Sun Java System Message Queue 3.7 UR1 관리 설명서

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

부품 번호: 820–0845
2007년 1월

Copyright 2006 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 모든 권리는 저작권자의 소유입니다.

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

미국 정부의 권리 – 상용 소프트웨어정부 사용자는 Sun Microsystems, Inc.의 표준 사용권 계약과 해당 FAR 규정 및 보충 규정을 준수해야 합니다.

이 배포판에는 타사에서 개발한 자료가 포함되어 있을 수 있습니다.

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

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

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

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

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

070531@17466

목차

머리말 21

제1부 Message Queue 관리 소개 31

1 관리 작업 및 도구 33
관리 작업 33
개발 환경에서의 관리 33
작업 환경에서의 관리 34
관리 도구 35
명령줄 유틸리티 35
관리 콘솔 36

2 빠른 시작 자습서 39
관리 콘솔 시작 40
관리 콘솔 온라인 도움말 41
브로커 작업 43
브로커 시작 43
관리 콘솔에 브로커 추가 43
▼ 관리 콘솔에 브로커를 추가하는 방법 43
브로커에 연결 45
▼ 브로커에 연결하는 방법 45
연결 서비스 보기 46
▼ 사용 가능한 연결 서비스를 보는 방법 46
물리적 대상 작업 48
물리적 대상 만들기 48
▼ 브로커에 물리적 대상을 추가하는 방법 48
물리적 대상 등록 정보 보기 49

▼ 물리적 대상의 등록 정보를 보거나 수정하는 방법 50
물리적 대상에서 메시지 제거 52
▼ 물리적 대상에서 메시지를 제거하는 방법 52
물리적 대상 삭제 53
▼ 물리적 대상을 삭제하는 방법 53
객체 저장소 작업 53
객체 저장소 추가 53
▼ 객체 저장소를 관리 콘솔에 추가하는 방법 54
객체 저장소에 연결 56
▼ 객체 저장소에 연결하는 방법 56
관리 객체 작업 56
연결 팩토리 추가 56
▼ 객체 저장소에 연결 팩토리를 추가하는 방법 56
대상 추가 58
▼ 객체 저장소에 대상을 추가하는 방법 58
관리 대상 객체 등록 정보 보기 60
▼ 관리 객체의 등록 정보를 보거나 수정하는 방법 60
관리 대상 객체 삭제 61
▼ 관리 객체를 삭제하는 방법 61
샘플 응용 프로그램 실행 61
▼ 샘플 응용 프로그램을 실행하는 방법 62

제2부 관리 작업 65

3 브로커 및 클라이언트 시작 67
시스템 자원 준비 67
시스템 클럭 동기화 67
파일 설명자 제한 설정 68
브로커 시작 68
대화식으로 브로커 시작 68
자동으로 브로커 시작 69
브로커 제거 72
Solaris 또는 Linux에서 브로커 제거 72
Windows 브로커 서비스 제거 73
클라이언트 시작 73

4 브로커 구성 75
브로커 서비스 75
연결 서비스 76
라우팅 서비스 78
지속성 서비스 79
보안 서비스 82
모니터링 서비스 85
브로커 등록 정보 설정 88
구성 파일 88
영구 데이터 저장소 구성 91
파일 기반 저장소 구성 91
JDBC 기반 저장소 구성 91
▼ JDBC 기반 데이터 저장소를 구성하는 방법 92
영구 데이터 보안 93

5 브로커 관리 95
전제 조건 95
imqcmd 유틸리티 사용 96
도움말 표시 96
제품 버전 표시 96
사용자 이름 및 비밀번호 지정 97
브로커 이름 및 포트 지정 97
예 97
브로커 정보 표시 98
브로커 등록 정보 업데이트 99
브로커 일시 중지 및 다시 시작 100
브로커 일시 중지 100
브로커 다시 시작 101
브로커 종료 및 다시 시작 101
브로커 메트릭 표시 102
연결 서비스 관리 103
연결 서비스 나열 103
연결 서비스 정보 표시 104
연결 서비스 등록 정보 업데이트 105
연결 서비스 메트릭 표시 105

연결 서비스 일시 중지 및 다시 시작 106
연결 정보 얻기 107
영구 가입 관리 108
트랜잭션 관리 109

6 물리적 대상 관리 113
명령 유틸리티 사용 114
하위 명령 114
물리적 대상 만들기 115
물리적 대상 나열 116
물리적 대상 정보 표시 117
물리적 대상 등록 정보 업데이트 118
물리적 대상 일시 중지 및 다시 시작 119
물리적 대상 제거 120
물리적 대상 완전 삭제 120
물리적 대상 압축 121
물리적 대상 디스크 사용률 모니터링 121
사용 불능 메시지 대기열 사용 123
사용 불능 메시지 대기열 사용 구성 123
사용 불능 메시지 대기열 관리 124
사용 불능 메시지 로깅 사용 125

7 보안 관리 127
사용자 인증 127
플랫 파일 사용자 저장소 사용 128
사용자 저장소에 LDAP 서버 사용 133
사용자 권한 부여: 액세스 제어 등록 정보 파일 135
액세스 제어 등록 정보 파일 작성 135
액세스 규칙 구문 136
사용 권한 계산 방법 137
연결 서비스에 대한 액세스 제어 138
물리적 대상에 대한 액세스 제어 139
자동 생성 물리적 대상에 대한 액세스 제어 140
메시지 암호화 141
자체 서명된 인증서 사용 141

▼ 자체 서명된 인증서를 사용하여 SSL 기반 연결 서비스를 설정하는 방법 142
서명된 인증서 사용 147
▼ 서명된 인증서를 사용하는 방법 147
비밀번호 파일 150
보안 문제 150
비밀번호 파일 내용 150
방화벽을 통해 연결 151
▼ 방화벽을 통해 브로커 연결을 활성화하는 방법 152
감사 기록 152

8 관리 대상 객체 관리 153
객체 저장소 153
LDAP 서버 객체 저장소 153
파일 시스템 객체 저장소 155
관리 객체 속성 156
연결 팩토리 속성 156
대상 속성 162
객체 관리자 유틸리티 사용 163
관리 객체 추가 164
관리 객체 삭제 165
관리 객체 나열 166
관리 객체 정보 보기 166
관리 객체 속성 수정 167
명령 파일 사용 167

9 브로커 클러스터 작업 171
클러스터 구성 등록 정보 171
개별 브로커에 대한 클러스터 등록 정보 설정 172
클러스터 구성 파일 사용 172
클러스터 관리 173
브로커 연결 173
클러스터에 브로커 추가 174
▼ 클러스터 구성 파일을 사용하여 클러스터에 새 브로커를 추가하는 방법 174
클러스터에서 브로커 제거 175
마스터 브로커 176

구성 변경 기록 관리 176
▼ 구성 변경 기록을 복원하는 방법 177
마스터 브로커를 사용할 수 없는 경우 177

10 브로커 모니터링 179
모니터링 도구 소개 179
브로커 로깅 구성 및 사용 181
기본 로깅 구성 181
로그 메시지 형식 182
로거 구성 변경 182
▼ 브로커의 로거 구성을 변경하는 방법 182
메트릭을 대화식으로 표시 186
imqcmd metrics 186
메트릭 하위 명령을 사용하여 메트릭 데이터 표시 188
▼ 메트릭 하위 명령을 사용하는 방법 188
메트릭 출력: imqcmd metrics 188
imqcmd query 189
브로커를 모니터하는 응용 프로그램 작성 190
메시지 기반 모니터링 설정 191
▼ 메시지 기반 모니터링을 설정하는 방법 191
보안 및 액세스 고려 사항 192
메트릭 출력: 메트릭 메시지 192

11 메시지 서비스 분석 및 조정 193
성능 정보 193
성능 조정 프로세스 193
성능 요소 194
벤치마크 194
기본 사용 패턴 195
성능에 영향을 미치는 요소 196
▼ 메시지 전달 단계 197
성능에 영향을 미치는 응용 프로그램 설계 요소 198
성능에 영향을 미치는 메시지 서비스 요소 202
성능 향상을 위한 구성 조정 206
시스템 조정 206

브로커 조정 209
클라이언트 런타임 메시지 흐름 조정 210

12 문제 해결 213
클라이언트가 연결을 설정할 수 없음 213
연결 처리 속도가 너무 느림 217
클라이언트가 메시지 생성자를 만들 수 없음 218
메시지 생성이 지연되거나 느림 219
메시지가 백로그됨 222
브로커 처리량이 일정하지 않음 225
메시지가 사용자에게 도달하지 않음 226
사용 불능 메시지 대기열에 메시지가 포함되어 있음 230

제3부 참조 235

13 명령줄 참조 237
명령줄 구문 237
브로커 유틸리티 238
명령 유틸리티 242
브로커 관리 244
연결 서비스 관리 245
연결 관리 246
물리적 대상 관리 246
영구 가입 관리 248
트랜잭션 관리 249
일반 명령 유틸리티 옵션 249
객체 관리자 유틸리티 250
데이터베이스 관리자 유틸리티 251
사용자 관리자 유틸리티 253
서비스 관리자 유틸리티 254
키 도구 유틸리티 255

14 브로커 등록 정보 참조 257
연결 등록 정보 257

라우팅 등록 정보 259
지속성 등록 정보 263
파일 기반 지속성 263
JDBC 기반 지속성 265
보안 등록 정보 268
모니터링 등록 정보 271
클러스터 구성 등록 정보 275
알파벳 순 브로커 등록 정보 목록 276

15 물리적 대상 등록 정보 참조 283
물리적 대상 등록 정보 283

16 관리 객체 속성 참조 287
연결 팩토리 속성 287
연결 처리 287
클라이언트 아이디 291
안정성 및 흐름 제어 291
대기열 브라우저 및 서버 세션 292
표준 메시지 등록 정보 293
메시지 헤더 대체 294
대상 속성 294
SOAP 종점 속성 295

17 JMS 자원 어댑터 등록 정보 참조 297
ResourceAdapter JavaBean 297
ManagedConnectionFactory JavaBean 299
ActivationSpec JavaBean 300

18 메트릭 참조 303
JVM 메트릭 303
브로커 전체 메트릭 304
연결 서비스 메트릭 305
대상 메트릭 307

제4부 부록 313

A 플랫폼별 Message Queue 데이터 위치 315
Solaris 315
Linux 316
Windows 317

B Message Queue 인터페이스의 안정성 319
분류 체계 319
인터페이스 안정성 320

C HTTP/HTTPS 지원 323
HTTP/HTTPS 지원 구조 323
HTTP 지원 활성화 325
▼ HTTP 지원을 활성화하는 방법 325
1단계. HTTP 터널 서블릿 배포 325
2단계. httpjms 연결 서비스 구성 328
▼ httpjms 연결 서비스를 활성화하는 방법 329
3단계. HTTP 연결 구성 330
HTTPS 지원 활성화 332
▼ HTTPS 지원을 활성화하는 방법 332
1단계. HTTPS 터널 서블릿에 대해 자체 서명된 인증서 생성 332
2단계. HTTP 터널 서블릿 .war 파일의 설명자 파일 수정 333
▼ HTTPS 터널 서블릿 .war 파일을 수정하는 방법 333
3단계. HTTP 터널 서블릿 배포 334
4단계. httpsjms 연결 서비스 구성 337
▼ httpsjms 연결 서비스를 활성화하는 방법 338
5단계. HTTPS 연결 구성 339
문제 해결 341
서버 또는 브로커 오류 342
터널 서블릿을 통한 클라이언트 연결 오류 342
▼ 클라이언트가 연결되지 않는 경우 342

D 자주 사용하는 명령 유틸리티 명령 343
구문 343
브로커 및 클러스터 관리 343
브로커 구성 등록 정보(-o 옵션) 344
서비스 및 연결 관리 344
영구 가입자 관리 345
트랜잭션 관리 345
대상 관리 345
대상 구성 등록 정보(-o 옵션) 345
메트릭 346

색인 347

그림

그림 1–1 로컬 및 원격 관리 유틸리티 37
그림 2–1 관리 콘솔 창 41
그림 2–2 관리 콘솔 도움말 창 42
그림 2–3 브로커 추가 대화 상자 44
그림 2–4 관리 콘솔 창에 표시된 브로커 45
그림 2–5 브로커에 연결 대화 상자 46
그림 2–6 연결 서비스 보기 47
그림 2–7 서비스 등록 정보 대화 상자 47
그림 2–8 브로커 대상 추가 대화 상자 49
그림 2–9 브로커 대상 등록 정보 대화 상자 51
그림 2–10 영구 가입 패널 52
그림 2–11 객체 저장소 추가 대화 상자 54
그림 2–12 관리 콘솔 창에 표시된 객체 저장소 55
그림 2–13 연결 팩토리 객체 추가 대화 상자 57
그림 2–14 대상 객체 추가 대화 상자 59
그림 2–15 관리 콘솔 창에 표시된 대상 객체 60
그림 4–1 영구 데이터 저장소 80
그림 4–2 보안 지원 83
그림 4–3 모니터링 지원 86
그림 4–4 브로커 구성 파일 89
그림 11–1 Message Queue 서비스를 통한 메시지 전달 197
그림 11–2 전송 프로토콜 속도 204
그림 C–1 HTTP/HTTPS 지원 구조 324

표

표 4–1 Message Queue 연결 서비스 76
표 4–2 메트릭 주제 대상 87
표 5–1 Message Queue 연결 서비스 103
표 5–2 imqcmd에서 업데이트하는 연결 서비스 등록 정보 105
표 6–1 명령 유틸리티의 물리적 대상 하위 명령 114
표 6–2 물리적 대상 디스크 사용률 메트릭 122
표 6–3 사용 불능 메시지 대기열에서 처리하는 표준 물리적 대상 등록 정보 124
표 7–1 사용자 저장소 초기 항목 128
표 7–2 imqusermgr 옵션 130
표 7–3 액세스 규칙의 구문 요소 136
표 7–4 물리적 대상 액세스 제어 규칙의 요소 140
표 7–5 자체 서명된 인증서에 필요한 고유 이름 정보 143
표 7–6 비밀번호를 사용하는 명령 150
표 7–7 비밀번호 파일의 비밀번호 151
표 7–8 정적 포트 주소에 대한 브로커 구성 등록 정보 151
표 8–1 LDAP 객체 저장소 속성 154
표 8–2 파일 시스템 객체 저장소 속성 155
표 10–1 메트릭 모니터링 도구의 장점 및 제한 180
표 10–2 로깅 수준 182
표 10–3 imqcmd metrics 하위 명령 구문 186
표 10–4 imqcmd metrics 하위 명령 옵션 187
표 10–5 imqcmd query 하위 명령 구문 190
표 10–6 메트릭 주제 대상 190
표 11–1 높은 안정성 및 높은 성능 시나리오 비교 199
표 13–1 브로커 유틸리티 옵션 238
표 13–2 브로커 관리에 대한 명령 유틸리티 하위 명령 244
표 13–3 연결 서비스 관리에 대한 명령 유틸리티 하위 명령 245
표 13–4 연결 서비스 관리에 대한 명령 유틸리티 하위 명령 246

표 13–5 물리적 대상 관리에 대한 명령 유틸리티 하위 명령 247
표 13–6 영구 가입 관리에 대한 명령 유틸리티 하위 명령 248
표 13–7 트랜잭션 관리에 대한 명령 유틸리티 하위 명령 249
표 13–8 일반 명령 유틸리티 옵션 249
표 13–9 객체 관리자 하위 명령 250
표 13–10 객체 관리자 옵션 250
표 13–11 데이터베이스 관리자 하위 명령 252
표 13–12 데이터베이스 관리자 옵션 252
표 13–13 사용자 관리자 하위 명령 253
표 13–14 일반 사용자 관리자 옵션 254
표 13–15 서비스 관리자 하위 명령 254
표 13–16 서비스 관리자 옵션 254
표 14–1 브로커 연결 등록 정보 257
표 14–2 브로커 라우팅 등록 정보 259
표 14–3 자동 생성된 대상에 대한 브로커 등록 정보 260
표 14–4 전역 브로커 지속성 등록 정보 263
표 14–5 파일 기반 지속성에 대한 브로커 등록 정보 264
표 14–6 JDBC 기반 지속성에 대한 브로커 등록 정보 265
표 14–7 브로커 보안 등록 정보 268
표 14–8 브로커 모니터링 등록 정보 272
표 14–9 클러스터 구성에 대한 브로커 등록 정보 275
표 14–10 알파벳 순 브로커 등록 정보 목록 277
표 15–1 물리적 대상 등록 정보 283
표 16–1 연결 처리에 대한 연결 팩토리 속성 288
표 16–2 메시지 브로커 주소 지정 체계 289
표 16–3 메시지 브로커 주소 예 290
표 16–4 클라이언트 식별에 대한연결 팩토리 속성 291
표 16–5 안정성 및 흐름 제어에 대한 연결 팩토리 속성 291
표 16–6 대기열 브라우저 및 서버 세션에 대한 연결 팩토리 속성 293
표 16–7 표준 메시지 등록 정보에 대한 연결 팩토리 속성 293
표 16–8 메시지 헤더 대체에 대한 연결 팩토리 속성 294
표 16–9 대상 속성 295
표 16–10 SOAP 종점 속성 295
표 17–1 자원 어댑터 등록 정보 298
표 17–2 관리 대상 연결 팩토리 등록 정보 299
표 17–3 활성화 사양 등록 정보 300

표 18–1 JVM 메트릭 303
표 18–2 브로커 전체 메트릭 304
표 18–3 연결 서비스 메트릭 305
표 18–4 대상 메트릭 307
표 A–1 Solaris 플랫폼의 Message Queue 데이터 위치 315
표 A–2 Linux 플랫폼의 Message Queue 데이터 위치 316
표 A–3 Windows 플랫폼의 Message Queue 데이터 위치 317
표 B–1 인터페이스 안정성 분류 체계 319
표 B–2 Message Queue 인터페이스의 안정성 320
표 D–1 브로커 구성 등록 정보(-o 옵션) 344
표 D–2 대상 구성 등록 정보(-o 옵션) 345

코드 예

예 3–1 브로커 서비스 시작 옵션 표시 72
예 8–1 연결 팩토리 추가 164
예 8–2 LDAP 객체 저장소에 대상 추가 165
예 8–3 파일 시스템 객체 저장소에 대상 추가 165
예 8–4 관리 객체 삭제 165
예 8–5 모든 관리 객체 나열 166
예 8–6 특정 유형의 관리 객체 나열 166
예 8–7 관리 객체 정보 보기 166
예 8–8 관리 대상 객체 속성 수정 167
예 8–9 객체 관리자 명령 파일 구문 167
예 8–10 명령 파일 예 168
예 8–11 부분 명령 파일 168
예 8–12 부분 명령 파일 사용 169

머리말

본 Sun Java System™ **Message Queue 관리 설명서**에서는 시스템 관리자가 Sun Java System Message Queue™ 메시징 시스템을 설정하고 관리하는 데 필요한 배경과 정보를 제공합니다.

대상

본 설명서는 Message Queue 관리 작업을 수행해야 하는 관리자와 응용 프로그램 개발자를 대상으로 합니다. Message Queue **관리자**는 Message Queue 메시징 시스템, 특히 이 시스템의 핵심인 메시지 브로커를 설정하고 관리합니다.

알아야 할 사항

본 설명서를 읽기 전에 **Sun Java System Message Queue 3.7 UR1 기술 개요**를 읽고 Java Message Service 사양의 Message Queue 구현, Message Queue 서비스 구성 요소 및 Message Queue 응용 프로그램의 개발, 배포, 관리의 기본 프로세스에 대해 숙지하고 있어야 합니다.

구성

표 P–1에서는 본 설명서의 내용을 간략하게 설명합니다.

표 P–1 본 설명서의 내용

부/장	설명
I부, Message Queue 관리 소개	
1장, 관리 작업 및 도구	Message Queue 관리 작업 및 도구에 대해 소개합니다.
2장, 빠른 시작 자습서	Message Queue 관리 콘솔을 익힐 수 있는 실습 자습서를 제공합니다.

표 P–1 본 설명서의 내용 (계속)

부/장	설명
II부, 관리 작업	
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.
6장, 물리적 대상 관리	물리적 대상과 관련된 관리 작업을 설명합니다.
7장, 보안 관리	비밀번호 파일 관리, 인증, 권한 부여, 암호화 등과 같은 보안 관련 작업에 대해 설명합니다.
8장, 관리 대상 객체 관리	객체 저장소를 설명하고 관리 객체(연결 팩토리와 대상)와 관련된 작업을 수행하는 방법을 보여줍니다.
9장, 브로커 클러스터 작업	Message Queue 브로커 클러스터 설정 및 관리 방법에 대해 설명합니다.
10장, 브로커 모니터링	Message Queue 모니터링 기능 설정 및 사용 방법에 대해 설명합니다.
11장, 메시지 서비스 분석 및 조정	메시지 서비스 성능을 분석 및 최적화하는 기술을 설명합니다.
12장, 문제 해결	일반적인 Message Queue 문제와 문제 해결을 위해 수행해야 할 조치를 결정하는 데 도움이 되는 제안 사항을 제공합니다.
III부, 참조	
13장, 명령줄 참조	Message Queue 명령줄 유틸리티에 대한 구문과 설명을 제공합니다.
14장, 브로커 등록 정보 참조	Message Queue 메시지 브로커의 구성 등록 정보를 설명합니다.
15장, 물리적 대상 등록 정보 참조	물리적 대상의 구성 등록 정보를 설명합니다.
16장, 관리 객체 속성 참조	관리 대상 객체(연결 팩토리 및 대상)의 구성 등록 정보를 설명합니다.
17장, JMS 자원 어댑터 등록 정보 참조	응용 프로그램 서버에서 사용할 Message Queue 자원 어댑터의 구성 등록 정보를 설명합니다.
18장, 메트릭 참조	Message Queue 메시지 브로커가 모니터링, 설정 및 진단을 위해 제공할 수 있는 메트릭 정보에 대해 설명합니다.
IV부, 부록	
부록 A, 플랫폼별 Message Queue 데이터 위치	지원되는 각 플랫폼에서 Message Queue 파일의 위치를 나열합니다.

표 P–1 본 설명서의 내용 *(계속)*

부/장	설명
부록 B, Message Queue 인터페이스의 안정성	다양한 Message Queue 인터페이스의 안정성에 대해 설명합니다.
부록 C, HTTP/HTTPS 지원	Message Queue 통신을 위해 HTTP(Hypertext Transfer Protocol)를 설정 및 사용하는 방법에 대해 설명합니다.
부록 D, 자주 사용하는 명령 유틸리티 명령	자주 사용하는 Message Queue 명령 유틸리티(`imqcmd`) 명령을 나열합니다.

관련 문서

Message Queue 설명서 세트를 구성하는 문서는 다음 표에서 일반적으로 사용되는 순서에 따라 나열되어 있습니다.

표 P–2 Message Queue 설명서 세트

문서	대상	설명
Sun Java System Message Queue 3.7 UR1 Installation Guide	개발자와 관리자	Solaris, Linux, Windows 플랫폼에서 Message Queue 소프트웨어를 설치하는 방법을 설명합니다.
Sun Java System Message Queue 3.7 UR1 릴리스 노트	개발자와 관리자	새로운 기능, 제한, 알려진 버그 및 기술 노트에 관한 설명이 포함되어 있습니다.
Sun Java System Message Queue 3.7 UR1 기술 개요	개발자와 관리자	Message Queue 개념, 기능 및 구성 요소에 대해 설명합니다.
Sun Java System Message Queue 3.7 UR1 Developer's Guide for Java Clients	개발자	JMS 또는 SOAP/JAXM API의 Message Queue 구현을 사용하는 Java 클라이언트 프로그램 개발자를 위한 빠른 시작 자습서와 프로그래밍 정보를 제공합니다.
Sun Java System Message Queue 3.7 UR1 관리 설명서	관리자, 개발자에게도 권장	Message Queue 관리 도구를 사용한 관리 작업 수행 시 필요한 배경 및 정보를 제공합니다.
Sun Java System Message Queue 3.7 UR1 Developer's Guide for C Clients	개발자	JMS API(C-API)의 Message Queue C 구현을 사용하는 C 클라이언트 프로그램 개발자를 위한 프로그래밍 및 참조 설명서를 제공합니다.

온라인 도움말

Message Queue 3.7 UR1은 Message Queue 메시지 서비스 관리 작업을 수행하는 명령줄 유틸리티를 포함합니다.

또한 Message Queue 3.7 UR1에는 그래픽 사용자 인터페이스(GUI) 관리 도구인 관리 콘솔(`imqadmin`)이 포함되어 있습니다. 상황에 맞는 도움말은 관리 콘솔에 포함되어 있습니다. 41 페이지 "관리 콘솔 온라인 도움말"을 참조하십시오.

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`

본 설명서는 모든 HTML 브라우저로 볼 수 있습니다. 표준 JMS API 설명서 및 Message Queue 관리 대상 객체에 대한 Message Queue 고유의 API를 포함하며, 이는 메시징 응용 프로그램 개발자에게 도움이 됩니다.

클라이언트 응용 프로그램 예

Message Queue는 개발자에게 도움이 되는 많은 클라이언트 응용 프로그램 예를 제공합니다.

Java 클라이언트 응용 프로그램 예

Java 클라이언트 응용 프로그램 예는 플랫폼에 따라 다음과 같은 위치에 있습니다. 해당 디렉토리와 각 하위 디렉토리에 있는 `README` 파일을 참조하십시오.

플랫폼	위치
Solaris	`/usr/demo/imq/`
Linux	`/opt/sun/mq/examples`
Windows	`IMQ_HOME/demo/`

C 클라이언트 프로그램 예

C 클라이언트 응용 프로그램 예는 플랫폼에 따라 다음과 같은 위치에 있습니다. 해당 디렉토리에 있는 README 파일을 참조하십시오.

플랫폼	위치
Solaris	/opt/SUNWimq/demo/C/
Linux	/opt/sun/mq/examples/C/
Windows	IMQ_HOME/demo/C/

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.

디렉토리 변수 규칙

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

주 – 표 P–3에 나오는 정보는 Message Queue의 독립 실행형 설치에만 적용됩니다. Message Queue가 Application Server 설치의 일부로 설치되어 실행되는 경우 디렉토리 변수 값이 다르게 설정됩니다. IMQ_HOME은 *appServer_install_dir*/imq(여기서 *appServer_install_dir*은 Application Server 설치 디렉토리)로 설정되고, IMQ_VARHOME은 *appServer_domainName_dir*/imq(여기서 *appServer_domainName_dir*은 Message Queue 브로커를 시작하는 도메인의 도메인 디렉토리)로 설정됩니다.

표 P–3 디렉토리 변수 규칙

변수	설명
IMQ_HOME	Message Queue 설명서에서 Message Queue 기본 디렉토리(루트 설치 디렉토리)를 참조할 때 사용됩니다. ■ Solaris 및 Linux에는 Message Queue 루트 설치 디렉토리가 없습니다. 따라서 Message Queue 설명서에서 Solaris 및 Linux의 파일 위치를 참조하는 경우에는 `IMQ_HOME`이 사용되지 않습니다. ■ Windows에서 루트 Message Queue 설치 디렉토리는 Message Queue 번들의 압축을 푸는 디렉토리로 설정됩니다.
IMQ_VARHOME	임시 또는 동적으로 작성된 Message Queue 구성 및 데이터 파일이 저장되는 `/var` 디렉토리입니다. 모든 디렉토리를 가리키는 환경 변수로 설정할 수 있습니다. ■ Solaris에서 `IMQ_VARHOME`의 기본값은 `/var/imq` 디렉토리입니다. ■ Solaris에서 Sun Java System Application Server 평가판의 `IMQ_VARHOME` 기본값은 `IMQ_HOME/var` 디렉토리입니다. ■ Linux에서 `IMQ_VARHOME`의 기본값은 `/var/opt/sun/mq` 디렉토리입니다. ■ Windows에서 `IMQ_VARHOME`의 기본값은 `IMQ_HOME\\var` 디렉토리입니다.

표 P–3 디렉토리 변수 규칙 *(계속)*

변수	설명
IMQ_JAVAHOME	Message Queue 실행 파일에 필요한 Java Runtime(JRE)의 위치를 가리키는 환경 변수입니다. ■ Solaris에서 IMQ_JAVAHOME은 다음과 같은 순서로 Java Runtime을 찾지만, 선택적으로 사용자는 필요한 JRE가 있는 적절한 위치로 값을 설정할 수 있습니다(Solaris 8 또는 9). /usr/jdk/entsys-j2se /usr/jdk/latest /usr/jdk/jdk1.5.* /usr/jdk/j2sdk1.5.* /usr/j2se Solaris 10: /usr/jdk/entsys-j2se /usr/jdk/java /usr/jdk/latest /usr/j2se ■ Linux에서 Message Queue가 먼저 다음 순서로 Java Runtime을 찾지만, 선택적으로 사용자는 IMQ_JAVAHOME의 값을 필요한 JRE가 있는 적절한 위치로 설정할 수 있습니다. /usr/jdk/entsys-j2se /usr/java/jre1.5. * /usr/java/jdk1.5. * /usr/java/jre1.4.2* /usr/java/j2sdk1.4.2* ■ Windows에서 IMQ_JAVAHOME은 지원되는 버전이 시스템에 있는 경우 기존 Java Runtime을 가리키도록 설정됩니다. 지원되는 버전이 없는 경우 새로운 버전이 설치됩니다.

본 설명서에서 IMQ_HOME, IMQ_VARHOME 및 IMQ_JAVAHOME은 플랫폼별 환경 변수 표시나 구문(예: UNIX의 $IMQ_HOME) **없이** 표시됩니다. 경로 이름에는 일반적으로 UNIX 디렉토리 구분자 표시(/)를 사용합니다.

표기 규칙

다음 표에서는 본 설명서에 사용되는 표기 규칙에 대해 설명합니다.

표 P–4 표기 규칙

서체	의미	예
`AaBbCc123`	명령, 파일 및 디렉토리의 이름, 그리고 컴퓨터 화면에 출력되는 내용입니다.	`.login` 파일을 편집합니다. `ls -a`를 사용하여 모든 파일을 나열합니다. `machine_name% you have mail.`
`AaBbCc123`	컴퓨터의 화면상 출력과는 달리 사용자가 직접 입력하는 사항입니다.	`machine_name%` **`su`** `Password:`
aabbcc123	위치 표시자: 실제 이름이나 값으로 대체됩니다.	파일 제거 명령은 `rm` *filename*입니다.
AaBbCc123	책 제목, 새로운 용어 및 강조 표시할 단어에 사용됩니다.	**사용 설명서**의 6장을 참조하십시오. *cache*는 로컬로 저장되는 복사본입니다. 파일을 저장하지 **마십시오**. **참고:**강조 표시된 일부 항목은 온라인에서 볼드로 표시됩니다.

명령 예의 쉘 프롬프트

C 쉘, Bourne 쉘, Korn 쉘 및 Windows 운영 체제에 대한 기본 UNIX® 시스템 프롬프트 및 수퍼유저 프롬프트는 다음 표와 같습니다.

표 P–5 쉘 프롬프트

쉘	프롬프트
C 쉘	`machine_name%`
수퍼유저용 C 쉘	`machine_name#`
Bourne 쉘 및 Korn 쉘	`$`
수퍼유저용 Bourne 쉘 및 Korn 쉘	`#`
Windows	C:\

기호 규칙

다음 표에서는 본 설명서에 사용될 수 있는 기호에 대해 설명합니다.

표 P–6 기호 규칙

기호	설명	예	의미
`[ ]`	선택 인수와 명령 옵션을 포함합니다.	`ls [-l]`	`-l` 옵션은 필수 항목이 아닙니다.
`{ | }`	필수 명령 옵션에 대한 일련의 선택 항목을 포함합니다.	`-d {y|n}`	`-d` 옵션에서는 `y` 인수 또는 `n` 인수를 사용해야 합니다.
`${ }`	변수 참조를 나타냅니다.	`${com.sun.javaRoot}`	`com.sun.javaRoot` 변수의 값을 참조합니다.
-	동시에 입력하는 여러 키를 결합합니다.	Ctrl-A	Ctrl 키를 누른 상태에서 A 키를 누릅니다.
+	연속해서 입력하는 여러 키를 결합합니다.	Ctrl+A+N	Ctrl 키를 눌렀다가 놓은 다음 후속 키들을 누릅니다.
→	그래픽 사용자 인터페이스의 메뉴 항목 선택을 나타냅니다.	파일 → 새로 만들기 → 템플리트	파일 메뉴에서 새로 만들기를 선택합니다. 새로 만들기 하위 메뉴에서 템플리트를 선택합니다.

설명서, 지원 및 교육

Sun 웹 사이트에서는 다음과 같은 추가 자원에 대한 정보를 제공합니다.

- 설명서(`http://www.sun.com/documentation/`)
- 지원(`http://www.sun.com/support/`)
- 교육(`http://www.sun.com/training/`)

Sun 제품 설명서 검색

docs.sun.com 웹 사이트에서 Sun 제품 설명서를 검색하는 이외에, 검색 필드에 다음 구문을 입력하여 검색 엔진을 사용할 수 있습니다.

search-term `site:docs.sun.com`

예를 들어, "브로커"를 검색하려면 다음과 같이 입력합니다.

```
broker site:docs.sun.com
```

Sun 웹 사이트(예: java.sun.com, www.sun.com 및 developers.sun.com)를 검색에 포함시키려면 검색 필드에서 "docs.sun.com" 대신 "sun.com"를 사용합니다.

타사 웹 사이트 참조

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

주 – Sun은 이 문서에 언급된 타사 웹 사이트의 사용 가능성에 대해 책임지지 않습니다. Sun은 해당 사이트 또는 자원에 있거나 사용 가능한 내용, 광고, 제품 또는 기타 자료에 대해 보증하지 않으며 책임 또는 의무를 지지 않습니다. Sun은 해당 사이트 또는 자원을 통해 사용 가능한 내용, 제품 또는 서비스의 사용과 관련해 발생하거나 발생했다고 간주되는 손해나 손실에 대해 책임이나 의무를 지지 않습니다.

사용자 의견 환영

Sun은 본 설명서의 개선을 위해 지속적으로 노력하고 있으며 고객의 의견과 제안을 환영합니다. 사용자 의견을 보내시려면 http://docs.sun.com에서 의견 보내기를 누릅니다. 온라인 양식에 전체 설명서 제목과 부품 번호를 입력합니다. 부품 번호는 해당 설명서의 제목 페이지나 문서의 URL에 있는 7자리 또는 9자리 숫자입니다. 예를 들어, 본 설명서의 부품 번호는 820-0845입니다.

1

Message Queue 관리 소개

- 1 장
- 2 장

◆ ◆ ◆ 1 장

1 관리 작업 및 도구

이 장에서는 명령줄 관리 유틸리티의 일반적인 기능에 중점을 두고 Sun Java™ System Message Queue™ 관리 작업과, 이 관리 작업을 수행하는 데 사용되는 도구에 대한 개요를 제공합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 33 페이지 "관리 작업"
- 35 페이지 "관리 도구"

관리 작업

수행할 일반적인 관리 작업은 Message Queue를 실행 중인 환경에 따라 다릅니다. Message Queue 응용 프로그램이 개발 및 테스트 중인 소프트웨어 개발 환경의 요구 사항은 유용한 작업을 수행하기 위해 해당 응용 프로그램이 배포되는 작업 환경의 요구 사항과는 다릅니다. 다음 절에서는 서로 다른 두 가지 환경 유형의 일반적인 관리 요구 사항을 간략하게 설명합니다.

개발 환경에서의 관리

개발 환경에서는 유연성을 강조합니다. 주로 Message Queue 메시지 서비스는 개발 시 응용 프로그램을 테스트하는 데 필요합니다. 흔히 프로그래머는 고유의 시스템을 관리하기 때문에 일반적으로 관리 작업이 최소화됩니다. 이와 같은 환경은 일반적으로 다음 특징을 통해 구별됩니다.

- 테스트에 사용하기 위해 브로커의 시작이 간단합니다.
- 관리 객체가 관리적인 방식으로 생성되기 보다는 클라이언트 코드로 인스턴스화됩니다.
- 자동 생성 대상
- 파일 시스템 객체 저장소
- 파일 기반 지속성

- 파일 기반 사용자 저장소
- 다중 브로커 클러스터에 마스터 브로커가 없습니다.

작업 환경에서의 관리

응용 프로그램을 안정적으로 배포하고 실행해야 하는 작업 환경에서는 관리가 훨씬 더 중요합니다. 수행할 관리 작업은 메시징 시스템 및 이 시스템에서 지원해야 하는 응용 프로그램의 복잡성에 따라 달라집니다. 이러한 작업은 설정 작업과 유지 보수 작업의 두 가지 일반적인 범주로 분류됩니다.

설정 작업

작업 환경에서의 관리 설정 작업은 일반적으로 다음 작업 전체 또는 일부를 포함합니다.

관리자 보안

- 기본 관리자(`admin`)의 비밀번호 설정(132 페이지 "기본 관리자 비밀번호 변경")
- 관리 연결 서비스(138 페이지 "연결 서비스에 대한 액세스 제어") 및 사용 불능 메시지 대기열(139 페이지 "물리적 대상에 대한 액세스 제어")에 대한 개인 또는 그룹 액세스 제어
- 파일 기반 또는 LDAP(Lightweight Directory Access Protocol) 사용자 저장소에 대한 관리 그룹 액세스 규제(130 페이지 "그룹", 134 페이지 "관리자에 대한 액세스 제어 설정")

일반 보안

- 파일 기반 사용자 저장소 내용 관리(131 페이지 "사용자 저장소 채우기 및 관리") 또는 기존 LDAP 사용자 저장소를 사용하도록 브로커 구성(133 페이지 "인스턴스 구성 파일 편집")
- 개별 사용자나 그룹에 수행할 수 있는 권한이 부여된 작업 제어(135 페이지 "사용자 권한 부여: 액세스 제어 등록 정보 파일")
- SSL(Secure Socket Layer)을 사용하는 암호화 서비스 설정(141 페이지 "메시지 암호화")

관리 대상 객체

- LDAP 객체 저장소 설정 및 구성(153 페이지 "LDAP 서버 객체 저장소")
- 연결 팩토리 및 대상 작성(164 페이지 "관리 객체 추가")

브로커 클러스터

- 클러스터 구성 파일 작성(172 페이지 "클러스터 구성 파일 사용")
- 마스터 브로커 지정(176 페이지 "마스터 브로커")

지속성

- 영구 저장소를 사용하도록 브로커 구성(91 페이지 "영구 데이터 저장소 구성").

메모리 관리

- 메모리 사용을 최적화하도록 대상의 구성 등록 정보 설정(15 장, 118 페이지 “물리적 대상 등록 정보 업데이트”)

유지 관리 작업

작업 환경에서는 응용 프로그램 성능, 안정성 및 보안이 중요하므로 메시지 서비스 자원이 다음을 비롯하여 진행 중인 관리 유지 보수 작업을 통해 철저하게 모니터링되고 제어되어야 합니다.

브로커 관리 및 조정

- 브로커 메트릭을 사용하여 브로커 조정 및 재구성(11 장)
- 브로커 메모리 자원 관리(78 페이지 “라우팅 서비스”)
- 브로커 클러스터를 작성 및 관리하여 메시지 로드 균형 조정(9 장)
- 오류가 발생한 브로커 복구(68 페이지 “브로커 시작”)

관리 대상 객체

- 클라이언트 응용 프로그램이 올바르게 작동하도록 연결 팩토리 속성 조정(156 페이지 “연결 팩토리 속성”)
- 물리적 대상 모니터링 및 관리(6 장)
- 대상에 대한 사용자 액세스 제어(139 페이지 “물리적 대상에 대한 액세스 제어”)

클라이언트 관리

- 영구 가입 모니터링 및 관리(108 페이지 “영구 가입 관리” 참조)
- 트랜잭션 모니터링 및 관리(109 페이지 “트랜잭션 관리” 참조)

관리 도구

Message Queue 관리 도구는 다음 두 범주로 구분됩니다.

- 명령줄 유틸리티
- 그래픽 관리 콘솔

명령줄 유틸리티

모든 Message Queue 유틸리티는 명령줄 인터페이스에서 액세스할 수 있습니다. 유틸리티 명령은 일반 형식, 구문 규칙 및 옵션을 공유합니다. 여기에는 다음 항목이 포함됩니다.

- **브로커 유틸리티**(`imqbrokerd`)는 브로커를 시작하고 여러 브로커를 클러스터로 연결하는 등 브로커 구성 등록 정보를 지정합니다.
- **명령 유틸리티**(`imqcmd`)는 브로커와 자원을 제어하고 물리적 대상을 관리합니다.

- **객체 관리자 유틸리티**(`imqobjmgr`)는 JNDI(Java Naming and Directory Interface)를 통해 액세스할 수 있는 객체 저장소의 공급자와 독립적인 **관리 대상 객체**를 관리합니다.
- **데이터베이스 관리자 유틸리티**(`imqdbmgr`)는 JNDI(Java Database Connectivity) 표준을 준수하는 영구 저장소의 데이터베이스를 작성 및 관리합니다.
- **사용자 관리자 유틸리티**(`imqusermgr`)는 사용자 인증 및 권한 부여를 위해 파일 기반 사용자 저장소를 채웁니다.
- **서비스 관리자 유틸리티**(`imqsvcadmin`)에서는 브로커를 Windows 서비스로 설치 및 관리합니다.
- **키 도구 유틸리티**(`imqkeytool`)는 SSL(Secure Socket Layer) 인증을 위해 자체 서명된 인증서를 생성합니다.

이러한 유틸리티 사용에 대한 자세한 내용은 13 장을 참조하십시오.

관리 콘솔

Message Queue **관리 콘솔**은 명령 및 객체 관리자 유틸리티의 기능 중 일부를 결합합니다. 관리 콘솔을 사용하면 다음 작업을 수행할 수 있습니다.

- 원격으로 브로커에 연결 및 제어
- 물리적 대상 작성 및 관리
- JNDI 객체 저장소의 관리 객체 작성 및 관리

하지만 관리 콘솔을 사용하여 브로커 시작, 브로커 클러스터 작성, JDBC 데이터베이스 또는 사용자 저장소 관리, 브로커를 Windows 서비스로 설치 또는 SSL 인증서 생성과 같은 작업을 수행할 수는 없습니다. 이러한 작업을 수행하려면 다른 명령줄 유틸리티(브로커, 데이터베이스 관리자, 사용자 관리자, 서비스 관리자 및 키 도구)가 필요합니다. 이러한 유틸리티는 원격으로 작동할 수 없으며 관리하는 브로커와 동일한 호스트에서 실행되어야 합니다(그림 1–1 참조).

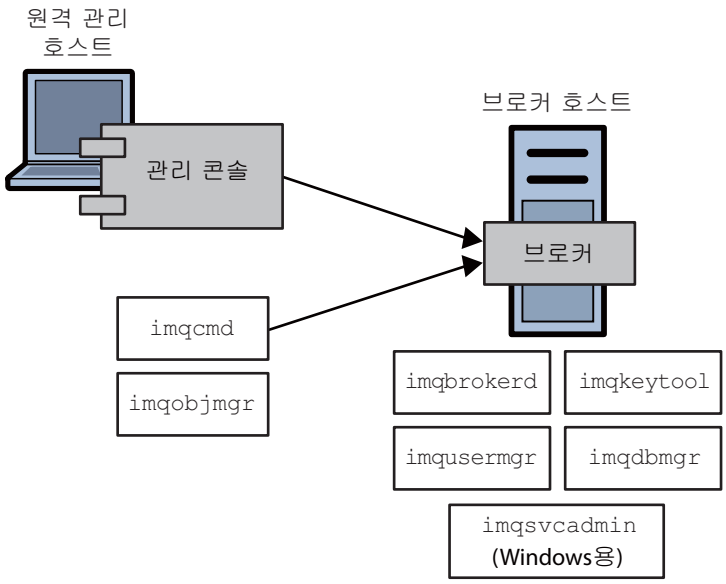


그림 1–1 로컬 및 원격 관리 유틸리티

관리 콘솔에 대한 간단한 소개를 보려면 2 장을 참조하십시오. 사용에 대한 자세한 내용은 고유의 도움말 기능을 통해 참조할 수 있습니다.

◆ ◆ ◆ 2 장

2 빠른 시작 자습서

이 빠른 시작 자습서는 메시지 브로커 및 객체 저장소를 관리하기 위한 그래픽 인터페이스인 Message Queue™ 관리 콘솔을 사용하여 일부 기본적인 관리 작업을 안내함으로써 Message Queue 관리를 간략하게 소개합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 40 페이지 "관리 콘솔 시작"
- 41 페이지 "관리 콘솔 온라인 도움말"
- 43 페이지 "브로커 작업"
- 48 페이지 "물리적 대상 작업"
- 53 페이지 "객체 저장소 작업"
- 56 페이지 "관리 객체 작업"
- 61 페이지 "샘플 응용 프로그램 실행"

이 자습서는 간단한 JMS 호환 응용 프로그램인 `HelloWorldMessageJNDI`를 실행하는 데 필요한 물리적 대상과 관리 대상 객체를 설정합니다. 응용 프로그램은 예제 응용 프로그램 디렉토리(Solaris 및 Windows 플랫폼의 경우 `demo`, Linux의 경우 `examples`, 부록 A 참조)의 `helloworld` 하위 디렉토리에서 사용할 수 있습니다. 자습서 마지막 부분에서 이 응용 프로그램을 실행합니다.

주 – 이 자습서를 진행하려면 Message Queue 제품이 설치되어 있어야 합니다. 필요한 경우 **Message Queue 설치 설명서**를 참조하십시오.

이 자습서는 기본 소개를 제공할 뿐 설명서의 전체 내용을 제공하지 않습니다. 이 자습서의 단계를 수행하여 다음 작업 방법을 학습할 수 있습니다.

- 메시지 브로커 시작
- 브로커에 연결 및 관리 콘솔을 사용하여 브로커 관리
- 브로커에 물리적 대상 만들기
- 객체 저장소를 생성하고 관리 콘솔을 사용하여 객체 저장소에 연결
- 관리 대상 객체를 객체 저장소에 추가하고 관리 객체의 등록 정보 보기

주 – 이 자습서에 있는 지침은 Windows 플랫폼용으로 작성되었습니다. 필요할 때마다 다른 플랫폼 사용자를 위한 보충 설명이 추가됩니다.

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

- 브로커 설정
- 브로커 클러스터 생성
- 특정 물리적 대상 등록 정보 구성
- 영구 저장소의 JDBC 데이터베이스 관리
- 사용자 저장소 관리
- 브로커를 Windows 서비스로 설치
- SSL 인증서 생성

본 설명서의 후속 장에서 이러한 모든 작업에 대해 설명합니다.

관리 콘솔 시작

관리 콘솔을 시작하려면 다음 방법 중 하나를 사용합니다.

- Solaris의 경우 다음 명령을 입력합니다.

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

- Linux의 경우 다음 명령을 입력합니다.

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

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

관리 콘솔 창이 표시되려면 몇 초 정도 기다려야 합니다(그림 2–1 참조).

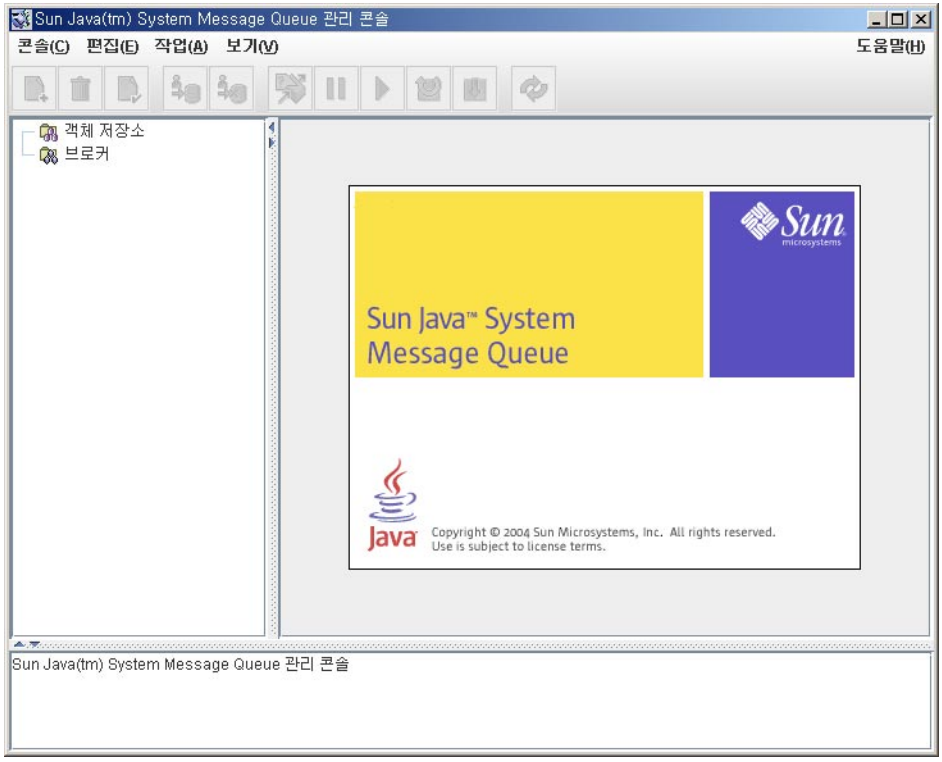


그림 2–1 관리 콘솔 창

몇 초 동안 관리 콘솔 창을 검사합니다. 콘솔 창에는 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.

