## W

W32Time 서비스 66  
Windows 서비스, 서비스(Windows) 참조

## X

xntpd 데몬 66