주 – 관리 콘솔에서 작업할 때 보기 메뉴의 갱신 명령을 사용하여 브로커나 객체 저장소 목록 등 어떤 요소나 요소 그룹의 시각적 표시를 업데이트할 수 있습니다.

관리 콘솔 온라인 도움말

관리 콘솔은 콘솔을 사용하여 관리 작업을 수행하는 방법에 대한 자세한 정보를 표시하는 도움말 기능을 제공합니다. 도움말 기능을 사용하려면 메뉴 모음의 오른쪽 끝에 있는 도움말 풀다운 메뉴를 누른 후 개요를 선택합니다. 관리 콘솔의 도움말 창(그림 2–2)이 표시됩니다.

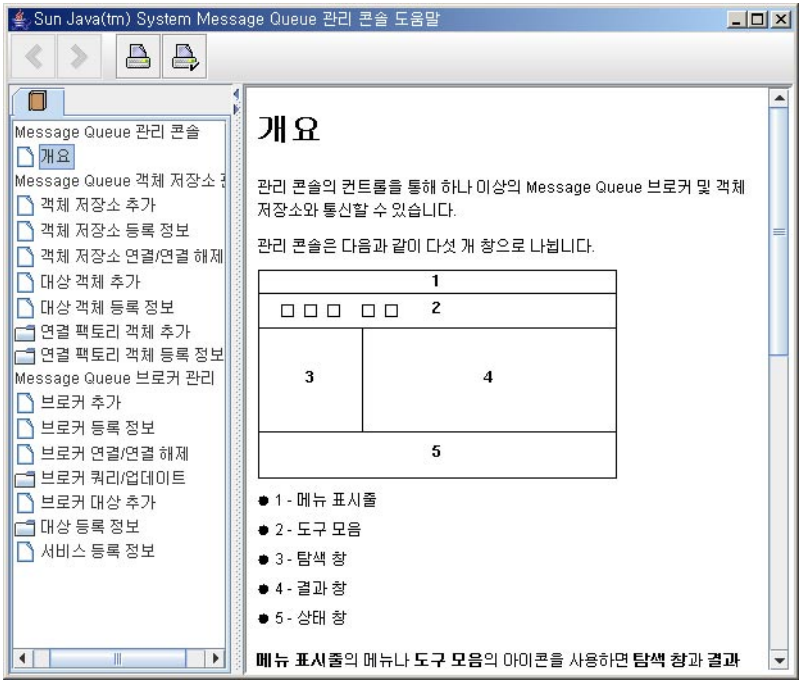


그림 2–2 관리 콘솔 도움말 창

도움말 창 왼쪽에 있는 탐색 창에서 항목은 Message Queue 관리 콘솔, Message Queue 객체 저장소 관리 및 Message Queue 브로커 관리의 세 가지 영역으로 분류됩니다. 각 영역 안에는 파일과 폴더가 있습니다. 폴더는 여러 탭을 포함하는 대화 상자, 간단한 대화 상자를 표시하는 파일 또는 개별 탭에 대한 도움말을 제공합니다. 탐색 창에서 항목 하나를 선택하면 오른쪽 결과 창에 해당 항목의 내용이 표시됩니다. 개요 항목이 선택된 상태에서 결과 창은 그림에 표시된 대로 창의 각 영역을 나타내는 관리 콘솔 창의 구조 보기를 표시합니다.

관리 콘솔을 사용하여 수행할 첫 번째 작업은 브로커의 참조를 만드는 것입니다. 그러나 시작하기 전에 도움말 창에서 관련 정보를 확인해야 합니다. 도움말 창의 탐색 창에 있는 브로커 추가 항목을 누릅니다. 그러면 결과 창의 내용이 바뀌어 브로커 추가의 의미와 브로커 추가 대화 상자의 각 필드의 사용을 설명하는 텍스트를 표시합니다. 도움말 텍스트를 자세히 읽어 본 후 도움말 창을 닫습니다.

브로커 작업

이 절에서는 관리 콘솔을 사용하여 메시지 브로커에 연결하고 관리하는 방법에 대해 설명합니다.

브로커 시작

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

- Solaris의 경우 다음 명령을 입력합니다.

  ```
  /usr/bin/imqbrokerd
  ```

- Linux의 경우 다음 명령을 입력합니다.

  ```
  /opt/sun/mq/bin/imqbrokerd
  ```

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

Windows 시작 메뉴를 사용한 경우에는 명령 창이 나타나고 브로커가 준비되었음을 나타내는 다음과 같은 행이 표시됩니다.

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

관리 콘솔 창을 다시 활성화합니다. 이제 콘솔에 브로커를 추가하여 연결할 준비가 되었습니다. 관리 콘솔에서 브로커에 대한 참조를 추가하기 전에 브로커를 시작할 필요는 없지만, 브로커에 연결하려면 먼저 브로커를 시작해야 합니다.

관리 콘솔에 브로커 추가

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

▼ 관리 콘솔에 브로커를 추가하는 방법

1. **관리 콘솔 창의 탐색 창에 있는 브로커 항목을 누르고 작업 메뉴에서 브로커 추가를 선택합니다.**

   또는, 브로커를 마우스 오른쪽 버튼으로 누르고 팝업 컨텍스트 메뉴에서 브로커 추가를 선택합니다. 두 경우 모두 브로커 추가 대화 상자(그림 2–3)가 나타납니다.

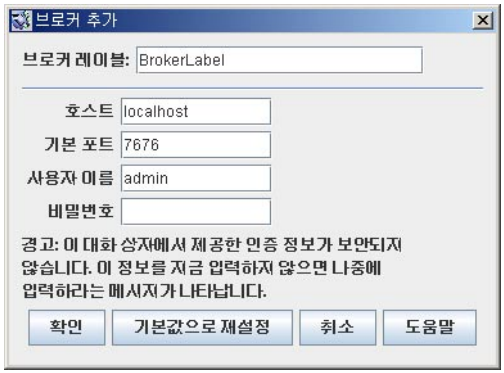


그림 2–3 브로커 추가 대화 상자

2 브로커 레이블 필드에 브로커 이름을 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 만드는 데 사용할 연결 팩토리를 구성할 때 지정해야 하는 값입니다.

이 연습에서는 브로커 레이블 필드에 이름 `MyBroker`를 입력합니다. 비밀번호 필드는 그대로 비워둡니다. 비밀번호는 연결할 때 지정하는 것이 더 안전합니다.

3 확인을 눌러 브로커를 추가하고 대화 상자를 닫습니다.

그림 2–4에 표시된 대로 새 브로커가 탐색 창의 브로커 아래에 나타납니다. 브로커 아이콘 위에 있는 빨간색 X 표시는 관리 콘솔에 현재 연결되어 있지 않음을 나타냅니다.

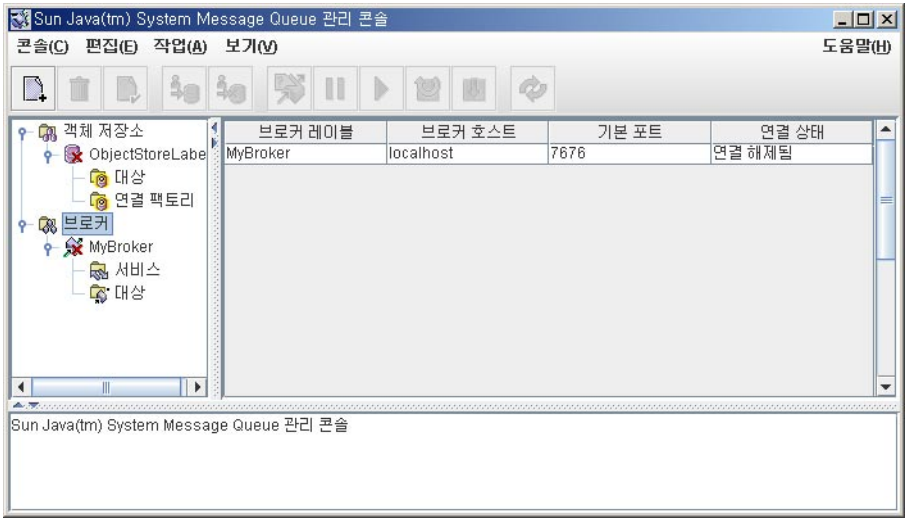


그림 2–4 관리 콘솔 창에 표시된 브로커

브로커를 추가한 후에는 작업 메뉴의 등록 정보 명령(또는 팝업 컨텍스트 메뉴)을 사용하여 43 페이지 "관리 콘솔에 브로커 추가"에 표시된 브로커 추가 대화 상자와 비슷한 브로커 등록 정보 대화 상자를 표시하여 등록 정보를 보거나 수정합니다.

브로커에 연결

관리 콘솔에 브로커를 추가했으면 계속해서 해당 브로커에 연결할 수 있습니다.

▼ 브로커에 연결하는 방법

1 **관리 콘솔 창의 탐색 창에 있는 브로커 항목을 누르고 작업 메뉴에서 브로커에 연결을 선택합니다.**

또는, 브로커의 이름을 마우스 오른쪽 버튼으로 누르고 팝업 컨텍스트 메뉴에서 브로커에 연결을 선택합니다. 두 경우 모두 브로커에 연결 대화 상자(그림 2–5)가 나타납니다.

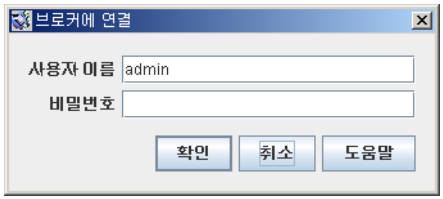


그림 2–5 브로커에 연결 대화 상자

2 **브로커에 연결하는 데 사용할 사용자 이름과 비밀번호를 입력합니다.**

이 대화 상자는 처음에 기본 사용자 이름인 admin을 표시합니다. 실제 환경에서는 되도록이면 안전한 사용자 이름 및 비밀번호를 설정해야 하며(127 페이지 "사용자 인증" 참조) 이 연습에서는 기본값을 사용합니다.

기본 사용자 이름과 관련된 비밀번호 역시 admin입니다. 이 비밀번호를 대화 상자의 비밀번호 필드에 입력합니다. 이렇게 하면 관리 권한으로 브로커에 연결됩니다.

3 **확인을 눌러 브로커에 연결하고 대화 상자를 닫습니다.**

브로커에 연결한 후에는 작업 메뉴의 명령(또는 컨텍스트 메뉴)을 사용하여 선택한 브로커의 다음 작업을 수행할 수 있습니다.

- 브로커 일시 중지는 실행 중인 브로커의 작업을 일시적으로 중단합니다.
- 브로커 다시 시작은 일시 중지된 브로커의 작업을 다시 시작합니다.
- 브로커 다시 작동은 브로커를 초기화하고 다시 시작합니다.
- 브로커 종료는 브로커의 작업을 종료합니다.
- 브로커 쿼리/업데이트는 브로커의 구성 등록 정보를 표시하거나 수정합니다.
- 브로커에서 연결 해제는 브로커와 관리 콘솔 간 연결을 종료합니다.

연결 서비스 보기

브로커는 해당 브로커에서 제공하는 연결 서비스와 지원하는 물리적 대상으로 구별됩니다.

▼ 사용 가능한 연결 서비스를 보는 방법

1 **관리 콘솔 창의 탐색 창에서 브로커 이름 아래에 있는 서비스를 선택합니다.**

각 서비스의 이름, 포트 번호 및 현재 상태를 보여주는 결과 창에 사용 가능한 서비스 목록이 나타납니다(그림 2–6 참조).

그림 2–6 연결 서비스 보기

2 결과 창에서 이름을 눌러 서비스를 선택합니다.

이 연습에서는 jms 이름을 선택합니다.

3 작업 메뉴에서 등록 정보를 선택합니다.

서비스 등록 정보 대화 상자(그림 2–7)가 나타납니다. 이 대화 상자를 사용하면 서비스를 정적 포트 번호에 할당하고 할당된 스레드의 최소/최대 수를 변경할 수 있습니다.

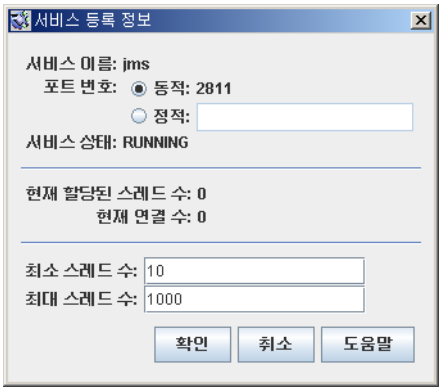


그림 2–7 서비스 등록 정보 대화 상자

이 연습에서는 연결 서비스 등록 정보를 변경하지 마십시오.

4 확인을 눌러 새 등록 정보 값을 적용하고 대화 상자를 닫습니다.

또한 작업 메뉴에는 서비스를 일시 중지하거나 다시 시작하는 명령이 포함되어 있습니다. 관리 서비스를 선택하고 작업 풀다운 메뉴를 누를 경우 서비스 일시 중지

명령을 사용할 수 없게 되는데,이는 관리 서비스가 브로커의 관리 콘솔 링크이기 때문입니다. 이 링크를 일시 중지하면 관리자는 더 이상 브로커에 액세스할 수 없습니다.

물리적 대상 작업

물리적 대상은 하나 이상의 메시지 사용자에게 나중에 전달하기 위해 메시지 생성자로부터 받은 메시지를 보관하는 메시지 브로커의 위치입니다. 사용 중인 **메시징 도메인**에 따라**대기열**(지점간 도메인)과 **주제**(게시/가입 도메인)의 두 종류의 대상이 있습니다. 메시징 도메인 및 해당 도메인과 관련된 대상에 대한 자세한 내용은 **Message Queue 기술 개요**를 참조하십시오.

물리적 대상 만들기

기본적으로 메시지 브로커는 메시지 생성자나 사용자가 존재하지 않는 대상에 액세스하려 할 때마다 새 물리적 대상을 자동으로 생성하도록 구성됩니다. 이러한 **자동 생성 대상**은 소프트웨어 개발 환경에서 클라이언트 코드를 테스트하는 중에 사용하기 편합니다. 그러나 작업 설정에서 대상 자동 생성을 사용할 수 없도록 설정하고 그 대신 관리자에게 모든 대상을 명시적으로 생성하게 하는 것이 좋습니다. 다음 절차에서는 브로커에 해당 **관리 작성 대상**을 추가하는 방법을 보여 줍니다.

▼ 브로커에 물리적 대상을 추가하는 방법

1 **관리 콘솔 창의 탐색 창에서 브로커 이름 아래에 있는 대상 항목을 누르고 작업 메뉴에서 브로커 대상 추가를 선택합니다.**

또는, 대상을 마우스 오른쪽 버튼으로 누르고 팝업 컨텍스트 메뉴에서 브로커 대상 추가를 선택합니다. 두 경우 모두 브로커 대상 추가 대화 상자(그림 2–8)가 나타납니다.

그림 2–8 브로커 대상 추가 대화 상자

2 **대상 이름 필드에 물리적 대상의 이름을 입력합니다.**

대상에 지정한 이름을 적어 두십시오. 나중에 이 물리적 대상에 해당되는 관리 대상 객체를 만들 때 이 이름이 필요합니다.

이 연습에서는 이름 `MyQueueDest`를 입력합니다.

3 **대기열 또는 주제 라디오 버튼을 선택하여 작성할 대상 유형을 지정합니다.**

이 연습에서는 대기열(선택되어 있지 않은 경우)을 선택합니다.

4 **확인을 눌러 물리적 대상을 추가하고 대화 상자를 닫습니다.**

이제 결과 창에 새 대상이 나타납니다.

물리적 대상 등록 정보 보기

관리 콘솔의 작업 메뉴에 있는 등록 정보 명령을 사용하면 물리적 대상의 등록 정보를 보거나 수정할 수 있습니다.

▼ 물리적 대상의 등록 정보를 보거나 수정하는 방법

1 **관리 콘솔 창의 탐색 창에서 브로커 이름 아래에 있는 대상을 선택합니다.**

각 대상의 이름, 유형 및 현재 상태를 보여주는 결과 창에 사용 가능한 물리적 대상 목록이 나타납니다.

2 **결과 창에서 이름을 눌러 물리적 대상을 선택합니다.**

3 **작업 메뉴에서 등록 정보를 선택합니다.**

선택한 물리적 대상에 대한 현재 상태 및 구성 정보를 보여주는 브로커 대상 등록 정보 대화 상자(그림 2–9)가 나타납니다. 이 대화 상자를 사용하면 대상에서 제공할 수 있는 메시지, 생성자 및 사용자의 최대 수와 같은 다양한 구성 등록 정보를 변경할 수 있습니다.

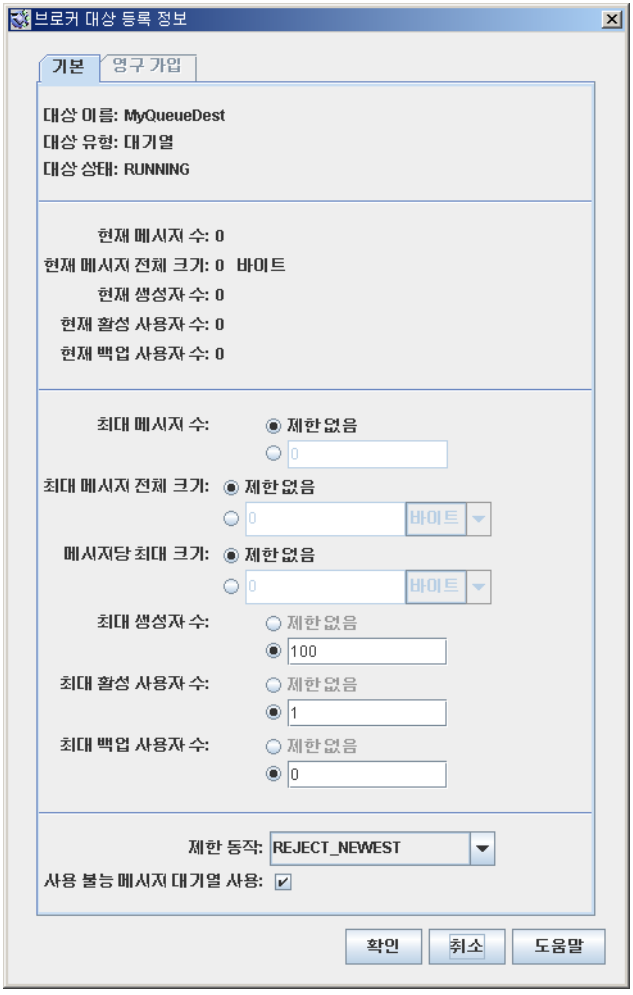


그림 2–9 브로커 대상 등록 정보 대화 상자

이 연습에서는 대상의 등록 정보를 변경하지 마십시오.

주제 대상의 경우 브로커 대상 등록 정보 대화 상자에 영구 가입이라는 추가 탭이 포함됩니다. 이 탭을 누르면 영구 가입 패널(그림 2–10)이 표시되는데, 이 패널은 현재 해당 주제와 관련된 모든 영구 가입에 대한 정보를 나열합니다.

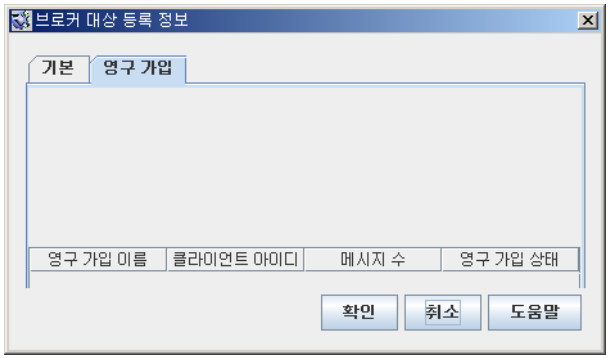


그림 2–10 영구 가입 패널

영구 가입 패널의 제거 및 삭제 버튼을 사용하여 다음을 수행할 수 있습니다.

- 영구 가입과 관련된 모든 보류 메시지 제거
- 주제에서 영구 가입 제거

대기열 대상의 경우 영구 가입 탭이 비활성화됩니다.

4 **확인을 눌러 새 등록 정보 값을 적용하고 대화 상자를 닫습니다.**

물리적 대상에서 메시지 제거

물리적 대상에서 메시지를 **제거**할 경우 대상과 관련된 모든 보류 중인 메시지가 제거되어 대상이 비워집니다.

▼ 물리적 대상에서 메시지를 제거하는 방법

1 **관리 콘솔 창의 탐색 창에서 브로커 이름 아래에 있는 대상을 선택합니다.**
각 대상의 이름, 유형 및 현재 상태를 보여주는 결과 창에 사용 가능한 물리적 대상 목록이 나타납니다.

2 **결과 창에서 이름을 눌러 대상을 선택합니다.**

3 **작업 메뉴에서 메시지 제거를 선택합니다.**
작업을 계속할지 묻는 확인 대화 상자가 나타납니다.

4 **예를 눌러 작업을 확인하고 확인 대화 상자를 닫습니다.**

물리적 대상 삭제

대상을 삭제하면 모든 메시지가 제거되고 대상이 속해 있는 브로커에서 대상이 영구적으로 제거되어 대상 자체가 완전 삭제됩니다.

▼ 물리적 대상을 삭제하는 방법

1. **관리 콘솔 창의 탐색 창에서 브로커 이름 아래에 있는 대상을 선택합니다.**
   각 대상의 이름, 유형 및 현재 상태를 보여주는 결과 창에 사용 가능한 대상 목록이 나타납니다.
2. **결과 창에서 이름을 눌러 대상을 선택합니다.**
3. **편집 메뉴에서 삭제를 선택합니다.**
   작업을 계속할지 묻는 확인 대화 상자가 나타납니다.
4. **예를 눌러 작업을 확인하고 확인 대화 상자를 닫습니다.**
   이 연습에서는 이전에 만든 `MyQueueDest` 대상을 삭제하지 마십시오. 삭제 작업을 수행하지 않고 확인 대화 상자를 닫으려면 아니요를 누릅니다.

객체 저장소 작업

객체 저장소는 Message Queue **관리 대상 객체**를 저장하는 데 사용됩니다. 관리 대상 객체는 특정 Message Queue 공급자에 대한 구현 및 구성 정보를 캡슐화합니다. 객체 저장소는 LDAP(Lightweight Directory Access Protocol) 디렉토리 서버 또는 로컬 파일 시스템의 디렉토리일 수 있습니다.

클라이언트 응용 프로그램 코드 내에서 관리 대상 객체를 직접 인스턴스화하고 구성할 수 있는 경우에도 일반적으로 이러한 객체를 관리자가 생성 및 구성하여 객체 저장소에 저장하도록 하는 것이 좋습니다. 객체 저장소에서는 클라이언트 응용 프로그램이 JNDI(Java Naming and Directory Interface)를 사용하여 이러한 객체에 액세스할 수 있습니다. 이렇게 하면 클라이언트 코드 자체가 공급자와 독립된 상태를 유지할 수 있습니다.

객체 저장소 추가

관리 콘솔을 사용하여 객체 저장소를 **관리**할 수 있지만, 객체 저장소를 **생성**할 수는 없습니다. 먼저 객체 저장소로 사용할 LDAP 서버 또는 파일 시스템 디렉토리가 있어야 합니다. 그런 다음, 기존 객체 저장소를 관리 콘솔에 추가하여 콘솔에서 객체 저장소에 대한 작업을 수행하는 데 사용할 수 있는 객체 저장소의 참조를 만듭니다.

주 – 이 장에서 사용하는 샘플 응용 프로그램에서는 객체 저장소가 C 드라이브의 `Temp` 디렉토리에 있다고 가정합니다. 다음 연습을 진행하기 전에 아직 C 드라이브에 `Temp` 폴더가 없으면 지금 만듭니다. Windows가 아닌 플랫폼에서는 이미 존재하는 `/tmp` 디렉토리를 사용할 수 있습니다.

▼ 객체 저장소를 관리 콘솔에 추가하는 방법

1 관리 콘솔 창의 탐색 창에 있는 객체 저장소를 누르고 작업 메뉴에서 객체 저장소 추가를 선택합니다.

또는, 객체 저장소를 마우스 오른쪽 버튼으로 누르고 팝업 컨텍스트 메뉴에서 객체 저장소 추가를 선택합니다. 두 경우 모두 객체 저장소 추가 대화 상자(그림 2–11)가 나타납니다.

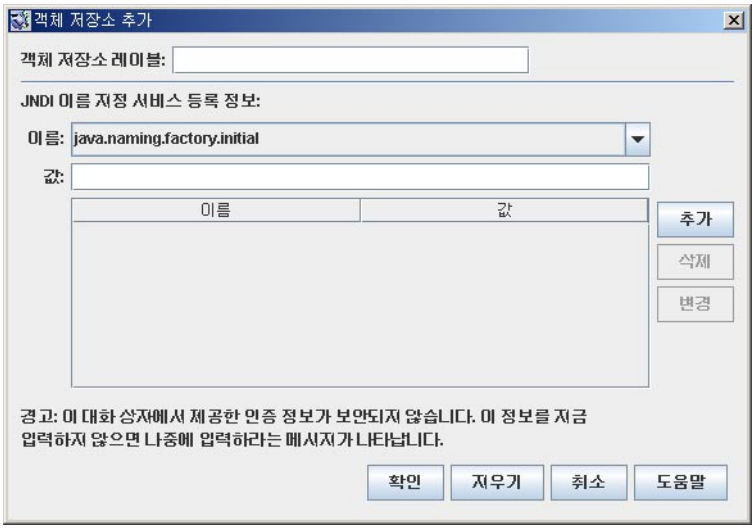


그림 2–11 객체 저장소 추가 대화 상자

2 객체 저장소 레이블 필드에 객체 저장소의 이름을 입력합니다.

관리 콘솔에서 이 객체 저장소를 식별하는 레이블을 제공합니다.

이 연습에서는 이름 `MyObjectStore`를 입력합니다.

3 관리 객체를 조회하는 데 사용할 JNDI 속성 값을 입력합니다.

a. 이름 풀다운 메뉴에서 지정할 속성의 이름을 선택합니다.

b. 속성 값을 값 필드에 입력합니다.

c. 추가 버튼을 눌러 지정한 속성 값을 추가합니다.

등록 정보와 등록 정보 값이 등록 정보 요약 창에 나타납니다.

설정해야 할 속성 수만큼 53 페이지 “객체 저장소 추가” "a"부터 "c" 단계를 반복합니다.

이 연습에서는 `java.naming.factory.initial` 속성을 다음으로 설정합니다.

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

그리고 `java.naming.provider.url` 속성을 다음으로 설정합니다.

```
file:///C:/Temp
```

(Solaris 또는 Linux 플랫폼의 경우 `file:///tmp`). 이 속성은 파일 시스템 객체 저장소에 대해 설정해야 하는 유일한 속성입니다. LDAP 저장소에 필요한 속성 값에 대해서는 153 페이지 “LDAP 서버 객체 저장소”를 참조하십시오.

4 확인을 눌러 객체 저장소를 추가하고 대화 상자를 닫습니다.

그림 2–12에 표시된 대로 새 객체 저장소가 탐색 창의 객체 저장소 아래에 나타납니다. 객체 저장소 아이콘 위에 있는 빨간색 X 표시는 관리 콘솔에 현재 연결되어 있지 않음을 나타냅니다.

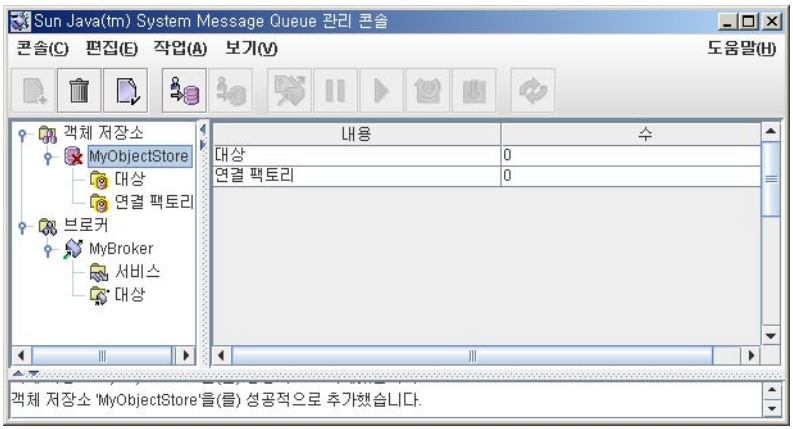


그림 2–12 관리 콘솔 창에 표시된 객체 저장소

탐색 창에서 객체 저장소를 누르면 내용이 결과 창에 나열됩니다. 관리 객체를 객체 저장소에 아직 추가하지 않았으므로 대상과 연결 팩토리의 수 열에 모두 0이 표시됩니다.

객체 저장소를 추가한 후에는 작업 메뉴(또는 팝업 컨텍스트 메뉴)의 등록 정보 명령을 사용하여 그림 2–11에 표시된 객체 저장소 추가 대화 상자와 비슷한 객체 저장소 등록 정보 대화 상자를 표시하여 등록 정보를 보거나 수정합니다.

객체 저장소에 연결

이제 객체 저장소가 관리 콘솔에 추가되었으며, 관리 객체를 추가하려면 객체 저장소에 연결해야 합니다.

▼ 객체 저장소에 연결하는 방법

- **관리 콘솔 창의 탐색 창에 있는 객체 저장소를 누르고 작업 메뉴에서 객체 저장소에 연결을 선택합니다.**
  또는, 객체 저장소의 이름을 마우스 오른쪽 버튼으로 누르고 팝업 컨텍스트 메뉴에서 객체 저장소에 연결을 선택합니다. 두 경우 모두 관리 콘솔에 연결되어 있음을 나타내며 객체 저장소 아이콘에서 빨간색 X 아이콘이 사라집니다.

관리 객체 작업

관리 콘솔에 객체 저장소를 연결한 후에는 관리 객체(연결 팩토리 및 대상)를 계속 추가할 수 있습니다. 이 절에서는 다음 방법에 대해 설명합니다.

주 – 관리 콘솔에는 Message Queue 관리 대상 객체만 표시됩니다. 객체 저장소에 추가할 관리 대상 객체와 동일한 조회 이름을 가진 비 Message Queue 객체를 포함하는 경우에 이 객체를 추가하려고 하면 오류 메시지가 표시됩니다.

연결 팩토리 추가

연결 팩토리는 클라이언트 응용 프로그램에서 브로커에 대한 연결을 만들 때 사용됩니다. 연결 팩토리를 구성하여 생성되는 연결의 등록 정보를 제어할 수 있습니다.

▼ 객체 저장소에 연결 팩토리를 추가하는 방법

1. **객체 저장소가 관리 콘솔에 연결되어 있는지 확인합니다(56 페이지 "객체 저장소에 연결" 참조).**

2. **관리 콘솔 창의 탐색 창에서 객체 저장소 이름 아래에 있는 연결 팩토리 항목을 누르고 작업 메뉴에서 연결 팩토리 객체 추가를 선택합니다.**
   또는, 연결 팩토리를 마우스 오른쪽 버튼으로 누르고 팝업 컨텍스트 메뉴에서 연결 팩토리 객체 추가를 선택합니다. 두 경우 모두 연결 팩토리 객체 추가 대화 상자(그림 2–13)가 나타납니다.

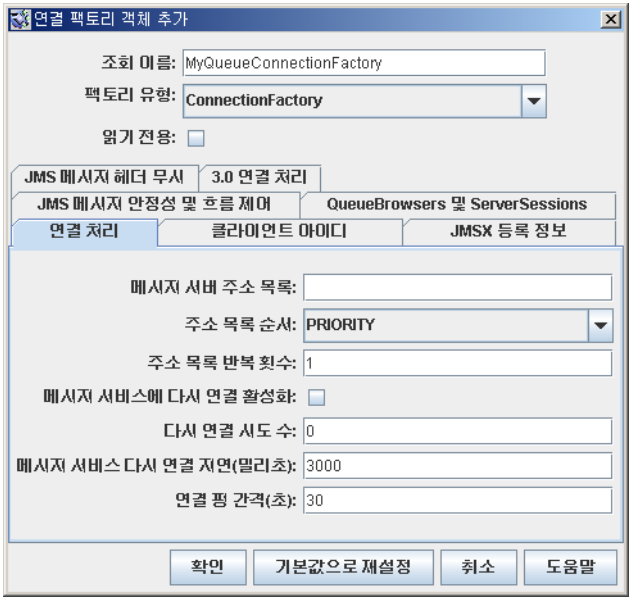


그림 2–13 연결 팩토리 객체 추가 대화 상자

3 조회 이름 필드에 연결 팩토리의 이름을 입력합니다.

이 이름은 클라이언트 응용 프로그램이 JNDI를 사용하여 연결 팩토리를 조회할 때 사용할 이름입니다.

이 연습에서는 이름 `MyQueueConnectionFactory`를 입력합니다.

4 팩토리 유형 풀다운 메뉴에서 생성할 연결 팩토리 유형을 선택합니다.

이 연습에서는 QueueConnectionFactory를 선택합니다.

5 연결 처리 탭을 누릅니다.

그림 2–13에 표시된 대로 연결 처리 패널이 나타납니다.

6 이 연결 팩토리에서 연결할 브로커의 주소를 메시지 서버 주소 목록 필드에 입력합니다.

주소 목록은 단일 브로커 또는 (브로커 클러스터의 경우) 여러 브로커로 구성될 수 있습니다. 각 브로커의 경우 브로커의 연결 서비스, 호스트 이름 및 포트 번호와 같은 정보를 지정합니다. 지정할 정보의 정확한 특성 및 구문은 사용할 연결 서비스에 따라 다릅니다. 자세한 내용은 287 페이지 “연결 처리”를 참조하십시오.

이 연습에서는 샘플 응용 프로그램 `HelloWorldMessageJNDI`에서 연결 팩토리가 기본적으로 구성되어 있는 표준 주소 목록 속성(연결 서비스 `jms`, 호스트 이름 `localhost` 및 포트 번호 7676)을 사용해야 하므로 메시지 서버 주소 목록 필드에 입력할 필요가 없습니다.

7 **필요에 따라 연결 팩토리의 기타 속성을 구성합니다.**

연결 팩토리 객체 추가 대화 상자에는 연결 처리 외에도 연결 팩토리의 다양한 속성을 구성하는 데 사용할 수 있는 여러 패널이 있습니다.

이 연습에서는 다른 속성 설정을 변경하지 마십시오. 그러나 지정할 수 있는 구성 정보가 있으면 다른 탭을 눌러 유용하게 사용할 수도 있습니다. 도움말 버튼을 사용하여 이러한 다른 구성 패널의 내용을 자세히 알아봅니다.

8 **해당되는 경우 읽기 전용 확인란을 누릅니다.**

그러면 연결 팩토리 객체의 구성 속성이 생성 시 지정했던 값으로 잠깁니다. 읽기 전용 관리 대상 객체의 속성은 클라이언트 코드를 통한 프로그래밍 방식으로든 명령줄을 통한 관리적인 방식으로든 무시할 수 없습니다.

이 연습에서는 읽기 전용을 선택하지 마십시오.

9 **확인을 눌러 연결 팩토리를 만들고 객체 저장소에 추가한 후 대화 상자를 닫습니다.**

이제 결과 창에 새 연결 팩토리가 나타납니다.

대상 추가

대상 관리 객체는 브로커의 물리적 대상을 나타내므로 클라이언트에서 공급자별 구성 및 이름 지정 구문과 상관 없이 물리적 대상으로 메시지를 보낼 수 있습니다. 클라이언트가 관리 대상 객체를 통해 주소 지정된 메시지를 보내는 경우 브로커가 메시지를 해당 물리적 대상(있는 경우)으로 전달합니다. 해당 물리적 대상이 없으면 48 페이지 "물리적 대상 만들기"에 설명한 대로 자동 생성을 사용하는 경우 브로커가 물리적 대상을 자동으로 만들고 메시지를 전달합니다. 그렇지 않으면 메시지를 전달할 수 없음을 나타내는 오류가 표시됩니다.

다음 절차는 기존 물리적 대상에 해당하는 객체 저장소에 대상 관리 객체를 추가하는 방법에 대해 설명합니다.

▼ 객체 저장소에 대상을 추가하는 방법

1 **객체 저장소가 관리 콘솔에 연결되어 있는지 확인합니다(56 페이지 "객체 저장소에 연결" 참조).**

2 **관리 콘솔 창의 탐색 창에서 객체 저장소 이름 아래에 있는 대상 항목을 누르고 작업 메뉴에서 대상 객체 추가를 선택합니다.**

또는, 대상을 마우스 오른쪽 버튼으로 누르고 팝업 컨텍스트 메뉴에서 대상 객체 추가를 선택합니다. 두 경우 모두 대상 객체 추가 대화 상자(그림 2–14)가 나타납니다.

그림 2–14 대상 객체 추가 대화 상자

3 조회 이름 필드에 대상 관리 객체의 이름을 입력합니다.

이 이름은 클라이언트 응용 프로그램이 JNDI를 사용하여 대상을 조회할 때 사용할 이름입니다.

이 연습에서는 이름 `MyQueue`를 입력합니다.

4 대기열 또는 주제 라디오 버튼을 선택하여 생성할 대상 객체 유형을 지정합니다.

이 연습에서는 대기열(선택되어 있지 않은 경우)을 선택합니다.

5 대상 이름 필드에 해당하는 물리적 대상의 이름을 입력합니다.

이 이름은 브로커에 물리적 대상을 추가했을 때 지정한 이름입니다(48 페이지 "물리적 대상 작업" 참조).

이 연습에서는 이름 `MyQueueDest`를 입력합니다.

6 선택적으로 대상 설명 필드에 대상에 대한 간략한 설명을 입력합니다.

이 필드의 내용은 클라이언트 작업에는 영향을 주지 않으며 사용자 참조용일 뿐입니다.

이 연습에서는 대상 설명 필드의 내용을 삭제하거나 다음과 같은 설명 텍스트를 입력합니다.

```
Example destination for MQ Admin Guide tutorial
```

7 해당되는 경우 읽기 전용 확인란을 누릅니다.

그러면 대상 객체의 구성 속성이 생성 시 지정했던 값으로 잠깁니다. 읽기 전용 관리 대상 객체의 속성은 클라이언트 코드를 통한 프로그래밍 방식으로든 명령줄을 통한 관리적인 방식으로든 무시할 수 없습니다.

이 연습에서는 읽기 전용을 선택하지 마십시오.

8 **확인을 눌러 대상 객체를 만들고 객체 저장소에 추가한 후 대화 상자를 닫습니다.**

그림 2–15에 표시된 대로 결과 창에 새 대상 객체가 나타납니다.

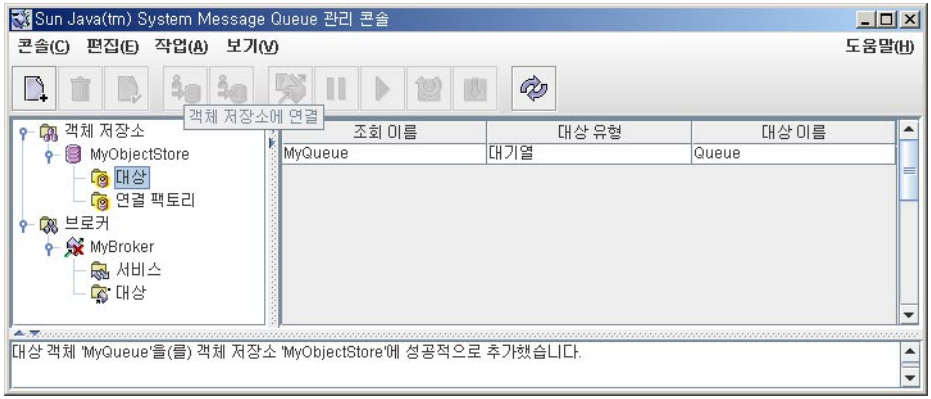


그림 2–15 관리 콘솔 창에 표시된 대상 객체

관리 대상 객체 등록 정보 보기

관리 콘솔의 작업 메뉴에 있는 등록 정보 명령을 사용하면 관리 대상 객체의 등록 정보를 보거나 수정할 수 있습니다.

▼ 관리 객체의 등록 정보를 보거나 수정하는 방법

1 **관리 콘솔 창의 탐색 창에서 객체 저장소 이름 아래에 있는 연결 팩토리 또는 대상을 선택합니다.**

각 항목의 조회 이름 및 유형과 대상 관리 객체의 경우에는 대상 이름을 보여주는 결과 창에 사용 가능한 연결 팩토리 또는 대상 관리 객체 목록이 나타납니다.

2 **결과 창에서 이름을 눌러 관리 객체를 선택합니다.**

3 **작업 메뉴에서 등록 정보를 선택합니다.**

연결 팩토리 객체 추가(그림 2–13) 또는 대상 객체 추가(그림 2–14) 대화 상자와 비슷한 연결 팩토리 객체 등록 정보 또는 대상 객체 등록 정보 대화 상자가 나타납니다. 이 대화 상자를 사용하면 선택한 객체의 구성 속성을 변경할 수 있습니다. 그러나, 객체의 조회 이름을 변경할 수는 없습니다. 이 이름을 변경하는 유일한 방법은 객체를 삭제한 후에 원하는 조회 이름의 새 관리 객체를 추가하는 것입니다.

4 **확인을 눌러 새 속성 값을 적용하고 대화 상자를 닫습니다.**

관리 대상 객체 삭제

관리 객체를 삭제하면 객체가 속해 있는 객체 저장소에서 객체가 영구적으로 제거됩니다.

▼ 관리 객체를 삭제하는 방법

1. **관리 콘솔 창의 탐색 창에서 객체 저장소 이름 아래에 있는 연결 팩토리 또는 대상을 선택합니다.**
   각 항목의 조회 이름 및 유형과 대상 관리 객체의 경우에는 대상 이름을 보여주는 결과 창에 사용 가능한 연결 팩토리 또는 대상 관리 객체 목록이 나타납니다.
2. **결과 창에서 이름을 눌러 관리 객체를 선택합니다.**
3. **편집 메뉴에서 삭제를 선택합니다.**
   작업을 계속할지 묻는 확인 대화 상자가 나타납니다.
4. **예를 눌러 작업을 확인하고 확인 대화 상자를 닫습니다.**
   이 연습에서는 이전에 만든 관리 대상 객체 `MyQueue` 또는 `MyQueueConnectionFactory`를 삭제하지 마십시오. 삭제 작업을 수행하지 않고 확인 대화 상자를 닫으려면 아니요를 누릅니다.

샘플 응용 프로그램 실행

이 자습서에서 사용할 수 있도록 샘플 응용 프로그램 `HelloWorldMessageJNDI`가 제공됩니다. 이 응용 프로그램은 사용자가 만든 물리적 대상과 관리 대상 객체를 사용합니다.

- 이름이 `MyQueueDest`인 대기열 물리적 대상
- JNDI 조회 이름이 `MyQueueConnectionFactory`인 대기열 연결 팩토리 관리 대상 객체
- JNDI 조회 이름이 `MyQueue`인 대기열 관리 대상 객체

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

응용 프로그램을 실행하기 전에 `HelloWorldMessageJNDI.java` 소스 파일을 열고 코드를 읽습니다. 이 프로그램은 간단하지만 확실하게 기록되어 있으므로 작동 방법을 손쉽게 이해할 수 있을 것입니다.

▼ 샘플 응용 프로그램을 실행하는 방법

1 사용 중인 플랫폼에 따라 다음 명령 중 하나를 사용하여 HelloWorldmessageJNDI 응용 프로그램을 포함하는 디렉토리를 현재 디렉토리로 설정합니다.

- Solaris:

```
cd /usr/demo/imq/helloworld/helloworldmessagejndi
```

- Linux:

```
cd /opt/sun/mq/examples/helloworld/helloworldmessagejndi
```

- Windows:

```
cd IMQ_HOME\\demo\\helloworld\\helloworldmessagejndi
```

HelloWorldMessageJNDI.class 파일이 있는지 확인해야 합니다. (응용프로그램을 변경하는 경우에는 **Java 클라이언트용 Message Queue 개발 안내서**에 제공된 클라이언트 응용 프로그램의 컴파일 절차를 사용하여 다시 컴파일해야 합니다.)

2 CLASSPATH 변수를 설정하여 HelloWorldMessageJNDI.class 파일이 들어 있는 현재 디렉토리와 Message Queue 제품에 포함되어 있는 다음 .jar 파일을 포함합니다.

```
jms.jar
imq.jar
jndi.jar
fscontext.jar
```

CLASSPATH 변수 설정에 대한 자세한 내용은 **Java 클라이언트용 Message Queue 개발 안내서**를 참조하십시오.

주 – jndi.jar 파일은 JDK 1.4와 함께 제공됩니다. 이전 버전의 JDK를 사용하지 않는 경우에는 CLASSPATH에 이 파일을 추가할 필요가 없습니다.

---

3 사용 중인 플랫폼에 따라 다음 명령 중 하나를 실행하여 HelloWorldMessageJNDI 응용 프로그램을 실행합니다.

- Solaris 또는 Linux:

```
% java HelloWorldMessageJNDI file:///tmp
```

- Windows:

```
java HelloWorldMessageJNDI
```

응용 프로그램이 성공적으로 실행되면 아래와 같이 결과가 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
```

2

관리 작업

- 3 장
- 4 장
- 5 장
- 6 장
- 7 장
- 8 장
- 9 장
- 10 장
- 11 장
- 12 장

◆ ◆ ◆ 3 장

3 브로커 및 클라이언트 시작

Sun Java System™ Message Queue™를 설치하고 몇 가지 준비 단계를 수행한 후에는 브로커와 클라이언트를 시작할 수 있습니다. 브로커의 구성은 일련의 구성 파일에 따라 결정되며, 브로커 유틸리티(`imqbrokerd`)로 전달되는 명령줄 옵션으로 대체될 수 있습니다. 자세한 내용은 4 장을 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

- 67 페이지 "시스템 자원 준비"
- 68 페이지 "브로커 시작"
- 72 페이지 "브로커 제거"
- 73 페이지 "클라이언트 시작"

시스템 자원 준비

브로커를 시작하기 전에 시스템 클럭을 동기화하고 파일 설명자 제한을 설정(Solaris 또는 Linux)하는 두 가지 예비 시스템 수준 작업을 수행해야 합니다. 다음 절에서는 이러한 작업에 대해 설명합니다.

시스템 클럭 동기화

브로커 또는 클라이언트를 시작하기 전에 Message Queue 시스템과 상호 작용하는 모든 호스트에서 클럭을 동기화해야 합니다. 동기화는 메시지 만료(수명)를 사용할 경우에 특히 중요합니다. 동기화되지 않은 클럭의 타임스탬프는 메시지 만료가 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`

서비스(Windows)에 의해 지원됩니다. 이러한 서비스 구성에 대한 자세한 내용은 운영 체제 설명서를 참조하십시오. 브로커가 실행된 이후에는 시스템 클럭을 역방향으로 설정하지 마십시오.

파일 설명자 제한 설정

Solaris와 Linux 플랫폼에서는 클라이언트 또는 브로커가 실행 중인 쉘에 프로세스가 사용할 수 있는 파일 설명자 수에 대한 소프트 제한이 있습니다. Message Queue에서는 클라이언트가 만들거나 브로커가 받아들이는 각 연결에 이러한 파일 설명자 중 하나를 사용합니다. 지속성 메시지를 갖는 각 물리적 대상 역시 파일 설명자를 사용합니다.

따라서 파일 설명자 제한은 브로커 또는 클라이언트가 가질 수 있는 연결 수를 제한합니다. 기본적으로 최대 연결 수는 Solaris의 경우 256이며 Linux의 경우에는 1024입니다. (지속성을 위해 파일 설명자를 사용하기 때문에 연결 제한은 실제로 더 낮음). 이 보다 더 많은 연결이 필요하면 클라이언트나 브로커가 실행될 각 쉘의 파일 설명자 제한을 늘려야 합니다. 파일 설명자 제한을 늘리는 방법에 대한 자세한 내용은 `ulimit` 설명서 페이지를 참조하십시오.

브로커 시작

Message Queue 명령줄 유틸리티나 Windows 시작 메뉴를 시용하여 브로커를 대화식으로 시작하거나 시스템 시작 시 자동으로 시작하도록 정렬할 수 있습니다. 다음 절에서는 브로커 시작 방법에 대해 설명합니다.

대화식으로 브로커 시작

브로커 유틸리티(`imqbrokerd`)를 사용하여 명령줄에서 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. 각 브로커 인스턴스는 자체 구성 등록 정보 및 파일 기반 메시지 저장소를 가지고 있습니다. 브로커 인스턴스가 처음 시작되면 Message Queue에서는 해당 사용자의 파일 작성 모드 마스크(`umask`)를 사용하여 해당 브로커 인스턴스의 구성 정보와 영구 데이터가 들어 있는 디렉토리에 대한 권한을 설정합니다.

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 매퍼가 있는 로컬 시스템에서 imqbroker라는 브로커 인스턴스가 시작됩니다(76 페이지 "포트 매퍼" 참조).

기본값이 아닌 인스턴스 이름을 지정하려면 -name 옵션을 imqbrokerd 명령에 사용합니다. 다음 명령은 인스턴스 이름이 myBroker인 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 설정하여 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 238 페이지 "브로커 유틸리티"를 참조하십시오. 이 정보에 대한 빠른 요약을 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. 다음과 같이 try를 사용권 이름으로 지정합니다.

```
imqbrokerd -license try
```

이 옵션은 브로커를 시작할 때마다 사용해야 하며 그렇지 않으면 브로커의 기본값이 표준 플랫폼판 사용권이 됩니다.

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 YES로 설정합니다.
- 비정상적인 종료 이후에 브로커가 자동으로 다시 시작되게 하려면 RESTART 등록 정보를 YES로 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. 그리고 나면 브로커가 시스템이 시작될 때 시작되어 시스템이 종료될 때까지 백그라운드에서 실행됩니다. 따라서 추가 인스턴스를 시작하는 경우가 아니면 `imqbrokerd` 명령을 사용해서 브로커를 시작하지 마십시오.

시스템에는 Windows 서비스로 실행 중인 브로커가 하나 뿐일 수 있습니다. 작업 관리자는 이러한 브로커를 두 개의 실행 가능한 프로세스로 나열합니다.

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

Message Queue를 Windows 시스템에 설치할 때 브로커를 서비스로 설치할 수 있습니다. 설치 이후에 서비스 관리자 유틸리티(`imqsvcadmin`)를 사용하여 다음 작업을 수행할 수 있습니다.

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

브로커에 시작 옵션을 전달하려면 `-args` 인수를 `imqsvcadmin` 명령에 사용합니다. 68 페이지 “브로커 시작”에서 설명한 대로 이 옵션은 `imqbrokerd` 명령의 `-D` 옵션과 동일하게 작동합니다. 명령 유틸리티(`imqcmd`)를 사용하여 보통의 경우와 같이 브로커 작업을 제어합니다.

`imqsvcadmin` 명령의 구문, 하위 명령 및 옵션에 대한 자세한 내용은 254 페이지 “서비스 관리자 유틸리티”를 참조하십시오.

브로커 서비스 재구성

설치된 브로커를 Windows 서비스로 재구성하는 절차는 다음과 같습니다.

▼ 실행 중인 브로커를 Windows 서비스로 재구성하는 방법

1 서비스를 정지합니다.

a. Windows 시작 메뉴의 설정 하위 메뉴에서 제어판을 선택합니다.

b. 관리 도구 제어판을 엽니다.

c. 아이콘을 선택한 후 파일 메뉴 또는 팝업 컨텍스트 메뉴에서 열기를 선택하거나 아이콘을 두 번 눌러 서비스 도구를 실행합니다.

d. 서비스(로컬)에서 Message Queue 브로커 서비스를 선택한 후 작업 메뉴에서 등록 정보를 선택합니다.

또는, Message Queue 브로커를 마우스 오른쪽 버튼으로 누른 다음 팝업 컨텍스트 메뉴에서 등록 정보를 선택하거나 Message Queue 브로커를 두 번 누릅니다. 두 경우 모두 Message Queue 브로커 등록 정보 대화 상자가 나타납니다.

e. 등록 정보 대화 상자의 일반 탭에서 브로커 서비스를 중지하려면 중지를 누릅니다.

2 서비스를 제거합니다.

명령줄에서 다음 명령을 입력합니다.

```
imqsvcadmin remove
```

3 `-args` 옵션으로 다른 브로커 시작 옵션을 지정하거나 `-vmargs` 옵션으로 다른 Java 버전 인수를 지정하여 서비스를 다시 설치합니다.

예를 들어, 서비스의 호스트 이름 및 포트 번호를 `broker1` 및 `7878`로 변경하려면 다음 명령을 사용합니다.

```
imqsvcadmin install -args "-name broker1 -port 7878"
```

대체 Java Runtime 사용

`imqsvcadmin` 명령의 `-javahome` 또는 `-jrehome` 옵션을 사용하여 대체 Java Runtime의 위치를 지정할 수 있습니다. 또한 서비스의 등록 정보 대화 창에 있는 일반 탭의 시작 매개변수 필드에서 해당 옵션을 지정할 수도 있습니다.

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

```
-javahome c:\\\\j2sdk1.4.0
```

브로커 서비스 시작 옵션 표시

브로커 서비스의 시작 옵션을 결정하려면 예 3–1에 표시된 대로 `imqsvcadmin` 명령에 `query` 옵션을 사용합니다.

예 3–1 브로커 서비스 시작 옵션 표시

```
imqsvcadmin query

Service Message Queue Broker is installed.
Display Name: Message Queue Broker
Start Type: Automatic
Binary location: C:\\Sun\\MessageQueue\\bin\\imqbrokersvc.exe
JavaHome: c:\\j2sdk1.4.0
Broker Args: -name broker1 -port 7878
```

서비스 시작 문제 해결

브로커를 Windows 서비스로 시작할 때 오류가 발생하는 경우 기록된 오류 이벤트를 볼 수 있습니다.

▼ 기록된 서비스 오류 이벤트를 보는 방법

1. **Windows 관리 도구 제어판을 엽니다.**
2. **이벤트 뷰어 도구를 시작합니다.**
3. **응용 프로그램 이벤트 로그를 선택합니다.**
4. **작업 메뉴에서 갱신을 선택하여 오류 이벤트를 표시합니다.**

브로커 제거

브로커를 다시 제거하는 절차는 다음 절에 설명된 대로 플랫폼마다 다릅니다.

Solaris 또는 Linux에서 브로커 제거

Solaris 또는 Linux 플랫폼에서 브로커 인스턴스를 제거하려면 imqbrokerd 명령에 `-remove` 옵션을 사용합니다. 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.

시스템을 시작할 때 브로커가 자동으로 시작되도록 설정한 경우에는 구성 파일 `/etc/imq/imqbrokerd.conf`(Solaris) 또는 `/etc/opt/sun/mq/imqbrokerd.conf`(Linux)를 편집하고 `AUTOSTART` 등록 정보를 `NO`로 설정합니다.

`imqbrokerd` 명령의 구문, 하위 명령 및 옵션에 대한 자세한 내용은 238 페이지 "브로커 유틸리티"를 참조하십시오. 이 정보에 대한 빠른 요약을 보려면 다음을 입력합니다.

Windows 브로커 서비스 제거

Windows 서비스로 실행 중인 브로커를 제거하려면 다음 명령을 사용하여

```
imqcmd shutdown bkr
```

브로커를 종료한 후, 다음 명령을 사용하여

```
imqsvcadmin remove
```

서비스를 제거합니다.

또는 **관리 도구** 제어판의 Windows **서비스** 도구를 사용하여 브로커 서비스를 중지하고 제거할 수도 있습니다.

브로커 서비스를 제거한 후에는 컴퓨터를 다시 시작합니다.

클라이언트 시작

클라이언트 응용 프로그램을 시작하기 전에 응용 프로그램 개발자로부터 시스템 설정 방법에 대한 정보를 얻습니다. Java 클라이언트 응용 프로그램을 시작할 경우 `CLASSPATH` 변수를 적절하게 설정하고 올바른 `.jar` 파일이 설치되었는지 확인해야 합니다. **Java 클라이언트용 Message Queue 개발 안내서**에 시스템 설정 관련 일반 단계에 대한 자세한 정보가 있으며 개발자에게는 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 프로그램 설명서는 응용 프로그램에서 설정하는 속성 값에 대한 정보를 제공해야 하며, 명령줄에서 이러한 속성 값 중 일부를 대체해야 하는 경우도 있습니다. Java 클라이언트의 명령줄에서 JNDI(Java Naming and Directory Interface) 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. 필요한 경우, 명령줄을 사용하여 해당 기본값을 대체할 수도 있습니다.

명령줄에서 Java 응용 프로그램에 대한 속성 값을 지정하려면 다음 구문을 사용합니다.

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

16 장에 표시된 대로 *attribute* 값은 연결 팩토리 관리 대상 객체 속성이어야 합니다. 값에 공백이 있는 경우 명령줄의 *attribute*= *value* 부분을 따옴표로 묶어야 합니다.

다음 예에서는 7677 포트에서 OtherHost 호스트의 브로커에 연결하는 MyMQClient라는 클라이언트 응용 프로그램을 시작합니다.

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

명령줄에서 지정한 호스트 이름 및 포트는 응용 프로그램 자체에서 설정된 호스트 이름 및 포트를 대체합니다.

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

◆ ◆ ◆ 4 장

4 브로커 구성

브로커의 구성은 시작 시 일련의 구성 파일 및 `imqbrokerd` 명령에 전달된 옵션에 따라 결정됩니다. 이 장에서는 사용 가능한 구성 등록 정보와 해당 등록 정보를 사용하여 브로커를 구성하는 방법에 대해 설명합니다.

이 장은 다음 내용으로 구성되어 있습니다.

- 75 페이지 "브로커 서비스"
- 88 페이지 "브로커 등록 정보 설정"
- 91 페이지 "영구 데이터 저장소 구성"

브로커 구성 등록 정보에 대한 자세한 내용은 14 장을 참조하십시오.

브로커 서비스

브로커 구성 등록 정보는 적용되는 서비스 또는 브로커 구성 요소에 따라 몇 개의 범주로 구분될 수 있습니다.

- **연결 서비스**는 보내고 받는 메시지를 전송하는 브로커와 해당 클라이언트 간의 물리적 연결을 관리합니다.
- **라우팅 서비스**는 JMS 페이로드 메시지의 라우팅과 전달을 지정할 뿐만 아니라, 메시지 서비스에서 사용하는 메시지를 제어하여 안정적인 전달을 지원합니다.
- **지속성 서비스**는 영구 저장소에 데이터를 기록하고 영구 저장소에서 데이터를 검색하는 작업을 관리합니다.
- **보안 서비스**는 브로커에 연결하는 사용자를 인증하고 작업 권한을 부여합니다.
- **모니터링 서비스**는 브로커 성능에 대한 메트릭 및 진단 정보를 생성합니다.

다음 절에서는 이러한 각각의 서비스와 특정 필요에 맞추어 이러한 서비스를 사용자 정의하는 데 사용하는 등록 정보에 대해 설명합니다.

연결 서비스

메시지 브로커는 다양한 전송 프로토콜을 사용하여 응용 프로그램과 관리 클라이언트를 모두 지원하는 다양한 **연결 서비스**를 제공할 수 있습니다. 연결 서비스와 관련된 브로커 구성 등록 정보는 257 페이지 "연결 등록 정보"에 나와 있습니다.

표 4–1은 다음의 두 가지 특성으로 구별되는 사용 가능한 연결 서비스를 보여줍니다.

- **서비스 유형**은 서비스가 JMS 메시지 전달(`NORMAL`)을 제공하는지 또는 Message Queue™ 관리 서비스(`ADMIN`)를 제공하는지 여부를 지정합니다.
- **프로토콜 유형**은 기본 전송 프로토콜을 지정합니다.

표 4–1 Message Queue 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)

브로커의 `imq.service.activelist` 등록 정보를 설정하면 이러한 연결 서비스 중 하나 또는 모두를 실행하도록 해당 등록 정보를 구성할 수 있습니다. 이 등록 정보의 값은 브로커가 시작될 때 활성화될 연결 서비스 목록입니다. 등록 정보가 명시적으로 지정되지 않은 경우 기본적으로 `jms` 및 `admin` 서비스가 활성화됩니다.

각 연결 서비스는 특정 인증 및 권한 부여 기능도 지원합니다. 자세한 내용은 82 페이지 "보안 서비스"를 참조하십시오.

포트 매퍼

각 연결 서비스는 호스트 이름(또는 IP 주소)과 포트 번호로 지정되는 특정 포트에 사용할 수 있습니다. 서비스의 정적 포트 번호를 명시적으로 지정하거나 브로커의 **포트 매퍼**를 동적으로 할당할 수 있습니다. 포트 매퍼 자체는 일반적으로 표준 포트 번호 7676에 있는 브로커의 **기본 포트**에 있습니다. 필요할 경우, 브로커 구성 등록 정보 `imq.portmapper.port`를 사용하여 다른 포트 번호로 대체할 수도 있습니다. 기본적으로 각 연결 서비스는 시작 시 포트 매퍼와 함께 자체적으로 등록됩니다. 클라이언트가 브로커에 대한 연결을 생성하면 Message Queue 클라이언트 런타임은 먼저 포트 매퍼에 연결하여 원하는 연결 서비스에 사용할 포트 번호를 요청합니다.

또는 포트 매퍼를 무시하고 imq.*serviceName.protocolType*.port 구성 등록 정보(여기서 *serviceName* 및 *protocolType*은 표 4–1에 나와 있는 특정 연결 서비스를 나타냄)를 사용하여 연결 서비스에 정적 포트 번호를 명시적으로 할당할 수 있습니다. jms, ssljms, admin 및 ssladmin 연결 서비스만 이러한 방식으로 구성할 수 있고, httpjms 및 httpsjms 서비스는 부록 C에 설명된 대로 다른 구성 등록 정보를 사용합니다. 그러나, 일반적으로 정적 포트는 방화벽을 통해 연결을 생성하는 경우(151 페이지 "방화벽을 통해 연결" 참조)처럼 특수한 경우에만 사용되며 일반 용도로는 바람직하지 않습니다.

주 – 두 개 이상의 호스트(예: 컴퓨터에 두 개 이상의 네트워크 카드가 설치된 경우)를 사용할 수 있는 경우에는 브로커 등록 정보를 사용하여 연결 서비스가 바인드해야 할 호스트를 지정할 수 있습니다. imq.hostname 등록 정보는 모든 연결 서비스를 위한 단일 기본 호스트를 지정합니다. 그리고 나서 필요할 경우 imq.*serviceName.protocolType*.hostname(jms, ssljms, admin 또는 ssl 관리 서비스용) 또는 imq.portmapper.hostname(포트 매퍼 자체용)을 사용하여 대체할 수 있습니다.

여러 포트 매퍼 요청을 동시에 받는 경우 이러한 요청은 작업을 기다리는 동안 운영 체제 백로그에 저장됩니다. imq.portmapper.backlog 등록 정보는 이와 같은 백로그 요청의 최대 수를 지정합니다. 이 제한을 초과하면 백로그가 줄어들 때까지 후속 요청이 거부됩니다.

스레드 풀 관리

각 연결 서비스는 다중 스레드 방식으로서, 다중 연결을 지원합니다. 이러한 연결에 필요한 스레드는 브로커에서 각 서비스의 개별 **스레드 풀**로 유지 관리됩니다. 연결에 필요한 스레드는 해당 연결을 지원하는 서비스의 스레드 풀에 추가됩니다.

선택한 스레딩 모델은 스레드가 단일 연결 전용인지 또는 여러 연결에서 공유할지 여부를 지정합니다.

- **전용 모델**에서 브로커에 대한 각 연결에는 두 개의 스레드(받는 메시지용 및 보내는 메시지용)가 필요합니다. 따라서 지원할 수 있는 연결 수는 제한되지만 성능은 향상됩니다.
- **공유 모델**에서 메시지를 보내고 받는 경우 공유 스레드에서 연결을 처리합니다. 각 연결에서는 전용 스레드를 필요로 하지 않기 때문에 이 모델은 사용 가능한 연결 수를 증가시키지만 스레드 관리에 필요한 추가 오버헤드로 인해 성능이 저하됩니다.

브로커의 imq.*serviceName*.threadpool_model 등록 정보는 두 모델 중에서 해당 연결 서비스에 사용할 모델을 지정합니다. 이 등록 정보는 두 가지 문자열 값(dedicated 또는 shared) 중 하나를 사용합니다. 등록 정보를 명시적으로 설정하지 않으면 기본적으로 dedicated로 간주됩니다.

또한 브로커 등록 정보 imq.*serviceName*.min_threads 및 imq.*serviceName*.max_threads를 설정하여 서비스 스레드 풀의 최대 및 최소 스레드 수를 지정할 수 있습니다. 사용 가능한 스레드 수가 지정된 최소 임계값을 초과하는 경우 Message Queue는 최소 임계값에 다시

도달할 때까지 스레드를 종료시켜 여유 스레드를 확보하는 방법으로 메모리 자원을 절약합니다. 로드량이 많은 경우 풀의 최대 수에 도달할 때까지 스레드 수가 증가할 수 있습니다. 이러한 경우 스레드를 사용할 수 있을 때까지 새 연결이 거부됩니다.

공유 스레딩 모델에서는 **분산자 스레드**를 사용하여 스레드를 활성 연결에 할당합니다. 브로커 등록 정보 `imq.shared.connectionMonitor_limit`는 단일 분산자 스레드에서 모니터링할 수 있는 최대 연결 수를 지정합니다. 이 등록 정보 값이 작을수록 스레드를 연결에 더 빨리 할당할 수 있습니다. `imq.ping.interval` 등록 정보는 브로커가 활성 여부를 확인하기 위해 연결을 주기적으로 테스트("핑")하는 시간 간격(초)을 지정합니다. 따라서 시도한 메시지 전송이 실패하기 전에 먼저 연결 실패를 감지할 수 있습니다.

라우팅 서비스

브로커에 클라이언트가 연결되면 메시지 경로 지정 및 전달이 수행될 수 있습니다. 이 단계에서 브로커는 여러 종류의 물리적 대상을 작성 및 관리하여 메시지의 원활한 흐름을 보장하고 자원을 효율적으로 관리합니다. 259 페이지 "라우팅 등록 정보"에 설명된 브로커 구성 등록 정보를 사용하여 응용 프로그램 요구에 맞는 방식으로 이러한 작업을 관리할 수 있습니다.

브로커의 성능과 안정성은 사용 가능한 시스템 자원(메모리 등)과 이러한 자원이 얼마나 효율적으로 활용되는가에 따라 달라집니다. 브로커에서 받는 메시지가 너무 많거나 메모리 부족이 발생하지 않도록 구성 정보를 설정할 수 있습니다. 이러한 등록 정보는 세 가지 서로 다른 수준에서 작동하여 자원이 부족해질 때에도 메시지 서비스의 작동 상태를 유지합니다.

- **시스템 수준 메시지 제한**은 시스템의 모든 물리적 대상에 적용됩니다. 여기에는 브로커에서 보유하는 최대 메시지 수(`imq.system.max_count`)와 해당 메시지가 차지하는 최대 총 바이트 수(`imq.system.max_size`)가 포함됩니다. 이 제한 중 하나에 도달하면 보류 중인 메시지가 제한 이하로 떨어질 때까지 브로커에서 새 메시지를 거부합니다. 또한 개별 메시지의 최대 크기(`imq.message.max_size`)와 만료된 메시지가 확보되는 시간 간격(`imq.message.expiration.interval`)에 대한 제한도 있습니다.
- **개별 대상 제한**은 특정 물리적 대상으로의 메시지 흐름을 규제합니다. 이러한 제한을 제어하는 구성 등록 정보는 15 장에서 설명합니다. 이 등록 정보에는 대상에서 보유하는 메시지 수와 크기, 대상에 대해 만들 수 있는 메시지 생성자와 사용자의 수, 그리고 대상에 전달하기 위해 함께 일괄 처리할 수 있는 메시지 수에 대한 제한이 포함됩니다.

  메시지 생성자를 통한 메시지 전달 속도를 낮추거나 새로운 받는 메시지를 거부하거나 가장 오래되었거나 우선 순위가 가장 낮은 기존 메시지를 삭제하여 메모리 제한에 응답하도록 대상을 구성할 수 있습니다. 이러한 방법으로 대상에서 삭제되는 메시지를 곧바로 삭제하지 않고 사용 불능 메시지 대기열로 옮길 수

있습니다. 브로커 등록 정보 imq.destination.DMQ.truncateBody는 사용 불능 메시지 대기열에 전체 메시지 본문을 저장할지 또는 헤더와 등록 정보 데이터만 저장할지 여부를 제어합니다.

응용 프로그램 개발과 테스트의 편의를 위해, 메시지 생성자 또는 사용자가 존재하지 않는 대상에 액세스를 시도할 때마다 새로운 물리적 대상을 자동으로 생성하도록 메시지 브로커를 구성할 수 있습니다. 표 14–3에 요약된 브로커 등록 정보는 방금 설명한 내용과 비슷하지만 관리상 생성된 대상이 아닌 **자동 생성 대상**에 적용됩니다.

- **시스템 메모리 임계값**은 브로커가 메모리 과부하를 방지하기 위해 조치 수위를 점차적으로 높이는 메모리 사용 수준을 정의합니다. 이러한 사용 수준은 네 가지로 정의됩니다.
  - **초록:**사용 가능한 메모리 충분
  - **노랑:**브로커 메모리 부족 시작
  - **주황:**브로커 메모리 부족
  - **빨강:**브로커 메모리 없음

    이러한 수준을 정의하는 메모리 사용률은 브로커 등록 정보 imq.green.threshold, imq.yellow.threshold, imq.orange.threshold 및 imq.red.threshold를 통해 지정되며, 각각의 기본값은 초록인 경우 0%, 노랑인 경우 80%, 주황인 경우 90%, 빨강인 경우 98%입니다.

    메모리 사용량이 한 수준에서 다음 수준으로 올라감에 따라, 브로커는 먼저 활성 메모리의 메시지를 영구 저장소로 옮기고 나서 비지속성 메시지의 생성자를 억제한 후 브로커의 메시지 흐름을 중지시킴으로써 단계적으로 응답합니다. 이러한 두 조치 모두 브로커 성능을 떨어뜨립니다. 메시지 생성 억제는 전달되는 각 일괄 처리의 크기를 등록 정보 imq.*resourceState* .count에 지정된 메시지 수로 제한함으로써 이루어집니다. 여기서 *resourceState*는 각각 green , yellow, orange 또는 red입니다.

이러한 시스템 메모리 임계값이 트리거되면 시스템 수준과 대상 메시지 제한이 너무 높게 설정되어 있다는 뜻입니다. 메모리 임계값이 항상 잠재적 메모리 과부하를 제때에 잡을 수 있는 것은 아니므로, 메모리 임계값만으로 메모리 사용을 제어하기보다는 메모리 자원을 최적화하도록 시스템 수준과 대상 제한을 다시 구성해야 합니다.

지속성 서비스

오류 발생 시 브로커를 복구하려면 메시지 전달 작업 상태를 다시 작성해야 합니다. 그러기 위해서는 브로커가 상태 정보를 **영구 데이터 저장소**에 저장해야 합니다. 브로커는 다시 시작할 때 저장된 데이터를 사용하여 대상 및 영구 가입을 다시 작성하고 지속성 메시지를 복구하며 열린 트랜잭션을 롤백하고 전달되지 못한 메시지의 라우팅 테이블을 다시 작성합니다. 그런 다음 메시지 전달을 다시 시작할 수 있습니다.

Message Queue는 파일 기반 및 JDBC 기반 지속성 모듈을 모두 지원합니다(그림 4–1 참조). 파일 기반 지속성에서는 개별 파일을 사용하여 영구 데이터를 저장합니다. JDBC 기반 지속성에서는 JDBC™(Java Database Connectivity) 인터페이스를 사용하여 브로커를

JDBC 호환 데이터 저장소에 연결합니다. 일반적으로 파일 기반 지속성은 JDBC 기반보다 빠르지만, JDBC 호환 저장소가 제공하는 중복 및 관리 제어 기능을 선호하는 사용자도 있습니다. 브로커 구성 등록 정보 `imq.persist.store`(표 14–4 참조)는 사용할 두 가지 형태의 지속성 중 하나를 지정합니다.

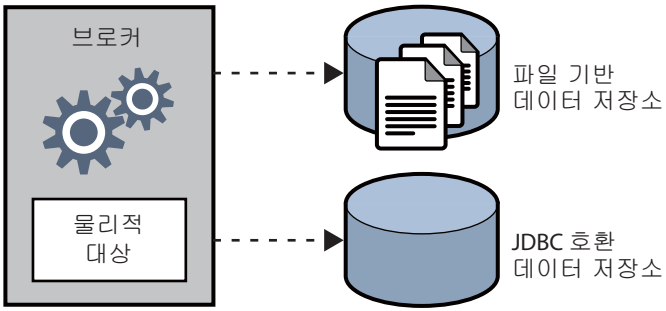


그림 4–1 영구 데이터 저장소

파일 기반 지속성

기본적으로 Message Queue에서는 파일 기반 영구 데이터 저장소를 사용하는데, 이 저장소에서는 개별 파일이 메시지, 대상, 영구 가입 및 트랜잭션과 같은 영구 데이터를 저장합니다. 파일 기반 지속성과 관련된 브로커 구성 등록 정보는 263 페이지 “파일 기반 지속성”에 나와 있습니다.

파일 기반 저장소는 데이터 저장소가 속한 브로커 인스턴스의 이름(*instanceName*)으로 식별되는 디렉토리에 있습니다.

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

stances 디렉토리의 위치는 부록 A를 참조하십시오. 브로커의 각 대상마다 해당 대상으로 전달된 메시지가 들어 있는 고유의 하위 디렉토리가 있습니다.

주 – 영구 데이터 저장소에는 중요 정보나 소유 정보를 포함하는 메시지가 있을 수 있기 때문에 `.../instances/` *instanceName*`/fs350/` 디렉토리를 인증되지 않은 액세스로부터 보호해야 합니다. 93 페이지 “영구 데이터 보안”를 참조하십시오.

메시지 이외의 모든 영구 데이터는 개별 파일(대상용 파일, 영구 가입용 파일 및 트랜잭션 상태 정보용 파일)에 저장됩니다. 대부분의 메시지는 가변 크기 레코드로 구성되는 단일 파일에 저장됩니다. 메시지를 추가 및 제거할 때 단편화를 줄이려면 이 파일을 압축할 수 있습니다(121 페이지 “물리적 대상 압축” 참조). 또한, 특정 임계값 크기를 초과하는 메시지는 가변 크기 레코드 파일보다는 고유의 개별 파일에 저장됩니다. 브로커 등록 정보 `imq.persist.file.message.max_record_size`를 사용하여 이 임계값 크기를 구성할 수 있습니다.

브로커는 더 이상 필요하지 않은 파일을 삭제하는 대신 나중에 다른 메시지에 다시 사용할 수 있도록 대상 디렉토리의 사용 가능 파일 풀로 반환하는 방식으로 개별 메시지 파일의 파일 풀을 관리합니다. 브로커 등록 정보 `imq.persist.file.destination.message.filepool.limit`는 풀의 최대 파일 수를 지정합니다. 대상의 개별 메시지 파일 수가 이 제한을 초과하면 파일이 더 이상 필요하지 않을 때 풀로 반환되지 않고 삭제됩니다.

파일을 파일 풀로 반환하는 경우 이전 내용을 삭제하지 않고 단순히 파일에 재사용 태그를 표시하여 저장소 공간을 절약할 수 있습니다. `imq.persist.file.message.filepool.cleanratio` 브로커 등록 정보를 사용하면 단순히 재사용 태그를 표시하지 않고 "깨끗한(빈)" 상태로 유지해야 하는 각 대상의 파일 풀에서 파일 비율을 지정할 수 있습니다. 이 값을 높게 설정할수록 파일 풀에 필요한 공간은 줄어들지만, 파일을 풀로 반환 시 파일 내용을 비우는 데 필요한 오버헤드는 더 커집니다. 브로커의 `imq.persist.file.message.cleanup` 등록 정보가 `true`이면 풀의 모든 파일이 브로커 종료 시 비워지므로 clean 상태로 유지됩니다. 이렇게 하면 저장소 공간은 절약되지만 종료 프로세스 속도는 느려집니다.

데이터를 영구 저장소에 기록할 때 운영 체제는 데이터를 동기식으로 쓰거나 "느리게(비동기식으로)" 씁니다. 저속 저장소는 데이터가 영구 저장소에 기록되지 않았는데도 브로커에서 기록되었다고 믿는 경우 시스템이 충돌하면 데이터가 손실될 수 있습니다. 성능을 조금 낮추더라도 절대 안정성을 유지하려는 경우에는 브로커 등록 정보 `imq.persist.file.sync.enabled`를 `true`로 설정하여 모든 데이터를 동기식으로 기록해야 합니다. 이 경우, 충돌 후 시스템이 다시 백업되면 데이터를 사용할 수 있으므로 브로커는 작업을 안정적으로 다시 시작할 수 있습니다. 그러나, 데이터가 손실되지 않았어도 클러스터화된 브로커에 현재 공유 데이터가 없으므로 클러스터의 다른 브로커는 사용할 수 없습니다.

JDBC 기반 지속성

파일 기반 지속성을 사용하는 대신 JDBC 호환 드라이버를 통해 액세스할 수 있는 데이터 저장소에 액세스하도록 브로커를 설정할 수 있습니다. 이 경우 해당 JDBC 관련 브로커 구성 등록 정보를 설정하고 데이터베이스 관리자 유틸리티(`imqdbmgr`)를 사용하여 적절한 스키마를 갖는 데이터베이스를 만들어야 합니다. 자세한 내용은 91 페이지 "JDBC 기반 저장소 구성"을 참조하십시오.

265 페이지 "JDBC 기반 지속성"에는 JDBC 데이터베이스를 사용하도록 브로커를 구성하기 위한 등록 정보가 나와 있습니다. 이러한 등록 정보는 각 브로커 인스턴스의 인스턴스 구성 파일(`config.properties`)이나 브로커 유틸리티(`imqbrokerd`) 또는 데이터베이스 관리자 유틸리티(`imqdbmgr`)의 `-D` 명령줄 옵션을 사용하여 지정할 수 있습니다.

`imq.persist.jdbc.driver` 등록 정보는 데이터베이스에 연결하는 데 사용할 JDBC 드라이버의 Java 클래스 이름을 제공합니다. 또한 기존 데이터베이스 연결(`imq.persist.jdbc.opendburl`), 새 데이터베이스 작성(`imq.persist.jdbc.createdburl`) 및 데이터베이스 연결 닫기(`imq.persist.jdbc.closedburl`) URL을 지정하는 등록 정보도 있습니다.

imq.persist.jdbc.user 및 imq.persist.jdbc.password 등록 정보는 데이터베이스에 액세스하는 데 사용되는 사용자 이름 및 비밀번호를 제공합니다. imq.persist.jdbc.needpassword는 비밀번호가 필요한지 여부를 지정하는 부울 플래그입니다. 보안상의 이유로 비밀번호는 -passfile 명령줄 옵션을 통해 지정된 비밀번호 파일에만 지정해야 합니다. imqbrokerd 및 imqdbmgr 명령은 대화식으로 비밀번호를 묻습니다. 이와 마찬가지로, 명령줄에서 imqbrokerd 명령의 -dbuser 옵션 또는 imqdbmgr 명령의 -u 옵션을 사용하여 사용자 이름을 입력할 수도 있습니다.

여러 브로커 인스턴스에서 공유하는 JDBC 데이터베이스에서 구성 등록 정보 imq.persist.jdbc.brokerid는 각각의 인스턴스에 대해 데이터베이스 테이블 이름에 추가할 고유한 인스턴스 식별자를 지정합니다. 보통 한 브로커 인스턴스에 대해서만 데이터를 저장하는 내장 데이터베이스에는 이 식별자가 필요하지 않습니다. 나머지 JDBC 관련 구성 등록 정보는 각 데이터베이스 테이블마다 한 개의 등록 정보로 구성된 데이터베이스 스키마를 생성하는 SQL 코드를 사용자 정의하는 데 사용됩니다. 예를 들어, imq.persist.jdbc.table.IMQSV35 등록 정보는 버전 테이블, imq.persist.jdbc.table.IMQCCREC35는 구성 변경 레코드 테이블, 그리고 imq.persist.jdbc.table.IMQDEST35는 대상 테이블을 생성하기 위한 SQL 명령을 제공합니다. 전체 목록은 표 14–6을 참조하십시오.

주 – 데이터베이스 시스템마다 필요한 정확한 SQL 구문이 다르기 때문에 자세한 내용은 데이터베이스 공급업체에서 제공하는 설명서를 참조하십시오.

보안 서비스

Message Queue는 사용자 액세스 제어(인증 및 권한 부여)와 암호화를 위한 보안 서비스를 제공합니다.

- **인증**은 검증된 사용자만 브로커에 연결할 수 있도록 합니다.
- **권한 부여**는 자원을 액세스하고 특정 작업을 수행할 권한이 있는 사용자 또는 그룹을 지정합니다.
- **암호화**는 연결을 통한 전달 중 메시지가 훼손되지 않게 보호합니다.

Message Queue 관리자는 브로커가 사용자를 인증하고 작업 권한을 부여하는 데 필요한 정보를 설정해야 합니다. 보안 서비스와 관련된 브로커 등록 정보는 268 페이지 “보안 등록 정보”에 나와 있습니다. 부울 등록 정보 imq.accesscontrol.enabled는 액세스 제어가 브로커 수준에 적용되는지 여부를 제어하는 마스터 스위치 역할을 합니다. 보다 자세한 제어를 위해 imq.*serviceName* .accesscontrol.enabled 등록 정보를 설정하여 연결 서비스에 대해 이 설정을 대체할 수 있습니다. 여기서 *serviceName*은 표 4–1에 표시된 대로 연결 서비스의 이름입니다. 예를 들면 imq.httpjms.accesscontrol.enabled와 같습니다.

그림 4–2는 브로커에서 인증 및 권한 부여 서비스를 제공하는 데 필요한 구성 요소를 보여줍니다. 이러한 서비스는 메시징 시스템 사용자에 대한 정보(이름, 비밀번호 및 그룹 멤버쉽)가 들어 있는 **사용자 저장소**에 따라 다릅니다. 또한, 사용자나 그룹에 특정 작업에

대한 권한을 부여하기 위해 브로커는 사용자나 그룹이 수행할 수 있는 작업을 지정하는 **액세스 제어 등록 정보 파일**을 참조합니다. 구성 등록 정보 `imq.accesscontrol.file.filename`을 사용하여 브로커 전체에 대해 단일 액세스 제어 등록 정보 파일을 지정하거나, `imq.`*serviceName*`.accesscontrol.file.filename`을 사용하여 단일 연결 서비스에 대해 지정할 수 있습니다.

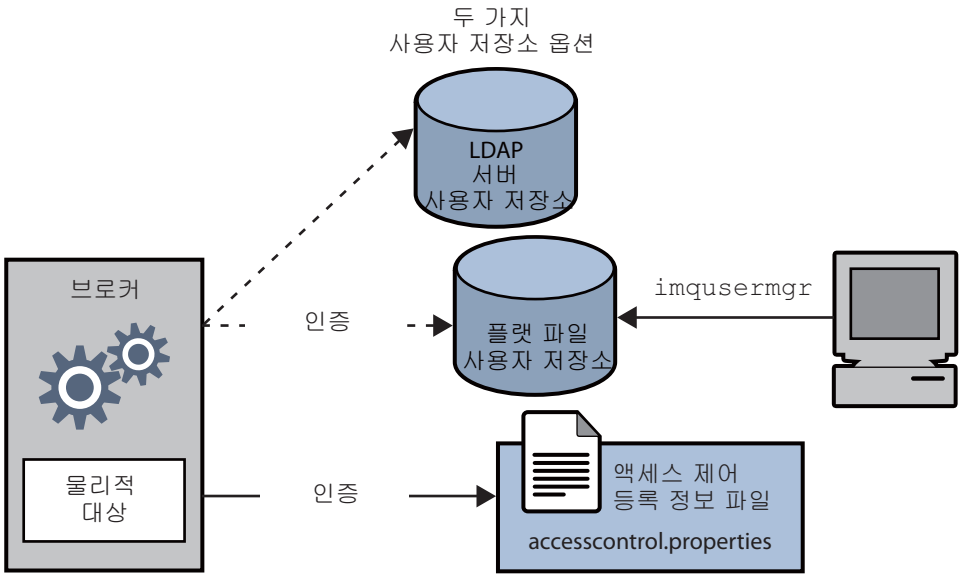


그림 4–2 보안 지원

그림 4–2에 표시된 대로, Message Queue 서비스와 함께 제공되는 플랫 파일 사용자 저장소에 사용자 데이터를 저장하거나 기존 LDAP(Lightweight Directory Access Protocol) 저장소에 플러그 인할 수 있습니다.

- 플랫 파일 저장소를 선택하는 경우에는 Message Queue 사용자 관리자 유틸리티(`imqusermgr`)를 사용하여 저장소를 관리해야 합니다. 이 옵션은 기본적으로 제공되며 사용하기 쉽습니다.
- 기존 LDAP 서버를 사용하려면 LDAP 공급업체에서 제공하는 도구를 사용하여 사용자 저장소를 채우고 관리합니다. 또한, 브로커에서 LDAP 서버에 사용자 및 그룹에 대한 정보를 쿼리할 수 있도록 브로커 인스턴스 구성 파일의 등록 정보를 설정해야 합니다.

브로커의 `imq.authentication.basic.user_repository` 등록 정보는 어떤 저장소 유형을 사용할지를 지정합니다. 일반적으로 LDAP 저장소는 확장성이 중요한 경우나 여러 브로커에서 저장소를 공유해야 하는 경우(예: 브로커 클러스터를 사용 중인 경우)에 권장됩니다. 플랫 파일 또는 LDAP 사용자 저장소 설정에 대한 자세한 내용은 127 페이지 “사용자 인증”을 참조하십시오.

인증

브로커에 연결을 요청하는 클라이언트는 브로커가 사용자 저장소에 저장된 항목과 비교할 사용자 이름 및 비밀번호를 제공해야 합니다. 클라이언트에서 브로커로 전송된 비밀번호는 기본 64 인코딩(플랫 파일 저장소용) 또는 메시지 다이제스트(MD5) 해싱(LDAP 저장소용)을 사용하여 인코딩됩니다. 이 옵션은 브로커 전체의 경우 `imq.authentication.type` 등록 정보를 통해 제어되거나 특정 연결 서비스의 경우 `imq.`*serviceName*`.authentication.type` 등록 정보를 통해 제어됩니다. `imq.authentication.client.response.timeout` 등록 정보는 인증 요청 시간 초과 간격을 설정합니다.

150 페이지 "비밀번호 파일"에 설명한 대로 비밀번호를 대화식으로 묻지 않고 **비밀번호 파일**에 비밀번호를 입력하도록 선택할 수도 있습니다. 부울 브로커 등록 정보 `imq.passfile.enabled`는 이 옵션을 제어합니다. 이 등록 정보가 true이면 `imq.passfile.dirpath` 및 `imq.passfile.name` 등록 정보에서 비밀번호 파일의 디렉토리 경로와 파일 이름을 제공합니다. `imq.imqcmd.password` 등록 정보(비밀번호 파일에 내장될 수 있음)는 관리자가 명령 유틸리티(`imqcmd`)를 사용하여 브로커, 연결 서비스, 연결, 물리적 대상, 영구 가입 및 트랜잭션을 관리하는 데 필요한 인증용 비밀번호를 지정합니다.

LDAP 기반 사용자 저장소를 사용하는 경우에는 LDAP 조회의 다양한 측면을 구성하는 데 사용할 수 있는 다양한 브로커 등록 정보가 있습니다. LDAP 서버 자체의 주소(호스트 이름 및 포트 번호)는 `imq.user_repository.ldap.server`를 통해 지정됩니다. `imq.user_repository.ldap.principal` 등록 정보는 LDAP 저장소에 바인드하는 데 필요한 고유 이름을 제공하며, `imq.user_repository.ldap.password`는 관련 비밀번호를 제공합니다. 기타 등록 정보는 개별 사용자 및 그룹 검색을 위한 디렉토리 기반 및 선택적 JNDI 필터, 사용자 및 그룹 이름의 공급자별 속성 식별자 등을 지정합니다. 자세한 내용은 268 페이지 "보안 등록 정보"를 참조하십시오.

권한 부여

인증된 사용자는 다양한 Message Queue 관련 작업을 수행할 수 있는 권한을 부여받게 됩니다. Message Queue 관리자는 사용자 그룹을 정의하고 그룹 내 개별 사용자 멤버쉽을 할당할 수 있습니다. 기본 액세스 제어 등록 정보 파일은 `admin`이라는 단일 그룹만 명시적으로 참조합니다(130 페이지 "그룹" 참조). 이 그룹의 사용자는 `admin` 연결 서비스용 연결 권한이 있으므로 대상 생성, 브로커 모니터링 및 제어와 같은 관리 작업을 수행할 수 있습니다. 다른 그룹으로 정의된 사용자는 기본적으로 `admin` 서비스에 연결할 수 없습니다.

사용자가 어떤 작업을 수행하려 하면 브로커는 사용자 저장소에 있는 사용자 이름 및 그룹 멤버쉽을 액세스 제어 등록 정보 파일에 있는 해당 작업에 액세스하도록 지정된 사용자 이름 및 그룹 멤버쉽과 대조 확인합니다. 액세스 제어 등록 정보 파일은 다음 작업에 대한 사용자 또는 그룹의 권한을 지정합니다.

- 브로커에 연결
- 대상에 액세스: 지정된 대상 또는 모든 대상에 대해 사용자, 생성자 또는 대기열 브라우저 생성

- 대상 자동 생성

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 설정하여 연결 수준에서 보안을 제공합니다.

SSL 기반 Message Queue 연결 서비스를 사용하려면 키 도구 유틸리티(`imqkeytool`)를 사용하여 공용 키/개인 키 쌍을 생성합니다. 이 유틸리티는 자체 서명된 인증서에 공용 키를 내장하고 이를 Message Queue 키 저장소에 저장합니다. 키 저장소 자체에 비밀번호가 설정되어 있으므로 잠금을 해제하려면 시작 시 `imq.keystore.password` 등록 정보에 지정된 키 저장소 비밀번호를 입력해야 합니다. 키 저장소 잠금이 해제되면 브로커는 연결을 요청하는 모든 클라이언트에게 인증서를 전달할 수 있습니다. 그리고 나서 클라이언트는 인증서를 사용하여 브로커에게 보낼 암호화된 연결을 설정합니다.

`imq.audit.enabled` 브로커 등록 정보는 Message Queue 브로커 로그 파일에 대한 감사 레코드 로깅을 제어합니다. 자세한 내용은 152 페이지 “감사 기록”을 참조하십시오.

모니터링 서비스

브로커는 응용 프로그램 및 브로커 성능을 모니터하고 진단할 구성 요소를 포함합니다. 여기에는 다음 항목이 포함됩니다.

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

그림 4–3은 일반 체계를 보여줍니다. 모니터링 서비스 구성을 위한 브로커 등록 정보는 271 페이지 “모니터링 등록 정보”에 나와 있습니다.

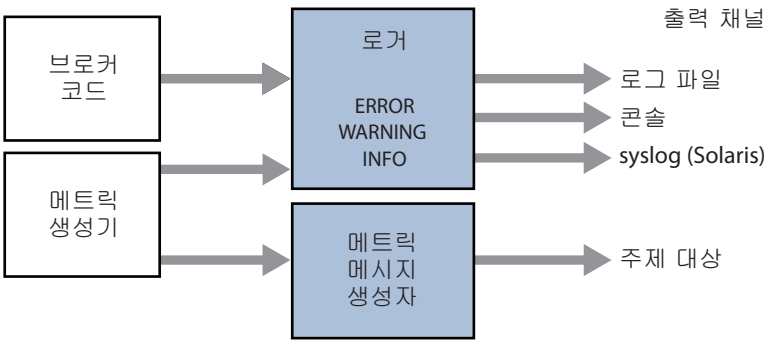


그림 4–3 모니터링 지원

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. 부울 브로커 등록 정보 `imq.metrics.enabled`는 이러한 정보를 기록할지 여부를 제어합니다. `imq.metrics.interval`는 빈도를 지정합니다.

로거

로거는 브로커 코드 및 메트릭 생성자가 생성한 정보를 가져와서 오류 발생 시 표준 출력(콘솔), 로그 파일, `syslog` 데몬 프로세스(Solaris 플랫폼용)에 해당 정보를 기록합니다.사용할 로그 파일은 `imq.log.file.dirpath` 및 `imq.log.file.filename` 브로커 등록 정보를 통해 식별됩니다. `imq.log.console.stream`는 콘솔 출력을 `stdout`로 전달할지 `stderr`로 전달할지 여부를 지정합니다.

`imq.log.level` 등록 정보는 로거에서 수집하는 메트릭 정보 범주 `ERROR`, `WARNING` 또는 `INFO`를 제어합니다. 각 수준에는 상위 수준이 포함됩니다. 예를 들어, `WARNING`을 로깅 수준으로 지정한 경우 오류 메시지도 기록됩니다. `imq.log.console.output` 및 `imq.log.file.output` 등록 정보는 지정된 범주 중에서 콘솔과 로그 파일에 각각 기록할 범주를 제어합니다. 그러나, 이 경우 범주에는 상위 수준이 포함되지 **않습니다**. 예를 들어, 오류와 경고를 로그 파일에 모두 기록하고 정보 메시지를 콘솔에 기록하려는 경우에는 `imq.log.file.output`을 `ERROR|WARNING`으로, `imq.log.console.output`을 `INFO`로 명시적으로 설정해야 합니다. Solaris 플랫폼의 경우 또 하나의 등록 정보인 `imq.log.syslog.output`은 `syslog` 데몬에 기록할 메트릭 정보 범주를 지정합니다. 사용 불능 메시지를 삭제하거나 사용 불능 메시지 대기열로 옮길 때 기록할지 여부를 지정하는 `imq.destination.logDeadMsgs` 등록 정보도 있습니다.

로그 파일의 경우 파일을 닫고 출력을 새 파일로 롤오버하는 지점을 지정할 수 있습니다. 로그 파일이 지정된 크기(`imq.log.file.rolloverbytes`)에 이르거나 지정된 기간(`imq.log.file.rolloversecs`)에 이르면 해당 로그 파일이 저장되고 새 로그 파일이 만들어집니다.

로깅 관련 추가 브로커 등록 정보는 271 페이지 "모니터링 등록 정보"을 참조하십시오. 로거 구성 방법과 로거를 사용하여 성능 정보를 얻는 방법에 대한 자세한 내용은 181 페이지 "브로커 로깅 구성 및 사용"을 참조하십시오.

메트릭 메시지 생성자(엔터프라이즈판)

메트릭 메시지 생성자는 주기적으로 메트릭 생성자로부터 정보를 받아서 해당 정보를 **메트릭 메시지**에 기록합니다. 그런 다음, 이 메트릭 메시지를 메시지에 포함된 메트릭 정보 유형에 따라 다양한 메트릭 주제 대상 중 하나로 전송합니다(표 4–2 참조). Message Queue 클라이언트는 메시지를 사용하거나 포함되어 있는 메트릭 데이터를 처리할 수 있습니다. 이렇게 하면 개발자는 사용자 정의 모니터링 도구를 작성하여 메시징 응용 프로그램을 지원할 수 있습니다. 각 메트릭 메시지 유형에서 보고하는 메트릭 수량에 대한 자세한 내용은 **Java 클라이언트용 Message Queue 개발 안내서**를 참조하십시오.

표 4–2 메트릭 주제 대상

주제 이름	메트릭 정보 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.`*queueName*	지정된 대기열의 대상 메트릭
`mq.metrics.destination.topic.`*topicName*	지정된 주제의 대상 메트릭

브로커 등록 정보 `imq.metrics.topic.enabled` 및 `imq.metrics.topic.interval`은 메시지를 메트릭 주제 대상으로 전송할지 여부와 전송 빈도를 각각 제어합니다. `imq.metrics.topic.timetolive` 및 `imq.metrics.topic.persist` 등록 정보는 이러한 메시지의 수명과 지속 여부를 지정합니다.

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

- 메시지 유형
- 메시지를 전송한 브로커의 주소(호스트 이름 및 포트 번호)
- 메트릭 샘플을 가져온 시간

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

브로커 등록 정보 설정

다음 두 가지 방법 중 하나로 브로커의 구성 등록 정보를 지정할 수 있습니다.

- 브로커 구성 파일 편집
- 명령줄에서 등록 정보 값 직접 입력

다음 두 절에서는 두 가지 브로커 구성 방법에 대해 설명합니다.

구성 파일

브로커 구성 파일에는 브로커 구성을 위한 등록 정보 설정이 포함되어 있습니다. 이러한 설정은 디렉토리 내에 보존되며, 이 디렉토리는 사용 중인 운영 체제 플랫폼에 따라 다릅니다. 자세한 내용은 부록 A를 참조하십시오. 이 디렉토리에는 다음과 같은 파일이 저장됩니다.

- 시작 시 로드되는 **기본 구성 파일** `default.properties`. 이 파일은 편집할 수 없지만, 이 파일을 참조하여 기본 설정을 확인하고 변경할 등록 정보의 정확한 이름을 찾을 수 있습니다.
- Message Queue가 설치될 때 지정된 등록 정보를 포함하는 **설치 구성 파일** `install.properties`. 설치한 후에는 이 파일을 편집할 수 없습니다.

또한, 각각의 개별 브로커 인스턴스마다 아래에 설명된 대로 고유의 **인스턴스 구성 파일**이 있습니다. 클러스터에서 브로커 인스턴스를 연결하는 경우에는 **클러스터 구성 파일**을 사용하여 클러스터 구성 정보를 지정해야 할 수도 있습니다. 자세한 내용은 275 페이지 “클러스터 구성 등록 정보”를 참조하십시오.

시작할 때 브로커는 다양한 구성 파일의 등록 정보 값을 병합합니다. 그림 4–4에 표시된 대로 파일은 인스턴스 구성 파일에 지정된 값이 설치 구성 파일에 지정된 값을 대체하고 이 값이 기본 구성 파일의 값을 대체하는 계층 형태로 이루어집니다. 계층의 맨 위에서는 `imqbrokerd` 명령의 명령줄 옵션을 사용하여 구성 파일에 지정된 등록 정보 값을 수동으로 대체할 수 있습니다.

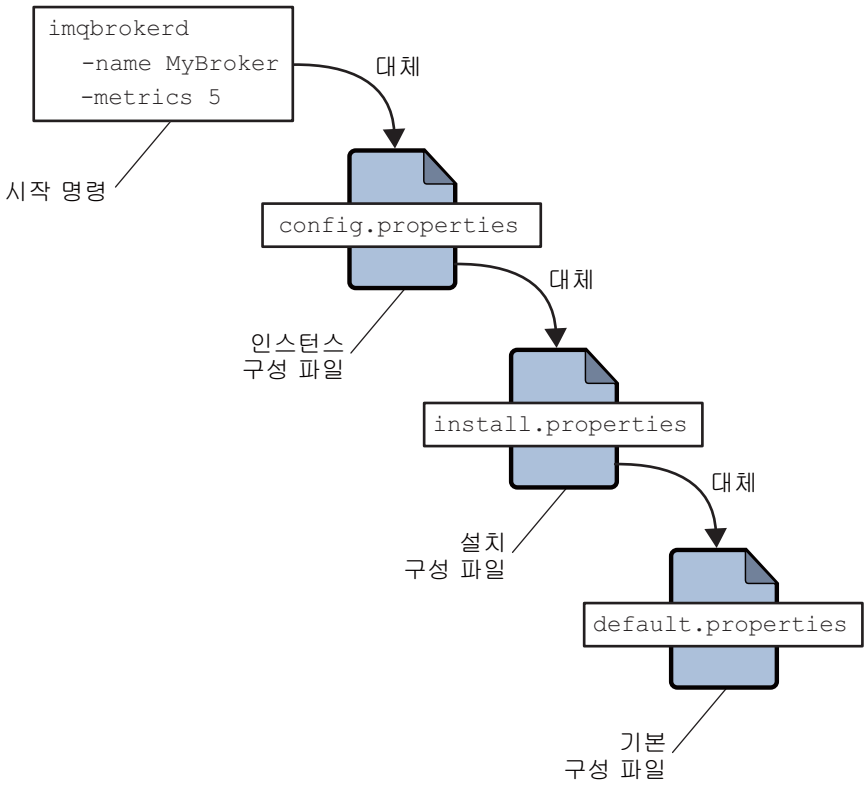


그림 4–4 브로커 구성 파일

인스턴스 구성 파일 편집

브로커를 처음 실행하면 해당 특정 브로커 인스턴스에 대한 구성 등록 정보를 포함하는 인스턴스 구성 파일이 만들어집니다. 인스턴스 구성 파일의 이름은 `config.properties`이며 해당 파일이 속해 있는 브로커 인스턴스의 이름으로 식별되는 디렉토리에 저장됩니다.

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

`instances` 디렉토리의 위치는 부록 A를 참조하십시오. 파일이 아직 없는 경우에는 브로커를 시작할 때 `-name` 옵션을 사용해야 합니다. Message Queue에서 파일을 만드는 데 사용할 수 있는 인스턴스 이름을 지정하려면 238 페이지 “브로커 유틸리티”를 참조하십시오.

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

인스턴스 구성 파일은 브로커 인스턴스에서 관리되며 Message Queue 관리 유틸리티를 사용하여 구성을 변경할 때 수정됩니다. 또한 인스턴스 구성 파일을 사용자가 직접 편집하여 브로커의 작동 및 자원 사용을 사용자 정의할 수 있습니다. 그렇게 하려면 `instances/` *instanceName* 디렉토리의 소유자이거나 `root`로 로그인하여 디렉토리의 액세스 권한을 변경해야 합니다.

브로커는 시작할 때만 인스턴스 구성 파일을 읽습니다. 브로커의 구성을 영구적으로 변경하려면 브로커를 종료하고 파일을 편집한 다음 브로커를 다시 시작해야 합니다. 파일(또는 구성 파일)의 등록 정보 정의에서는 다음 구문을 사용합니다.

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
```

사용 가능한 브로커 구성 등록 정보 및 기본값은 75 페이지 “브로커 서비스” 및 14 장을 참조하십시오.

명령줄에서 구성 옵션 설정

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

시작할 때 브로커 유틸리티(`imqbrokerd`)를 사용하여 브로커 인스턴스를 시작합니다. 이 명령의 `-D` 옵션을 사용하여 브로커 구성 등록 정보와 값을 지정할 수 있습니다. 자세한 내용은 68 페이지 “브로커 시작” 및 238 페이지 “브로커 유틸리티”를 참조하십시오. 서비스 관리자 유틸리티(`imqsvcadmin`)를 사용하여 브로커를 Windows 서비스로 시작하는 경우 `-args` 옵션을 사용하여 시작 구성 등록 정보를 지정할 수 있습니다. 254 페이지 “서비스 관리자 유틸리티”를 참조하십시오.

브로커 인스턴스를 실행하는 동안 특정 브로커 등록 정보를 변경할 수도 있습니다. 실행 중인 브로커의 구성을 수정하려면 명령 유틸리티의 `imqcmd update bkr` 명령을 사용합니다. 99 페이지 “브로커 등록 정보 업데이트” 및 244 페이지 “브로커 관리”를 참조하십시오.

영구 데이터 저장소 구성

A broker's 브로커의 영구 데이터 저장소는 물리적 대상, 영구 가입, 메시지, 트랜잭션 및 확인에 대한 정보가 저장되어 있습니다. Message Queue 브로커는 기본적으로 파일 기반 영구 저장소를 사용하도록 구성되어 있지만, JDBC 호환 드라이버를 통해 액세스할 수 있는 데이터 저장소에 플러그 인 되도록 다시 구성할 수도 있습니다.브로커 구성 등록 정보 `imq.persist.store`(표 14–4 참조)는 사용할 두 가지 형태의 지속성 중 하나를 지정합니다.

이 절에서는 영구 저장소를 사용하도록 브로커를 구성하는 방법에 대해 설명합니다. 이 장은 다음 항목으로 구성되어 있습니다.

- 91 페이지 "파일 기반 저장소 구성"
- 91 페이지 "JDBC 기반 저장소 구성"
- 93 페이지 "영구 데이터 보안"

파일 기반 저장소 구성

브로커 인스턴스를 만들면 파일 기반 데이터 저장소가 자동으로 생성됩니다. 이 저장소는 브로커의 인스턴스 디렉토리에 있습니다. 정확한 위치는 부록 A를 참조하십시오.

기본적으로 Message Queue는 디스크에 대한 비동기 쓰기 작업을 수행합니다. 운영 체제는 성능 향상을 위해 이러한 작업을 버퍼링할 수 있습니다. 하지만 쓰기 작업 중에 운영 체제에 예기치 않은 오류가 발생할 경우 메시지가 손실될 수 있습니다. 성능이 조금 저하되더라도 안정성을 향상시키려면 데이터를 비동기식으로 쓰도록 브로커 등록 정보 `imq.persist.file.sync`를 설정할 수 있습니다. 이 등록 정보에 대한 자세한 내용은 80 페이지 "파일 기반 지속성" 및 표 14–5를 참조하십시오.

브로커 인스턴스를 시작할 때 `imqbrokerd` 명령의 `-reset` 옵션을 사용하여 파일 시스템을 지울 수 있습니다. 이 옵션 및 하위 옵션에 대한 자세한 내용은 238 페이지 "브로커 유틸리티"를 참조하십시오.

JDBC 기반 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.또한, 데이터베이스 관리자를 사용하여 데이터베이스에서 손상된 테이블을 삭제하거나 다른 데이터베이스를 데이터 저장소로 사용할 수도 있습니다. 자세한 내용은 251 페이지 "데이터베이스 관리자 유틸리티"를 참조하십시오.

주 – Oracle 및 PointBase 데이터베이스 제품의 구성 예를 사용할 수 있습니다. 이 파일의 위치는 플랫폼마다 다르며부록 A의 관련 테이블에서 "응용 프로그램 및 구성 예"에 나와 있습니다. 또한 PointBase 내장 버전, PointBase 서버 버전 및 Oracle의 예는 인스턴스 구성 파일 `config.properties`에서 주석 처리된 값으로 제공됩니다.

▼ JDBC 기반 데이터 저장소를 구성하는 방법

1 브로커의 구성 파일에서 JDBC 기반 등록 정보를 설정합니다.

관련 등록 정보는 81 페이지 "JDBC 기반 지속성"과 표 14–6에서 설명합니다. 특히, 브로커의 `imq.persist.store` 등록 정보를 `jdbc`로 설정해야 합니다(263 페이지 "지속성 등록 정보" 참조).

2 다음 위치에 JDBC 드라이버 `.jar` 파일의 사본 또는 심볼릭 링크를 넣습니다.

- Solaris:

  ```
  /usr/share/lib/imq/ext/
  ```

- Linux:

  ```
  /opt/sun/mq/share/lib/
  ```

- Windows:

  ```
  IMQ_VARHOME\\lib\\ext
  ```

예를 들어, Solaris 시스템에서 PointBase를 사용하는 경우 다음 명령을 사용하여 드라이버의 `.jar` 파일을 해당 위치로 복사합니다.

```
% cp j2eeSDKInstallDirectory/pointbase/lib/pointbase.jar /usr/share/lib/imq/ext
```

대신, 다음 명령은 심볼링 링크를 만듭니다.

```
% ln -s j2eeSDKID/lib/pointbase/pointbase.jar /usr/share/lib/imq/ext
```

3 Message Queue 지속성에 필요한 데이터베이스 스키마를 만듭니다.

`imqdbmgr create all` 명령(내장 데이터베이스용) 또는 `imqdbmgr create tbl` 명령(외부 데이터베이스용)을 사용합니다(251 페이지 "데이터베이스 관리자 유틸리티" 참조).

a. `imqdbmgr`이 위치한 디렉토리로 변경합니다.

- Solaris:

  ```
  cd /usr/bin
  ```

- Linux:

```
cd /opt/sun/mq/bin
```

- Windows:

```
cd IMQ_HOME\\bin
```

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 이름과 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.

영구 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 파일 기반 또는 JDBC 기반 데이터 저장소의 데이터를 보호하는 방법에 대해 설명합니다.

파일 기반 저장소 보안

파일 기반 지속성을 사용하는 브로커는 영구 데이터를 플랫폼마다 다른 위치에 있는 플랫 파일 데이터 저장소에 씁니다(부록 A 참조).

…`/instances/` *instanceName*`/fs350/`

여기서 *instanceName*은 브로커 인스턴스를 식별하는 이름입니다.

instanceName`/fs350/` 디렉토리는 브로커 인스턴스가 처음으로 시작될 때 생성됩니다. 이 디렉토리 보안 절차는 브로커가 실행 중인 운영 체제 플랫폼에 따라 다릅니다.

- Solaris와 Linux에서는 디렉토리의 권한이 브로커 인스턴스를 시작한 사용자의 파일 모드 작성 마스크(`umask`)에 따라 결정됩니다. 따라서, 마스크를 적절하게 설정하여 브로커 인스턴스를 시작하고 지속성 파일 읽기 권한을 제한할 수 있습니다. 또는, 관리자(수퍼유저)가 `instances` 디렉토리에 대한 권한을 `700`으로 설정하여 영구 데이터를 보호할 수 있습니다.
- Windows에서는 Windows 운영 체제에서 제공한 메커니즘을 사용하여 디렉토리 권한을 설정할 수 있습니다. 여기에는 일반적으로 디렉토리에 대한 등록 정보 대화 상자 열기가 포함됩니다.

JDBC 기반 저장소 보안

JDBC 기반 지속성을 사용하는 브로커는 영구 데이터를 JDBC 호환 데이터베이스에 씁니다. 데이터베이스 서버에서 관리하는 데이터베이스(예: Oracle)의 경우 Message Queue 데이터베이스 테이블(이름이 `IMQ`로 시작하는 테이블)에 액세스하는 데 사용할 사용자 이름과 비밀번호를 만드는 것이 좋습니다. 데이터베이스에서 개별 테이블을 보호할 수 없는 경우 Message Queue 브로커 전용 데이터베이스를 만듭니다. 사용자 이름/비밀번호 액세스 작성 방법에 대한 자세한 내용은 데이터베이스 공급업체에서 제공하는 설명서를 참조하십시오.

브로커가 데이터베이스 연결을 여는 데 필요한 사용자 이름과 암호는 브로커 구성 등록 정보에서 제공될 수 있습니다. 그러나 `imqbrokerd` 명령의 `-dbuser` 및 `-dbpassword` 옵션을 사용하여 명령줄 옵션으로 제공하는 것이 보다 안전합니다(238 페이지 "브로커 유틸리티" 참조).

브로커가 데이터베이스의 JDBC 드라이버를 통해 직접 액세스하는 내장 데이터베이스의 경우 93 페이지 "파일 기반 저장소 보안"에서 설명한 대로, 일반적으로 영구 데이터가 저장되는 디렉토리에 대한 파일 권한을 설정하여 보안을 제공합니다. 하지만 브로커와 데이터베이스 관리자 유틸리티를 모두 사용하여 데이터베이스를 읽기 및 쓰기 가능하게 하려면 해당 브로커와 유틸리티를 모두 동일한 사용자가 실행해야 합니다.

◆ ◆ ◆ 5 장

5 브로커 관리

이 장에서는 imqcmd 유틸리티를 사용하여 브로커 및 해당 서비스를 관리하는 방법에 대해 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 95 페이지 “전제 조건” 95 페이지 “전제 조건”
- 96 페이지 “imqcmd 유틸리티 사용” 96 페이지 “imqcmd 유틸리티 사용”
- 98 페이지 “브로커 정보 표시”
- 99 페이지 “브로커 등록 정보 업데이트”
- 100 페이지 “브로커 일시 중지 및 다시 시작”
- 101 페이지 “브로커 종료 및 다시 시작”
- 102 페이지 “브로커 메트릭 표시”
- 103 페이지 “연결 서비스 관리”
- 107 페이지 “연결 정보 얻기”
- 108 페이지 “영구 가입 관리”
- 109 페이지 “트랜잭션 관리”

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

- 브로커에서 물리적 대상 관리. 물리적 대상 만들기, 표시, 업데이트 및 완전 삭제 방법과 사용 불능 메시지 대기열 사용 방법은 6 장을 참조하십시오.
- 브로커에 대한 보안 설정. 사용자 인증, 액세스 제어, 암호화, 비밀번호 파일, 감사 로깅 등과 같은 항목에 대한 자세한 내용은 7 장을 참조하십시오.

전제 조건

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

- imqbrokerd 유틸리티 명령을 사용하여 브로커를 시작합니다. 브로커가 실행될 때까지는 다른 명령줄 유틸리티를 사용할 수 없습니다.

- Message Queue™ 관리자를 설정할지 아니면 기본 계정을 사용할지를 결정합니다. 관리 명령을 사용하려면 사용자 이름과 비밀번호를 지정해야 합니다.

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

  작업 시스템을 설정하는 경우 관리자에 대한 인증과 권한 부여를 설정해야 합니다. 파일 기반 사용자 저장소 설정 또는 LDAP 디렉토리 서버 사용 구성에 대한 자세한 내용은 7 장을 참조하십시오. 작업 환경에서는 보안을 위해 기본이 아닌 사용자 이름과 비밀번호를 사용하는 것이 좋습니다.
- 브로커에 대한 보안 연결을 사용하려면 대상 브로커 인스턴스에서 ssladmin 서비스를 설정하고 활성화합니다. 자세한 내용은 141 페이지 “메시지 암호화”를 참조하십시오.

imqcmd 유틸리티 사용

imqcmd 유틸리티를 사용하여 브로커와 해당 서비스를 관리할 수 있습니다.

imqcmd 명령의 구문, 하위 명령 및 옵션에 대한 자세한 내용은 13 장을 참조하십시오. 물리적 대상 관리에 대한 자세한 내용은 별도의 장으로 제공되는 15 장을 참조하십시오.

도움말 표시

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

예를 들어, 다음 명령은 imqcmd에 대한 도움말을 표시합니다.

```
imqcmd -H
```

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

사용자 이름 및 비밀번호 지정

각 imqcmd 하위 명령은 사용자 저장소에 대해 인증되므로 사용자 이름과 비밀번호가 필요합니다. -h 또는 -H 옵션을 사용하여 도움말을 표시하는 명령과, -v 옵션을 사용하여 제품 버전을 표시하는 명령만 제외됩니다.

사용자 이름 지정

-u 옵션을 사용하여 관리자 이름을 지정합니다. 관리자 이름을 생략하면 이름을 묻는 명령 프롬프트가 표시됩니다. 예를 들어, 다음 명령은 기본 브로커에 대한 정보를 표시합니다.

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

이 장의 예에서는 쉽게 이해할 수 있도록 기본 사용자 이름 admin을 -u 옵션의 인수로 표시합니다. 작업 환경에서는 사용자 정의 사용자 이름을 사용하게 됩니다.

비밀번호 지정

다음 방법 중 하나를 사용하여 비밀번호를 지정합니다.

- 비밀번호 파일(passfile)을 만들고 해당 파일에 비밀번호를 입력합니다. 명령줄에서 -passfile 옵션을 사용하여 비밀번호 파일의 이름을 입력합니다.
- 비밀번호를 묻는 명령 프롬프트를 표시하게 합니다.

Message Queue의 이전 버전에서는 imqcmd 명령줄에서 -p 옵션을 사용하여 비밀번호를 지정할 수 있었습니다. 이 옵션은 더 이상 사용되지 않으며 향후 버전에서는 제거됩니다.

브로커 이름 및 포트 지정

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

원격 호스트에서 실행 중인 브로커나 기본 포트가 아닌 포트에서 수신 중인 브로커 또는 둘 다에 명령을 실행하는 경우 -b 옵션을 사용하여 브로커의 호스트 및 포트를 지정해야 합니다.

예

이 절의 예에서는 imqcmd 사용 방법을 설명합니다.

첫 번째 예에서는 localhost의 포트 7676에서 실행 중인 브로커의 등록 정보를 나열하므로 -b 옵션이 필요하지 않습니다. 이 명령에서는 기본 관리자 이름(admin)을 사용하고 비밀번호를 생략했기 때문에 비밀번호를 묻는 명령 프롬프트가 표시됩니다.

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

다음 예에서는 myserver 호스트의 1564 포트에서 실행 중인 브로커의 등록 정보를 나열합니다. 사용자 이름은 aladdin입니다. (이 명령을 실행하려면 사용자 저장소를 업데이트하여 사용자 이름 aladdin을 admin 그룹에 추가해야 함).

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

다음 예에서는 localhost의 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 myPassfile이라는 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 옵션을 지정하면 서비스가 구성되고 시작된 경우 imqcmd에서 ssladmin 서비스가 사용됩니다.

브로커 정보 표시

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

다음은 query bkr 하위 명령 구문입니다.

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

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에 해당하는 현재 등록 정보 설정을 나열합니다. 지정한 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                                    imqbroker
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 명령 구문입니다.

imqcmd update bkr [-b *hostName*:
portNumb er]-o *attribute=value*

```
 [[-o attribute=value1]
...]
```

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에 해당하는 지정된 속성을 변경합니다. 예를 들어, 다음 명령은 대기열 대상의 자동 생성을 해제합니다.

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

등록 정보에 대한 자세한 내용은 14 장을 참조하십시오.

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 수신 또는 디스패치할 수 없습니다.

그러나 브로커를 일시 중지하더라도 `admin` 연결 서비스는 지연되지 않으므로 브로커에 대한 메시지 흐름을 규제하는 데 필요한 관리 작업은 수행할 수 있습니다. 또한 브로커를 일시 중지해도 `cluster` 연결 서비스는 지연되지 않습니다. 그러나 클러스터 내의 메시지 전달은 클러스터의 여러 브로커에서 수행하는 전달 기능에 따라 다릅니다. 따라서 클러스터에서 브로커를 일시 중지하면 일부 메시지 트래픽이 느려질 수 있습니다.

다음은 `pause bkr` 하위 명령 구문입니다.

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

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

다음 명령은 `myhost`의 포트 `1588`에서 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 106 페이지 “연결 서비스 일시 중지 및 다시 시작” 및 119 페이지 “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 명령 구문입니다.

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

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

다음 명령은 localhost의 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 명령 구문입니다.

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

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

다음 명령은 ctrlsrv의 포트 1572에서 실행 중인 브로커를 종료합니다.

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

restart bkr 하위 명령을 사용하여 브로커를 종료하고 다시 시작합니다. 다음은 restart bkr 하위 명령 구문입니다.

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

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.

브로커 메트릭 표시

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

다음은 `metrics bkr` 하위 명령 구문입니다.

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

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에 대한 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 결과가 출력됩니다.

```
-------------------------------------------------------
 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
```

브로커에서 수집하여 보고하는 데이터에 대한 자세한 설명은 304 페이지 “브로커 전체 메트릭”을 참조하십시오.

연결 서비스 관리

imqcmd 유틸리티에는 다음과 같은 연결 서비스 관리 작업을 수행할 수 있는 하위 명령이 포함되어 있습니다.

- 103 페이지 "연결 서비스 나열"
- 104 페이지 "연결 서비스 정보 표시"
- 105 페이지 "연결 서비스 등록 정보 업데이트"
- 105 페이지 "연결 서비스 메트릭 표시"
- 106 페이지 "연결 서비스 일시 중지 및 다시 시작"

브로커는 응용 프로그램 클라이언트 및 관리 클라이언트 모두와의 연결을 지원합니다. 현재 Message Queue 브로커에서 사용 가능한 연결 서비스는 표 5–1에서 확인할 수 있습니다. 표에 나타난 것과 같이 각 서비스는 NORMAL(응용 프로그램 클라이언트) 또는 ADMIN(관리 클라이언트) 중에서 사용하는 서비스 유형과 기본 전송 프로토콜에 연결됩니다.

표 5–1 Message Queue 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)

imqcmd 하위 명령을 사용하면 연결 서비스 전체를 관리하거나 특정 연결 서비스를 관리할 수 있습니다. 하위 명령의 대상이 특정 서비스인 경우 -n 옵션을 사용하여 표 5–1의 서비스 이름 열에 나열된 이름 중 하나를 지정합니다.

연결 서비스 나열

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

다음은 list svc 하위 명령 구문입니다.

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

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 연결 서비스를 모두 나열합니다.

다음 명령은 localhost의 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 명령 구문입니다.

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

query svc 하위 명령은 기본 브로커 또는 지정한 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 imqcmd에서 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 명령 구문입니다.

```
imqcmd update svc -n serviceName [-b
hostName:portNumber]
          -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 자세한 내용은 257 페이지 “연결 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.

다음은 metrics 하위 명령 구문입니다.

```
imqcmd metrics svc -n serviceName [-b
hostName:portNumber] [-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.

- **ttl** 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 ttl -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
```

`imqcmd`를 사용하여 연결 서비스 메트릭을 보고하는 방법에 대한 자세한 내용은 305 페이지 “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 명령 구문입니다.

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

이 하위 명령은 기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 실행 중인 지정된 서비스를 일시 중지합니다.예를 들어, 다음 명령은 기본 프로커에서 실행 중인 `httpjms` 서비스를 일시 중지합니다.

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

서비스를 일시 중지하면 다음과 같은 현상이 나타납니다.

- 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 명령 구문입니다.

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

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

연결 정보 얻기

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

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

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

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 들어, 다음 명령은 기본 브로커에 대한 모든 연결을 나열합니다.

```
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:portNumber]
```

이 하위 명령은 기본 브로커 또는 지정한 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 사용하면 다음 중 하나를 수행하여 브로커의 영구 가입을 관리할 수 있습니다.

- 영구 가입 나열
- 영구 가입에 대한 모든 메시지 제거
- 영구 가입 완전 삭제

영구 가입은 클라이언트에 영구로 등록된 주제에 가입하는 것입니다. 여기에는 고유한 아이디가 있으며, 사용자가 비활성 상태인 동안에도 브로커에서 가입에 대한 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 명령 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   Durable Sub
                            Messages      State
----------------------------------------------------------------
myDurable   myClientID       1           INACTIVE
```

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

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

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

destroy dur 하위 명령은 지정한 클라이언트 식별자에 해당하는 지정된 영구 가입을 완전 삭제합니다. 다음은 destroy dur 하위 명령 구문입니다.

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

예를 들어, 다음 명령은 영구 가입 myDurable과 클라이언트 아이디 myClientID를 완전 삭제합니다.

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

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 아이디**가있습니다. 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 모니터하고 롤백 또는 완결해야 합니다.

브로커가 추적 중인 모든 트랜잭션을 나열하려면 `list txn` 명령을 사용합니다. 다음은 `list tx` 하위 명령 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, 사용자 이름, 메시지 또는 확인 응답의 수, 작성 시간을 반환합니다. 예를 들면 다음과 같습니다.

```
-----------------------------------------------------------------
Transaction ID  State     User name   # Msgs/   Creation time
                                      # Acks
-----------------------------------------------------------------

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 false로 설정되어 있으면(표 14–2 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 클라이언트 아이디, 연결 아이디, 분산 트랜잭션 아이디(XID) 등의 여러 추가 값을 볼 수 있습니다. 다음은 `query txn` 하위 명령 구문입니다.

```
imqcmd query txn -n transactionID
```

예를 들어, 다음 예는 아래와 같은 출력을 생성합니다.

```
imqcmd query txn -n 64248349708800

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 명령을 사용하여 분산 트랜잭션을 완결 또는 롤백할 수 있습니다. 앞에서 설명한 것과 같이, PREPARED 상태에 있는 트랜잭션만 완결 또는 롤백할 수 있습니다.

다음은 commit 하위 명령 구문입니다.

```
imqcmd commit txn -n transactionID
```

예를 들면 다음과 같습니다.

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

다음은 rollback 하위 명령 구문입니다.

```
imqcmd rollback txn -n transactionID
```

자세한 내용은 표 14–2의 imq.transaction.autorollback 등록 정보를 참조하십시오.

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

◆ ◆ ◆ 6 장

6 물리적 대상 관리

이 장에서는 imqcmd 유틸리티를 사용하여 물리적 대상을 관리하는 방법에 대해 설명합니다. Message Queue™ 메시지는 브로커의 물리적 대상을 통해 사용자 클라이언트로 라우팅됩니다. 브로커는 물리적 대상에 연결된 메모리와 영구 저장소를 관리하고 해당 동작을 설정합니다.

브로커 클러스터에서 브로커 하나에 물리적 대상을 만들면 클러스터가 해당 물리적 대상을 모든 다른 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 장은 다음 내용으로 구성되어 있습니다.

- 114 페이지 "명령 유틸리티 사용"
- 115 페이지 "물리적 대상 만들기"
- 116 페이지 "물리적 대상 나열"
- 117 페이지 "물리적 대상 정보 표시"
- 118 페이지 "물리적 대상 등록 정보 업데이트"
- 119 페이지 "물리적 대상 일시 중지 및 다시 시작"
- 120 페이지 "물리적 대상 제거"
- 120 페이지 "물리적 대상 완전 삭제"
- 121 페이지 "물리적 대상 압축"
- 123 페이지 "사용 불능 메시지 대기열 사용"

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

물리적 대상에 대한 소개는 **Message Queue 기술 개요**를 참조하십시오.

주 – 클라이언트 응용 프로그램은 물리적 대상과 상호 작용할 때마다 `Destination` 객체를 사용합니다. 공급자 독립성과 이식성을 위해 클라이언트는 대상 관리 객체라는 관리자 생성 대상 객체를 일반적으로 사용합니다. 8 장에서 설명한 대로 클라이언트 응용 프로그램에 사용하도록 관리 대상 객체를 구성할 수 있습니다.

명령 유틸리티 사용

Message Queue 명령 유틸리티(`imqcmd`)를 사용하여 물리적 대상을 관리합니다. `imqcmd` 명령 구문은 다른 브로커 서비스 관리에 사용할 때의 구문과 동일합니다.

`imqcmd`와 해당 하위 명령과 옵션에 대한 자세한 내용은 13 장에서 확인할 수 있습니다.

하위 명령

표 6–1에는 이 장에서 설명하는 `imqcmd` 하위 명령이 나열되어 있습니다. 이러한 하위 명령에 대한 자세한 내용은 246 페이지 “물리적 대상 관리”를 참조하십시오.

표 6–1 명령 유틸리티의 물리적 대상 하위 명령

하위 명령 및 인수	설명
`compact dst`	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 등록 정보를 업데이트합니다.

물리적 대상 만들기

물리적 대상을 만들려면 imqcmd create 하위 명령을 사용합니다. 다음은 create 하위 명령 구문입니다.

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

예를 들어, 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 대상을 만들 때 다음을 지정해야 합니다.

- 물리적 대상 유형, t(주제) 또는 q(대기열).
- 물리적 대상 이름. 이름 지정 규칙은 다음과 같습니다.
  - 이름은 영숫자만 포함할 수 있으며공백을 포함할 수 없습니다.
  - 이름은 영문자, 밑줄 문자(_) 또는 달러 기호($)로 시작할 수 있습니다. mq 문자열로는 시작할 수 없습니다.

  기본값이 아닌 모든 물리적 대상 등록 정보 값

물리적 대상을 업데이트할 때 등록 정보도 설정할 수 있습니다.

많은 물리적 대상 등록 정보가 브로커 메모리 자원과 메시지 흐름에 영향을 미칩니다. 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. 이러한 제한은 브로커 구성 등록 정보를 통해 제어되는 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 15 장을 참조하십시오.

자동 생성 대상의 경우 브로커의 인스턴스 구성 파일에 기본 등록 정보 값을 설정합니다. 자동 생성 등록 정보에 대한 자세한 내용은 표 14–3을 참조하십시오.

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 list dst 하위 명령을 사용하여 브로커의 모든 물리적 대상을 나열합니다. 다음은 list dst 하위 명령 구문입니다.

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

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

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, myHost의 포트 4545에서 실행 중인 브로커의 모든 물리적 대상을 나열하려면 다음 명령을 입력합니다.

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

주제만 포함하도록 t 대상 유형을 지정하지 않는 한 mq.sys.dmq 사용 불능 메시지 대기열에 대한 정보가 항상 다른 물리적 대상과 함께 표시됩니다.

물리적 대상 정보 표시

물리적 대상의 현재 등록 정보에 대한 정보를 보려면 query dst 하위 명령을 사용합니다. 다음은 query dst 하위 명령 구문입니다.

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

이 명령은 지정한 유형과 이름의 대상에 대한 정보를 나열합니다.예를 들어, 다음 명령은 XQueue 대기열 대상에 대한 정보를 표시합니다.

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

다음과 같은 결과가 출력됩니다.

```
-----------------------------------
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(118 페이지 "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 명령 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 15–1에 나열된 등록 정보 중 하나일 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 정보 목록은 15 장을 참조하십시오.

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

주 – 사용 불능 메시지 대기열은 등록 정보가 다른 대상과 약간 차이가 나는 특수한 물리적 대상입니다. 자세한 내용은 123 페이지 "사용 불능 메시지 대기열 사용"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 대상을 압축하려면 먼저 일시 중지해야 합니다.

물리적 대상에서 주고 받는 메시지 전달을 일시 중지하려면 pause dst 하위 명령을 사용합니다. 다음은 pause dst 하위 명령 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 명령 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 대상에 저장된 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 명령 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 명령 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 불능 메시지 대기열은 완전 삭제할 수 없습니다.

물리적 대상 압축

파일 기반 데이터 저장소를 메시지의 영구 저장소로 사용하는 경우, 필요할 때마다 디스크 사용률을 모니터하고 디스크를 압축할 수 있습니다.

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

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 모니터하고 사용률이 떨어지면 사용 가능한 디스크 공간을 확보합니다.

물리적 대상 디스크 사용률 모니터링

물리적 대상의 디스크 사용률을 모니터하려면 다음과 같은 명령을 사용합니다.

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

다음과 같은 결과가 출력됩니다.

```
--------------------------------------
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 명령 구문입니다.

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

이 하위 명령은 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(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

사용 불능 메시지 대기열 mq.sys.dmq는 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 useDMQ를 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 관리

Message Queue 명령 유틸리티(imqcmd)를 사용하면 약간의 차이는 있겠지만 다른 대기열을 관리하는 것처럼 사용 불능 메시지 대기열을 관리할 수 있습니다. 예를 들어, 사용 불능 메시지 대기열은 시스템에서 생성되기 때문에 사용자가 생성, 일시 중지 또는 완전 삭제할 수 없습니다. 또한 표 6–3에서 표시된 것처럼 사용 불능 메시지 대기열의 기본값은 경우에 따라 일반 대기열의 기본값과 다를 수도 있습니다.

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 1000입니다. 다른 대기열의 기본값은 -1(무제한)입니다.
maxNumProducers	사용 불능 메시지 대기열에 적용되지 않습니다.
maxTotalMsgBytes	사용 불능 메시지 대기열의 기본값은 10MB입니다. 다른 대기열의 기본값은 -1(무제한)입니다.
isLocalOnly	브로커 클러스터에서 사용 불능 메시지 대기열은 항상 전역 물리적 대상이며 이 등록 정보는 false로 영구 설정됩니다.

메시지 내용

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

사용 불능 대기열의 크기를 줄이거나 사용 불능 메시지를 복원하지 않으려면 imq.destination.DMQ.truncateBody 브로커 등록 정보를 true로 설정하는 것을 고려해 보십시오.

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

그러면 메시지 본문이 삭제되고 헤더와 등록 정보 데이터만 유지됩니다.

사용 불능 메시지 로깅 사용

사용 불능 메시지 로깅은 기본적으로 비활성화됩니다. 사용 불능 메시지 로깅을 사용하면 브로커에서 다음 이벤트를 기록할 수 있습니다.

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

다음 명령은 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.

◆ ◆ ◆ 7 장

7 보안 관리

사용자 인증에 사용할 사용자 저장소를 구성하고 액세스 제어를 정의하며 클라이언트 브로커 통신을 암호화하는 SSL(Secure Socket Layer) 연결 서비스를 구성하고 브로커를 시작할 때 사용할 비밀번호 파일을 설정함으로써 보안을 관리합니다.

이 장은 다음 내용으로 구성되어 있습니다.

- 127 페이지 "사용자 인증"
- 135 페이지 "사용자 권한 부여: 액세스 제어 등록 정보 파일"
- 141 페이지 "메시지 암호화"
- 150 페이지 "비밀번호 파일"
- 152 페이지 "감사 기록"

사용자 인증

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

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.

저장소 유형은 다음 중 하나가 될 수 있습니다.

- Message Queue™와 함께 제공되는 플랫 파일 저장소

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

  사용자 저장소의 설정 및 관리에 대한 자세한 내용은 128 페이지 "플랫 파일 사용자 저장소 사용"을 참조하십시오.

- LDAP 서버

  LDAP v2 또는 v3 프로토콜을 사용하는 기존 또는 새 LDAP 디렉토리 서버가 될 수 있습니다. 플랫 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 133 페이지 "사용자 저장소에 LDAP 서버 사용"을 참조하십시오.

플랫 파일 사용자 저장소 사용

Message Queue에서는 플랫 파일 사용자 저장소와 플랫 파일 사용자 저장소를 채우고 관리할 때 사용할 수 있는 명령줄 도구인 사용자 관리자 유틸리티(`imqusermgr`)를 제공합니다. 다음 절에서는 플랫 파일 사용자 저장소에 대해 설명하고 사용자 관리자 유틸리티를 사용하여 해당 저장소를 채우고 관리하는 방법을 설명합니다.

사용자 저장소 만들기

플랫 파일 사용자 저장소는 인스턴스마다 다릅니다. 기본 사용자 저장소(`passwd`)는 사용자가 시작하는 각 브로커 인스턴스에 대해 자동으로 만들어집니다. 이 사용자 저장소는 저장소와 연관된 브로커 인스턴스의 이름으로 식별되는 디렉토리에 저장됩니다(부록 A 참조).

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

저장소는 이 두 항목을 사용하여 만들어집니다. 표 7–1의 각 행은 하나의 항목을 나타냅니다.

표 7–1 사용자 저장소 초기 항목

사용자 이름	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 `guest` 사용자 항목을 사용하면 클라이언트가 기본 `guest` 사용자 이름과 비밀번호를 사용하여 브로커 인스턴스에 연결할 수 있습니다.
- 초기 `admin` 사용자 항목을 사용하면 `imqcmd` 명령을 사용하여 기본 `admin` 사용자 이름과 비밀번호로 브로커 인스턴스를 관리할 수 있습니다. 비밀번호를 변경하려면 이 초기 항목을 업데이트해야 합니다(132 페이지 "기본 관리자 비밀번호 변경" 참조).

다음 절에서는 플랫 파일 사용자 저장소를 채우고 관리하는 방법을 설명합니다.

사용자 관리자 유틸리티

Message Queue 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 13 장을 참조하십시오.

사용자 관리자를 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 필요합니다. 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 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 명령은 지정한(또는 기본) 브로커 인스턴스 저장소에서 지정한 사용자를 삭제합니다. 하위 명령의 구문은 다음과 같습니다.

  `delete` [`-i` *instanceName*] `-u` *userName* [`-s`] [`-f`]
- `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 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 7–2에서는 `imqusermgr` 명령의 옵션이 나열되어 있습니다.

표 7–2 imqusermgr 옵션

옵션	설명
-a *activeState*	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 *instanceName*	명령이 적용될 브로커 인스턴스 이름을 지정합니다. 지정하지 않으면 기본 인스턴스 이름 imqbroker인 것으로 가정합니다.
-p *passwd*	사용자의 비밀번호를 지정합니다.
-g *group*	사용자 그룹을 지정합니다. 유효한 값에는 admin, user, anonymous가 있습니다.
-s	자동 모드를 설정합니다.
-u *userName*	사용자 이름을 지정합니다.
-v	버전 정보를 표시합니다. 명령줄에 있는 명령은 실행되지 않습니다.

그룹

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

- **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 사용자에 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 이름을 사용하지 않는(클라이언트 응용 프로그램에서 사용해야 할 실제 사용자 이름을 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. user 그룹과 비교하여 이 그룹의 액세스 권한을 제한하거나, 배포 시 그룹에서 사용자를 제거해야 합니다.

사용자의 그룹을 변경하려면, 사용자 항목을 삭제한 후 그 사용자에 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 135 페이지 “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 이름을 추가하지 않게 됩니다.

사용자 이름 및 비밀번호 형식

사용자 이름과 비밀번호는 다음과 같은 지침을 따라야 합니다.

- 사용자 이름에는 별표(*), 쉼표(,), 콜론(:), 줄바꿈 또는 캐리지 리턴 문자가 포함되지 않아야 합니다.
- 사용자 이름 또는 비밀번호는 한 문자 이상이어야 합니다.
- 사용자 이름 또는 비밀번호에 공백이 포함된 경우에는 이름 또는 비밀번호 전체를 따옴표로 묶어야 합니다.
- 명령 쉘에서 명령줄에 입력할 수 있는 최대 문자 수가 제한되어 있는 것 외에는 비밀번호 또는 사용자 이름에 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 sesame인 사용자 Katharine을 추가합니다.

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

저장소에서 사용자를 삭제하려면 delete 하위 명령을 사용합니다. 예를 들어, 다음 명령은 Bob이라는 사용자를 삭제합니다.

```
imqusermgr delete -u Bob
```

사용자의 비밀번호 또는 상태를 변경하려면 update 하위 명령을 사용합니다. 예를 들어, 다음 명령은 Katharine의 비밀번호를 aladdin으로 변경합니다.

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

한 사용자 또는 모든 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 `admin`의 기본 비밀번호를 자신만 아는 비밀번호로 변경해야 합니다. 다음 명령은 `mybroker` 브로커 인스턴스의 기본 관리자 비밀번호를 `admin`에서 `grandpoobah`로 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`)가 올바르게 작동해야 하고 이전 비밀번호는 실패해야 합니다.

비밀번호를 변경한 후에 관리 콘솔을 포함한 Message Queue 관리 도구를 사용하려면 새 비밀번호를 입력해야 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. 이 등록 정보를 사용하면 사용자가 브로커 인스턴스에 연결하거나 메시징 작업을 수행하려 할 때마다 브로커 인스턴스가 LDAP 서버로부터 사용자 및 그룹에 대한 정보를 쿼리할 수 있습니다.

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 A를 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 인코딩하는 데 base-64(`basic`)를 사용할지 또는 MD5(`digest`)를 사용할지 여부를 결정합니다. 사용자 저장소에 LDAP 디렉토리를 사용하는 경우에는 인증 유형을 `basic`으로 설정해야 합니다. 예를 들면 다음과 같습니다.

```
imq.authentication.type=basic
```

3 LDAP 액세스를 제어하는 브로커 등록 정보도 설정해야 합니다. 이러한 등록 정보는 브로커의 인스턴스 구성 파일에 저장됩니다. 등록 정보는 82 페이지 "보안 서비스"에서 설명되고 268 페이지 "보안 등록 정보"에 요약되어 있습니다.

Message Queue는 JNDI API를 사용하여 LDAP 디렉토리 서버와 통신합니다. 이러한 등록 정보에 사용되는 구문과 용어에 대한 자세한 내용은 JNDI 설명서를 참조하십시오. Message Queue는 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.

4 **필요한 경우에는 액세스 제어 등록 정보 파일에서 사용자/그룹과 규칙을 편집해야 합니다. 액세스 제어 등록 정보 파일 사용에 대한 자세한 내용은 135 페이지 "사용자 권한 부여: 액세스 제어 등록 정보 파일"을 참조하십시오.**

5 **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 등록 정보 `imq.user_repository.ldap.ssl.enabled`를 `true`로 설정합니다.

  여러 LDAP 디렉토리 서버를 사용하는 경우 `ldap://`를 사용하여 각 디렉토리 서버를 추가로 지정합니다. 예를 들면 다음과 같습니다.

  `imq.user_repository.ldap.server` = *myHost*`:7878 ldap://` *otherHost*`:7878` …

  추가된 각 디렉토리 서버를 공백으로 구분합니다. 목록의 모든 디렉토리 서버는 기타 LDAP 관련 등록 정보에 대해 동일한 값을 사용해야 합니다.

관리자에 대한 액세스 제어 설정

관리자를 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.

▼ 관리자 설정 방법

1 **브로커 등록 정보 `imq.accesscontrol.enabled`를 `true`(기본값)로 설정하여 액세스 제어 파일을 사용 가능하게 합니다.**

`imq.accesscontrol.enabled` 등록 정보는 액세스 제어 파일을 사용 가능하게 합니다.

2 **액세스 제어 파일 `accesscontrol.properties`를 엽니다. 파일 위치는 부록 A에 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 이름을 입력합니다.**

connection.ADMIN.allow.user= *userName*[[,*userName2*] …]

4 **그룹에 Message Queue 관리자 권한을 부여하려면 다음과 같이 그룹 이름을 입력합니다.**

connection.ADMIN.allow.group= *groupName*[[,*groupName2*] …]

사용자 권한 부여: 액세스 제어 등록 정보 파일

access control properties file(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.

ACL 파일은 사용자 정보가 플랫 파일 사용자 저장소에 있는지 또는 LDAP 사용자 저장소에 있는지 여부와 상관 없이 모두 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 새로운 설정이 적용됩니다. 파일을 편집한 후 브로커를 다시 시작할 필요가 없습니다.

액세스 제어 등록 정보 파일 작성

ACL 파일은 인스턴스에 고유합니다. 브로커 인스턴스를 시작할 때마다 기본 파일 accesscontrol.properties가 인스턴스 디렉토리에 생성됩니다. 파일에 대한 경로 형식은 다음과 같습니다(부록 A 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 생성 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 개의 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
resourceType	`connection`, `queue` 또는 `topic` 중 하나입니다.
resourceVariant	*resourceType*에서 지정한 유형의 인스턴스입니다. 예를 들면 `myQueue`가 있습니다. 와일드카드 문자(*)를 사용하여 모든 연결 서비스 유형 또는 모든 물리적 대상을 나타낼 수 있습니다.
operation	값은 표현할 액세스 규칙의 종류에 따라 달라집니다.
access	One of the following: `allow` 또는 `deny` 중 하나입니다.
principalType	One of the following: `user` 또는 `group` 중 하나입니다. 자세한 내용은 130 페이지 "그룹"을 참조하십시오.

표 7–3 액세스 규칙의 구문 요소 *(계속)*

요소	설명
principals	규칙의 왼쪽에 지정한 액세스를 갖는 사람입니다. `principalType`이 `user`이면 개별 사용자 또는 사용자 목록(쉼표로 구분)이 될 수 있고, `principalType`이 `group`이면 한 그룹 또는 그룹 목록(쉼표로 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 다음을 참조하십시오.

`http://java.sun.com/j2se/1.4/docs/guide/intl/faq.html`

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 `tq1`에 전송할 수 없습니다.

```
queue.*.produce.allow.user=*
queue.tq1.produce.deny.user=Bob
```

- 명시된 *principal*로 지정한 액세스 권한은 * *principal*로 지정한 액세스 권한을 대체합니다. 다음 규칙은 `tq1`에서 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 `tq1`에 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 `tq1`에 메시지를 생성할 수 없습니다. User의 다른 모든 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 `q1`을 찾아볼 수 없습니다.

```
queue.q1.browse.allow.user=Bob
queue.q1.browse.deny.user=Bob
```

다음 두 규칙은 User 그룹이 `q5`에서 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 `NORMAL`과 `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 `admin`이 사용자 관리자 유틸리티를 통해 만들어집니다. 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 전송, 주제에 메시지 게시, 대기열에서 메시지 수신, 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
resourceType	`queue` 또는 `topic`일 수 있습니다.
resourceVariant	물리적 대상 이름 또는 모든 대기열이나 모든 주제를 나타내는 모든 물리적 대상(*)입니다.
operation	`produce`, `consume` 또는 `browse` 중 하나일 수 있습니다.
access	`allow` 또는 `deny`일 수 있습니다.
principalType	`user` 또는 `group`일 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 모든 구성원에 대해 `Admissions` 주제에 가입하지 못하게 합니다.

```
topic.Admissions.consume.deny.group=user
```

자동 생성 물리적 대상에 대한 액세스 제어

ACL 등록 정보 파일의 마지막 섹션에는 브로커에서 물리적 대상을 자동 생성하는 사용자 또는 그룹을 지정하는 액세스 규칙이 있습니다.

사용자가 아직 존재하지 않는 물리적 대상에서 생성자 또는 사용자를 만든 경우 브로커의 자동 생성 등록 정보가 활성화되어 있으면 브로커가 해당 대상을 만듭니다.

기본적으로 모든 사용자 또는 그룹은 브로커가 물리적 대상을 자동 생성하도록 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 생성 액세스 규칙의 일반 구문은 다음과 같습니다.

resourceType`.create.`*access*`.`*principalType*`=`*principals*

여기서 *resourceType*은 `queue` 또는 `topic`입니다.

예를 들어, 다음 규칙은 브로커가 Snoopy를 제외한 모든 사람에 대해 주제 대상을 자동 생성하도록 허용합니다.

```
topic.create.allow.user=*
topic.create.deny.user=Snoopy
```

물리적 대상 자동 생성 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 생성은 **허용**했다면, 브로커는 물리적 대상이 없는 경우 물리적 대상을 만들기는 하지만 메시지를 그 물리적 대상으로 전달하지는 **않습니다**.

메시지 암호화

이 절에서는 클라이언트와 브로커 간에 암호화된 메시지를 보내는 SSL(Secure Socket Layer) 표준을 기반으로 연결 서비스를 설정하는 방법에 대해 설명합니다. Message Queue는 다음과 같은 SSL 기반 연결 서비스를 지원합니다.

- `ssljms` 서비스는 TCP/IP 전송 프로토콜을 사용하여 클라이언트와 브로커 간에 암호화된 보안 메시지를 전달합니다.
- `ssladmin` 서비스는 TCP/IP 전송 프로토콜을 사용하여 Message Queue 명령 유틸리티(`imqcmd`)와 브로커 간에 암호화된 보안 연결을 생성합니다. 관리 콘솔(`imqadmin`)에서는 암호화된 연결이 지원되지 않습니다.
- `cluster` 서비스는 TCP/IP 전송 프로토콜을 사용하여 클러스터의 브로커 간에 암호화된 보안 통신을 제공합니다.
- `httpsjms` 서비스는 HTTP 전송 프로토콜과 HTTPS 터널 서블릿을 사용하여 클라이언트와 브로커 간에 암호화된 보안 메시지를 전달합니다.

이 절의 나머지 부분에서는 `ssljms`, `ssladmin` 및 `cluster` 연결 서비스를 사용하여 TCP/IP를 통해 보안 연결을 설정하는 방법에 대해 설명합니다. `httpsjms` 서비스를 사용하여 HTTP를 통해 보안 연결을 설정하는 방법은 부록 C을 참조하십시오.

자체 서명된 인증서 사용

TCP/IP를 통해 SSL 기반 연결 서비스를 사용하려면 키 도구 유틸리티(`imqkeytool`)를 사용하여 공개/개인 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. 이 절에서는 그런 자체 서명된 인증서를 사용하여 SSL 기반 서비스를 설정하는 방법에 대해 설명합니다.

인증 수준을 강화하려면 인증 기관이 확인한 서명된 인증서를 사용할 수 있습니다. 서명된 인증서를 사용하려면 자체 서명된 인증서에 필요한 단계 이외에 몇 가지 추가 단계를 수행해야 합니다. 이 절에서 설명하는 단계를 수행한 다음 147 페이지 “서명된 인증서 사용”에 설명된 추가 단계를 수행해야 합니다.

자체 서명된 인증서를 사용한 Message Queue의 SSL 지원은 클라이언트가 알려지고 신뢰할 수 있는 서버와 통신한다는 가정 하에 전송 데이터를 보호하기 위한 것입니다. 다음 절차에서는 자체 서명된 인증서를 사용하도록 SSL 기반 연결 서비스를 설정하는 데 필요한 단계를 보여 줍니다. 다음에 오는 하위 절에서는 각 단계에 대해 자세히 설명합니다.

▼ 자체 서명된 인증서를 사용하여 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.**
   이 단계는 `ssljms` 연결 서비스에만 적용되고 `ssladmin` 또는 `cluster`에는 적용되지 않습니다.

자체 서명된 인증서 생성

키 도구 유틸리티(`imqkeytool`)를 실행하여 브로커에 대한 자체 서명된 인증서를 생성합니다. (UNIX® 시스템에서 키 저장소를 만들 권한을 가지려면 유틸리티를 수퍼유저(`root`)로 실행해야 할 수도 있음). `ssljms`, `ssladmin` 또는 `cluster` 연결 서비스에 동일한 인증서를 사용할 수 있습니다.

명령 프롬프트에서 다음을 입력합니다.

```
imqkeytool -broker
```

키 도구 유틸리티가 키 저장소 비밀번호를 묻습니다.

```
Generating keystore for the broker ...
Enter keystore password:
```

그런 다음 이 인증서가 속하는 브로커를 식별하는 정보를 묻습니다. 입력한 정보를 기반으로 X.500 고유 이름이 생성됩니다. 표 7–5에서는 프롬프트와 각 프롬프트에 대해 제공할 값을 보여 줍니다. 값은 대소문자가 구분되고 공백을 포함할 수 있습니다.

표 7–5 자체 서명된 인증서에 필요한 고유 이름 정보

프롬프트	X.500 속성	설명	예
What is your first and last name?	commonName (CN)	브로커를 실행하는 서버의 정규화된 이름	mqserver.sun.com
What is the name of your organizational unit?	organizationalUnit (OU)	부서 이름	purchasing
What is the name of your organization?	organizationName (ON)	회사, 정부 기관 등과 같이 더 큰 조직의 이름	My Company, Inc.
What is the name of your city or locality?	localityName (L)	구/군/시 이름	San Francisco
What is the name of your state or province?	stateName (ST)	약어를 사용하지 않은 시/도의 전체 이름	California
What is the two-letter country code for this unit?	country (C)	표준 2자 국가 코드	US

정보 입력이 끝나면 키 도구 유틸리티에서 확인을 위해 해당 정보를 표시합니다. 예를 들면 다음과 같습니다.

```
Is CN=mqserver.sun.com, OU=purchasing, ON=My Company, Inc.,
L=San Francisco, ST=California, C=US correct?
```

현재 값을 적용하고 계속하려면 yes를 입력하고, 값을 다시 입력하려면 기본값을 적용하거나 no를 입력합니다. 확인이 끝난 후 키 쌍이 생성되는 동안 유틸리티가 일시 중지됩니다.

그런 다음 키 쌍을 잠글 비밀번호(키 비밀번호)를 묻습니다. 키 비밀번호 및 키 저장소 비밀번호와 동일한 비밀번호를 사용하려면 이 프롬프트에서 Return 키를 누릅니다.

주 – 지정한 비밀번호를 기억해 두십시오. 브로커를 시작할 때 이 비밀번호를 입력해야 브로커가 키 저장소를 열 수 있습니다. 비밀번호 파일에 키 저장소 비밀번호를 저장할 수 있습니다(150 페이지 “비밀번호 파일” 참조).

키 도구 유틸리티는 자체 서명된 인증서를 생성한 다음 Message Queue의 키 저장소에 넣습니다. 부록 A에 표시된 것처럼 키 저장소는 운영 체제에 따라 다른 디렉토리에 있습니다.

SSL 기반 연결 서비스의 Message Queue 키 저장소에 대해 구성 가능한 등록 정보는 다음과 같습니다.

- `imq.keystore.file.dirpath`: 키 저장소 파일을 포함하는 디렉토리 경로입니다. 기본값은 부록 A를 참조하십시오.

- `imq.keystore.file.name`: 키 저장소 파일의 이름입니다.
- `imq.keystore.password`: 키 저장소 비밀번호입니다.

특정 문제를 해결하기 위해 키 쌍을 재생성해야 하는 경우도 있습니다. 예를 들어, 키 저장소 비밀번호를 잊은 경우나 브로커를 시작할 때 SSL 기반 서비스가 초기화되지 않고 예외가 발생되는 경우가 있습니다.

```
java.security.UnrecoverableKeyException:Cannot recover key
```

이 예외는 자체 서명된 인증서를 생성할 때의 키 저장소 비밀번호가 지정한 키 비밀번호와 다른 경우 발생할 수 있습니다.

▼ 키 쌍을 다시 생성하는 방법

1 부록 A에 나와 있는 위치에 있는 브로커의 키 저장소를 제거합니다.

2 위에서 설명한 것처럼 `imqkeytool`을 다시 실행하여 새 키 쌍을 생성합니다.

SSL 기반 연결 서비스 활성화

브로커에서 SSL 기반 연결 서비스를 활성화하려면 `ssljms` 또는 `ssladmin`을 `imq.service.activelist` 등록 정보에 추가해야 합니다.

▼ 브로커에서 SSL 기반 서비스를 활성화하는 방법

1 브로커의 인스턴스 구성 파일을 엽니다.

인스턴스 구성 파일은 구성 파일이 연결되어 있는 브로커 인스턴스의 이름(*instanceName*)으로 식별되는 디렉토리에 있습니다(부록 A 참조).

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

2 `imq.service.activelist` 등록 정보에 항목을 추가하고(항목이 아직 없는 경우) 목록에 원하는 SSL 기반 서비스를 포함시킵니다.

기본적으로 이 등록 정보에는 `jms` 및 `admin` 연결 서비스가 포함됩니다. 활성화할 SSL 기반 서비스(`ssljms`, `ssladmin` 또는 둘 다)를 추가합니다.

```
imq.service.activelist=jms,admin,ssljms,ssladmin
```

주 – SSL 기반 `cluster` 연결 서비스는 `imq.cluster.transport` 등록 정보를 `imq.service.activelist` 등록 정보 대신 사용하여 활성화합니다. 자세한 내용은 173 페이지 “브로커 연결”을 참조하십시오.

---

3 인스턴스 구성 파일을 저장하고 닫습니다.

브로커 시작

브로커를 시작하고 키 저장소 비밀번호를 제공합니다. 다음과 같은 두 가지 방법으로 비밀번호를 제공할 수 있습니다.

- 브로커를 시작할 때 비밀번호를 묻는 프롬프트가 표시되게 합니다.

  ```
  imqbrokerd
  Please enter Keystore password:
  ```

- 150 페이지 "비밀번호 파일"에서 설명한 대로 비밀번호 파일에 비밀번호를 입력합니다. 그런 다음 `imq.passfile.enabled` 등록 정보를 `true`로 설정하고 다음 중 하나를 수행합니다.
  - 비밀번호 파일의 위치를 `imqbrokerd` 명령에 전달합니다.

    ```
    imqbrokerd -passfile /passfileDirectory/passfileName
    ```

  - `-passfile` 옵션 없이 브로커를 시작하되, 다음의 두 브로커 구성 등록 정보를 사용하여 비밀번호 파일의 위치를 지정합니다.

    ```
    imq.passfile.dirpath=/passfileDirectory

    imq.passfile.name=passfileName
    ```

주 – SSL을 사용하는 브로커나 클라이언트를 시작할 때 몇 초간 CPU 사용량이 갑자기 증가하는 경우가 있을 수 있습니다. 이것은 Message Queue에서 난수를 생성하는 데 사용하는 JSSE(Java Secure Socket Extension) 메서드 `java.security.SecureRandom`이 초기 난수 시드를 생성하는 데 많은 시간이 소요되기 때문입니다. 시드를 만들고 나면 CPU 사용 수준이 정상으로 돌아갑니다.

SSL 기반 클라이언트 구성 및 실행

SSL 기반 연결 서비스를 사용하도록 클라이언트를 구성하는 절차는 `ssljms` 연결 서비스를 사용하는 응용 프로그램 클라이언트인지 `ssladmin` 연결 서비스를 사용하는 Message Queue 관리 클라이언트(예: `imqcmd`)인지 여부에 따라 다릅니다.

응용 프로그램 클라이언트

응용 프로그램 클라이언트의 경우 다음과 같은 `.jar` 파일이 `CLASSPATH` 변수에 지정되어 있어야 합니다.

```
imq.jar
jms.jar
```

Java 2 Software Development Kit(J2SDK) 1.4 이전 버전을 사용할 경우 다음과 같은 JSSE(Java Secure Socket Extension) 및 JNDI(Java Naming and Directory Interface) `.jar` 파일을 포함시켜야 합니다.

```
jsse.jar
jnet.jar
jcert.jar
jndi.jar
```

J2SDK 1.4 이상에서는 JSSE 및 JNDI를 기본적으로 지원하므로 이러한 파일을 포함시킬 필요가 없습니다.

CLASSPATH 파일을 제대로 지정한 경우 다음과 같은 명령을 입력하여 클라이언트를 시작하고 브로커의 ssljms 연결 서비스에 연결할 수 있습니다.

```
java -DimqConnectionType=TLS clientAppName
```

그러면 연결에서 SSL 기반 연결 서비스를 사용하게 됩니다.

관리 클라이언트

관리 클라이언트의 경우 imqcmd 명령을 호출할 때 -secure 옵션을 포함시켜 보안 연결을 설정할 수 있습니다. 예를 들면 다음과 같습니다.

```
imqcmd list svc -b hostName:portNumber -u adminName -secure
```

여기서 *adminName*은 Message Queue 사용자 저장소에 유효한 항목입니다. 이 명령은 비밀번호를 묻는 프롬프트를 표시합니다(플랫 파일 저장소를 사용하는 경우 132 페이지 "기본 관리자 비밀번호 변경" 참조).

연결 서비스를 나열하는 것은 ssladmin 서비스가 실행 중이며 다음 출력과 같이 보안 관리 연결을 성공적으로 설정할 수 있다는 것을 확인하는 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

서명된 인증서는 자체 서명된 인증서보다 더 높은 수준의 서버 인증을 제공합니다. 서명된 인증서는 클라이언트와 브로커 간에만 구현 가능하고 동일 클러스터 내의 브로커 간에는 구현할 수 없습니다. 그럴 경우 위에서 설명한 자체 서명된 인증서 구성 단계 이외에 다음과 같은 추가 단계를 수행해야 합니다. 이러한 단계에 대해서는 다음에 나오는 하위 절에 자세히 설명되어 있습니다.

▼ 서명된 인증서를 사용하는 방법

1 키 저장소에 인증서를 설치합니다.

2 브로커에 SSL 기반 연결을 설정할 때 서명된 인증서를 요구하도록 Message Queue 클라이언트를 구성합니다.

서명된 인증서 가져오기 및 설치

다음 절차에서는 서명된 인증서를 가져와서 설치하는 방법에 대해 설명합니다.

▼ 서명된 인증서를 가져오는 방법

1 J2SE `keytool` 명령을 사용하여 이전 절에서 생성한 자체 서명된 인증서에 대한 CSR(Certificate Signing Request)을 생성합니다.

`keytool` 명령에 대한 자세한 내용은 을 참조하십시오.

`http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html`

예를 들면 다음과 같습니다.

```
keytool -certreq -keyalg RSA -alias imq -file certreq.csr
        -keystore /etc/imq/keystore -storepass myStorePassword
```

이 명령은 지정된 파일(이 예의 경우 `certreq.csr`)에서 인증서를 캡슐화하는 CSR을 생성합니다.

2 CSR을 사용하여 서명된 인증서를 생성하거나 요청합니다.

이 작업은 다음 중 한 가지 방법으로 수행할 수 있습니다.

- Thawte 또는 Verisign과 같이 공신력 있는 인증 기관(CA)에서 서명한 인증서를 가져옵니다. 이 작업을 수행하는 방법에 대한 자세한 내용은 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 파일에 저장합니다.**

아래 지침에서는 `broker.cer`이라는 이름 예를 사용하여 브로커 인증서를 나타냅니다.

▼ 서명된 인증서 설치 방법

1 **J2SE가 기본적으로 인증 기관을 지원하는지 여부를 확인합니다.**

다음 명령은 시스템 키 저장소에 있는 루트 CA를 나열합니다.

```
keytool -v -list -keystore $JAVA_HOME/lib/security/cacerts
```

해당 CA가 목록에 있는 경우 다음 단계를 건너뜁니다.

2 **인증 기관이 J2SE에서 지원되지 않는 경우 CA의 루트 인증서를 Message Queue 키 저장소로 가져옵니다.**

예를 들면 다음과 같습니다.

```
keytool -import -alias ca -file ca.cer -noprompt -trustcacerts
        -keystore /etc/imq/keystore -storepass myStorePassword
```

여기서 `ca.cer`은 CA에서 가져온 루트 인증서가 들어 있는 파일입니다.

CA 테스트 인증서를 사용하는 경우 테스트 CA 루트 인증서를 가져와야 할 수 있습니다. CA에는 복사본을 가져오는 방법에 대한 지침이 있어야 합니다.

3 **서명된 인증서를 키 저장소로 가져와서 자체 서명된 원본 인증서를 대체합니다.**

예를 들면 다음과 같습니다.

```
keytool -import -alias imq -file broker.cer -noprompt -trustcacerts
        -keystore /etc/imq/keystore -storepass myStorePassword
```

여기서 `broker.cer`은 CA로부터 받은 서명된 인증서가 들어 있는 파일입니다.

Message Queue 키 저장소에는 이제 SSL 연결에 사용할 서명된 인증서가 포함되어 있습니다.

서명된 인증서를 요구하도록 Message Queue 클라이언트 런타임 구성

서명된 인증서를 요구하도록 Message Queue 클라이언트 런타임을 구성하고 인증서에 서명한 인증 기관을 신뢰하는지를 확인해야 합니다.

▼ 서명된 인증서를 요구하도록 클라이언트 런타임을 구성하는 방법

1 **연결 팩토리의 `imqSSLIsHostTrusted` 속성을 `false`로 설정합니다.**

기본적으로 클라이언트가 브로커 연결을 설정하는 데 사용할 연결 팩토리 객체의 `imqSSLIsHostTrusted` 속성은 `true`로 설정되어 있습니다. 즉, 클라이언트 런타임에서

제공된 인증서를 수용합니다. 클라이언트 런타임이 제공되는 모든 인증서를 검증하도록 이 값을 false로 변경해야 합니다. 인증서 서명자가 클라이언트의 트러스트 저장소에 없는 경우 검증이 실패합니다.

2 **서명 기관이 클라이언트의 트러스트 저장소에 등록되어 있는지 여부를 확인합니다.**

클라이언트가 인증 기관에서 서명한 인증서를 수용하는지 여부를 테스트하려면 145 페이지 "SSL 기반 클라이언트 구성 및 실행"에서 설명한 것처럼 SSL 연결 설정을 시도합니다. CA가 클라이언트의 트러스트 저장소에 있으면 성공적으로 연결되고 다음 단계를 건너뛸 수 있습니다. 연결이 실패하고 인증서 검증 오류가 발생하면 다음 단계로 이동합니다.

3 **서명 CA의 루트 인증서를 클라이언트의 트러스트 저장소에 설치합니다.**

클라이언트는 기본적으로 키 저장소 파일 cacerts 및 jssecacerts를 검색하므로 인증서를 이러한 파일에 설치할 때 추가 구성이 필요하지 않습니다. 다음 예에서는 Verisign 인증 기관의 테스트 루트 인증서를 testrootca.cer 파일에서 기본 시스템 인증서 파일 cacerts에 설치합니다.이 예제에서는 J2SE가 $JAVA_HOME/usr/j2se 디렉토리에 설치된다고 가정합니다.

```
keytool -import -keystore /usr/j2se/jre/lib/security/cacerts
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

대체(권장) 옵션으로 루트 인증서를 대체 시스템 인증서 파일 jssecacerts에 설치할 수 있습니다.

```
keytool -import -keystore /usr/j2se/jre/lib/security/jssecacerts
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

세 번째 방법으로 루트 인증서를 다른 키 저장소 파일에 설치하고 해당 저장소 파일을 트러스트 저장소로 사용하도록 클라이언트를 구성할 수 있습니다. 다음 예에서는 /home/smith/.keystore 파일에 설치합니다.

```
keytool -import -keystore /home/smith/.keystore
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

클라이언트는 기본적으로 이 키 저장소를 검색하지 않기 때문에 클라이언트에서 트러스트 저장소로 사용하도록 해당 위치를 명시적으로 지정해야 합니다. 이 작업은 클라이언트가 다음을 실행 중일 때 Java 시스템 등록 정보 javax.net.ssl.trustStore를 설정하여 수행합니다.

```
javax.net.ssl.trustStore=/home/smith/.keystore
```

비밀번호 파일

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
`imqbrokerd`	브로커를 시작합니다.	JDBC 기반 영구 데이터 저장소, SSL 인증서 키 저장소 또는 LDAP 사용자 저장소에 액세스합니다.
`imqcmd`	브로커 관리	명령 사용 권한이 있는 관리자를 인증합니다.
`imqdbmgr`	JDBC 기반 데이터 저장소 관리	데이터 저장소에 액세스합니다.

비밀번호 파일에 이러한 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. 현재 릴리스에서는 이러한 옵션 중 하나에 대한 명령줄의 값이 비밀번호 파일의 관련 값보다 우선합니다.

보안 문제

모니터를 다른 사람이 보고 있지 않다면 프롬프트에 응답하여 비밀번호를 대화식으로 지정하는 것이 가장 안전한 비밀번호 지정 방법입니다. 명령줄에서 비밀번호 파일을 지정할 수도 있습니다. 명령을 비대화식으로 사용할 경우 비밀번호 파일을 사용해야 합니다.

비밀번호 파일은 암호화되지 않기 때문에 사용 권한을 설정하여 인증되지 않은 액세스로부터 보호해야 합니다. 브로커를 시작하는 사용자가 파일을 볼 수 있지만 읽기 액세스만 허용되도록 사용 권한을 설정합니다.

비밀번호 파일 내용

비밀번호 파일은 일련의 등록 정보와 값이 들어 있는 간단한 텍스트 파일입니다. 각 값은 명령에 사용되는 비밀번호입니다.

비밀번호 파일에는 표 7–7에 표시된 비밀번호를 포함할 수 있습니다.

표 7–7 비밀번호 파일의 비밀번호

비밀번호	영향 받는 명령	설명
imq.imqcmd.password	+-imqcmd	imqcmd 명령줄에 대한 관리자 비밀번호를 지정합니다. 비밀번호는 각 명령에 대해 인증됩니다.
imq.keystore.password	imqbrokerd	SSL 기반 서비스의 키 저장소 비밀번호를 지정합니다.
imq.persist.jdbc.password	imqbrokerd imdbmgr	필요한 경우 데이터베이스 연결을 여는 비밀번호를 지정합니다.
imq.user_repository.ldap.password	imqbrokerd	구성된 LDAP 사용자 저장소에 바인드할 수 있도록 브로커에 할당한 고유 이름에 연결되는 비밀번호를 지정합니다.

샘플 비밀번호 파일은 Message Queue 제품의 일부입니다. 샘플 파일의 위치는 부록 A를 참조하십시오.

방화벽을 통해 연결

방화벽을 사용하여 클라이언트 응용 프로그램을 브로커와 분리하면 연결 설정을 위해 특별한 조치가 필요합니다. 한 가지 방법으로 httpjms 또는 httpsjms 연결 서비스를 사용하여 방화벽을 통해 "터널"을 생성할 수 있습니다. 자세한 내용은 부록 C을 참조하십시오. HTTP 연결은 다른 연결 서비스보다 더 느립니다. 그러나 그 보다 빠른 방법은 Message Queue 포트 매퍼를 우회하고 원하는 연결 서비스에 정적 포트 주소를 명시적으로 할당한 다음 방화벽에서 해당 포트를 여는 것입니다. 이 방법을 사용하면 jms 또는 ssljms 연결 서비스(또는 admin 또는 ssladmin)를 사용하여 방화벽을 통해 연결할 수 있습니다.

표 7–8 정적 포트 주소에 대한 브로커 구성 등록 정보

연결 서비스	구성 등록 정보
jms	imq.jms.tcp.port
ssljms	imq.ssljms.tls.port
admin	imq.admin.tcp.port
ssladmin	imq.ssladmin.tls.port

▼ 방화벽을 통해 브로커 연결을 활성화하는 방법

1 사용할 연결 서비스에 정적 포트 주소를 할당합니다.

포트 매퍼를 우회하고 연결 서비스에 정적 포트 번호를 직접 할당하려면 브로커 구성 등록 정보 `imq.`*serviceName*`.`*protocolType*`.port`를 설정합니다. 여기서 *serviceName*은 연결 서비스의 이름이고 *protocolType*은 해당 프로토콜 유형입니다(표 7–8 참조). 브로커 구성 등록 정보와 마찬가지로 브로커를 시작할 때 명령줄에서 또는 브로커의 인스턴스 구성 파일에서 이 등록 정보를 지정할 수 있습니다. 예를 들어, 포트 번호 `10234`를 `jms` 연결 서비스에 할당하려면

```
imq.jms.tcp.port=10234
```

행을 구성 파일에 포함시키거나 다음 명령을 사용하여 브로커를 시작합니다.

```
imqbrokerd  -name myBroker  -Dimq.jms.tcp.port=10234
```

2 연결 서비스에 할당된 포트 번호에 연결할 수 있도록 방화벽을 구성합니다.

또한 방화벽을 통해 Message Queue의 포트 매퍼 포트(포트 매퍼를 다른 포트에 다시 할당하지 않은 경우 일반적으로 `7676`)에 연결할 수 있어야 합니다. 위 예에서 포트 `10234` 및 `7676`에 대해 방화벽을 열어야 합니다.

감사 기록

Message Queue는 엔터프라이즈판에서만 감사 기록을 지원합니다. 감사 로깅이 활성화되면 Message Queue는 다음과 같은 이벤트 유형에 대한 레코드를 생성합니다.

- 브로커 인스턴스 시작, 종료, 다시 시작 및 제거
- 사용자 인증 및 권한 부여
- 영구 저장소 재설정
- 물리적 대상 생성, 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.

◆ ◆ ◆ 8 장

8 관리 대상 객체 관리

관리 대상 객체는 공급자별 구성 및 이름 지정 정보를 캡슐화하여 하나의 JMS 공급자에서 다른 공급자로 이식할 수 있는 클라이언트 응용 프로그램을 개발할 수 있도록 합니다. Message Queue™ 관리자는 일반적으로 클라이언트 응용 프로그램이 메시지 송/수신을 위해 브로커에 연결하는 데 사용할 관리 대상 객체를 만듭니다.

이 장에서는 객체 관리자 유틸리티(`imqobjmgr`)를 사용하여 생성 및 관리 대상 객체를 관리하는 방법에 대해 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 153 페이지 "객체 저장소"
- 156 페이지 "관리 객체 속성"
- 163 페이지 "객체 관리자 유틸리티 사용"
- 164 페이지 "관리 객체 추가"
- 166 페이지 "관리 객체 나열"
- 166 페이지 "관리 객체 정보 보기"
- 167 페이지 "관리 객체 속성 수정"

객체 저장소

관리 대상 객체는 클라이언트 응용 프로그램에서 JNDI(Java Naming and Directory Interface)를 통해 액세스할 수 있도록 미리 만들어진 객체 저장소에 있습니다. 사용할 수 있는 객체 저장소에는 표준 LDAP(Lightweight Directory Access Protocol) 디렉토리 서버와 로컬 파일 시스템의 디렉토리의 두 가지 유형이 있습니다.

LDAP 서버 객체 저장소

LDAP 서버는 작업 메시징 시스템에 권장되는 객체 저장소입니다. LDAP 서버는 분산 시스템에서 사용할 수 있도록 디자인되었으며 작업 환경에서 유용한 보안 기능을 제공합니다.

LDAP 구현은 여러 공급업체에서 제공합니다. Message Queue 관리 도구를 사용하여 LDAP 서버의 객체 저장소를 관리하려면 먼저 Java 객체를 저장할 서버를 구성한 다음 JNDI 조회를 수행해야 합니다. 자세한 내용은 LDAP 구현과 함께 제공된 설명서를 참조하십시오.

LDAP 서버를 객체 저장소로 사용하려면 표 8–1에 표시된 속성을 지정해야 합니다. 이 속성들은 다음 범주로 구분됩니다.

- **초기 컨텍스트**. `java.naming.factory.initial` 속성은 서버에서 JNDI 조회를 위한 초기 컨텍스트를 지정합니다. 이 속성 값은 지정된 LDAP 객체 저장소에 대해 고정되어 있습니다.
- **위치**.`java.naming.provider.url` 속성은 LDAP 서버의 URL과 디렉토리 경로를 지정합니다. 지정한 디렉토리 경로가 있는지 확인해야 합니다.
- **보안**.`java.naming.security.principal`, `java.naming.security.credentials` 및 `java.naming.security.authentication` 속성은 객체 저장소에 액세스를 시도하는 호출자의 인증을 관리합니다. 이러한 속성의 정확한 형식 및 값은 LDAP 서비스 공급자에 따라 다릅니다. 보안 정보가 모든 작업에 필요한지 또는 저장된 데이터를 변경하는 작업에만 필요한지 여부를 결정하거나 자세한 내용을 보려면 해당 LDAP 구현에 제공된 설명서를 참조하십시오.

표 8–1 LDAP 객체 저장소 속성

속성	설명
`java.naming.factory.initial`	JNDI 조회를 위한 초기 컨텍스트 예: `com.sun.jndi.ldap.LdapCtxFactory`
`java.naming.provider.url`	서버 URL 및 디렉토리 경로 예: `ldap://myD.com:389/ou=mq1,o=App` 여기서 관리 대상 객체는 `/App/mq1` 디렉토리에 저장됩니다.
`java.naming.security.principal`	호출자를 인증할 때 사용되는 principal의 아이디 이 속성의 형식은 인증 방법에 따라 달라집니다. 예를 들면 다음과 같습니다. `uid=homerSimpson,ou=People,o=mq` 이 속성을 지정하지 않으면 LDAP 서비스 공급자가 동작을 결정합니다.

표 8–1 LDAP 객체 저장소 속성 *(계속)*

속성	설명
java.naming.security.credentials	인증 principal 자격 증명 이 속성의 값은 인증 방법에 따라 달라집니다. 예를 들어 해시 비밀번호, 단순 텍스트 비밀번호, 키 또는 자격 증명 등이 될 수 있습니다. 이 등록 정보를 지정하지 않으면 LDAP 서비스 공급자가 동작을 결정합니다.
java.naming.security.authentication	인증 보안 수준 이 속성의 값은 none, simple 또는 strong 키워드 중 하나입니다. 예를 들어, simple을 지정하면 누락된 principal 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 관리 대상 객체에 대한 객체 저장소로 로컬 파일 시스템의 디렉토리 사용을 지원합니다. 이 방법은 작업 시스템에는 바람직하지 않지만, 개발 환경에서는 사용하기 쉬운 장점이 있습니다. 그러나, 여러 컴퓨터 노드에 배포된 클라이언트의 중앙 집중식 객체 저장소로 사용할 디렉토리의 경우에는 해당 모든 클라이언트가 디렉토리에 액세스할 수 있어야 합니다. 또한 해당 디렉토리에 액세스할 수 있는 모든 사용자는 Message Queue 관리 도구를 사용하여 관리 대상 객체를 만들고 관리할 수 있습니다.

파일 시스템 디렉토리를 객체 저장소로 사용하려면 표 8–2에 표시된 속성을 지정해야 합니다. LDAP 객체 저장소의 경우 이러한 속성은 위에서 설명한 것과 동일한 일반적인 의미를 지닙니다. 특히 java.naming.provider.url 속성은 객체 저장소를 포함하는 디렉토리의 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	JNDI 조회를 위한 초기 컨텍스트 예: com.sun.jndi.fscontext.RefFSContextFactory

표 8–2 파일 시스템 객체 저장소 속성 (계속)

속성	설명
java.naming.provider.url	디렉토리 경로 예: file:///C:/myapp/mqobjs

관리 객체 속성

Message Queue 관리 대상 객체에는기본적으로 다음과 같은 두 가지 종류가 있습니다.

- **연결 팩토리**는 클라이언트 응용 프로그램에서 브로커에 대한 연결을 만들 때 사용됩니다.
- **대상**은 클라이언트 응용 프로그램이 메시지를 교환(전송/검색)할 수 있는 브로커 상의 위치를 나타냅니다.

주 – SOAP 메시징에서는 특수한 *SOAP* **종점 관리 대상 객체**를 사용합니다. 자세한 내용은 **Java 클라이언트용 Message Queue 개발 안내서**를 참조하십시오.

각 관리 대상 객체 유형에는 객체의 등록 정보와 동작을 결정하는 특정 속성이 있습니다. 이 절에서는 객체 관리자 명령줄 유틸리티(imqobjmgr)를 사용하여 이러한 속성을 설정하는 방법에 대해 설명합니다. 또한 56 페이지 “관리 객체 작업”에 설명된 대로 GUI 관리 콘솔을 사용하여 설정할 수도 있습니다.

연결 팩토리 속성

클라이언트 응용 프로그램에서는 **연결 팩토리** 관리 대상 객체를 사용하여 메시지를 브로커와 교환하는 데 사용할 연결을 만듭니다. 연결 팩토리의 속성은 생성되는 모든 연결의 등록 정보를 정의합니다. 연결이 만들어지고 나면 등록 정보를 변경할 수 없습니다. 따라서 연결 등록 정보를 구성하는 유일한 방법은 연결을 만드는 데 사용되는 연결 팩토리의 속성을 설정하는 것입니다.

Message Queue는 두 종류의 연결 팩토리 객체를 정의합니다.

- ConnectionFactory 객체는 일반 메시징과 비분산 트랜잭션을 지원합니다.
- XAConnectionFactory 객체는 분산 트랜잭션을 지원합니다.

두 종류 모두 자원, 성능 및 메시지 처리량을 최적화하는 데 사용할 수 있는 동일한 구성 속성을 공유합니다. 이러한 속성은 16 장에 나열되어 있으며 다음 절에서 설명합니다.

- 157 페이지 “연결 처리”
- 159 페이지 “클라이언트 아이디”

- 160 페이지 “안정성 및 흐름 제어”
- 161 페이지 “대기열 브라우저 및 서버 세션”
- 161 페이지 “표준 메시지 등록 정보”
- 162 페이지 “메시지 헤더 대체”

연결 처리

연결 처리 속성은 연결할 브로커 주소를 지정하고, 필요한 경우 연결 실패를 찾아내어 재연결을 시도하는 방법을 지정합니다. 이러한 속성은 표 16–1에 요약되어 있습니다.

브로커 주소 목록

가장 중요한 연결 처리 속성은 `imqAddressList`로서, 이 속성은 브로커 또는 연결을 설정할 브로커를 지정합니다. 이 속성의 값은 하나의 브로커 주소 또는 (브로커 클러스터의 경우) 쉼표로 구분된 여러 주소를 포함하는 문자열입니다. 브로커 주소는 사용할 연결 서비스(76 페이지 “연결 서비스” 참조)와 연결 설정 방법에 따라 다양한 주소 지정 체계를 사용할 수 있습니다.

- `mq`는 브로커의 포트 매퍼를 사용하여 `jms` 또는 `ssljms` 연결 서비스를 위해 포트를 동적으로 할당합니다.
- `mqtcp`는 포트 매퍼를 우회하고 `jms` 연결 서비스를 사용하여 지정된 포트로 직접 연결합니다.
- `mqssl`은 `ssljms` 연결 서비스를 사용하여 지정된 포트에 SSL(Secure Socket Layer) 연결을 설정합니다.
- `http`는 `httpjms` 연결 서비스를 사용하여 지정된 URL에서 Message Queue 터널 서블릿으로 HTTP(Hypertext Transport Protocol) 연결을 설정합니다.
- `https`는 `httpsjms` 연결 서비스를 사용하여 지정된 URL에서 Message Queue 터널 서블릿으로 HTTPS(Secure Hypertext Transport Protocol) 연결을 설정합니다.

이러한 주소 지정 체계는 표 16–2에 요약되어 있습니다.

각 브로커 주소의 일반 형식은 다음과 같습니다.

scheme`://`*address*

여기서 *scheme*은 위에 나열된 주소 지정 체계 중 하나이며 *address*는 브로커 주소 자체를 나타냅니다. 표 16–2의 마지막 열에 표시된 대로, 주소를 지정하는 정확한 구문은 주소 지정 체계에 따라 다릅니다. 표 16–3에서는 다양한 주소 형식의 예를 보여줍니다.

다중 브로커 클러스터 환경의 주소 목록에는 두 개 이상의 브로커 주소가 포함될 수 있습니다. 첫 번째 연결 시도가 실패하면 Message Queue 클라이언트 런타임에서 목록에 있는 다른 주소로 연결을 시도하며, 연결되지 않은 경우 목록의 마지막 항목에 이를 때까지 계속 시도합니다. 두 개의 추가 연결 팩토리 속성이 이러한 연결 시도 방법을 제어합니다.

- imqAddressListBehavior는 지정한 주소로 연결을 시도하는 순서를 지정합니다. 이 속성이 PRIORITY 문자열로 설정된 경우 주소에 대한 연결 시도는 주소 목록에 나타나는 순서대로 실행됩니다. 속성 값이 RANDOM이면 임의의 순서대로 주소를 선택하여 연결을 시도합니다. 이 방법은 많은 Message Queue 클라이언트가 동일한 연결 팩토리 객체를 공유하는 경우 모든 클라이언트가 동일한 브로커 주소로의 연결을 시도하지 않도록 하는 데 유용합니다.
- imqAddressListIterations는 연결을 포기하고 실패를 보고하기 전에 목록 전체를 몇 번씩 반복할지 지정합니다. 값 -1은 무제한 반복을 의미합니다. 연결이 설정되거나 종료 시간까지(둘 중 먼저 발생하는 항목) 클라이언트 런타임에서 연결을 계속 시도합니다.

자동 재연결

연결 팩토리의 imqReconnectEnabled 속성을 true로 설정하면 연결에 실패한 경우 클라이언트가 브로커에 자동으로 다시 연결할 수 있습니다. imqReconnectAttempts 속성은 지정된 브로커 주소에 대한 재연결 시도 횟수를 제어하고, imqReconnectInterval 속성은 시도 간격(초)을 지정합니다.

브로커 주소 목록(imqAddressList)에 여러 주소가 지정되어 있는 브로커 클러스터에서 실패한 연결은 원래의 브로커에서뿐만 아니라 클러스터의 다른 브로커에서도 복원될 수 있습니다. 원래 브로커로의 재연결이 실패하면 클라이언트 런타임은 목록의 다른 주소로 시도합니다. imqAddressListBehavior 및 imqAddressListIterations 속성은 이전 절에 설명한 대로 연결 시도되는 주소의 순서와 목록 반복 횟수를 제어합니다. 각 주소에 대해 imqReconnectInterval 밀리초 간격으로 반복을 시도하며, imqReconnectAttempts를 통해 지정된 최대 시도 횟수까지 반복을 시도합니다.

자동 재연결은 메시지 사용에 대한 모든 클라이언트 확인 모드를 지원합니다. 연결이 다시 설정된 후 브로커는 이전에 전달한 확인되지 않은 모든 메시지를 다시 전달하며 해당 메시지에 Redeliver 플래그를 표시합니다. 응용 프로그램 코드에서는 이 플래그를 사용하여 이미 사용되었지만 아직 확인되지 않은 메시지가 있는지 확인합니다. 그러나 비영구 가입자의 경우 연결이 닫히게 되면 브로커에 메시지가 저장되지 않습니다. 따라서 연결이 종료된 상태에서 해당 가입자에 대해 생성된 메시지는 재연결 후에도 전달할 수 없으므로 손실됩니다.자동 재연결 중에는 메시지 생성이 차단됩니다. 메시지 생성자는 연결이 다시 설정될 때까지 브로커로 메시지를 보낼 수 없습니다.

자동 재연결은 연결 페일오버를 제공하지만 데이터 페일오버는 제공하지 않습니다. 실패한 브로커 또는 연결이 끊긴 브로커가 가지고 있는 지속성 메시지 및 기타 상태 정보는 클라이언트가 다른 브로커 인스턴스에 재연결될 때 손실될 수 있습니다. 연결 재설정을 시도하는 동안 Message Queue에서는 클라이언트 런타임이 제공한 객체(세션, 메시지 사용자 및 메시지 생성자)를 유지합니다. 연결 실패 시 잠시 동안 임시 대상도 유지 관리됩니다. 클라이언트가 다시 연결하여 임시 대상에 다시 액세스할 수도 있기 때문입니다. 클라이언트에게 다시 연결하여 이러한 대상을 사용할 수 있는 시간을 제공한 후 브로커는 해당 대상을 삭제합니다. 재연결 시 클라이언트측 상태를 브로커에서 완전히 복원할 수 없는 경우(예: 연결 시간 동안에만 존재하는 트랜잭션된 세션 사용 시) 자동 재연결이 수행되지 않으며 연결의 예외 처리기가 대신 호출됩니다. 그리고 나면 응용 프로그램 코드에 따라 예외, 재연결 및 복원 상태를 파악하게 됩니다.

주기적 연결 테스트(핑)

연결을 주기적으로 테스트하거나 "핑"을 실행하도록 Message Queue 클라이언트 런타임을 구성하면 시도한 메시지 전송이 실패하기 전에 먼저 연결 실패를 감지할 수 있습니다. 이러한 테스트는 메시지를 사용하기만 하고 생성하지는 않는 클라이언트 응용 프로그램에 특히 중요합니다. 그렇지 않으면 연결이 실패한 경우 이러한 응용 프로그램을 찾아낼 수 없기 때문입니다. 메시지를 가끔씩만 생성하는 클라이언트에도 이 기능을 유용하게 사용할 수 있습니다.

연결 팩토리 속성 `imqPingInterval`은 연결을 핑할 빈도(초)를 지정합니다. 기본적으로 이 간격은 30초로 설정되며 값 `-1`은 핑 작업을 수행할 수 없습니다.

실패한 핑에 대한 응답은 운영 체제 플랫폼마다 다릅니다. 일부 운영 체제에서는 클라이언트 응용 프로그램의 예외 수신기에 예외가 즉시 발생합니다. 클라이언트에 예외 수신기가 없는 경우에는 다음 번 연결 사용 시도가 실패합니다. 다른 시스템에서는 핑이 성공하거나 버퍼 오버플로가 발생할 때까지 연속적으로 핑을 버퍼링하여 브로커에 대한 연결 설정을 계속해서 시도하기도 합니다.

클라이언트 아이디

표 16–4에 나열된 연결 팩토리 속성은 클라이언트 인증과 영구 가입자의 클라이언트 식별자 설정을 지원합니다.

클라이언트 인증

메시지 서비스에서 관리하는 사용자 저장소에 대해 사용자 이름 및 비밀번호를 사용하여 브로커에 연결하려는 모든 시도를 인증해야 합니다. 연결 팩토리 속성 `imqDefaultUsername` 및 `imqDefaultPassword`는 연결을 생성할 때 클라이언트가 명시적으로 제공하지 않을 경우 사용할 기본 사용자 이름 및 비밀번호를 지정합니다.

응용 프로그램 개발 및 테스트 중 사용자 저장소 채우기를 번거롭게 여기는 개발자의 편의를 제공하기 위해 Message Queue는 사용자 이름과 비밀번호가 모두 `guest`인 guest 사용자 계정을 제공합니다. 또한 이 값은 `imqDefaultUsername` 및 `imqDefaultPassword` 속성의 기본값이기도 하므로, 명시적으로 지정되지 않은 경우 클라이언트는 항상 guest 계정으로 연결을 설정할 수 있습니다. 작업 환경에서 브로커 연결 액세스는 사용자 저장소에 명시적으로 등록되어 있는 사용자로만 제한되어야 합니다.

클라이언트 식별자

Java Message Service 사양에서는 브로커가 클라이언트를 대신하여 지속성 상태를 유지해야 할 때마다 연결이 고유한 **클라이언트 식별자**를 제공하도록 규정합니다. Message Queue는 이러한 클라이언트 식별자를 사용하여 주제 대상에 대한 영구 가입자를 추적합니다. 영구 가입자가 비활성 상태인 경우 브로커는 주제에 대한 모든 수신 메시지를 보관했다가 가입자가 다시 활성화되면 전달합니다. 브로커는 클라이언트 식별자로 가입자를 식별합니다.

클라이언트 응용 프로그램이 연결 객체의 setClientID 메소드를 사용하여 프로그래밍 방식으로 고유의 클라이언트 식별자를 설정할 수 있지만, 각 식별자가 고유하도록 클라이언트 식별자를 조정하기는 어렵습니다. 일반적으로는 클라이언트를 대신하여 연결을 생성하는 경우 Message Queue에서 고유한 식별자를 자동으로 할당하게 하는 편이 더 좋습니다. 그러기 위해서는 연결 팩토리의 imqConfiguredClientID 속성을 다음 형식의 값으로 설정하면 됩니다.

${u}*factoryID*

${u} 문자는 속성 값의 처음 네 문자여야 합니다. 중괄호 안에 u 이외의 문자를 지정할 경우 연결 생성 시 예외가 발생하게 됩니다. 이러한 문자가 다른 위치에서 사용되는 경우에는 특별한 의미 없이 일반 텍스트로 간주됩니다. *factoryID*의 값은 연결 팩토리 객체와 관련된 고유 문자열입니다.

특정 클라이언트에 대한 연결을 생성하는 경우 Message Queue는 ${u} 문자를 u:*userName*으로 바꾸어 클라이언트 식별자를 구성합니다. 여기서 *userName*은 연결 인증에 사용되는 사용자 이름입니다. 따라서 해당 연결 팩토리가 생성한 연결은 다른 모든 측면에서는 동일하더라도 각 연결마다 고유한 클라이언트 식별자를 갖게 됩니다. 예를 들어, 사용자 이름이 Calvin이고 연결 팩토리의 imqConfiguredClientID 속성에 대해 지정된 문자열이 ${u}Hobbes이면 지정된 클라이언트 식별자는 u:CalvinHobbes가 됩니다.

주 – 두 클라이언트가 모두 기본 사용자 이름 guest를 사용하여 연결을 시도하는 경우에는 각 클라이언트마다 동일한 ${u} 구성 요소를 가진 클라이언트 식별자를 갖게 되므로 이 방법이 적용되지 않습니다. 이 경우, 연결을 요청한 첫 번째 클라이언트만 연결됩니다. Message Queue에서 클라이언트 식별자가 동일한 두 개의 연결을 생성할 수는 없으므로 두 번째 클라이언트의 연결 시도는 실패하게 됩니다.

---

imqConfiguredClientID를 사용하여 클라이언트 식별자를 지정한 경우에도 클라이언트 응용 프로그램은 setClientID 연결 메소드를 사용하여 이 설정을 무시할 수 있습니다. 연결 팩토리의 imqDisableSetClientID 속성을 true로 설정하면 이 작업을 막을 수 있습니다. 영구 가입자를 사용하는 응용 프로그램의 경우에는 imqConfiguredClientID를 사용한 관리 방식이나 setClientID를 사용한 프로그램 방식으로 클라이언트 식별자를 설정**해야** 합니다.

안정성 및 흐름 제어

클라이언트에서 송/수신되는 "페이로드" 메시지와 Message Queue에서 자체적으로 사용하는 제어 메시지(브로커 확인 등)는 동일한 클라이언트 브로커 연결을 생략하기 때문에 과도한 페이로드 트래픽 수준이 제어 메시지 전달을 방해할 수 있습니다. 이 문제를 줄이려면 표 16–5에 나열된 연결 팩토리 속성을 사용하여 두 가지 메시지 유형의 상대적 흐름을 관리할 수 있습니다. 이 속성들은 다음 네 가지 범주로 분류됩니다.

- **확인 시간 초과**는 예외 발생 전 브로커 확인을 기다리는 최대 시간(imqAckTimeout)을 지정합니다.

- **연결 흐름 측정**은 페이로드 메시지 전송을 지정된 크기의 일괄 처리(imqConnectionFlowCount)로 제한하여 누적된 제어 메시지를 주기적으로 전달할 수 있게 합니다.
- **연결 흐름 제어**는 사용될 때까지 기다리고 있는 연결에 대해 보류 중일 수 있는 페이로드 메시지의 수(imqConnectionFlowLimit)를 제한합니다. 제한에 도달하면 사용 대기 중인 메시지 수가 제한보다 적을 때까지 연결에 대한 페이로드 메시지 전달이 일시 중단됩니다. 이 기능의 사용은 부울 플래그(imqConnectionFlowLimitEnabled)를 통해 제어됩니다.
- **사용자 흐름 제어**는 사용될 때까지 기다리고 있는 단일 사용자에 대해 보류 중일 수 있는 페이로드 메시지의 수(imqConsumerFlowLimit)를 제한합니다. 이 제한은 consumerFlowLimit와 같은 특정 대기열 대상의 등록 정보로 지정할 수도 있습니다. 제한에 도달하면 사용 대기 중인 메시지 수를 나타내는 imqConsumerFlowLimit의 비율이 imqConsumerFlowThreshold 속성에서 지정한 제한보다 낮을 때까지 사용자에 대한 페이로드 메시지의 전달이 일시 중단됩니다. 따라서 여러 사용자가 동일한 연결에 집중되지 않도록 함으로써 사용자 간 로드 균형 유지를 향상시킬 수 있습니다.

이러한 흐름 제어 기술 중 하나를 사용하려면 신뢰성과 처리량을 적절히 조정해야 합니다. 자세한 내용은 210 페이지 “클라이언트 런타임 메시지 흐름 조정”을 참조하십시오.

대기열 브라우저 및 서버 세션

표 16–6은 클라이언트 대기열 찾아보기와 서버 세션에 영향을 주는 연결 팩토리 속성을 나열합니다. imqQueueBrowserMaxMessagesPerRetrieve 속성은 대기열 대상 내용을 찾아볼 때 한 번에 검색할 수 있는 최대 메시지 수를 지정합니다. imqQueueBrowserRetrieveTimeout은 메시지 검색 시 최대 대기 시간을 지정합니다. imqQueueBrowserMaxMessagesPerRetrieve는 검색되는 총 메시지 수에는 영향을 주지 않고 클라이언트 런타임에 전달하기 위해 메시지가 청크되는 방법에만 영향을 줍니다. 청크가 클수록 메시지 수는 적고, 청크가 작을수록 메시지 수는 많아집니다. 클라이언트 응용 프로그램은 항상 대기열의 모든 메시지를 받습니다. 속성 값을 변경하면 성능에 영향을 줄 수 있지만, 검색되는 총 데이터 양에는 영향을 주지 않습니다.부울 속성 imqLoadMaxToServerSession은 응용 프로그램 서버 세션에서 연결 사용자의 동작을 제어합니다. 이 속성의 값이 true이면 클라이언트는 서버 세션에 최대 메시지 수까지 로드할 수 있습니다. false이면 한 번에 하나의 메시지만 로드합니다.

표준 메시지 등록 정보

JMS(Java Message Service) 사양은 JMS 공급자(Message Queue 등)가 필요에 따라 지원을 선택할 수 있는 특정 표준 메시지 등록 정보를 정의합니다. 편의상 이러한 표준 등록 정보의 이름은 모두 JMSX로 시작됩니다. 표 16–7에 나열된 연결 팩토리 속성은 Message Queue 클라이언트 런타임에서 특정 표준 등록 정보를 설정할지 여부를 제어합니다. 생성된 메시지의 경우에는 다음 등록 정보를 포함합니다.

- JMSXUserID 메시지를 보내는 사용자의 아이디
- JMSXAppID 메시지를 보내는 응용 프로그램의 아이디

- JMSXProducerTXID 메시지가 생성된 트랜잭션의 트랜잭션 식별자

사용된 메시지의 경우에는 다음 등록 정보를 포함합니다.

- JMSXConsumerTXID 메시지가 사용된 트랜잭션의 트랜잭션 식별자
- JMSXRcvTimestamp 메시지가 사용자에게 전달된 시간

메시지 헤더 대체

표 16–8에 나열된 연결 팩토리 속성을 사용하여 특정 JMS 메시지 헤더 필드에 대해 클라이언트가 설정한 값을 무시할 수 있습니다. 사용자가 지정한 설정은 연결 팩토리에서 가져온 연결에서 생성된 모든 메시지에 사용됩니다. 이 방법으로 무시할 수 있는 헤더 필드는 다음과 같습니다.

- JMSDeliveryMode 전달 모드(지속성 또는 비지속성
- JMSExpiration 만료 시간
- JMSPriority 우선 순위 수준

이러한 각 필드에는필드를 무시할 수 있는지 여부를 제어하는 부울 속성과 값을 지정하기 위한 부울 속성의 두 가지 속성이 있습니다. 예를 들어, 우선 순위 수준을 설정하기 위한 속성은 imqOverrideJMSPriority와 imqJMSPriority입니다. 임시 대상에 적용되는 값을 무시할지 여부를 제어하는 imqOverrideJMSHeadersToTemporaryDestinations 속성도 있습니다.

주 – 무시되는 메시지 헤더는 특정 응용 프로그램의 요구에 맞지 않을 수도 있으므로 이 속성은 반드시 응용 프로그램 설계자 또는 사용자와 상의하여 사용해야 합니다.

대상 속성

물리적 대기열 또는 주제 대상을 식별하는 **대상** 관리 객체는 표 16–9에 나열된 두 가지 속성만 갖습니다. 중요한 한 가지 속성은 imqDestinationName으로, 이 속성은 관리 대상 객체가 나타내는 물리적 대상의 이름을 제공합니다. 이 이름은 물리적 대상을 만든 imqcmd create dst 명령에 -n 옵션을 사용하여 지정된 이름입니다. 대상 관리 객체와 해당 객체가 나타내는 물리적 대상이 반드시 일대일 관계가 아니어도 됩니다. 하나의 물리적 대상을 여러 관리 대상 객체가 참조할 수도 있고 물리적 대상을 참조하는 관리 대상 객체가 없을 수도 있습니다.또한 imqDestinationDescription이라는 선택적 설명 문자열도 있습니다. 이 문자열을 사용하면 대상 객체를 식별하거나 사용자가 만든 대상 객체를 다른 대상 객체와 구분하기가 쉽습니다.

객체 관리자 유틸리티 사용

Message Queue 객체 관리자 유틸리티(`imqobjmgr`)를 사용하면 관리 대상 객체를 만들고 관리할 수 있습니다. `imqobjmgr` 명령은 관리 대상 객체에 대한 다양한 작업을 수행하기 위해 다음과 같은 하위 명령을 제공합니다.

`add`	관리 대상 객체를 객체 저장소에 추가
`delete`	객체 저장소에서 관리 대상 객체 삭제
`list`	객체 저장소에 기존 관리 대상 객체 나열
`query`	관리 대상 객체에 대한 정보 표시
`update`	관리 대상 객체의 속성 수정

`imqobjmgr` 명령의 구문, 하위 명령 및 옵션에 대한 자세한 내용은 250 페이지 "객체 관리자 유틸리티"를 참조하십시오.

대부분의 객체 관리자 작업의 경우 `imqobjmgr` 명령 옵션으로 다음 정보를 지정해야 합니다.

- 관리 대상 객체의 **JNDI 조회 이름**(`-l`)

  이 이름은 클라이언트 응용 프로그램에서 JNDI(Java Naming and Directory Interface)를 사용하여 객체 저장소의 관리 객체를 조회할 수 있는 논리적 이름입니다.
- **JNDI 객체 저장소의 속성**(`-j`)

  사용 가능한 속성 및 값에 대한 자세한 내용은 153 페이지 "객체 저장소"를 참조하십시오.
- 관리 대상 객체의 **유형**(`-t`)

  사용 가능한 유형은 다음과 같습니다.

`q`	대기열 대상
`t`	주제 대상
`cf`	연결 팩토리
`qf`	대기열 연결 팩토리
`tf`	주제 연결 팩토리
`xcf`	분산 트랜잭션의 연결 팩토리
`xqf`	배포 트랜잭션의 대기열 연결 팩토리
`xtf`	분산 트랜잭션의 주제 연결 팩토리
`e`	SOAP 종점
- 관리 대상 객체의 **속성**(`-o`)

사용 가능한 속성 및 값에 대한 자세한 내용은 156 페이지 “관리 객체 속성”을 참조하십시오.

관리 객체 추가

imqobjmgr 명령의 add 하위 명령은 연결 팩토리의 관리 대상 객체나 주제 또는 대기열 대상을 객체 저장소에 추가합니다. LDAP 객체 저장소에 저장된 관리 대상 객체에는 cn=; 접두사로 시작하는 조회 이름이 지정되어야 하며, 파일 시스템 객체 저장소의 조회 이름은 특정 접두사로 시작할 필요는 없지만 슬래시 문자(/)를 포함할 수 없습니다.

주 – 객체 관리자는 Message Queue 관리 대상 객체만 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 브로커 연결을 생성할 수 있도록 하려면 생성할 연결 유형(대기열 연결 팩토리 또는 주제 연결 팩토리)에 대해 연결 팩토리 관리 대상 객체를 추가합니다. 예 8–1은 대기열 연결 팩토리(관리 대상 객체 유형 qf)를 LDAP 객체 저장소에 추가하는 명령을 보여줍니다. 객체에는 조회 이름 cn=myQCF가 지정되며 jms 연결 서비스를 사용하여 7272 포트 번호를 통해 myHost 호스트에서 실행 중인 브로커에 연결합니다.

예 8–1 연결 팩토리 추가

```
imqobjmgr add
   -l "cn=myQCF"
   -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=homerSimpson,ou=People,o=imq"
   -j "java.naming.security.credentials=doh"
   -j "java.naming.security.authentication=simple"
   -t qf
   -o "imqAddressList=mq://myHost:7272/jms"
```

대상 추가

대상을 나타내는 관리 객체를 만드는 경우 관리 객체를 객체 저장소에 추가하기 전에 먼저 물리적 대상을 만드는 것이 좋습니다. 115 페이지 “물리적 대상 만들기”에 설명한 대로 명령 유틸리티(imqcmd)를 사용하여 물리적 대상을 만듭니다.

예 8–2에 나와 있는 명령은 조회 이름 myTopic을 사용하여 주제 대상을 나타내는 LDAP 객체 저장소에 관리 대상 객체를 추가하며, 물리적 대상 이름은 physTopic입니다. 대기열

대상을 추가하는 명령도 이와 비슷하며, 관리 대상 객체 유형(-t 옵션)으로 t("주제 대상") 대신에 q("대기열 대상")를 사용하는 것이 다릅니다.

예 8–2 LDAP 객체 저장소에 대상 추가

```
imqobjmgr add
   -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=homerSimpson,ou=People,o=imq"
   -j "java.naming.security.credentials=doh"
   -j "java.naming.security.authentication=simple"
   -t t
   -o "imqDestinationName=physTopic"
```

예 8–3은 LDAP 서버 대신에 Solaris 파일 시스템에 저장된 관리 대상 객체를 사용하는 동일한 명령을 보여줍니다.

예 8–3 파일 시스템 객체 저장소에 대상 추가

```
imqobjmgr add
   -l "cn=myTopic"
   -j "java.naming.factory.initial=
          com.sun.jndi.fscontext.RefFSContextFactory"
   -j "java.naming.provider.url=file:///home/foo/imq_admin_objects"
   -t t
   -o "imqDestinationName=physTopic"
```

관리 객체 삭제

객체 저장소에서 관리 대상 객체를 삭제하려면 imqobjmgr 명령의 delete 하위 명령을 사용하고 삭제할 객체의 조회 이름, 유형 및 위치를 지정합니다. 예 8–4에 표시된 명령은 위 164 페이지 “대상 추가”에서 추가된 객체를 삭제합니다.

예 8–4 관리 객체 삭제

```
imqobjmgr delete
   -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=homerSimpson,ou=People,o=imq"
   -j "java.naming.security.credentials=doh"
   -j "java.naming.security.authentication=simple"
   -t t
```

관리 객체 나열

객체 관리자의 `list` 하위 명령을 사용하면 객체 저장소의 모든 관리 대상 객체 또는 특정 유형의 관리 대상 객체 목록을 가져올 수 있습니다. 예 8–5는 LDAP 서버의 모든 관리 대상 객체를 나열하는 방법을 보여줍니다.

예 8–5 모든 관리 객체 나열

```
imqobjmgr list
   -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=homerSimpson,ou=People,o=imq"
   -j "java.naming.security.credentials=doh"
   -j "java.naming.security.authentication=simple"
```

예 8–6은 모든 대기열 대상(유형 `q`)을 나열합니다.

예 8–6 특정 유형의 관리 객체 나열

```
imqobjmgr list
   -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=homerSimpson,ou=People,o=imq"
   -j "java.naming.security.credentials=doh"
   -j "java.naming.security.authentication=simple"
   -t q
```

관리 객체 정보 보기

`query` 하위 명령은 객체의 조회 이름과 객체를 포함하는 객체 저장소의 속성을 통해 식별되는 지정된 관리 대상 객체에 대한 정보를 표시합니다. 예 8–7은 조회 이름이 `cn=myTopic`인 객체에 대한 정보를 표시합니다.

예 8–7 관리 객체 정보 보기

```
imqobjmgr query
   -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=homerSimpson,ou=People,o=imq"
   -j "java.naming.security.credentials=doh"
   -j "java.naming.security.authentication=simple"
```

관리 객체 속성 수정

관리 대상 객체의 속성을 수정하려면 `imqobjmgr update` 하위 명령을 사용합니다. 객체의 조회 이름 및 위치를 입력하고 `-o` 옵션을 사용하여 새 속성 값을 지정합니다.

167 페이지 "관리 객체 속성 수정"에서는 예 8–8에서 객체 저장소에 추가된 대기열 연결 팩토리에 대한 `imqReconnectAttempts` 속성 값을 변경합니다.

예 8–8 관리 대상 객체 속성 수정

```
imqobjmgr update
   -l "cn=myQCF"
   -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=homerSimpson,ou=People,o=imq"
   -j "java.naming.security.credentials=doh"
   -j "java.naming.security.authentication=simple"
   -t qf
   -o "imqReconnectAttempts=3"
```

명령 파일 사용

`imqobjmgr` 명령에 대한 `-i` 옵션을 사용하면 하위 명령 절의 전체 또는 일부를 나타내기 위해 Java 등록 정보 파일 구문을 사용하는 명령 파일의 이름을 지정할 수 있습니다. 이 기능은 일반적으로 입력을 많이 해야 하거나 `imqobjmgr`의 여러 호출에서 동일한 내용을 입력해야 하는 객체 저장소 속성을 지정하는 데 특히 유용합니다. 명령 파일을 사용하면 명령줄에 허용되는 최대 문자 수를 초과하는 일을 방지할 수도 있습니다.

예 8–9는 객체 관리자 명령 파일의 일반 구문을 보여줍니다. `version` 속성은 명령줄 옵션이 아닙니다. 이 속성은 Message Queue 제품 버전이 아니라 명령 파일 자체의 버전을 참조하므로 값을 `2.0`으로 설정해야 합니다.

예 8–9 객체 관리자 명령 파일 구문

```
version=2.0
cmdtype=[ add | delete | list | query | update ]
obj.lookupName=lookup name
objstore.attrs.objStoreAttrName1=value1
objstore.attrs.objStoreAttrName2=value2
   . . .
objstore.attrs.objStoreAttrNameN=valueN
obj.type=[ q | t | cf | qf | tf | xcf | xqf | xtf | e ]
obj.attrs.objAttrName1=value1
obj.attrs.objAttrName2=value2
   . . .
```

예 8–9 객체 관리자 명령 파일 구문 *(계속)*

```
obj.attrs.objAttrNameN=valueN
```

예를 들어, 대기열 연결 팩토리를 LDAP 객체 저장소에 추가한 이전의 예 8–1에서 객체 관리자 명령을 살펴보겠습니다. 이 명령을 예 8–10에 표시된 대로 명령 파일에 캡슐화할 수 있습니다. 명령 파일의 이름이 `MyCmdFile`이면 명령줄에서 다음 명령을 실행할 수 있습니다.

```
imqobjmgr -i MyCmdFile
```

예 8–10 명령 파일 예

```
version=2.0
cmdtype=add
obj.lookupName=cn=myQCF
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=homerSimpson,ou=People,o=imq
objstore.attrs.java.naming.security.credentials=doh
objstore.attrs.java.naming.security.authentication=simple
obj.type=qf
obj.attrs.imqAddressList=mq://myHost:7272/jms
```

명령 파일을 사용하면 나머지 부분을 명령줄에 명시적으로 제공하면서 `imqobjmgr` 하위 명령 절의 일부만 지정할 수도 있습니다. 예를 들어, 예 8–11에 표시된 명령 파일은 LDAP 객체 저장소의 속성 값만 지정합니다.

예 8–11 부분 명령 파일

```
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=homerSimpson,ou=People,o=imq
objstore.attrs.java.naming.security.credentials=doh
objstore.attrs.java.naming.security.authentication=simple
```

그리고 나면 이 명령 파일을 사용하여 예 8–12에 표시된 대로 나머지 옵션을 명시적으로 제공하면서 `imqobjmgr` 명령에 객체 저장소를 지정할 수 있습니다.

예 8–12 부분 명령 파일 사용

```
imqobjmgr add
   -l "cn=myQCF"
   -i MyCmdFile
   -t qf
   -o "imqAddressList=mq://myHost:7272/jms"
```

사용자의 플랫폼에 따라 다음 위치에서 명령 파일의 예를 추가로 확인할 수 있습니다.

Solaris:`/usr/demo/imq/imqobjmgr` **Linux:**`/opt/sun/mq/examples/imqobjmgr`
Windows:`IMQ_HOME/demo/imqobjmgr`

◆ ◆ ◆ 9 장

9 브로커 클러스터 작업

Message Queue™ Enterprise Edition은 메시지 전달 서비스를 클라이언트에 제공하기 위해 함께 작업하는 브로커 그룹인 **브로커 클러스터**의사용을 지원합니다. 클러스터를 사용하여 메시지 서비스는 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.

- 171 페이지 "클러스터 구성 등록 정보"
- 173 페이지 "클러스터 관리"
- 176 페이지 "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 표 14–9에서 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 cluster 연결 서비스의 호스트 이름 또는 IP 주소를 지정합니다. 이 설정은 컴퓨터에 여러 개의 네트워크 인터페이스 카드가 있을 때처럼사용 가능한 호스트가 두 개 이상인 경우에 유용합니다.
- imq.cluster.port cluster 연결 서비스에 포트 번호를 제공합니다.
- imq.cluster.transport cluster 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 transport는 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), cluster 연결 서비스에 사용할 전송 프로토콜(imq.cluster.transport) 및 마스터 브로커의 주소(imq.cluster.masterbroker)와 같은 공유 구성 등록 정보를 정의합니다. 다음 코드는 마스터 브로커 역할을 하는 ctrlhost에서 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

일반적으로 명령줄에서 (`-cluster` 옵션 사용) 또는 클러스터 구성 파일에서 `imq.cluster.brokerlist` 등록 정보를 설정하는 두 가지 방법으로 클러스터에 브로커를 연결할 수 있습니다. 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.

명령줄에서 브로커 클러스터를 구성하려면 `-cluster` 옵션을 `imqbrokerd` 명령에 사용하여 클러스터 내의 전체 브로커 목록을 지정합니다. 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.

사용하는 방법에 관계없이 네트워크 루프백 IP 주소(`127.0.0.1`)로 확인되는 주소가 클러스터의 브로커에 지정되지 않아야 합니다. 이 주소로 구성된 브로커는 클러스터의 다른 브로커에 연결할 수 없습니다.

주 – 일부 Linux 설치 프로그램은 `localhost` 항목을 네트워크 루프백 주소로 자동으로 설정합니다. 그러한 시스템에서 사용자는 클러스터의 모든 브로커 주소가 제대로 지정될 수 있도록 시스템 IP 주소를 수정해야 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 로컬 `hosts` 파일을 참조하기 전에 DNS 조회를 수행하도록 항목의 순서를 변경합니다. `/etc/nsswitch.conf`의 행은 다음과 같아야 합니다.

```
hosts:dns files
```

클러스터 내의 브로커 간에 암호화된 보안 메시지를 전달하려면 SSL 기반 전송 프로토콜을 사용하도록 `cluster` 연결 서비스를 구성합니다. 141 페이지 “메시지 암호화”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 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. 클러스터의 모든 브로커에 대해 이 명령을 실행할 필요는 없습니다. 한 브로커에 대해 명령을 실행하면 모든 브로커에서 클러스터 구성이 다시 로드됩니다.

3 *(선택 사항)* 클러스터 구성 파일을 가리키도록 브로커의 `config.properties` 파일에서 `imq.cluster.url` 등록 정보 값을 설정합니다.

4 **새 브로커를 시작합니다.**

174 페이지 “클러스터에 브로커 추가”를 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 `cluster` 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 `imqbrokerd` 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 **`imqcmd` 명령을 사용하여 클러스터에서 각 브로커를 중지합니다.**

2 **`imqbrokerd` 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.

```
imqbrokerd -cluster A,B,
C
```

클러스터에서 *A* 브로커를 제거하려면 다음 명령을 사용하여 *B*와 *C* 브로커를 다시 시작합니다.

```
imqbrokerd -cluster B,C
```

클러스터 구성 파일을 사용하여 브로커 제거

처음에 중앙 클러스터 구성 파일의 `imq.cluster.brokerlist` 등록 정보를 사용하여 구성원 브로커를 지정해 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 `imq.cluster.brokerlist` 등록 정보에 지정된 목록에서 제외된 브로커를 제거합니다.**

2. **클러스터에 남아 있는 각 브로커에 대해 다음 명령을 실행합니다.**

   ```
   imqcmd 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 177 페이지 "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

발생된 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. 예:

```
imqbrokerd -backup mybackuplog
```

▼ 구성 변경 기록을 복원하는 방법

1. **클러스터에 있는 모든 브로커를 종료합니다.**
2. **다음 명령을 사용하여 백업 파일로부터 마스터 브로커의 구성 변경 기록을 복원합니다.**

   ```
   imqbrokerd -restore mybackuplog
   ```

3. **마스터 브로커에 새 이름이나 포트 번호를 할당하는 경우 클러스터 구성 파일에서 `imq.cluster.brokerlist` 및 `imq.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 `imqcmd` 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.

◆ ◆ ◆ 10 장

10 브로커 모니터링

이 장에서는 브로커를 모니터하는 데 사용 가능한 도구와 메트릭 데이터를 가져오는 방법에 대해 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 179 페이지 "모니터링 도구 소개"
- 181 페이지 "브로커 로깅 구성 및 사용"
- 186 페이지 "메트릭을 대화식으로 표시"
- 190 페이지 "브로커를 모니터하는 응용 프로그램 작성"

특정 메트릭에 대한 자세한 내용은 18 장을 참조하십시오.

모니터링 도구 소개

Message Queue™ 정보에 대한 모니터링 인터페이스에는 로그 파일, 대화식 명령 및 메트릭을 얻을 수 있는 클라이언트 API가 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 살피거나 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 체킹에 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 18 장을 참조하십시오.

브로커 로깅 구성 및 사용

Message Queue 로거는 브로커 코드, 디버거, 메트릭 생성기에서 생성한 정보를 가져와서이 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 참조).

```
.../instances/instanceName/log
```

주 – 라이프사이클이 Application Server에 의해 제어되는 브로커의 경우 해당 브로커가 시작된 도메인에 대한 도메인 디렉토리의 하위 디렉토리에 로그 파일이 있습니다.

```
.../appServer_domainName_dir/imq/instances/imqbroker/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 9개의 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 표 14–8에 설명되어 있는 imq.log.timezone 등록 정보에 대한 정보를 참조하십시오.

로거 구성 변경

로그 관련 등록 정보에 대한 설명은 표 14–8에 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 -D를 사용하여 로거 등록 정보(또는 **모든** 브로커 등록 정보)를 변경할 수도 있습니다.

명령줄에 전달되는 옵션은 브로커 인스턴스 구성 파일에서 지정한 등록 정보를 대체합니다. 로깅에 영향을 주는 imqbrokerd 옵션은 다음과 같습니다.

`-metrics` *interval*	브로커 메트릭 로깅 간격(초)
`-loglevel` *level*	로깅 수준(ERROR, WARNING, INFO 또는 NONE)
`-silent`	자동 모드(콘솔에 기록하지 않음)
`-tty`	모든 메시지를 콘솔에 기록

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 syslog에 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 ERROR로 설정하고 imq.log.console.output 등록 정보를 WARNING으로 설정한 경우에는 WARNING 메시지의 로깅을 활성화하지 않았기 때문에 메시지가 기록되지 않습니다.

로그 파일 롤오버 기준 변경

로그 파일의 롤오버 기준에는시간과 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 181 페이지 "브로커 로깅 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`

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
```

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 18 장을 참조하십시오.

사용 불능 메시지 기록

브로커에 대한 사용 불능 메시지 로깅을 사용하여 물리적 대상을 모니터할 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 큽니다.
  - 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.

다음은 사용 불능 메시지에 대한 로그 형식의 예입니다.

```
[29/Mar/2006:15:35:39 PST] [B1147]: Message 8-129.145.180.87(e7:6b:dd:5d:98:aa)-
35251-1143675279400 from destination Q:q0 has been placed on the DMQ because
[B0053]: Message on destination Q:q0 Expired: expiration time 1143675279402,
```

```
arrival time 1143675279401, JMSTimestamp 1143675279400
```

사용 불능 메시지 로깅은 기본적으로 비활성화됩니다. 사용 불능 메시지 로깅을 사용하려면 `imq.destination.logDeadMsgs` 브로커 속성을 설정합니다.

메트릭을 대화식으로 표시

Message Queue 브로커는 다음과 같은 이벤트 유형을 보고할 수 있습니다.

- **JVM(Java 가상 머신) 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 자원에 대한 정보 및 특정 연결 서비스의 메시지 흐름에 대한 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 `imqcmd`의 `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 189 페이지 “imqcmd query”를 참조하십시오.

imqcmd metrics

`imqcmd metrics`의 구문과 옵션은 각각 표 10–3 및 표 10–4에 나타나 있습니다.

표 10–3 `imqcmd metrics` 하위 명령 구문

하위 명령 구문	제공되는 메트릭 데이터

`metrics bkr`
`[-b` *hostName*`:`*portNumber*`]`
`[-m` *metricType*`]`
`[-int` *interval*`]`
`[-msp` *numSamples*`]`
	기본 브로커 또는 지정한 호스트 및 포트의 브로커 메트릭을 표시합니다.

표 10–3 imqcmd metrics 하위 명령 구문 *(계속)*

하위 명령 구문	제공되는 메트릭 데이터

`metrics svc -n` *serviceName*
`[-b` *hostName*`:`*portNumber*`]`
`[-m` *metricType*`]`
`[-int` *interval*`]`
`[-msp` *numSamples*`]`
	기본 브로커 또는 지정한 호스트 및 포트의 브로커에서 지정된 서비스의 메트릭을 표시합니다.

`metrics dst -t` *destType*
`-n` *destName*
`[-b` *hostName*`:`*portNumber*`]`
`[-m` *metricType*`]`
`[-int` *interval*`]`
`[-msp` *numSamples*`]`
	지정한 유형과 이름의 물리적 대상에 대한 메트릭 정보를 표시합니다.

표 10–4 imqcmd metrics 하위 명령 옵션

하위 명령 옵션	설명
`-b` *hostName*: *portNumber*	메트릭 데이터를 보고하는 브로커의 호스트 이름과 포트를 지정합니다. 기본값은 `localhost:7676`입니다.
`-int` *interval*	메트릭을 표시할 간격(초)을 지정합니다. 기본값은 5초입니다.
`-m` *metricType*	표시할 메트릭 유형을 지정합니다. **`ttl`** 브로커, 서비스 또는 대상에 유입되고 유출되는 메시지와 패킷에 대한 메트릭을 표시합니다(기본 메트릭 유형). **`rts`** 브로커, 연결 서비스 또는 대상에 유입 및 유출되는 메시지와 패킷의 메트릭을 초당 속도로 표시합니다. **`cxn`** 연결, 가상 메모리 힙 및 스레드를 표시합니다(브로커 및 연결 서비스에만 해당). **`con`** 사용자 관련 메트릭을 표시합니다(대상에만 해당). **`dsk`** 디스크 사용 메트릭을 표시합니다(대상에만 해당).
`-msp` *numSamples*	출력에 표시되는 샘플 수를 지정합니다. 기본값은 무제한 수입니다(무한).
`-n` *destName*	메트릭 데이터가 보고되는 물리적 대상(있는 경우)의 이름을 지정합니다. 기본값이 없습니다.
`-n` *serviceName*	메트릭 데이터가 보고되는 연결 서비스(있는 경우)를 지정합니다. 기본값이 없습니다.
`-t` *destType*	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.**
68 페이지 “브로커 시작”을 참조하십시오.

2 **표 10–3과 표 10–4에 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 결과를 출력합니다(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 결과를 출력합니다(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 결과를 출력합니다(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 결과를 출력합니다(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 10–5에 나와 있습니다.

표 10–5 imqcmd query 하위 명령 구문

하위 명령 구문	제공되는 메트릭 데이터
query bkr [-b *hostName*: *portNumber*]	브로커 메모리와 영구 저장소에 저장되어 있는 현재의 메시지 수와 메시지 바이트에 대한 정보(98 페이지 "브로커 정보 표시" 참조)
또는	
query svc -n *serviceName* [-b *hostName*:*portNumber*]	지정한 연결 서비스에 대한 현재의 할당된 스레드 수와 연결 수에 대한 정보(104 페이지 "연결 서비스 정보 표시" 참조)
또는	
query dst -t *destType* -n *destName* [-b *hostName*:*portNumber*]	지정한 대상의 메모리와 영구 저장소에 저장되어 있는 생성자, 활성 및 백업 사용자, 메시지 및 메시지 바이트의 현재 수에 대한 정보(117 페이지 "물리적 대상 정보 표시" 참조)

주 – imqcmd query에서 제공하는 메트릭 데이터는 제한되어 있기 때문에 이 도구는 18장에 제시된 표에 설명되어 있지 않습니다.

브로커를 모니터하는 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 10–6에 그 이름과 함께 각 대상에 전달되는 메트릭 메시지 유형이 표시되어 있습니다.

표 10–6 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 대상 유형 목록

표 10–6 메트릭 주제 대상 (계속)

주제 이름	메트릭 메시지 유형
mq.metrics.destination.queue.*monitoredDestinationName*	지정된 이름의 대기열에 대한 대상 메트릭
mq.metrics.destination.topic.*monitoredDestinationName*	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 **Java 클라이언트용 Message Queue 개발 안내서**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.

`imq.metrics.topic.persist`를 설정합니다.

기본값은 `false`입니다.

d. 메트릭 메시지가 삭제되기 전까지 해당 대상에 남아 있는 기간을 지정합니다.

`imq.metrics.topic.timetolive`를 설정합니다.

기본값은 300초입니다.

3 메트릭 주제 대상에 대한 액세스 제어가 필요한 경우 설정합니다.

아래의 192 페이지 “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.

135 페이지 “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 user1과 user2를 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 모니터할 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 141 페이지 “메시지 암호화”를 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 18 장을 참조하십시오.

◆ ◆ ◆ 11 장

11 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.

- 193 페이지 "성능 정보"
- 196 페이지 "성능에 영향을 미치는 요소"
- 206 페이지 "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 서비스가 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 수입니다.
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 측정한 것입니다.

이러한 성능의 여러 요소들은 일반적으로 상호 연관됩니다. 메시지 처리량이 높으면 메시지가 브로커에서 백로그될 가능성이 적어지므로 대기 시간이 짧아집니다(단일 메시지가 매우 빠르게 전달될 수 있음). 하지만 대기 시간은 통신 연결 속도, 브로커 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 모니터하여 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(198 페이지 “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, 파일 기반 지속성 대 JDBC 기반 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 요소에 의해 변동됩니다.

기본 사용 패턴을 설정하려면 메시지 서비스를 오랫동안 모니터하여 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 브로커 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. 아래는 안정적으로 전달되는 지속성 메시지인 경우에 대해 이러한 단계를 보여 줍니다. 단계는 다음 그림과 같습니다.

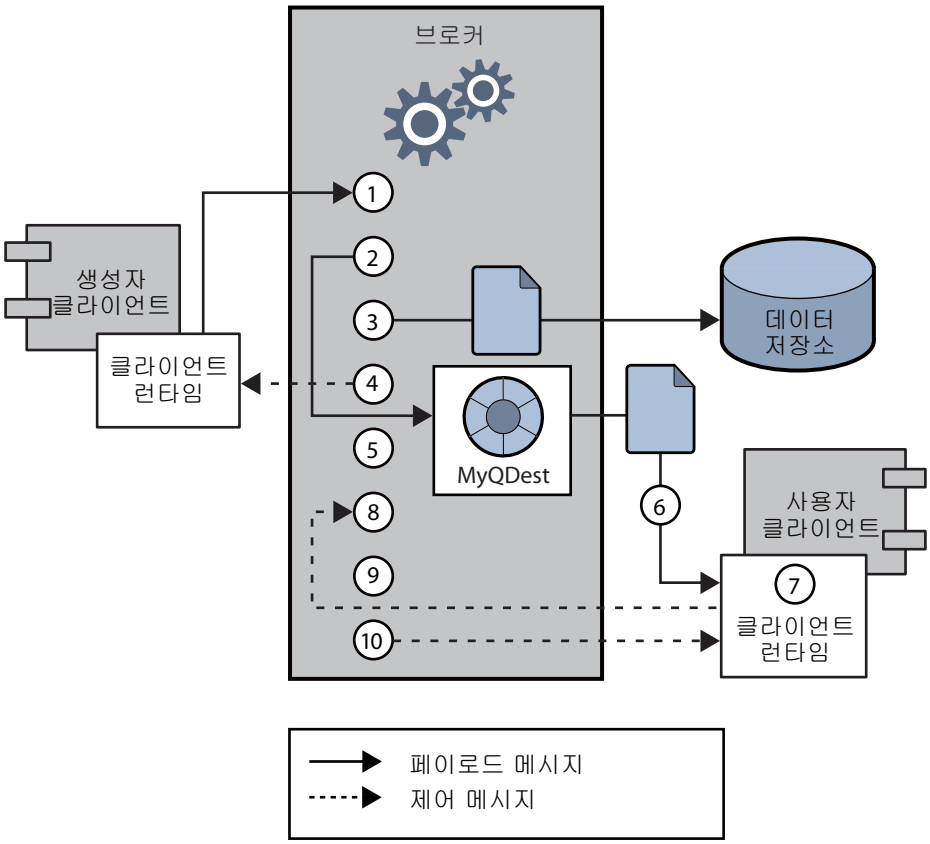


그림 11–1 Message Queue 서비스를 통한 메시지 전달

▼ 메시지 전달 단계

1. 메시지가 생성자 클라이언트에서 브로커로 전달됩니다.
2. 브로커가 메시지를 읽습니다.
3. 안정성을 위해 메시지가 영구 저장소에 저장됩니다.
4. 안정성을 위해 브로커가 메시지 수신을 확인합니다.
5. 브로커가 메시지 경로 지정을 결정합니다.
6. 브로커가 메시지를 씁니다.

7 **메시지가 브로커에서 사용자 클라이언트로 전달됩니다.**

8 **안정성을 위해 사용자 클라이언트가 메시지 수신을 확인합니다.**

9 **안정성을 위해 브로커가 클라이언트 확인을 처리합니다.**

10 **브로커가 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 서비스 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.

- 199 페이지 “전달 모드(지속성/비지속성 메시지)”
- 199 페이지 “트랜잭션 사용”
- 200 페이지 “확인 모드”
- 201 페이지 “영구 및 비영구 가입”

성능에 영향을 미치는 다른 응용 프로그램 설계 요소는 다음과 같습니다.

- 201 페이지 “선택기 사용(메시지 필터링)”
- 201 페이지 “메시지 크기”
- 202 페이지 “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 11–1은 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 or 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

전달 모드(지속성/비지속성 메시지)

지속성 메시지는 브로커에 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.

영구 가입자가 있는 대기열과 주제는 비영구 메시지에 비해 속도가 약 40% 향상되었습니다. 이는 10k 크기 메시지와 AUTO_ACKNOWLEDGE 모드에서 생성된 결과입니다.

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 일반적으론 그럴 수 없는데도 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 브로커가 클라이언트에 전달한 메시지 사용에 대해 클라이언트가 확인 응답을 보내는 것입니다.

클라이언트의 메시지 확인 없이 세션이 닫히거나 확인이 처리되기 전에 브로커에 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 브로커가 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` 및 `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` 및 `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 및 비영구 가입

주제 대상의 가입자는 영구 가입자와 비영구 가입자의 두 가지 범주로 구분됩니다.

영구 가입은 다음과 같은 이유로 안정성이 향상되지만 처리량이 떨어집니다.

- Message Queue 메시지 서비스는 브로커에 오류가 발생하더라도 복구 후 목록을 사용할 수 있도록 각 영구 가입에 할당된 메시지 목록을 영구 저장해야 합니다.
- 브로커에 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. 브로커에 오류가 발생하는 경우 해당 사용자 연결이 끊기며 메시지가 전달되지 않습니다.

지속성 및 비지속성 10k 크기 메시지의 두 경우에 대한 영구 가입자와 비영구 가입자의 성능을 비교했습니다. 두 가지 경우 모두 AUTO_ACKNOWLEDGE 확인 모드를 사용합니다. 지속성 메시지의 경우에만 약 30%의 성능 감소 효과가 있음을 확인했습니다.

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 사용자에 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 >1은 NumberOfOrders 등록 정보 값이 2이상인 메시지만 전달합니다.

선택기를 사용하여 사용자를 생성하면 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 메시징 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.

AUTO_ACKNOWLEDGE 확인 모드에서 대기열 대상에 1k, 10k 및 100k 크기의 메시지를 전달할 때의 초당 처리량 비교 테스트에서, 비영구 메시지의 경우 1k 메시지는 약 50%, 10k 메시지는 약 20%, 100k 메시지는 약 5% 더 빨라졌습니다. 메시지 크기는 영구 메시지와

비영구 메시지 모두에서 성능에 큰 영향을 미쳤습니다. 100k 메시지는 10k 메시지보다 약 10배 더 빠르지만, 10k 메시지는 1k 메시지보다 약 5배 더 빠릅니다.

메시지 본문 유형

JMS는 복잡성 순서에 따라 아래에 대략적으로 표시된 다섯 개의 메시지 본문 유형을 지원합니다.

- `BytesMessage`는 응용 프로그램에서 지정하는 형식의 바이트 집합을 포함합니다.
- `TextMessage`는 간단한 Java 문자열입니다.
- `StreamMessage`는 Java 프리미티브 값의 스트림을 포함합니다.
- `MapMessage`는 일련의 이름-값 쌍을 포함합니다.
- `ObjectMessage`는 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.

- 202 페이지 “하드웨어”
- 203 페이지 “운영 체제”
- 203 페이지 “Java 가상 머신(JVM)”
- 203 페이지 “연결”
- 205 페이지 “브로커 제한 및 동작”
- 204 페이지 “메시지 서비스 구조”
- 205 페이지 “데이터 저장소 성능”
- 205 페이지 “클라이언트 런타임 구성”

아래의 절에서는 이러한 각 요소가 메시징 성능에 미치는 영향을 설명합니다.

하드웨어

Message Queue 브로커나 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 일반적으로 하드웨어 업그레이드만으로 병목 현상을 극복하려면 비용이 많이 듭니다.

운영 체제

하드웨어 플랫폼이 같은 경우라도 각기 다른 운영 체제의 효율성으로 인해 성능은 다양할 수 있습니다. 예를 들어, 운영 체제에서 사용하는 스레드 모델은 브로커가 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)

브로커는 호스트 JVM에서 실행되고 지원되는 Java 프로세스입니다. 결과적으로 JVM 처리는 브로커가 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 서비스가 처리할 수 있는 메시지 수와 메시지 전달 속도에 영향을 미칠 수 있습니다.

브로커 연결 제한

브로커에 대한 모든 액세스는 연결을 통해서 이루어집니다. 동시 연결 수에 대한 제한은 브로커를 동시에 사용할 수 있는 생성자나 사용자 클라이언트 수에 영향을 미칠 수 있습니다.

일반적으로 브로커에 대한 연결 수는 사용 가능한 스레드 수에 의해 제한됩니다. 전용 스레드 모델이나 공유 스레드 모델 중 하나를 지원하도록 Message Queue를 구성할 수 있습니다(77 페이지 “스레드 풀 관리” 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 브로커와 통신할 수 있습니다. Message Queue는 76 페이지 “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.

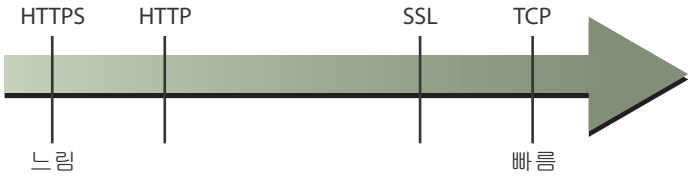


그림 11–2 전송 프로토콜 속도

테스트를 통해 높은 안정성 시나리오(영구 가입이 있고 AUTO_ACKNOWLEDGE 확인 모드를 사용하는 주제 대상에 1k의 지속성 메시지를 보냄)와 높은 성능 시나리오(영구 가입이 없고 DUPS_OK_ACKNOWLEDGE 확인 모드를 사용하는 주제 대상에 1k의 비지속성 메시지를 보냄)의 두 가지 경우에 대해 TCP와 SSL의 처리량을 비교했습니다.

일반적으로 프로토콜은 높은 안정성 사례에서 미치는 영향이 더 적습니다. 이는 아마도 높은 안정성 사례에서 필요한 지속성 오버헤드가 처리량을 제한하는 데 프로토콜 속도보다 더 중요한 요소이기 때문일 것입니다. 또는

- 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.
- 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 서비스는 단일 브로커로 구현되거나 다중 상호 연결 브로커 인스턴스로 구성된 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 서비스에 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 9 장을 참조하십시오.

브로커 제한 및 동작

브로커가 처리해야 하는 메시지 처리량은 브로커가 지원하는 메시징 응용 프로그램의 사용 패턴에 달려 있습니다. 하지만 브로커는 메모리, CPU 사이클 등 자원이 제한되어 있습니다. 따라서 브로커가 넘치게 되어 응답하지 않거나 불안정하게 될 수 있습니다.

Message Queue 메시지 브로커에는 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 모니터하고 조정하면 메시지 유입과 유출의 균형을 유지하여 시스템 과부하가 발생할 수 없도록 하는 데 사용할 수 있습니다. 이 메커니즘은 오버헤드를 사용하고 메시지 처리량을 제한하는 단점이 있지만 운영 무결성을 유지할 수 있습니다.

데이터 저장소 성능

Message Queue는 파일 기반 및 JDBC 기반 지속성 모듈을 모두 지원합니다. 파일 기반 지속성에서는 개별 파일을 사용하여 영구 데이터를 저장합니다. JDBC 기반 지속성은 JDBC™(Java Database Connectivity) 인터페이스를 사용하고 JDBC 호환 데이터 저장소가 필요합니다. 일반적으로 파일 기반 지속성은 JDBC 기반보다 빠르지만, JDBC 호환 저장소가 제공하는 중복 및 관리 제어 기능을 선호하는 사용자도 있습니다.

파일 기반 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. 이 클라이언트 런타임은 클라이언트가 물리적 대상에게 메시지를 보내고 대상으로부터 메시지를 받는 데 필요한 모든 작업을 지원합니다. 연결 팩토리 속성 값을 설정하여 클러이언트 런타임을 구성할 수 있으므로 연결 흐름 측정, 사용자 흐름 제한 및 연결 흐름 제한과 같은 다양한 동작 측면을 제어하여 성능 및 메시지 처리량을 향상시킬 수 있습니다. 이러한 기능과 기능을 구성하는 데 사용되는 속성에 대한 자세한 내용은 210 페이지 “클라이언트 런타임 메시지 흐름 조정”을 참조하십시오.

성능 향상을 위한 구성 조정

다음 절에서는 구성 조정이 성능에 미치는 영향에 대해 설명합니다.

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 비해 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 `imqsvcadmin install` 명령에 `-vmargs` 옵션을 사용하여 JVM 인수를 지정합니다. 13 장의 254 페이지 “서비스 관리자 유틸리티”를 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.`*protocolType* `.nodelay`
- `imq.protocol.`*protocolType* `.inbufsz`
- `imq.protocol.`*protocolType* `.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(208 페이지 “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 1KB를 더한 크기(가장 가까운 KB값으로 반올림)로 설정하는 것이 좋습니다. 예를 들어, 브로커가 본문 크기가 1KB인 패킷을 받는 경우 패킷의 전체 크기(메시지 본문 + 헤더 + 등록 정보)는 약 1200바이트입니다. inbufsz가 2KB(2048바이트)인 경우 적당한 성능을 제공합니다. inbufsz나 outbufsz를 이 크기보다 더 늘리면 성능은 약간 늘어날 수 있지만 각 연결에 필요한 메모리가 늘어나게 됩니다.

HTTP/HTTPS 조정

앞의 두 절에서 설명한 일반 등록 정보 이외에도 HTTP/HTTPS 성능은 클라이언트가 Message Queue 터널 서블릿을 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 폴하는 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 폴하는 간격을 지정합니다.

pullPeriod 값이 -1(기본값)인 경우 클라이언트 런타임은 이전 요청이 반환되자마자 서버를 폴하여 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 79 페이지 "지속성 서비스"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 6 장을 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.

메시지를 보낼 때 예외가 반환되면 클라이언트는 잠시 동안 일시 중지하고 전송을 다시 시도해야 합니다. 브로커에서 메시지를 받지 못해서 발생하는 예외는 없습니다. 예외가 발생된다면 송신측의 클라이언트에서 발생되는 예외뿐입니다.

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 Technical Overview**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 동작에 대해 설명합니다(205 페이지 “클라이언트 런타임 구성” 참조). 이러한 동작은 연결 팩토리 관리 객체의 속성으로 구성됩니다. 연결 팩토리 속성 설정에 대한 자세한 내용은 8 장을 참조하십시오.

메시지 흐름 측정

클라이언트가 보내고 받는 메시지(**페이로드 메시지**)와 Message Queue 제어 메시지는 동일한 클라이언트 브로커 연결을 통해 전달됩니다. 제어 메시지가 페이로드 메시지 전달로 인해 일시 중단되면 브로커 확인과 같은 제어 메시지 전달이 지연될 수 있습니다. 이러한 유형의 혼잡을 막기 위해 Message Queue는 연결에서 페이로드 메시지의 흐름을 측정합니다.

페이로드 메시지는 imqConnectionFlowCount 연결 팩토리 속성에서 지정한 대로 설정된 수만 전달되도록 일괄 처리됩니다. 일괄 처리가 전달된 후에는 페이로드 메시지 전달이 일시 중지되고 보류 중인 제어 메시지만 전달됩니다. 페이로드 메시지의 추가 일괄 처리가 전달된 다음 보류 중인 제어 메시지가 전달되는 식으로 이러한 주기가 반복됩니다.

클라이언트가 브로커에 많은 응답을 요구하는 작업을 수행 중인 경우 imqConnectionFlowCount 값을 유지해야 합니다. 예를 들어, 클라이언트가 CLIENT_ACKNOWLEDGE or AUTO_ACKNOWLEDGE 모드, 지속성 메시지, 트랜잭션 또는 대기열 브라우저를 사용하고 있거나 클라이언트가 사용자를 추가 또는 제거하고 있는 경우가 있습니다. 반면 클라이언트에 연결에서 DUPS_OK_ACKNOWLEDGE 모드를 사용하는 단순 사용자만 있는 경우 성능을 저하시키지 않고 imqConnectionFlowCount를 늘릴 수 있습니다.

메시지 흐름 제한

메모리와 같은 로컬 자원 제한이 발생하기 전에 Message Queue 클라이언트 런타임이 처리할 수 있는 페이로드 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 흐름 제한

클라이언트 런타임에 전달된 페이로드 메시지 수가 사용자의 imqConsumerFlowLimit 값을 초과하면 해당 사용자에 대해 메시지 전달이 중지됩니다. 해당 사용자의 사용되지 않은 메시지 수가 imqConsumerFlowThreshold에 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 - (current number of pending msgs in buffer
)
```

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%에 근접) 브로커가 다음 집합을 전달하기 전에 클라이언트가 버퍼링된 나머지 메시지를 모두 처리할 수 있으므로 메시지 처리량이 다시 저하됩니다. 일반적으로 특정 성능이나 안정성 문제가 있는 경우가 아니라면 imqConsumerFlowThreshold 속성의 기본값을 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 흐름 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** 사용자에 대해 버퍼링되는 전체 메시지 수를 제한합니다. 이 수가 imqConnectionFlowLimit의 값을 초과하면 전체 수가 연결 제한 아래로 떨어질 때까지 해당 연결을 통한 메시지 전달이 중지됩니다(imqConnectionFlowLimitEnabled를 true로 설정하는 경우에만 imqConnectionFlowLimit 속성을 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.

◆ ◆ ◆ 12장

12 문제 해결

이 장에서는 다음과 같은 문제를 이해하고 해결하는 방법을 설명합니다.

- 213 페이지 "클라이언트가 연결을 설정할 수 없음"
- 217 페이지 "연결 처리 속도가 너무 느림"
- 218 페이지 "클라이언트가 메시지 생성자를 만들 수 없음"
- 219 페이지 "메시지 생성이 지연되거나 느림"
- 222 페이지 "메시지가 백로그됨"
- 225 페이지 "브로커 처리량이 일정하지 않음"
- 226 페이지 "메시지가 사용자에게 도달하지 않음"
- 230 페이지 "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 일치하는지 확인합니다. 문제를 Sun에 보고할 때도 버전 번호가 필요합니다. 버전 번호를 확인하려면 다음 명령을 실행합니다.

```
imqcmd -v
```

클라이언트가 연결을 설정할 수 없음

증상:

- 클라이언트가 새 연결을 설정할 수 없습니다.
- 클라이언트가 실패한 연결을 자동으로 다시 연결할 수 없습니다.

가능한 원인:

- 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 플랫폼에서 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.

가능한 원인: 클라이언트 응용 프로그램이 연결을 닫을 수 없어 연결 수가 자원 제한을 초과합니다.

문제의 원인을 확인하는 방법: 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 해결하는 방법: 문제가 있는 클라이언트를 다시 작성하여 사용되지 않는 연결을 닫습니다.

가능한 원인: 브로커가 실행 중이지 않거나 네트워크 연결에 문제가 있습니다.

문제의 원인을 확인하는 방법:

- 브로커의 기본 포트(예: 기본값 7676)로 텔넷하고 브로커가 포트 매퍼 출력으로 응답하는지 확인합니다.
- 브로커 프로세스가 호스트에서 실행 중인지 확인합니다.

문제를 해결하는 방법:

- 브로커를 시작합니다.
- 네트워크 연결 문제를 해결합니다.

가능한 원인: 연결 서비스가 비활성 상태이거나 일시 중지되어 있습니다.

문제의 원인을 확인하는 방법: 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 해결하는 방법:

- 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 다음과 같은 항목과

  ```
  ERROR [B3009]: Unable to start service ssljms: [B4001]: Unable to open
  protocol tls for ssljms service...
  ```

  이 예외의 근본 원인에 대한 설명을 포함시키게 됩니다.

  SSL 서비스를 올바르게 구성하려면 141 페이지 “메시지 암호화”를 참조하십시오.

- 연결 서비스의 상태가 paused로 표시되어 있는 경우 서비스를 다시 시작합니다(106 페이지 "연결 서비스 일시 중지 및 다시 시작" 참조).

가능한 원인: 필요한 연결 수에 비해 사용 가능한 스레드 수가 너무 적습니다.

문제의 원인을 확인하는 방법: 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(77 페이지 "스레드 풀 관리" 참조).

문제를 해결하는 방법:

- 전용 스레드 풀 모델을 사용하고 있는 경우(imq. *serviceName*.threadpool_model=dedicated) 최대 연결 수는 스레드 풀의 최대 스레드 수의 반입니다. 따라서 연결 수를 늘리려면 스레드 풀의 크기를 늘리거나(imq. *serviceName*.max_threads) 공유 스레드 풀 모델로 전환합니다.
- 공유 스레드 풀 모델을 사용하고 있는 경우(imq. *serviceName*.threadpool_model=shared) 최대 연결 수는 연결 모니터 제한(imq.*serviceName*.connectionMonitor_limit)과 최대 스레드 수(imq. *serviceName*.max_threads)를 곱한 수의 반입니다. 따라서 연결 수를 늘리려면 스레드 풀의 크기를 늘리거나 연결 모니터 제한을 늘립니다.
- 결국 지원 가능한 연결 수(또는 연결의 처리량)는 입출력 제한에 도달합니다. 그런 경우 다중 브로커 클러스터를 사용하여 클러스터 내의 브로커 인스턴스로 연결을 분산합니다.

가능한 원인: Solaris나 Linux 플랫폼에서 필요한 연결 수에 비해 파일 설명자가 너무 적습니다.

이 문제에 대한 자세한 내용은 68 페이지 "파일 설명자 제한 설정"을 참조하십시오.

문제의 원인을 확인하는 방법: 브로커 로그에 다음과 유사한 항목이 있는지 확인합니다.

```
Too many open files
```

문제를 해결하는 방법: ulimit 설명서 페이지에 설명한 대로 파일 설명자 제한을 늘립니다.

가능한 원인: 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 플랫폼의 경우 하드 코드된 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 확인하는 방법: 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 해결하는 방법:

- 클라이언트가 시도했던 연결을 잠시 후에 다시 시도하도록 프로그래밍합니다(이 문제가 본래 임시적이기 때문에 이렇게 하면 대개 제대로 작동됨).
- `imq.portmapper.backlog` 값을 늘립니다.
- 클라이언트가 너무 자주 연결을 닫은 다음 열고 있지는 않은지 확인합니다.

가능한 원인: 운영 체제가 동시 연결 수를 제한합니다.

Windows 운영 체제 사용권에서는 지원되는 동시 원격 연결 수를 제한합니다.

문제의 원인을 확인하는 방법: 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 해결하는 방법:

- 더 많은 연결을 허용하도록 Windows 사용권을 업그레이드합니다.
- 다중 브로커 클러스터를 설정하여 연결을 여러 브로커 인스턴스에 분산합니다.

가능한 원인: 사용자의 인증 또는 권한 부여가 실패합니다.

다음과 같은 이유로 인해 인증이 실패할 수 있습니다.

- 잘못된 비밀번호
- 사용자 저장소에 사용자 항목 없음
- 사용자에게 연결 서비스에 대한 액세스 권한이 없음

문제의 원인을 확인하는 방법: 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 사용자 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 정보 파일에 연결 서비스 액세스에 대한 제한이 있는지 확인합니다.

문제를 해결하는 방법:

- 잘못된 비밀번호가 사용된 경우 올바른 비밀번호를 제공합니다.
- 사용자 저장소에 사용자 항목이 없는 경우 사용자를 추가합니다(131 페이지 "사용자 저장소 채우기 및 관리" 참조).
- 사용자가 연결 서비스에 대한 액세스 권한이 없는 경우 해당 권한을 허용하도록 액세스 제어 등록 정보 파일을 편집합니다(138 페이지 "연결 서비스에 대한 액세스 제어" 참고).

연결 처리 속도가 너무 느림

증상:

- 메시지 처리량이 기대에 미치지 못합니다.
- 브로커에 대해 지원되는 연결 수가 213 페이지 "클라이언트가 연결을 설정할 수 없음"에 설명된 대로 제한되는 것이 아니라 메시지 입출력 속도에 의해 제한됩니다.

가능한 원인:

- 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.
- 연결 처리량이 느린 것처럼 보이는 것은 실제로는 메시지 전달 프로세스의 어떤 단계에 병목 현상이 있는 것입니다.

가능한 원인: 네트워크 연결 또는 WAN이 너무 느립니다.

문제의 원인을 확인하는 방법:

- 네트워크를 핑하여 핑이 반환되는 데 걸리는 시간을 확인한 다음 네트워크 관리자에게 문의합니다.
- 로컬 클라이언트를 사용하여 메시지를 보내고 받은 다음 이 전달 시간을 네트워크 링크를 사용하는 원격 클라이언트의 전달 시간과 비교할 수 있습니다.

문제를 해결하는 방법: 네트워크 링크를 업그레이드합니다.

가능한 원인: 연결 서비스 프로토콜이 기본적으로 TCP에 비해 느립니다.

예를 들어 SSL 기반 또는 HTTP 기반 프로토콜이 TCP보다 느립니다(203 페이지 "전송 프로토콜" 참조).

문제의 원인을 확인하는 방법: SSL 기반 또는 HTTP 기반 프로토콜을 사용하는 경우 TCP를 사용해보고 전달 시간을 비교합니다.

문제를 해결하는 방법: 일반적으로 응용 프로그램 요구 사항에 따라 사용되는 프로토콜이 지정되므로 207 페이지 "전송 프로토콜 조정"에 설명된 대로 프로토콜을 조정해보는 것 이외에 할 수 있는 작업이 거의 없습니다.

가능한 원인: 연결 서비스 프로토콜이 최적으로 조정되어 있지 않습니다.

문제의 원인을 확인하는 방법: 프로토콜을 조정하여 어떻게 달라지는지 확인합니다.

문제를 해결하는 방법: 207 페이지 "전송 프로토콜 조정"에 설명된 대로 프로토콜을 조정합니다.

가능한 원인: 메시지가 너무 커서 너무 많은 대역폭을 사용합니다.

문제의 원인을 확인하는 방법: 작은 크기의 메시지를 사용하여 벤치마크를 실행해 봅니다.

문제를 해결하는 방법:

- 응용 프로그램 개발자에게 응용 프로그램을 수정하여 메시지 압축 기능을 사용하게 합니다. 이 방법에 대해서는 **Java 클라이언트용 Message Queue 개발 안내서**에 설명되어 있습니다.
- 메시지를 보낼 데이터의 알림으로 사용하고 데이터는 다른 프로토콜을 사용하여 이동합니다.

가능한 원인: 연결 처리량이 느린 것처럼 보이는 것은 실제로는 메시지 전달 프로세스의 어떤 단계에 병목 현상이 있는 것입니다.

문제의 원인을 확인하는 방법: 위의 원인 중 어떤 것으로도 느린 연결 처리량에 대해 설명할 수 없는 경우 196 페이지 "성능에 영향을 미치는 요소"를 참조하여 다른 병목 현상이 있을 수 있는지 확인하고 다음 문제와 관련된 증상이 있는지 확인합니다.

- 219 페이지 "메시지 생성이 지연되거나 느림"
- 222 페이지 "메시지가 백로그됨"
- 225 페이지 "브로커 처리량이 일정하지 않음"

문제를 해결하는 방법: 위에 나열된 문제 해결 절에 제공되어 있는 문제 해결 지침을 따릅니다.

클라이언트가 메시지 생성자를 만들 수 없음

증상:

- 물리적 대상에 대해 메시지 생성자를 만들 수 없기 때문에 클라이언트에서 예외가 발생합니다.

가능한 원인:

- 물리적 대상이 제한된 생성자 수만 허용하도록 구성되었습니다.
- 액세스 제어 등록 정보 파일의 설정으로 인해 사용자가 메시지 생성자를 만들 수 있는 권한이 없습니다.

가능한 원인: 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 확인하는 방법: 물리적 대상을 확인합니다.

```
imqcmd query dst
```

(117 페이지 "물리적 대상 정보 표시" 참조). 출력에 현재 생성자 수와 maxNumProducers 값이 표시됩니다. 두 값이 같은 경우 생성자의 수가 구성된 제한에 도달한 것입니다. 브로커가 새 생성자를 거부할 때 다음과 같은 예외를 반환하고

```
ResourceAllocationException [C4088]: A JMS destination limit was reached
```

브로커 로그에 다음 항목을 생성합니다.

```
[B4183]: Producer can not be added to destination
```

문제를 해결하는 방법: maxNumProducers 속성 값을 늘립니다(118 페이지 "물리적 대상 등록 정보 업데이트" 참조).

가능한 원인: 액세스 제어 등록 정보 파일의 설정으로 인해 사용자가 메시지 생성자를 만들 수 있는 권한이 없습니다.

문제의 원인을 확인하는 방법: 브로커가 새 생성자를 거부할 때 다음과 같은 예외를 반환하고

```
JMSSecurityException [C4076]: Client does not have permission to create producer
on destination
```

브로커 로그에 다음 항목을 생성합니다.

```
[B2041]: Producer on destination denied[B4051]: Forbidden guest.
```

문제를 해결하는 방법: 사용자가 메시지를 생성할 수 있도록 액세스 제어 등록 정보를 변경합니다(139 페이지 "물리적 대상에 대한 액세스 제어" 참조).

메시지 생성이 지연되거나 느림

증상:

- 지속성 메시지를 보낼 때 send 메소드가 반환되지 않고 클라이언트가 차단됩니다.
- 지속성 메시지를 보낼 때 클라이언트에 예외가 발생합니다.
- 생성자 클라이언트가 느려집니다.

가능한 원인:

- 브로커가 백로그되고 느린 메시지 생성자로 응답했습니다.
- 브로커가 지속성 메시지를 데이터 저장소에 저장할 수 없습니다.
- 브로커 확인 시간 제한이 너무 짧습니다.
- 생성자 클라이언트에서 JVM 제한이 발생했습니다.

가능한 원인: 브로커가 백로그되고 느린 메시지 생성자로 응답했습니다.

백로그된 브로커가 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 과부하를 막기 위해 점점 더 중대한 조치를 취합니다. 이러한 조치로는 메시지 생성자 억제가 있습니다.

문제의 원인을 확인하는 방법: 구성된 메시지 제한으로 인해 브로커가 메시지를 거부할 경우 브로커는 다음과 같은 예외를 반환합니다.

```
JMSException [C4036]: A server error occurred
```

브로커 로그에 다음 항목을 생성합니다.

```
[B2011]: Storing of JMS message from IMQconn failed
```

이 메시지 다음에는 제한에 도달했음을 나타내는 메시지가 표시됩니다.

```
[B4120]: Cannot store message on destination destName because capacity of
maxNumMsgs would be exceeded.
```

이는 초과된 메시지 제한이 물리적 대상에 있거나

```
[B4024]: The maximum number of messages currrently in the system has been
exceeded, rejecting message.
```

제한이 브로커 전체에 해당되는 경우에 표시됩니다.

좀 더 일반적으로 다음과 같은 방법으로 거부가 발생하기 전에 메시지 제한 조건을 확인할 수 있습니다.

- 물리적 대상과 브로커를 쿼리하여 각각에 구성된 메시지 제한 설정을 검사합니다.
- 적절한 `imqcmd` 명령을 사용하여 물리적 대상이나 브로커 전체에서 현재 메시지 수나 메시지 바이트 수를 모니터합니다. 모니터할 수 있는 메트릭 및 해당 메트릭을 가져오는 데 사용하는 명령에 대한 자세한 내용은 18 장을 참조하십시오.

문제를 해결하는 방법:

- 메모리 자원을 초과하지 않도록 주의하면서 물리적 대상 또는 브로커 전체에 대한 메시지 제한을 수정합니다.

  일반적으로는 브로커 전체 메시지 제한에 도달하는 일이 없도록 개별 대상 단위로 메모리를 관리해야 합니다. 자세한 내용은 209 페이지 “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(표 15–1 참조).

가능한 원인: 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 확인하는 방법: 브로커가 데이터 저장소에 기록할 수 없는 경우 브로커는 브로커 로그에 다음 항목 중 하나를 만듭니다.

`[B2011]: Storing of JMS message from` *connectionID* `failed [B4004]: Failed to persist message` *messageID*

문제를 해결하는 방법:

- 파일 기반 지속성의 경우 파일 기반 데이터 저장소의 디스크 공간을 늘립니다.
- JDBC 호환 데이터 저장소의 경우 JDBC 기반 지속성이 제대로 구성되어 있는지 확인합니다(91 페이지 “영구 데이터 저장소 구성” 참조). 제대로 구성되어 있는 경우 데이터베이스 관리자에게 문의하여 다른 데이터베이스 문제를 해결합니다.

가능한 원인: 브로커 확인 시간 제한이 너무 짧습니다.

느린 연결 또는 CPU 사용률이 높거나 메모리 자원이 부족하여 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 확인하는 방법: imqAckTimeout 값이 초과되면 브로커가 예외를 반환합니다.

`JMSException [C4000]: Packet acknowledge failed`

문제를 해결하는 방법: imqAckTimeout 연결 팩토리 속성 값을 변경합니다(160 페이지 “안정성 및 흐름 제어” 참조).

가능한 원인: 생성자 클라이언트에서 JVM 제한이 발생했습니다.

문제의 원인을 확인하는 방법:

- 클라이언트 응용 프로그램이 메모리 부족 오류를 수신하는지 확인합니다.
- freeMemory, maxMemory 및 totalMemory와 같은 **런타임 메서드를** 사용하여 JVM **힙에서** 사용 가능한 **메모리를** 확인합니다.

문제를 해결하는 방법: JVM을 조정합니다(206 페이지 “Java 가상 머신 조정” 참조).

메시지가 백로그됨

증상:

- 메시지 생성이 지연되거나 생성된 메시지가 브로커에서 거부됩니다.
- 메시지가 사용자에게 도달하기까지 비정상적으로 오래 걸립니다.
- 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 다음 항목을 확인합니다.

```
[B2011]: Storing of JMS message from ... failed.
```

이 항목 다음에는 초과된 제한에 대해 설명하는 다른 항목이 표시됩니다.

가능한 원인:

- 주제 대상에 비활성 영구 가입이 있습니다.
- 대기열에서 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 메시지 생성 속도를 따라갈 수 없습니다.
- 클라이언트 코드에 결함이 있습니다. 사용자가 메시지를 확인하지 않습니다.

가능한 원인: 주제 대상에 비활성 영구 가입이 있습니다.

영구 가입이 비활성 상태인 경우 해당 사용자가 활성화되어 메시지를 사용할 수 있을 때까지 대상에 메시지가 저장됩니다.

문제의 원인을 확인하는 방법: 각 주제 대상에서 영구 가입 상태를 확인합니다.

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

문제를 해결하는 방법:

- 문제가 있는 영구 가입의 모든 메시지를 제거합니다(108 페이지 “영구 가입 관리” 참조).

- 주제에 대해 메시지 제한 및 제한 동작 속성을 지정합니다(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(120 페이지 “물리적 대상 제거” 참조).
- 생성자 클라이언트를 다시 작성하여 각 메시지의 수명 값을 설정함으로써 메시지가 메모리에 남아 있을 수 있는 시간을 제한합니다. imqOverrideJMSExpiration 및 imqJMSExpiration 연결 팩토리 속성을 설정하여 연결을 공유하는 모든 생성자에 대해 이러한 설정을 대체할 수 있습니다(294 페이지 “메시지 헤더 대체” 참조).

가능한 원인: 대기열에서 메시지를 사용할 수 있는 사용자 수가 너무 적습니다.

메시지를 전달할 수 있는 활성 사용자의 수가 너무 적은 경우 메시지가 누적될 때 대기열 대상이 백로그될 수 있습니다. 이런 상태는 다음과 같은 이유 중 하나 때문에 발생할 수 있습니다.

- 대상에 활성 사용자의 수가 너무 적습니다.
- 사용자 클라이언트가 연결을 설정하지 못했습니다.
- 대기열의 메시지에 일치하는 선택기를 사용하는 활성 사용자가 없습니다.

문제의 원인을 확인하는 방법: 사용자가 대기열의 메시지를 사용할 수 없는 원인을 확인하려면 대상에서 활성 사용자의 수를 확인합니다.

```
imqcmd metrics dst - n destName -t q -m con
```

문제를 해결하는 방법: 사용자가 사용할 수 없는 원인에 따라

- 추가 사용자 클라이언트를 시작하여 대기열에 대한 활성 사용자를 더 많이 만듭니다.
- imq.consumerFlowLimit 브로커 등록 정보를 조정하여 여러 사용자에 대한 대기열 전달을 최적화합니다(210 페이지 “다중 사용자 대기열 성능” 참조).
- 대기열에 대해 메시지 제한 및 제한 동작 속성을 지정합니다(표 15–1 참조). 예를 들어, 메모리에 누적되는 메시지를 삭제하는 REMOVE_OLDEST 및 REMOVE_LOW_PRIOROTY 제한 동작을 지정할 수 있습니다.
- 해당 대상에서 모든 메시지를 제거합니다(120 페이지 “물리적 대상 제거” 참조).
- 생성자 클라이언트를 다시 작성하여 각 메시지의 수명 값을 설정함으로써 메시지가 메모리에 남아 있을 수 있는 시간을 제한합니다. imqOverrideJMSExpiration 및 imqJMSExpiration 연결 팩토리 속성을 설정하여 연결을 공유하는 모든 생성자에 대해 이러한 설정을 대체할 수 있습니다(294 페이지 “메시지 헤더 대체” 참조).

가능한 원인: 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 확인하는 방법:메지지가 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 해결하는 방법:

- 사용자 클라이언트 코드를 최적화합니다.
- 대기열 대상의 경우 활성 사용자의 수를 늘립니다(210 페이지 "다중 사용자 대기열 성능" 참조).

가능한 원인: 클라이언트 확인 처리가 메시지 사용을 느리게 합니다.

다음과 같은 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 확인하는 방법:

- 패킷 흐름을 기준으로 메시지 흐름을 확인합니다. 초당 패킷 수와 메시지 수의 비율이 맞지 않으면 클라이언트 확인에 문제가 있을 수 있습니다.
- 클라이언트가 다음과 같은 예외를 받았는지 확인합니다:

```
JMSException [C4000]: Packet acknowledge failed
```

문제를 해결하는 방법:

- 클라이언트가 사용하는 확인 모드를 수정합니다. 예를 들어, `DUPS_OK_ACKNOWLEDGE` 또는 `CLIENT_ACKNOWLEDGE`로 전환합니다.
- `CLIENT_ACKNOWLEDGE` 또는 트랜잭션된 세션을 사용 중인 경우 많은 수의 메시지를 하나의 확인으로 그룹화합니다.
- 사용자 및 연결 흐름 제어 매개 변수를 조정합니다(210 페이지 "클라이언트 런타임 메시지 흐름 조정" 참조).

가능한 원인: 브로커가 메시지 생성 속도를 따라갈 수 없습니다.

이 경우 브로커가 메시지를 라우팅하여 사용자에게 전달하는 것보다 더 빨리 메시지가 브로커로 유입됩니다. 브로커의 지체는 다음 중 하나 또는 모두에 대한 제한 때문일 수 있습니다.

- CPU
- 네트워크 소켓 읽기/쓰기 작업
- 디스크 읽기/쓰기 작업
- 메모리 페이징
- 영구 저장소
- JVM 메모리 제한

문제의 원인을 확인하는 방법: 이 문제를 일으키는 다른 원인은 없는지 확인합니다.

문제를 해결하는 방법:

- 컴퓨터 또는 데이터 저장소의 속도를 업그레이드합니다.
- 브로커 클러스터를 사용하여 로드를 여러 브로커 인스턴스에 분산합니다.

가능한 원인: 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` 확인 모드 또는 트랜잭션된 세션을 사용하는 사용자가 `Session.acknowledge` 또는 `Session.commit`을 정기적으로 호출하고 있지 않을 수 있습니다.
- `AUTO_ACKNOWLEDGE` 확인 모드를 사용하는 사용자가 어떤 이유로 중지되어 있을 수 있습니다.

문제의 원인을 확인하는 방법: 먼저 이 절에 나열된 모든 다른 가능한 원인을 확인하고다음 명령을 사용하여 대상을 나열합니다.

```
imqcmd list dst
```

`UnAcked` 헤더에 나열된 메시지 수가 대상의 메시지 수와 동일한지를 확인합니다. 이 헤더에 있는 메시지가 사용자에게 전송되었지만 확인되지 않았습니다. 이 수가 총 메시지 수와 같은 경우 브로커가 모든 메시지를 전송하고 확인을 대기 중인 것입니다.

문제를 해결하는 방법: 이 문제를 디버깅할 때 응용 프로그램 개발자에게 도움을 요청하십시오.

브로커 처리량이 일정하지 않음

증상:

- 메시지 처리량이 산발적으로 떨어지다가 정상 성능을 회복합니다.

가능한 원인:

- 브로커의 메모리 자원이 매우 적습니다.
- JVM 메모리 재생 이용(가비지 컬렉션)이 발생합니다.
- JVM이 성능을 높이기 위해 JIT(Just-In-Time) 컴파일러를 사용하고 있습니다.

가능한 원인: 브로커의 메모리 자원이 매우 적습니다.

대상 및 브로커 제한이 제대로 설정되어 있지 않기 때문에 브로커가 메모리 과부하를 막기 위해 점점 더 중대한 조치를 취하게 되어 메시지 백로그를 지울 때까지 브로커가 아주 느려질 수 있습니다.

문제의 원인을 확인하는 방법: 브로커 로그에서 부족한 메모리 상태에 대한 항목 및

```
[B1089]: In low memory condition, broker is attempting to free up resources
```

새 메모리 상태와 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 해결하는 방법:

- JVM을 조정합니다(206 페이지 "Java 가상 머신 조정" 참조).
- 시스템 스왑 공간을 늘립니다.

가능한 원인: 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 확인하는 방법: 컴퓨터의 CPU 사용을 모니터합니다. 메모리 재생 이용이 발생하면 CPU 사용량이 떨어집니다.

또한 다음 명령줄 옵션을 사용하여 브로커를 시작합니다.

```
- vmargs -verbose:gc
```

표준 출력에 메모리 재생 이용이 발생하는 시간이 표시됩니다.

문제를 해결하는 방법: 다중 CPU 컴퓨터에서는 메모리 재생 이용이 병렬로 발생되도록 설정합니다.

```
-XX:+UseParallelGC=true
```

가능한 원인: JVM이 성능을 높이기 위해 JIT(Just-In-Time) 컴파일러를 사용하고 있습니다.

문제의 원인을 확인하는 방법: 이 문제를 일으키는 다른 원인은 없는지 확인합니다.

문제를 해결하는 방법: 잠시 동안 시스템이 실행되도록 놓아두면 성능이 향상됩니다.

메시지가 사용자에게 도달하지 않음

증상:

- 생성자가 보낸 메시지를 사용자가 받지 못합니다.

가능한 원인:

- 제한 동작으로 인해 메시지가 브로커에서 삭제됩니다.
- 메시지 시간 초과 값이 만료됩니다.
- 클럭이 동기화되지 않습니다.
- 사용자 클라이언트가 연결에서 메시지 전달을 시작하지 못했습니다.

가능한 원인: 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 삭제합니다.
- REMOVE_LOW_PRIORITY: 메시지의 보존 기간을 기준으로 우선 순위가 가장 낮은 메시지를 삭제합니다.
- REJECT_NEWEST: 새 지속성 메시지를 거부합니다.

문제의 원인을 확인하는 방법: 230 페이지 “사용 불능 메시지 대기열에 메시지가 포함되어 있음”에서 설명한 대로 사용 불능 메시지 대기열을 확인합니다. 특히, "메시지 수 또는 메시지 크기가 대상 제한을 초과합니다."의 지침을 사용합니다.REMOVE_OLDEST 또는 REMOVE_LOW_PRIORITY 원인을 찾습니다.

문제를 해결하는 방법: 대상 제한을 늘립니다. 예를 들면 다음과 같습니다.

```
imqcmd update dst -n MyDest - o maxNumMsgs=1000
```

가능한 원인: 메시지 시간 초과 값이 만료됩니다.

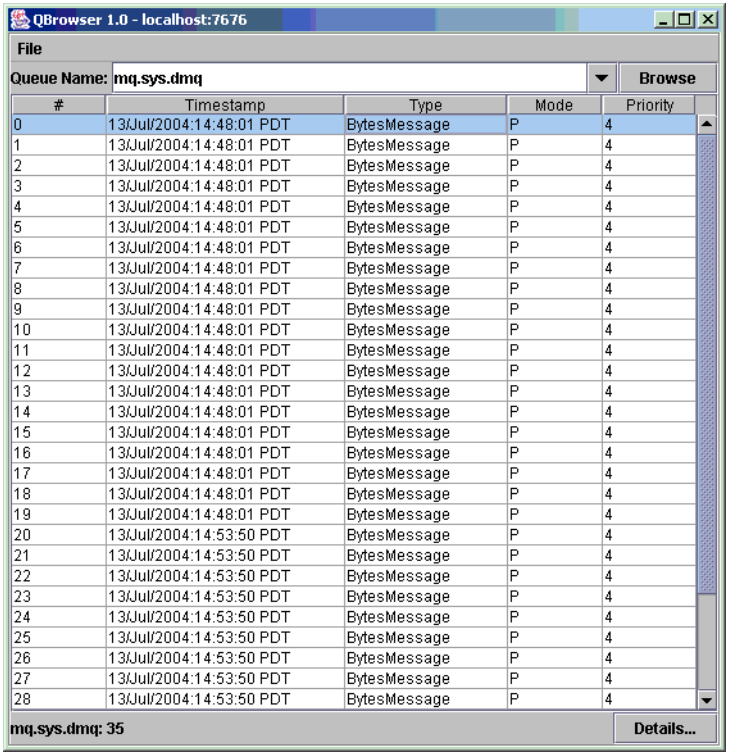
브로커가 시간 초과 값이 만료된 메시지를 삭제합니다. 대상이 충분히 메시지로 백로그된 경우 수명 값이 너무 짧은 메시지는 삭제될 수 있습니다.

문제의 원인을 확인하는 방법:QBrowser 데모 응용 프로그램을 사용하여 사용 불능 메시지 대기열의 내용을 살펴보고 메시지 시간이 초과되었는지 확인합니다. QBrowser 데모의 플랫폼별 위치는 부록 A를 참조하십시오. 또한 "예제 응용 프로그램 및 위치" 표를 참조하십시오.

다음은 Windows 플랫폼에서의 호출 예입니다.

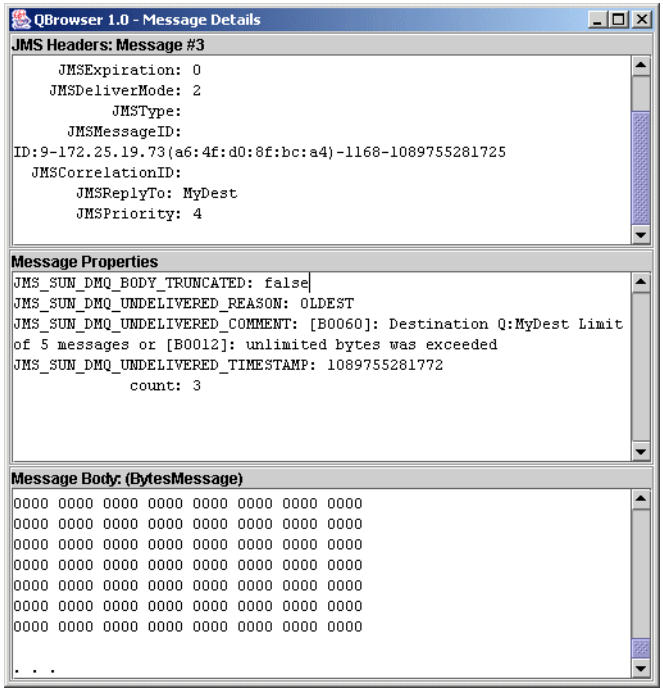
```
cd \MessageQueue3\demo\applications\qbrowser java QBrowser
```

QBrowser 기본 창이 나타나면 대기열 이름 mq.sys.dmq를 선택한 다음 Browse를 누릅니다. 다음과 같은 목록이 나타납니다.



#	Timestamp	Type	Mode	Priority
0	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
1	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
2	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
3	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
4	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
5	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
6	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
7	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
8	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
9	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
10	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
11	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
12	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
13	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
14	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
15	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
16	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
17	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
18	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
19	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4
20	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4
21	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4
22	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4
23	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4
24	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4
25	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4
26	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4
27	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4
28	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4

메시지를 두 번 누르면 해당 메시지에 대한 세부 정보가 표시됩니다.



메시지에 대한 JMS_SUN_DMQ_UNDELIVERED_REASON 등록 정보 값이 EXPIRED인지 확인합니다.

문제를 해결하는 방법: 응용 프로그램 개발자에게 문의하여 수명 값을 늘립니다.

가능한 원인: 클럭이 동기화되지 않습니다.

클럭이 동기화되어 있지 않은 경우 브로커의 메시지 수명 계산 오류로 인해 메시지가 만료 시간을 초과하여 삭제될 수 있습니다.

문제의 원인을 확인하는 방법: 브로커 로그 파일에서 B2102, B2103, B2104 메시지 중 하나가 있는지 확인합니다. 이러한 모든 메시지는 가능한 클럭 스큐가 감지되었음을 보고합니다.

문제를 해결하는 방법: 67 페이지 "시스템 자원 준비"에 설명된 대로 시간 동기화 프로그램이 실행 중인지 확인합니다.

가능한 원인: 사용자 클라이언트가 연결에서 메시지 전달을 시작하지 못했습니다.

클라이언트 코드가 연결을 설정하고 해당 연결에서 메시지 전달을 시작할 때까지 메시지를 전달할 수 없습니다.

문제의 원인을 확인하는 방법: 클라이언트 코드가 연결을 설정하고 메시지 전달을 시작하는지 확인합니다.

문제를 해결하는 방법: 연결을 설정하고 메시지 전달을 시작하도록 클라이언트 코드를 다시 작성합니다.

사용 불능 메시지 대기열에 메시지가 포함되어 있음

증상:

- 대상을 나열하면 사용 불능 메시지 대기열에 메시지가 포함되어 있습니다. 예를 들어, 다음과 같은 명령을 실행합니다.

  ```
  imqcmd list dst
  ```

  사용자 이름과 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.dmq  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.

가능한 원인:

- 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.
- 사용자가 메시지를 확인할 수 없습니다.
- 영구 사용자가 비활성 상태입니다.
- 예기치 않은 브로커 오류가 발생했습니다.

가능한 원인: 메시지 수 또는 메시지 크기가 대상 제한을 초과합니다.

문제의 원인을 확인하는 방법: QBrowser 데모 응용 프로그램을 사용하여 사용 불능 메시지 대기열의 내용을 살펴봅니다. QBrowser 데모의 플랫폼별 위치는 부록 A를 참조하십시오. 또한 "예제 응용 프로그램 및 위치" 표를 참조하십시오.

다음은 Windows 플랫폼에서의 호출 예입니다.

```
cd \MessageQueue3\demo\applications\qbrowser java QBrowser
```

QBrowser 기본 창이 나타나면 대기열 이름 `mq.sys.dmq`를 선택한 다음 Browse를 누릅니다. "메시지 시간 초과 값이 만료됩니다."에서 이전에 표시된 것과 같은 목록이

나타납니다. "메시지 시간 초과 값이 만료됩니다."에 표시된 것처럼 아무 메시지나 두 번 누르면 해당 메시지에 대한 세부 정보가 표시됩니다

다음 메시지 등록 정보 값을 확인합니다.

- JMS_SUN_DMQ_UNDELIVERED_REASON
- JMS_SUN_DMQ_UNDELIVERED_COMMENT
- JMS_SUN_DMQ_UNDELIVERED_TIMESTAMP

JMS Headers에서 JMSDestination 값을 검토하여 메시지가 사용 불능 상태인 대상을 확인합니다.

문제를 해결하는 방법: 대상 제한을 늘립니다. 예를 들면 다음과 같습니다.

```
imqcmd update dst - n MyDest -o maxNumMsgs=1000
```

가능한 원인: 브로커 클럭과 생성자 클럭이 동기화되어 있지 않습니다.

문제의 원인을 확인하는 방법: 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 파일에서 B2102, B2103, B2104 메시지 중 하나가 있는지 확인합니다. 이러한 모든 메시지는 가능한 클럭 스큐가 감지되었음을 보고합니다.

문제를 해결하는 방법: 67 페이지 "시스템 자원 준비"에 설명된 대로 시간 동기화 프로그램이 실행 중인지 확인합니다.

가능한 원인: 메시지 시간이 초과되기 전에 사용자가 메시지를 받지 못합니다.

문제의 원인을 확인하는 방법: 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
```

Current Number of Active Consumers에 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 해결하는 방법: 응용 프로그램 개발자에게 메시지 수명 값을 늘리도록 요청합니다.

가능한 원인: 사용자 수에 비해 생성자가 너무 많습니다.

문제의 원인을 확인하는 방법: 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 해결하는 방법: 다음과 같은 명령을 사용하여 사용자 클라이언트를 추가하거나 사용자 제한 동작을 FLOW_CONTROL(사용 속도를 사용하여 생성 속도 제어)로 설정합니다.

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

가능한 원인: 생성자가 사용자보다 더 빠릅니다.

문제의 원인을 확인하는 방법: 사용자가 느리기 때문에 생성자가 느려지는지 확인하려면 다음과 같은 명령을 사용하여 대상 제한 동작을 FLOW_CONTROL(사용 속도를 사용하여 생성 속도 제어)로 설정합니다.

```
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: 브로커가 제거하는 초 당 메시지 수를 나타냅니다. 브로커는 모든 사용자가 메시지 수신을 확인하면 메시지를 제거하기 때문에 메트릭이 사용 속도에 영향을 줍니다.
- Msgs/sec In: 브로커가 생성자로부터 수신하는 초 당 메시지 수를 나타냅니다. 따라서 메트릭이 생성 속도에 영향을 미칩니다.

흐름 제어는 생성을 소비에 맞추므로 생성이 느려지거나 중단되었는지 확인합니다. 그럴 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 해결하는 방법: 생성 속도가 사용 속도보다 일관되게 더 빠를 경우 흐름 제어를 정기적으로 사용하여 시스템을 정렬된 상태로 유지하십시오. 또한 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.

가능한 원인: 사용자가 너무 느립니다.

문제의 원인을 확인하는 방법: "생성자가 사용자보다 더 빠릅니다."에서 설명한 대로 메트릭을 사용하여 생성 속도와 사용 속도를 확인합니다.

문제를 해결하는 방법:

- 다음과 같은 명령을 사용하여 대상의 제한 동작을 FLOW_CONTROL로 설정합니다.

  ```
  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 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 항목에 설명된 것처럼 대상 제한을 늘립니다.
- 메시지 수나 바이트 수 또는 둘 다를 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 기간에 메시지 손실을 허용할 수 있는지 여부를 고려합니다.

가능한 원인: 클라이언트가 메시지를 완결하지 못합니다.

문제의 원인을 확인하는 방법: 응용 프로그램 개발자에게 문의하여 응용 프로그램이 트랜잭션을 사용하는지 여부를 확인합니다. 그럴 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 해결하는 방법: 코딩 오류를 수정하려면 응용 프로그램 개발자에게 문의하십시오.

가능한 원인: 사용자가 메시지를 확인할 수 없습니다.

문제의 원인을 확인하는 방법: 응용 프로그램 개발자에게 문의하여 응용 프로그램이 시스템 기반 확인을 사용하는지 클라이언트 기반 확인을 사용하는지 확인합니다. 응용 프로그램이 시스템 기반 확인을 사용하는 경우 이 절을 건너뜁니다. 응용 프로그램이 클라이언트 기반 확인을 사용하는 경우(CLIENT_ACKNOWLEDGE) 다음과 같은 명령을 사용하여 먼저 클라이언트에 저장된 메시지 수를 줄입니다.

```
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 이름과 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 해결하는 방법:코딩 오류를 수정하려면 응용 프로그램 개발자에게 문의하십시오.

가능한 원인: 영구 사용자가 비활성 상태입니다.

문제의 원인을 확인하는 방법: 다음 명령 형식을 사용하여 주제의 영구 가입자를 확인합니다.

```
imqcmd list dur -d topicName
```

문제를 해결하는 방법:

- imqcmd purge dur 명령을 사용하여 영구 사용자를 제거합니다.
- 사용자 응용 프로그램을 다시 시작합니다.

가능한 원인: 예기치 않은 브로커 오류가 발생했습니다.

문제의 원인을 확인하는 방법: "Producers are faster than consumers"에서 설명한 대로 QBrowser를 사용하여 메시지를 확인합니다.JMS_SUN_DMQ_UNDELIVERED_REASON 값이 ERROR이면 브로커 오류가 발생한 것입니다.

문제를 해결하는 방법:

- 브로커 로그 파일을 검토하여 관련 오류를 확인합니다.
- Sun 기술 지원부에 연락하여 브로커 문제를 보고합니다.

3

참조

- 13 장
- 14 장
- 15 장
- 16 장
- 17 장
- 18 장

◆ ◆ ◆ 13장

13 명령줄 참조

이 장에서는 Message Queue™ 명령줄 관리 유틸리티 사용에 대한 참조 정보를 제공합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 237 페이지 "명령줄 구문"
- 238 페이지 "브로커 유틸리티"
- 242 페이지 "명령 유틸리티"
- 250 페이지 "객체 관리자 유틸리티"
- 251 페이지 "데이터베이스 관리자 유틸리티"
- 253 페이지 "사용자 관리자 유틸리티"
- 254 페이지 "서비스 관리자 유틸리티"
- 255 페이지 "키 도구 유틸리티"

명령줄 구문

Message Queue 명령줄 유틸리티는 쉘 명령입니다. 유틸리티 이름이 명령이고 하위 명령 또는 옵션은 해당 명령에 전달되는 인수입니다. 별도의 명령을 사용하여 유틸리티를 시작하거나 종료할 필요가 없습니다.

모든 명령줄 유틸리티는 다음과 같은 명령 구문을 공유합니다.

utilityName [*subcommand*] [*commandArgument*] [[*-optionName* [*optionArgument*]] ...]

여기서 *utilityName*은 다음 중 하나입니다.

- `imqbrokerd`(브로커 유틸리티)
- `imqcmd`(명령 유틸리티)
- `imqobjmgr`(객체 관리자 유틸리티)
- `imqdbmgr`(데이터베이스 관리자 유틸리티)
- `imqusermgr`(사용자 관리자 유틸리티)
- `imqsvcadmin`(서비스 관리자 유틸리티)
- `imqkeytool`(키 도구 유틸리티)

하위 명령 및 명령 수준의 인수(있는 경우)는 모든 옵션과 인수 앞에 와야 합니다. 옵션 자체는 임의 순서대로 나타날 수 있습니다. 모든 하위 명령, 명령 인수, 옵션 및 옵션 인수는 공백으로 구분됩니다. 옵션 인수 값에 공백이 포함되어 있는 경우 전체 값을 따옴표로 묶어야 합니다. 일반적으로 속성, 값 쌍을 따옴표로 묶는 것이 가장 안전합니다.

기본 브로커를 시작하는 다음 명령은 하위 명령절이 없는 명령줄의 예입니다.

```
imqbrokerd
```

다음은 전체 명령줄의 예입니다.

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

이 명령은 `myQueue`라는 대기열 대상(대상 유형 `q`)을 완전 삭제합니다. 인증은 사용자 이름 `admin`에 대해 수행됩니다. 비밀번호를 묻는 메시지가 표시됩니다. 이 명령은 확인 메시지를 표시하지 않고(`-f` 옵션) 자동 모드로 수행되며 어떠한 출력도 표시하지 않습니다(`-s` 옵션).

브로커 유틸리티

브로커 유틸리티(`imqbrokerd`)는 브로커를 시작합니다. 명령줄 옵션은 현재 브로커 세션에 대해서만 브로커 구성 파일의 값을 무시합니다.

표 13–1에서는 `imqbrokerd` 명령 옵션과 각 옵션으로 대체되는 구성 등록 정보(있는 경우)를 보여줍니다.

표 13–1 브로커 유틸리티 옵션

옵션	대체되는 등록 정보	설명
`-name` *instanceName*	`imq.instancename`	브로커의 인스턴스 이름 동일한 호스트에서 실행 중인 여러 브로커 인스턴스에는 서로 다른 인스턴스 이름이 있어야 합니다. **기본값:** `imqbroker`
`-port` *portNumber*	`imq.portmapper.port`	브로커의 포트 매퍼 포트 번호 Message Queue 클라이언트는 이 포트 번호를 사용하여 브로커에 연결합니다. 동일한 호스트에서 실행 중인 여러 브로커 인스턴스에는 서로 다른 포트 매퍼 포트 번호가 있어야 합니다. **기본값:** 7676

표 13–1 브로커 유틸리티 옵션 (계속)

옵션	대체되는 등록 정보	설명
`-cluster` *broker1* [[, *broker2*] …]	`imq.cluster.brokerlist`	브로커를 클러스터에 연결[1] 지정된 브로커는 `imq.cluster.brokerlist` 등록 정보의 목록과 병합됩니다. 각 브로커 인수는 다음 형태 중 하나입니다. *hostName*: *portNumber* *hostName* :*portNumber* *hostName*을 생략하는 경우에는 기본값이 `localhost`입니다. *portNumber*를 생략하는 경우에는 기본값이 `7676`입니다.
`-D`*property*`=` *value*	인스턴스 구성 파일의 해당 등록 정보	구성 등록 정보 설정 브로커 구성 등록 정보에 대한 자세한 내용은 14 장을 참조하십시오. **주의:** 이 옵션을 사용하여 설정한 등록 정보의 형식과 맞춤법에 주의하십시오. 잘못된 값은 알림이나 경고 없이 무시됩니다.
`-reset props`	없음	구성 등록 정보 재설정 브로커의 기존 인스턴스 구성 파일 `config.properties`를 빈 파일로 바꿉니다. 모든 등록 정보에 기본값이 설정됩니다.
`-reset store`	없음	영구 데이터 저장소 재설정 데이터 저장소의 모든 영구 데이터(지속성 메시지, 영구 가입 및 트랜잭션 정보 포함)를 삭제하여 빈 상태에서 브로커 인스턴스를 시작할 수 있도록 합니다. 이후에 다시 시작할 때 영구 저장소가 재설정되지 않도록 하려면 `-reset` 옵션을 지정하지 않고 브로커 인스턴스를 다시 시작합니다. 지속성 메시지 또는 영구 가입만 삭제하려면 `-reset messages` 또는 `-reset durables`를 대신 사용합니다.
`-reset messages`	없음	데이터 저장소에서 지속성 메시지 삭제
`-reset durables`	없음	데이터 저장소에서 영구 가입 삭제
`-backup` *fileName*	없음	구성 변경 레코드를 파일로 백업[1] 자세한 내용은 176 페이지 “구성 변경 기록 관리”를 참조하십시오.

[1] 브로커 클러스터에만 적용

표 13–1 브로커 유틸리티 옵션 *(계속)*

옵션	대체되는 등록 정보	설명
`-restore` *fileName*	없음	백업 파일에서 구성 변경 레코드 복원[1] `-backup` 옵션을 사용하여 백업 파일을 미리 작성해 두어야 합니다. 자세한 내용은 176 페이지 "구성 변경 기록 관리"를 참조하십시오.
`-remove instance`	없음	브로커 인스턴스 제거[2] 인스턴스 구성 파일, 로그 파일, 영구 저장소 및 인스턴스와 관련된 기타 파일 및 디렉토리를 삭제합니다.
`-password` *keyPassword*	`imq.keystore.password`	SSL 인증서 키 저장소의 비밀번호[3]
`-dbuser` *userName*	`imq.persist.jdbc.user`	JDBC 기반 영구 데이터 저장소의 사용자 이름
`-dbpassword` *dbPassword*	`imq.persist.jdbc.password`	JDBC 기반 영구 데이터 저장소의 비밀번호[3]
`-ldappassword` *ldapPassword*	`imq.user_repository.ldap.password`	LDAP 사용자 저장소의 비밀번호[3]
`-passfile` *filePath*	`imq.passfile.enabledimq.passfile.dirpathimq.passfile.name`	비밀번호 파일 위치 브로커의 `imq.passfile.enabled` 등록 정보를 `true`로 설정하고 `imq.passfile.dirpath`를 비밀번호 파일을 포함하는 경로로 설정하며 `imq.passfile.name`을 파일 이름 자체로 설정합니다. 자세한 내용은 150 페이지 "비밀번호 파일"을 참조하십시오.
`-shared`	`imq.jms.threadpool_model`	공유 스레드 풀 모델을 사용하여 `jms` 연결 서비스 구현 연결 간에 예외 스레드가 공유되므로 지원되는 연결 수가 증가합니다. 브로커의 `imq.jms.threadpool_model` 등록 정보를 `shared`로 설정합니다.
`-javahome` *path*	없음	대체 Java Runtime 위치 **기본 동작:** 시스템에 설치된 런타임 또는 Message Queue와 함께 번들로 제공된 런타임을 사용합니다.

[1] 브로커 클러스터에만 적용

[2] `-force` 옵션을 함께 지정한 경우가 아니면 사용자의 확인이 필요합니다.

[3] 이 옵션은 더 이상 사용되지 않으며 추후에 제거될 것입니다. 비밀번호를 생략하거나(대화식으로 비밀번호를 묻는 메시지 표시) `-passfile` 옵션을 사용하여 비밀번호를 포함하는 파일을 지정합니다.

표 13–1 브로커 유틸리티 옵션 *(계속)*

옵션	대체되는 등록 정보	설명
`-vmargs` *arg1* [[*arg2*] …]	없음	인수를 Java 가상 머신에 전달 인수는 공백으로 구분됩니다. 두 개 이상의 인수를 전달하거나 공백을 포함하는 인수를 전달하려면 인수 목록을 따옴표로 묶습니다. VM 인수는 명령줄을 통해서만 전달할 수 있으며, 인스턴스 구성 파일에는 관련 구성 등록 정보가 없습니다.
`-license` [*licenseName*]	없음	로드할 사용권이 설치된 Message Queue 제품의 기본값과 다를 경우: `pe`: 기본 기능 사용 플랫폼판 `try`: 엔터프라이즈 기능 사용 플랫폼판(90일 시험 사용) `unl`: 엔터프라이즈판 사용권 이름이 지정되지 않은 경우에는 시스템에 설치된 모든 사용권이 나열됩니다.
`-upgrade-store-nobackup`	없음	Message Queue 3.5 또는 3.5 SP*x*로 업그레이드할 때 호환되지 않는 버전의 이전 데이터 저장소를 자동으로 제거[2] 자세한 내용은 **Message Queue 설치 설명서**를 참조하십시오.
`-force`	없음	사용자 확인 없이 작업 수행 이 옵션은 일반적으로 확인을 필요로 하는 `-remove instance` 및 `-upgrade-store-nobackup` 옵션에만 적용됩니다.

`-loglevel` *level*	`imq.broker.log.level`	로깅 수준:
		`NONE`
		`ERROR`
		`WARNING`
		`INFO`
		
		기본값: `INFO`

`-metrics` *interval*	`imq.metrics.interval`	브로커 메트릭 로깅 간격(초)

[2] `-force` 옵션을 함께 지정한 경우가 아니면 사용자의 확인이 필요합니다.

표 13–1 브로커 유틸리티 옵션 (계속)

옵션	대체되는 등록 정보	설명
-tty	imq.log.console.output	모든 메시지를 콘솔에 기록 브로커의 imq.log.console.output 등록 정보를 ALL로 설정합니다. 지정하지 않을 경우 오류 및 경고 메시지만 기록됩니다.
-s | -silent	imq.log.console.output	자동 모드(콘솔에 기록하지 않음) 브로커의 imq.log.console.output 등록 정보를 NONE으로 설정합니다.
-version	없음	버전 정보 표시[4]
-h | -help	없음	사용 도움말 표시[4]

[4] 명령줄에 지정된 다른 옵션은 무시됩니다.

명령 유틸리티

명령 유틸리티(imqcmd)는 브로커, 연결 서비스, 연결, 물리적 대상, 영구 가입 및 트랜잭션을 관리하는 데 사용됩니다.

모든 imqcmd 명령은 하위 명령을 포함해야 합니다. 단, -v 또는 -h 옵션을 사용하여 제품 버전 정보 또는 사용 도움말을 표시하는 경우는 제외됩니다. 다음은 사용 가능한 하위 명령 목록이며 각각의 하위 명령에 대해서는 다음 절에서 자세히 설명합니다. 모든 경우에 하위 명령에서 브로커 주소(-b 옵션)를 받으며 호스트 이름 또는 포트 번호가 지정되지 않으면 값 localhost 및 7676을 기본값으로 간주합니다.

244 페이지 “브로커 관리”

shutdown bkr	브로커 종료
restart bkr	브로커 다시 작동
pause bkr	브로커 일시 중지
resume bkr	브로커 다시 시작
update bkr	브로커 등록 정보 설정
reload cls	클러스터 구성 다시 로드
query bkr	브로커 등록 정보 값 나열
metrics bkr	브로커 메트릭 표시

페이지 "연결 서비스 관리"	
pause svc	연결 서비스 일시 중지
resume svc	연결 서비스 다시 시작
update svc	연결 서비스 등록 정보 설정
list svc	브로커에서 사용할 수 있는 연결 서비스 나열
query svc	연결 서비스 등록 정보 값 나열
metrics svc	연결 서비스 메트릭 표시
246 페이지 "연결 관리"	
list cxn	브로커 연결 나열
query cxn	연결 정보 표시
246 페이지 "물리적 대상 관리"	
create dst	물리적 대상 작성
destroy dst	물리적 대상 완전 삭제
pause dst	물리적 대상에 대한 메시지 전달 일시 중지
resume dst	물리적 대상에 대한 메시지 전달 다시 시작
update dst	물리적 대상 등록 정보 설정
purge dst	물리적 대상에서 모든 메시지 제거
compact dst	물리적 대상 압축
list dst	물리적 대상 나열
query dst	물리적 대상 등록 정보 값 나열
metrics dst	물리적 대상 메트릭 표시
248 페이지 "영구 가입 관리"	
destroy dur	영구 가입 완전 삭제
purge dur	영구 가입에 대한 모든 메시지 제거
list dur	주제에대한 영구 가입 나열
249 페이지 "트랜잭션 관리"	
commit txn	트랜잭션 완결
rollback txn	트랜잭션 롤백
list txn	브로커에서 추적 중인 트랜잭션 나열

`query txn`	트랜잭션 정보 표시

브로커 관리

명령 유틸리티를 사용하여 브로커를 시작할 수 없습니다. 대신, 브로커 유틸리티(`imqbrokerd`)를 사용하십시오. 브로커가 시작된 후에는 표 13–2에 나열된 `imqcmd` 하위 명령을 사용하여 브로커를 관리 및 제어할 수 있습니다.

표 13–2 브로커 관리에 대한 명령 유틸리티 하위 명령

구문	설명
`shutdown bkr` [`-b` *hostName* `:`*portNumber*]	브로커 종료
`restart bkr` [`-b` *hostName* `:`*portNumber*]	브로커 다시 시작 브로커를 종료하고 처음 시작했을 때 지정했던 것과 동일한 옵션을 사용하여 다시 시작합니다.
`pause bkr` [`-b` *hostName* `:`*portNumber*]	브로커 일시 중지 자세한 내용은 100 페이지 “브로커 일시 중지”를 참조하십시오.
`resume bkr` [`-b` *hostName* `:`*portNumber*]	브로커 다시 시작
`update bkr` [`-b` *hostName* `:`*portNumber*] `-o` *property1*`=` *value1* [[`-o` *property2* `=`*value2*] …]	브로커 등록 정보 설정 브로커 등록 정보에 대한 자세한 내용은 14 장을 참조하십시오.
`reload cls`	클러스터 구성 다시 로드[1] 모든 지속성 정보를 최신 상태로 유지합니다.
`query bkr -b` *hostName* `:`*portNumber*	브로커 등록 정보 값 나열 또한 클러스터 내 지정된 브로커에 연결된 모든 실행 중인 브로커를 나열합니다.

[1] 브로커 클러스터에만 적용

표 13–2 브로커 관리에 대한 명령 유틸리티 하위 명령 *(계속)*

구문	설명
`metrics bkr` [`-b` *hostName* :*portNumber*] [`-m` *metricType*] [`-int` *interval*] [`-msp` *numSamples*]	브로커 메트릭 표시 `-m` 옵션은 표시할 메트릭 유형을 지정합니다. `ttl`: 브로커에 유입 및 유출되는 메시지와 패킷 `rts`: 브로커에 유입 및 유출되는 메시지와 패킷의 초당 속도 `cxn`: 연결, 가상 메모리 힙 및 스레드 **기본값:** `ttl` `-int` 옵션은 메트릭 표시 간격(초)을 지정합니다. **기본값:** `5` `-msp` 옵션은 표시할 샘플 수를 지정합니다. **기본값:** 무제한(지정되지 않음)

연결 서비스 관리

표 13–3에서는 연결 서비스를 관리하기 위한 `imqcmd` 하위 명령을 나열합니다.

표 13–3 연결 서비스 관리에 대한 명령 유틸리티 하위 명령

구문	설명
`pause svc -n` *serviceName* [`-b` *hostName*: *portNumber*]	연결 서비스 일시 중지 `admin` 연결 서비스는 일시 중지할 수 없습니다.
`resume svc -n` *serviceName* [`-b` *hostName*: *portNumber*]	연결 서비스 다시 시작
`update svc -n` *serviceName* [`-b` *hostName*: *portNumber*] `-o` *property1*= *value1* [[`-o` *property2* =*value2*] …]	연결 서비스 등록 정보 설정 연결 서비스 등록 정보에 대한 자세한 내용은 257 페이지 “연결 등록 정보”를 참조하십시오.
`list svc` [`-b` *hostName* :*portNumber*]	브로커에서 사용할 수 있는 연결 서비스 나열
`query svc -n` *serviceName* [`-b` *hostName*: *portNumber*]	연결 서비스 등록 정보 값 나열

표 13–3 연결 서비스 관리에 대한 명령 유틸리티 하위 명령 *(계속)*

구문	설명
`metrics svc -n` *serviceName* [`-b` *hostName*: *portNumber*] [`-m` *metricType*] [`-int` *interval*] [`-msp` *numSamples*]	연결 서비스 메트릭 표시 `-m` 옵션은 표시할 메트릭 유형을 지정합니다. `ttl`: 브로커에 유입 및 유출되는 메시지와 패킷의 메트릭을 지정된 연결 서비스를 통해 표시 `rts`: 브로커에 메시지와 패킷이 유입 및 유출되는 초당 속도를 지정된 연결 방식으로 표시 `cxn`: 연결, 가상 메모리 힙 및 스레드 **기본값:** `ttl` `-int` 옵션은 메트릭 표시 간격(초)을 지정합니다. **기본값:** `5` `-msp` 옵션은 표시할 샘플 수를 지정합니다. **기본값:** 무제한(지정되지 않음)

연결 관리

표 13–4에서는 연결을 관리하기 위한 `imqcmd` 하위 명령을 나열합니다.

표 13–4 연결 서비스 관리에 대한 명령 유틸리티 하위 명령

구문	설명
`list cxn` [`-svn` *serviceName*] [`-b` *hostName*: *portNumber*]	브로커에 대한 연결 나열 지정된 연결 서비스에 대한 브로커의 모든 연결을 나열합니다. 연결 서비스를 지정하지 않는 경우 모든 연결이 나열됩니다.
`query cxn -n` *connectionID* [`-b` *hostName*: *portNumber*]	연결 정보 표시

물리적 대상 관리

표 13–5에서는 물리적 대상을 관리하기 위한 `imqcmd` 하위 명령을 나열합니다. `-t`(대상 유형) 옵션은 언제나 두 가지 값 중 하나를 사용할 수 있습니다.

`q`: 대기열 대상
`t`: 주제 대상

표 13–5 물리적 대상 관리에 대한 명령 유틸리티 하위 명령

구문	설명
`create dst -t` *destType* `-n` *destName* `[-o` *property1* `=`*value1*`]` `[ [ -o` *property2* `=`*value2*`] … ]`	물리적 대상 작성[1] 대상 이름 *destName*에는 영숫자 문자(공백 없음)만 사용할 수 있으며 영문자, 밑줄 문자(_) 또는 달러 기호($) 문자로 시작해야 합니다. 대상 이름은 `mq`로 시작할 수 없습니다.
`destroy dst -t` *destType* `-n` *destName*	물리적 대상 완전 삭제[1] 이 작업은 사용 불능 메시지 대기열처럼 시스템에서 생성된 대상에는 적용할 수 없습니다.
`pause dst [-t` *destType* `-n` *destName*`]` `[-pst` *pauseType*`]`	물리적 대상에 대한 메시지 전달 일시 중지 `-t` 및 `-n` 옵션에서 지정한 물리적 대상에 대해 메시지 전달을 일시 중지합니다. 이 옵션을 지정하지 않으면 모든 대상이 일시 중지됩니다. `-pst` 옵션은 일시 중지할 메시지 전달의 유형을 지정합니다. `CONSUMERS`: 메시지 사용자로의 전달 일시 중지 `PRODUCERS`: 메시지 생성자로의 전달 일시 중지 `ALL`: 모든 메시지 전달 일시 중지 **기본값:** `ALL`
`resume dst [-t` *destType* `-n`*destName*`]`	물리적 대상에 대한 메시지 전달 다시 시작 `-t` 및 `-n` 옵션을 통해 지정한 물리적 대상에 대해 메시지 전달을 다시 시작합니다. 이 옵션을 지정하지 않을 경우 모든 대상이 다시 시작됩니다.
`update dst -t` *destType* `-n` *destName* `-o` *property1*`=` *value1* `[ [ -o` *property2* `=`*value2*`] … ]`	물리적 대상 등록 정보 설정 물리적 대상 등록 정보에 대한 자세한 내용은 15 장을 참조하십시오.
`purge dst -t` *destType* `-n` *destName*	물리적 대상에서 모든 메시지 제거
`compact dst [-t` *destType* `-n` *destName*`]`	물리적 대상 압축 `-t` 및 `-n` 옵션을 통해 지정하는 물리적 대상에 대한 파일 기반 영구 데이터 저장소를 압축합니다. 이 옵션을 지정하지 않을 경우 모든 대상이 압축됩니다. 대상을 압축하려면 먼저 대상을 일시 중지해야 합니다.

[1] 마스터 브로커를 일시적으로 사용할 수 없는 클러스터에서는 수행할 수 없습니다.

표 13–5 물리적 대상 관리에 대한 명령 유틸리티 하위 명령 *(계속)*

구문	설명
`list dst` [`-t` *destType*] [`-tmp`]	물리적 대상 나열 `-t` 옵션을 통해 지정된 유형의 모든 물리적 대상을 나열합니다. 대상 유형을 지정하지 않을 경우 대기열 및 주제 대상이 모두 나열됩니다. `-tmp` 옵션을 지정하는 경우 임시 대상도 나열됩니다.
`query dst -t` *destType* `-n` *destName*	물리적 대상 등록 정보 값 나열
`metrics dst -t` *destType* `-n` *destName* [`-m` *metricType*] [`-int` *interval*] [`-msp` *numSamples*]	물리적 대상 메트릭 표시 `-m` 옵션은 표시할 메트릭 유형을 지정합니다. `ttl`: 대상에 유입 및 유출되고 메모리에 있는 메시지와 패킷 `rts`: 브로커에 유입 및 유출되는 메시지와 패킷의 초당 속도(기타 속도 정보와 함께 표시) `con`: 메시지 사용자 관련 메트릭 `dsk`: 디스크 사용량 **기본값:** `ttl` `-int` 옵션은 메트릭 표시 간격(초)을 지정합니다. **기본값:** `5` `-msp` 옵션은 표시할 샘플 수를 지정합니다. **기본값:** 무제한(지정되지 않음)

영구 가입 관리

표 13–6에서는 영구 가입을 관리하기 위한 `imqcmd` 하위 명령을 나열합니다.

표 13–6 영구 가입 관리에 대한 명령 유틸리티 하위 명령

구문	설명
`destroy dur -c` *clientID* `-n` *subscriberName*	영구 가입 완전 삭제[1]
`purge dur -c` *clientID* `-n` *subscriberName*	영구 가입에 대한 모든 메시지 제거
`list dur -d` *topicName*	주제에 대한 영구 가입 나열

[1] 마스터 브로커를 일시적으로 사용할 수 없는 클러스터에서는 수행할 수 없습니다.

트랜잭션 관리

표 13–7에서는 트랜잭션을 관리하기 위한 imqcmd 하위 명령을 나열합니다.

표 13–7 트랜잭션 관리에 대한 명령 유틸리티 하위 명령

구문	설명
`commit txn -n` *transactionID*	트랜잭션 완결
`rollback txn -n` *transactionID*	트랜잭션 롤백
`list txn`	브로커에서 추적 중인 트랜잭션 나열
`query txn -n` *transactionID*	트랜잭션 정보 표시

일반 명령 유틸리티 옵션

표 13–8에 나열된 추가 옵션은 imqcmd 명령의 모든 하위 명령에 적용할 수 있습니다.

표 13–8 일반 명령 유틸리티 옵션

옵션	설명
`-secure`	ssladmin 연결 서비스를 사용하는 브로커에 보안 연결 사용
`-u` *userName*	인증에 사용되는 사용자 이름 이 옵션을 생략하면 명령 유틸리티에서 대화식으로 사용자 이름을 묻는 메시지를 표시합니다.
`-p` *password*	인증에 사용되는 비밀번호[1]
`-passfile` *path*	비밀번호 파일의 위치 자세한 내용은 150 페이지 “비밀번호 파일”을 참조하십시오.
`-rtm` *timeoutInterval*	초기 시간 초과 간격(초) 요청을 다시 시도하기 전에 명령 유틸리티가 브로커로부터 응답을 기다리는 초기 시간 길이입니다. 각 후속 재시도에서는 이 초기 간격의 배수가 되는 시간 초과 간격을 사용합니다. **기본값:** `10`
`-rtr` *numRetries*	브로커 요청 시간이 초과된 후 시도할 재시도 횟수 **기본값:** `5`

[1] 이 옵션은 더 이상 사용되지 않으며 추후에 제거될 것입니다. 비밀번호를 생략하거나(대화식으로 비밀번호를 묻는 메시지 표시) `-passfile` 옵션을 사용하여 비밀번호를 포함하는 파일을 지정합니다.

표 13–8 일반 명령 유틸리티 옵션 *(계속)*

옵션	설명
`-javahome` *path*	대체 Java Runtime 위치 **기본 동작:** 시스템에 설치된 런타임 또는 Message Queue와 함께 번들로 제공된 런타임을 사용합니다.
`-f`	사용자 확인 없이 작업 수행
`-s`	자동 모드(출력이 표시되지 않음)
`-v`	버전 정보 표시[2,3]
`-h`	사용 도움말 표시[2, 3]
`-H`	속성 목록과 예를 포함하여 자세한 사용 도움말을 표시합니다.[2,3]

[2] 명령줄에 지정된 다른 옵션은 무시됩니다.

[3] 사용자 이름과 비밀번호가 필요하지 않음

객체 관리자 유틸리티

객체 관리자 유틸리티(`imqobjmgr`)는 Message Queue 관리 대상 객체를 만들고 관리합니다. 표 13–9에서는 사용 가능한 하위 명령을 나열합니다.

표 13–9 객체 관리자 하위 명령

하위 명령	설명
`add`	객체 저장소에 관리 객체 추가
`delete`	객체 저장소에서 관리 객체 삭제
`list`	객체 저장소의 관리 객체 나열
`query`	관리 객체 정보 표시
`update`	관리 객체 수정

표 13–10에서는 `imqobjmgr` 명령에 대한 옵션을 나열합니다.

표 13–10 객체 관리자 옵션

옵션	설명
`-l` *lookupName*	관리 대상 객체의 JNDI 조회 이름
`-j` *attribute*= *value*	JNDI 객체 저장소 속성(153 페이지 "객체 저장소" 참조)

표 13–10 객체 관리자 옵션 *(계속)*

옵션	설명
`-t` *objectType*	관리 객체 유형: `q`: 대기열 대상 `t`: 주제 대상 `cf`: 연결 팩토리 `qf`: 대기열 연결 팩토리 `tf`: 주제 연결 팩토리 `xcf`: 분산 트랜잭션의 연결 팩토리 `xqf`: 배포 트랜잭션의 대기열 연결 팩토리 `xtf`: 분산 트랜잭션의 주제 연결 팩토리 `e`: SOAP 종점(**Java 클라이언트용 Message Queue 개발 안내서** 참조)
`-o` *attribute*`=` *value*	관리 대상 객체 속성(156 페이지 “관리 객체 속성” 및 16 장 참조)
`-r` *readOnlyState*	관리 객체가 읽기 전용입니까? 값이 `true`이면 클라이언트에서 객체 속성을 수정할 수 없습니다. **기본값:** `false`
`-i` *fileName*	하위 명령절 전체 또는 일부를 포함하는 명령 파일의 이름
`-pre`	명령을 수행하지 않고 결과 미리 보기 이 옵션은 기본 속성의 값을 확인할 때 유용합니다.
`-javahome` *path*	대체 Java Runtime 위치 **기본 동작:** 시스템에 설치된 런타임 또는 Message Queue와 함께 번들로 제공된 런타임을 사용합니다.
`-f`	사용자 확인 없이 작업 수행
`-s`	자동 모드(출력이 표시되지 않음)
`-v`	버전 정보 표시[1]
`-h`	사용 도움말 표시[1]
`-H`	속성 목록과 예를 포함하여 자세한 사용 도움말 표시[1]

[1] 명령줄에 지정된 다른 옵션은 무시됩니다.

데이터베이스 관리자 유틸리티

데이터베이스 관리자 유틸리티(`imqdbmgr`)는 JDBC 기반 영구 데이터 저장소의 데이터베이스 스키마를 설정합니다. 이 유틸리티를 사용하여 손상된 Message Queue 데이터베이스 테이블을 삭제하거나 데이터 저장소를 변경할 수 있습니다. 표 13–11에서는 사용 가능한 하위 명령을 나열합니다.

표 13–11 데이터베이스 관리자 하위 명령

하위 명령	설명
`create all`	새 데이터베이스와 영구 저장소 스키마 만들기 내장 데이터베이스 시스템에 사용됩니다. 브로커 등록 정보 `imq.persist.jdbc.createdburl`을 지정해야 합니다.
`create tbl`	기존 데이터베이스의 영구 저장소 스키마 만들기 외부 데이터베이스 시스템에 사용됩니다.
`delete tbl`	현재 영구 저장소에서 Message Queue 데이터베이스 테이블 삭제
`delete oldtbl`	이전 버전 영구 저장소에서 Message Queue 데이터베이스 테이블 삭제 영구 저장소가 현재 버전의 Message Queue로 자동 마이그레이션된 후에 사용됩니다.
`recreate tbl`	영구 저장소 스키마 다시 만들기 현재의 영구 저장소에서 모든 기존 Message Queue 데이터베이스 테이블을 삭제하고 스키마를 다시 만듭니다.
`reset lck`	영구 저장소 잠금 재설정 영구 저장소 데이터베이스를 다른 프로세스에서 사용할 수 있도록 잠금을 재설정합니다.

표 13–12에서는 `imqdbmgr` 명령에 대한 옵션을 나열합니다.

표 13–12 데이터베이스 관리자 옵션

옵션	설명
`-b` *instanceName*	브로커의 인스턴스 이름
`-D`*property*`=` *value*	브로커 구성 등록 정보 설정 지속성 관련 브로커 구성 등록 정보에 대한 자세한 내용은 263 페이지 “지속성 등록 정보”를 참조하십시오. **주의:** 이 옵션을 사용하여 설정한 등록 정보의 형식과 맞춤법에 주의하십시오. 잘못된 값은 알림이나 경고 없이 무시됩니다.
`-u` *name*	인증에 사용되는 사용자 이름
`-p` *password*	인증에 사용되는 비밀번호[1]
`-passfile` *path*	비밀번호 파일의 위치 자세한 내용은 150 페이지 “비밀번호 파일”을 참조하십시오.

[1] 이 옵션은 더 이상 사용되지 않으며 추후에 제거될 것입니다. 비밀번호를 생략하거나(대화식으로 비밀번호를 묻는 메시지 표시) `-passfile` 옵션을 사용하여 비밀번호를 포함하는 파일을 지정합니다.

표 13–12 데이터베이스 관리자 옵션 *(계속)*

옵션	설명
`-v`	버전 정보 표시[2]
`-h`	사용 도움말 표시[2]

[2] 명령줄에 지정된 다른 옵션은 무시됩니다.

사용자 관리자 유틸리티

사용자 관리자 유틸리티(`imqusermgr`)는 플랫 파일 사용자 저장소를 채우거나 편집하는 데 사용됩니다. 이 유틸리티는 브로커가 설치된 위치와 동일한 호스트에서 실행되어야 합니다. 브로커별 사용자 저장소가 아직 없으면 먼저 해당 브로커 인스턴스를 시작하여 사용자 저장소를 만들어야 합니다. 또한 저장소에 쓸 수 있는 적절한 권한도 필요합니다. 즉, Solaris 및 Linux 플랫폼의 경우에는 해당 브로커 인스턴스를 처음으로 만든 사용자나 루트 사용자여야 합니다.

표 13–13에서는 `imqusermgr` 명령에 사용 가능한 하위 명령을 나열합니다. 이 경우, `-i` 옵션은 명령의 사용자 저장소가 적용되는 브로커의 인스턴스 이름을 지정합니다. 지정하지 않을 경우 기본 이름 `imqbroker`가 사용됩니다.

표 13–13 사용자 관리자 하위 명령

구문	설명

`add` [`-i` *instanceName*]
`-u` *userName* `-p` *password*
[`-g` *group*]
	사용자 및 비밀번호를 저장소에 추가
	선택적 `-g` 옵션은 현재 사용자를 할당할 그룹을 지정합니다.
	`admin`
	`user`
	`anonymous`

`delete` [`-i` *instanceName*] `-u` *userName*	저장소에서 사용자 삭제
`update` [`-i` *instanceName*] `-u` *userName* `-p` *password* [`-a` *activeState*] `update` [`-i` *instanceName*] `-u` *userName* `-a` *activeState* [`-p` *password*]	사용자의 비밀번호 또는 활성 상태 설정(또는 모두) `-a` 옵션은 사용자를 활성화할지(`true`) 비활성화할지(`false`) 여부를 지정하는 부울 값을 사용합니다. **기본값:** `true`
`list` [`-i` *instanceName*] [`-u` *userName*]	사용자 정보 표시 사용자 이름을 지정하지 않을 경우 저장소의 모든 사용자가 나열됩니다.

또한 표 13–14에 나열된 옵션은 imqusermgr 명령의 모든 하위 명령에 적용할 수 있습니다.

표 13–14 일반 사용자 관리자 옵션

옵션	설명
-f	사용자 확인 없이 작업 수행
-s	자동 모드(출력이 표시되지 않음)
-v	버전 정보 표시[1]
-h	사용 도움말 표시[1]

[1] 명령줄에 지정된 다른 옵션은 무시됩니다.

서비스 관리자 유틸리티

서비스 관리자 유틸리티(imqsvcadmin)는 브로커를 Windows 서비스로 설치합니다. 표 13–15에서는 사용 가능한 하위 명령을 나열합니다.

표 13–15 서비스 관리자 하위 명령

하위 명령	설명
install	서비스 설치
remove	서비스 제거
query	시작 옵션 표시 시작 옵션에는 서비스의 수동 또는 자동 시작, 위치, Java Runtime 위치, 시작할 때 브로커에 전달되는 인수 값이 포함됩니다(표 13–16 참조).

표 13–16에서는 imqsvcadmin 명령에 대한 옵션을 나열합니다.

표 13–16 서비스 관리자 옵션

옵션	설명
-javahome *path*	대체 Java Runtime 위치 **기본 동작:** 시스템에 설치된 런타임 또는 Message Queue와 함께 번들로 제공된 런타임을 사용합니다.
-jrehome *path*	대체 JRE(Java Runtime Environment) 위치

표 13–16 서비스 관리자 옵션 *(계속)*

옵션	설명
`-vmargs` *arg1* [[*arg2*] …]	브로커 서비스를 실행 중인 Java 가상 머신(JVM)에 전달할 추가 인수[1] **예:** `imqsvcadmin install -vmargs "-Xms16m -Xmx128m"`
`-args` *arg1* [[*arg2*] …]	브로커 서비스에 전달할 추가 명령줄 인수[1] **예:** `imqsvcadmin install -args "-passfile d:\\imqpassfile"` 브로커 명령줄 인수에 대한 자세한 내용은 238 페이지 "브로커 유틸리티"를 참조하십시오.
`-h`	사용 도움말 표시[2]

[1] 이러한 인수는 서비스의 등록 정보 창(Windows 관리 도구 제어판의 서비스 도구)에 있는 일반 탭의 시작 매개 변수 필드에서도 지정할 수 있습니다.

[2] 명령줄에 지정된 다른 옵션은 무시됩니다.

`-javahome`, `-vmargs` 및 `-args` 옵션을 사용하여 지정한 모든 정보는 Windows 레지스트리의 다음 경로에 있는 `JREHome`, `JVMArgs` 및 `ServiceArgs` 키 아래에 저장됩니다.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\iMQ_Broker\Parameters
```

키 도구 유틸리티

키 도구 유틸리티(`imqkeytool`)는 브로커에 대해 자체 서명한 인증서를 생성합니다. 이 인증서는 `ssljms`, `ssladmin` 또는 `cluster` 연결 서비스에 사용할 수 있습니다. 구문은 다음과 같습니다.

```
imqkeytool -broker
```

UNIX 시스템의 경우 superuser(`root`) 계정에서 이 유틸리티를 실행해야 할 수도 있습니다.

◆ ◆ ◆ 14장

14 브로커 등록 정보 참조

이 장에서는 메시지 브로커의 구성 등록 정보에 대한 참조 정보를 제공합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 257 페이지 "연결 등록 정보"
- 259 페이지 "라우팅 등록 정보"
- 263 페이지 "지속성 등록 정보"
- 268 페이지 "보안 등록 정보"
- 271 페이지 "모니터링 등록 정보"
- 275 페이지 "클러스터 구성 등록 정보"
- 276 페이지 "알파벳 순 브로커 등록 정보 목록"

연결 등록 정보

표 14–1은 연결 서비스 관련 브로커 등록 정보를 나열합니다.

표 14–1 브로커 연결 등록 정보

등록 정보	유형	기본값	설명
`imq.service.activelist`	문자열	`jms,admin`	브로커가 시작될 때 활성화될 쉼표로 구분된 연결 서비스 목록
`imq.hostname`	문자열	사용 가능한 모든 IP 주소	모든 연결 서비스의 기본 호스트 이름 또는 IP 주소
`imq.portmapper.hostname`	문자열	없음	포트 매퍼의 호스트 이름 또는 IP 주소 이 등록 정보를 지정한 경우 `imq.hostname`을 무시합니다.

표 14–1 브로커 연결 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
`imq.portmapper.port`[1]	정수	`7676`	포트 매퍼의 포트 번호 **주 –** 여러 브로커 인스턴스가 동일한 호스트에서 실행 중이면 각 호스트마다 고유한 포트 매퍼 포트가 할당되어야 합니다.
`imq.`*serviceName* `.`*protocolType*`.hostname`[2]	문자열	없음	연결 서비스의 호스트 이름 또는 IP 주소 지정할 경우, 지정된 연결 서비스의 `imq.hostname`을 무시합니다.
`imq.`*serviceName* `.`*protocolType*`.port`[2]	정수	`0`	연결 서비스의 포트 번호 값 `0`은 포트 번호를 포트 매퍼에서 동적으로 할당하도록 지정합니다.
`imq.portmapper.backlog`	정수	`50`	운영 체제 백로그에서 보류 중인 최대 포트 매퍼 요청 수
`imq.`*serviceName* `.threadpool_model`[3]	문자열	`dedicated`	스레드 풀 관리를 위한 스레딩 모델: `dedicated`: 연결당 2개의 전용 스레드(받는 메시지와 보내는 메시지에 대해 각각 하나씩) `shared`: 메시지를 보내거나 받을 때 공유 스레드에서 처리되는 연결 전용 모델은 지원할 수 있는 연결 수를 제한하지만 보다 뛰어난 성능을 제공합니다. 공유 모델은 사용 가능한 연결 수를 늘리지만 스레드 관리에 필요한 추가 오버헤드로 인해 성능이 다소 저하됩니다.
`imq.`*serviceName* `.min_threads`	정수	`jms: 10ssljms : 10httpjms: 10 httpsjms: 10 admin: 4ssladmin: 4`	연결 서비스의 스레드 풀에서 관리되는 최소 스레드 수 사용 가능한 스레드 수가 이 임계값을 초과하면 최소 수에 도달할 때까지 스레드를 종료시켜 여유 스레드를 확보합니다. 기본값은 표시된 대로 연결 서비스에 따라 다릅니다.
`imq.`*serviceName* `.max_threads`	정수	`jms: 1000 ssljms: 500httpjms: 500httpsjms: 500 admin: 10ssladmin: 10`	이 스레드 수를 초과하면 스레드 풀에 명명된 연결 서비스용으로 새 스레드가 더 이상 추가되지 않는 스레드 수 `0`보다 크고 `imq.`*serviceName*`.min_threads` 값보다 커야 합니다. 기본값은 표시된 대로 연결 서비스에 따라 다릅니다.

[1] `imqcmd update bkr` 명령과 함께 사용할 수 있습니다.

[2] `jms`, `ssljms`, `admin` 및 `ssladmin` 서비스에만 해당됩니다. `httpjms` 및 `httpsjms` 서비스 구성에 대한 자세한 내용은 부록 C을 참조하십시오.

[3] `jms` 및 `admin` 서비스에만 해당

표 14–1 브로커 연결 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.shared.connectionMonitor_limit[4]	정수	Solaris: 512Linux: 512Windows: 64	분산자 스레드에서 모니터링되는 최대 연결 수 시스템에서 모든 연결을 모니터하기에 충분한 수의 분산자 스레드를 할당합니다. 이 등록 정보 값이 작을수록 스레드를 활성 연결에 더 빨리 할당할 수 있습니다. 값 -1은 스레드당 무제한 연결 수를 의미합니다. 기본값은 표시된 대로 운영 체제 플랫폼에 따라 다릅니다.
imq.ping.interval	정수	120	클라이언트와 브로커 간 연결을 테스트하는 간격(초) 값 0 또는 -1을 설정하면 연결 테스트를 주기적으로 실행할 수 없습니다.

[4] 공유 스레딩 모델에만 해당

라우팅 등록 정보

표 14–2은 라우팅 서비스 관련 브로커 등록 정보를 나열합니다. 표 14–3에는 대상 자동 생성을 구성하는 등록 정보가 나열되어 있습니다.

표 14–2 브로커 라우팅 등록 정보

등록 정보	유형	기본값	설명
imq.system.max_count[1]	정수	-1	브로커에서 보유하는 최대 메시지 수 값 -1은 무제한 메시지 수를 의미합니다.
imq.system.max_size[1]	문자열	-1	브로커에서 보유하는 메시지의 최대 총 크기 값은 다음 접미사를 사용하여 바이트, 킬로바이트 또는 메가바이트 단위로 표현할 수 있습니다. b: 바이트 k: 킬로바이트(1024바이트) m: 메가바이트(1024 × 1024 = 1,048,576바이트) 접미사가 붙지 않은 값은 바이트로 표현됩니다. 값 -1은 무제한 메시지 용량을 의미합니다.

[1] imqcmd update bkr 명령과 함께 사용할 수 있습니다.

표 14–2 브로커 라우팅 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
			예: 1600: 1600바이트 1600b: 1600바이트 16k: 16킬로바이트(= 16,384바이트) 16m: 16메가바이트(= 16,777,216바이트) -1: 제한 없음
imq.message.max_size[1]	문자열	70m	단일 메시지 본문의 최대 크기 구문은 imq.system.max_size의 구문과 같습니다(위 항목 참조).
imq.message.expiration.interval	정수	60	만료된 메시지가 확보되는 간격(초)
imq.*resourceState* .threshold	정수	green: 0 yellow: 80orange: 90red: 98	메모리 자원 상태가 트리거되는 사용률(여기서 *resourceState*는 green, yellow, orange 또는 red)
imq.*resourceState* .count	정수	green: 5000 yellow: 500orange: 50red: 0	메모리 자원 상태 임계값에 도달했는지 여부를 확인하기 전에 일괄 처리에서 허용되는 최대 받는 메시지 수(여기서 *resourceState*는 green, yellow, orange 또는 red) 이 제한은 시스템 메모리가 점점 부족해지면 메시지 생성자를 억제합니다.
imq.destination.DMQ.truncateBody[1]	부울	false	사용 불능 메시지 대기열에 저장하기 전에 메시지 본문을 제거하시겠습니까? true이면 메시지 헤더와 등록 정보 데이터가 저장됩니다.
imq.transaction.autorollback	부울	false	브로커 시작 시 준비된 상태로 있는 분산된 트랜잭션을 자동으로 롤백하시겠습니까? false이면 트랜잭션을 수동으로 완결하거나 명령 유틸리티(imqcmd)를 사용하여 롤백해야 합니다.

[1] imqcmd update bkr 명령과 함께 사용할 수 있습니다.

표 14–3 자동 생성된 대상에 대한 브로커 등록 정보

등록 정보	유형	기본값	설명
imq.autocreate.queue[1,2]	부울	true	대기열 대상을 자동으로 생성하시겠습니까?
imq.autocreate.topic[3]	부울	true	주제 대상을 자동으로 생성하시겠습니까?

[1] imqcmd update bkr 명령과 함께 사용할 수 있습니다.

[2] 대기열 대상에만 해당

[3] 주제 대상에만 해당

표 14–3 자동 생성된 대상에 대한 브로커 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.autocreate.destination.maxNumMsgs	정수	100000	사용되지 않은 메시지의 최대 개수 값 -1은 무제한 메시지 수를 의미합니다.
imq.autocreate.destination.maxBytesPerMsg	문자열	10k	단일 메시지의 최대 크기(바이트) 값은 다음 접미사를 사용하여 바이트, 킬로바이트 또는 메가바이트 단위로 표현할 수 있습니다. b: 바이트 k: 킬로바이트(1024바이트) m: 메가바이트(1024 × 1024 = 1,048,576바이트) 접미사가 붙지 않은 값은 바이트로 표현됩니다. 값 -1은 무제한 메시지 크기를 의미합니다. **예:** 1600: 1600바이트 1600b: 1600바이트 16k: 16킬로바이트(= 16,384바이트) 16m: 16메가바이트(= 16,777,216바이트) -1: 제한 없음
imq.autocreate.destination.maxTotalMsgBytes	문자열	10m	사용되지 않은 메시지의 최대 총 메모리(바이트) 구문은 imq.autocreate.destination.maxBytesPerMsg의 구문과 같습니다(위 항목 참조).
imq.autocreate.destination.limitBehavior	문자열	REJECT_NEWEST	메모리 제한 임계값에 도달했을 때 브로커 동작: FLOW_CONTROL: 생성자 속도 늦추기 REMOVE_OLDEST: 가장 오래된 메시지 제거 REMOVE_LOW_PRIORITY: 메시지의 보존 기간을 기준으로 우선 순위가 가장 낮은 메시지 삭제(생성자 클라이언트에 알리지 않음) REJECT_NEWEST: 최근 메시지 거부(지속성 메시지의 경우에만 생성자 클라이언트에 예외 알림)

표 14–3 자동 생성된 대상에 대한 브로커 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
			값이 REMOVE_OLDEST 또는 REMOVE_LOW_PRIORITY이고 imq.autocreate.destination.useDMQ 등록 정보가 true이면 초과 메시지가 사용 불능 메시지 대기열로 이동됩니다.
imq.autocreate.destination.maxNumProducers	정수	100	대상에 대한 최대 메시지 생성자 수 이 제한에 도달하면 새로운 생성자가 생성되지 않습니다. 값 -1은 무제한 생성자 수를 의미합니다.
imq.autocreate.queue.maxNumActiveConsumers[2]	정수	1	대기열 대상에서 로드 균형 조정 전달 시 최대 활성 메시지 사용자 수 값 -1은 무제한 사용자 수를 의미합니다.
imq.autocreate.queue.maxNumBackupConsumers[2]	정수	0	대기열 대상에서 로드 균형 조정 전달 시 최대 백업 메시지 사용자 수 값 -1은 무제한 사용자 수를 의미합니다.
imq.autocreate.queue.consumerFlowLimit[2]	정수	1000	단일 일괄 처리에서 대기열 사용자에게 전달되는 최대 메시지 수 로드 균형 조정된 대기열 전달에서 이 수는 로드 균형 조정을 시작하기 전에 활성 사용자에게 라우팅되는 초기 대기 메시지 수입니다. 대상 사용자는 연결에 대해 낮은 값을 지정하여 이 제한을 무시할 수 있습니다. 값 -1은 무제한 메시지 수를 의미합니다.
imq.autocreate.topic.consumerFlowLimit[3]	정수	1000	단일 일괄 처리에서 주제 사용자에게 전달되는 최대 메시지 수 값 -1은 무제한 사용자 수를 의미합니다.
imq.autocreate.destination.isLocalOnly	부울	false	로컬 전달만 하시겠습니까? 이 등록 정보는 브로커 클러스터의 대상에만 적용되며 대상을 만들고 나면 변경할 수 없습니다. 이 값이 true이면 대상을 다른 브로커에 복사하지 않으므로 메시지 전달이 로컬 사용자(대상이 생성된 브로커에 연결된 사용자)에게만 제한됩니다.

[2] 대기열 대상에만 해당

[3] 주제 대상에만 해당

표 14–3 자동 생성된 대상에 대한 브로커 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.autocreate.queue.localDeliveryPreferred[2]	부울	false	로컬 전달을 선호합니까? 이 등록 정보는 브로커 클러스터의 로드 균형 조정된 대기열 전달에만 적용됩니다. 이 값이 true이면 로컬 브로커에 사용자가 없는 경우에만 메시지가 원격 사용자에게 전달됩니다. 대상이 로컬 전용 전달로 제한되지 않아야 합니다(imq.autocreate.destination.isLocalOnly가 false여야 함).
imq.autocreate.destination.useDMQ	부울	true	사용 불능 메시지를 사용 불능 메시지 대기열로 보내겠습니까? false이면 사용 불능 메시지가 삭제됩니다.

[2] 대기열 대상에만 해당

지속성 등록 정보

Message Queue™는 영구 데이터 저장소의 파일 기반 및 JDBC 기반 모델을 모두 지원합니다. 브로커 등록 정보 imq.persist.store(표 14–4)는 사용할 모델을 지정합니다. 다음 절에서는 두 모델의 브로커 구성 등록 정보에 대해 설명합니다.

표 14–4 전역 브로커 지속성 등록 정보

등록 정보	유형	기본값	설명
imq.persist.store	문자열	file	영구 데이터 저장소 모델: file: 파일 기반 지속성 jdbc: JDBC 기반 지속성

파일 기반 지속성

표 14–5는 파일 기반 지속성 관련 브로커 등록 정보를 나열합니다.

표 14–5 파일 기반 지속성에 대한 브로커 등록 정보

등록 정보	유형	기본값	설명
`imq.persist.file.message.max_record_size`	문자열	`1m`	메시지 저장소 파일에 추가할 최대 크기 메시지 이 크기를 초과하는 모든 메시지는 고유의 개별 파일에 저장됩니다. 값은 다음 접미사를 사용하여 바이트, 킬로바이트 또는 메가바이트 단위로 표현할 수 있습니다. `b`: 바이트 `k`: 킬로바이트(1024바이트) `m`: 메가바이트(1024 × 1024 = 1,048,576바이트) 접미사가 붙지 않는 값은 바이트로 표현됩니다. **예:** `1600`: 1600바이트 `1600b`: 1600바이트 `16k`: 16킬로바이트(= 16,384바이트) `16m`: 16메가바이트(= 16,777,216바이트)
`imq.persist.file.destination.message.filepool.limit`	정수	`100`	대상 파일 풀에서 재사용할 수 있는 최대 사용 가능 파일 수 이 제한을 초과하는 사용 가능 파일은 삭제됩니다. 이 제한을 초과할 경우 브로커는 필요에 따라 추가 파일을 작성하고 삭제합니다. 이 제한이 클수록 브로커에서 영구 데이터를 더 빠르게 처리할 수 있습니다.
`imq.persist.file.message.filepool.cleanratio`	정수	`0`	빈 상태로 유지 관리할 사용 가능 파일 풀의 파일 비율 이 값이 클수록 작업 중 파일을 지우는 데 필요한 오버헤드가 늘어나지만, 파일 풀에 필요한 디스크 공간은 줄어듭니다.
`imq.persist.file.message.cleanup`	부울	`false`	종료 시 사용 가능 파일 풀의 파일을 지우겠습니까? 이 등록 정보를 `true`로 설정하면 파일 저장소에 필요한 디스크 공간을 절약할 수 있지만 브로커 종료가 느려집니다.

표 14–5 파일 기반 지속성에 대한 브로커 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.persist.file.sync.enabled	부울	false	메모리 상태를 물리적 저장소 장치와 동기화하시겠습니까? 이 등록 정보를 true로 설정하면 시스템 충돌로 인한 데이터 손실이 없어지지만 성능이 저하됩니다. **주** – Sun Cluster 및 Sun Cluster Data Service for Message Queue를 실행하는 경우 모든 클러스터 노드의 브로커에 대해 이 등록 정보를 true로 설정합니다.

JDBC 기반 지속성

표 14–6은 JDBC 기반 지속성 관련 브로커 등록 정보를 나열합니다. 다음은 DataMirror Mobile Solutions, Inc.의 데이터베이스 제품 중 PointBase® 제품군에 대한 예입니다.

표 14–6 JDBC 기반 지속성에 대한 브로커 등록 정보

등록 정보	예	설명
imq.persist.jdbc.brokerid	PointBase 내장 버전에는 필요하지 않음	*(**선택 사항**)* 브로커 인스턴스 식별자 *n* – 12자 이하의 영숫자 문자열이어야 합니다. 여기서 *n*은 데이터베이스에서 허용되는 최대 테이블 이름 길이입니다. 두 개 이상의 브로커 인스턴스가 영구 데이터 저장소로 같은 데이터베이스를 사용 중인 경우에는 데이터베이스 테이블 이름에 이 식별자가 추가되어 해당 이름을 고유하게 나타냅니다. 보통 한 브로커 인스턴스에 대해서만 데이터를 저장하는 내장 데이터베이스에는 이 식별자가 필요하지 않습니다.
imq.persist.jdbc.driver	com.pointbase.jdbc.jdbcUniversalDriver	데이터베이스에 연결하기 위한 JDBC 드라이버의 Java 클래스 이름

표 14–6 JDBC 기반 지속성에 대한 브로커 등록 정보 *(계속)*

등록 정보	예	설명

`imq.persist.jdbc.opendburl`	`jdbc:pointbase:embedded:`*dbName* `;`
	`database.home=`
	`.../instances/`*instanceName*`/dbstore`
		기존 데이터베이스로의 연결을 여는 데 필요한 URL

`imq.persist.jdbc.createdburl`	`jdbc:pointbase:embedded:`*dbName* `;`
	`new,database.home=`
	`.../instances/`*instanceName*`/dbstore`
		(선택 사항) 새 데이터베이스를 작성하는 데 필요한 URL
		Message Queue 데이터베이스 관리자 유틸리티(`imqdbmgr`)를 사용하여 데이터베이스를 만드는 경우에만 필요합니다.

`imq.persist.jdbc.closedburl`	PointBase에는 필요하지 않음	***(선택 사항)*** 데이터베이스 연결을 닫는 데 필요한 URL
`imq.persist.jdbc.user`		***(선택 사항)*** 필요할 경우 데이터베이스 연결을 여는 데 필요한 사용자 이름 보안상의 이유로 `imqbrokerd -dbuser` 및 `imqdbmgr - u` 명령줄 옵션을 사용해서 값을 지정할 수도 있습니다.
`imq.persist.jdbc.needpassword`		***(선택 사항)*** 브로커에서 데이터베이스에 액세스하려면 비밀번호가 필요합니까? `true`이면 `-passfile` 옵션을 사용하여 비밀번호가 들어 있는 비밀번호 파일을 지정하지 않는 경우 `imqbrokerd` 및 `imqdbmgr` 명령이 비밀번호를 묻는 프롬프트 메시지를 표시합니다.
`imq.persist.jdbc.password`[1]		***(선택 사항)*** 데이터베이스 연결을 여는 데 필요한 비밀번호

`imq.persist.jdbc.table.IMQSV35`	`CREATE TABLE ${`*name*`}`
	`(STOREVERSION INTEGER NOT NULL,`
	`BROKERID VARCHAR(100))`
		버전 테이블을 작성하는 SQL 명령

`imq.persist.jdbc.table.IMQCCREC35`	`CREATE TABLE ${`*name*`}`
	`(RECORDTIME BIGINT NOT NULL,`
	`RECORD BLOB(10k))`
		구성 변경 레코드 테이블을 작성하는 SQL 명령

[1] 비밀번호 파일에만 사용해야 합니다.

표 14–6 JDBC 기반 지속성에 대한 브로커 등록 정보 *(계속)*

등록 정보	예	설명
imq.persist.jdbc.table.IMQDEST35	CREATE TABLE ${*name*} (DID VARCHAR(100) NOT NULL, DEST BLOB(10k), *primaryKey*(DID))	대상 테이블을 작성하는 SQL 명령
imq.persist.jdbc.table.IMQINT35	CREATE TABLE ${*name*} (CUID BIGINT NOT NULL, INTEREST BLOB(10k), *primaryKey*(CUID))	인터레스트 테이블을 작성하는 SQL 명령
imq.persist.jdbc.table.IMQMSG35	CREATE TABLE ${*name*} (MID VARCHAR(100) NOT NULL, DID VARCHAR(100), MSGSIZE BIGINT, MSG BLOB(1m), *primaryKey*(MID))	메시지 테이블을 작성하는 SQL 명령 MSG 열의 기본 최대 길이는 1메가바이트(1m)입니다. 이보다 더 긴 메시지가 있을 것으로 예상되면 길이를 해당 크기에 맞게 설정합니다. 테이블이 이미 작성된 경우에는 해당 테이블을 다시 작성하여 최대 메시지 길이를 변경해야 합니다.
imq.persist.jdbc.table.IMQPROPS35	CREATE TABLE ${*name*} (PROPNAME VARCHAR(100) NOT NULL, PROPVALUE BLOB(10k), *primaryKey*(PROPNAME))	등록 정보 테이블을 작성하는 SQL 명령
imq.persist.jdbc.table.IMQILIST35	CREATE TABLE ${*name*} (MID VARCHAR(100) NOT NULL, CUID BIGINT, DID VARCHAR(100), STATE INTEGER, *primaryKey*(MID, CUID))	인터레스트 상태 테이블을 작성하는 SQL 명령
imq.persist.jdbc.table.IMQTXN35	CREATE TABLE ${*name*} (TUID BIGINT NOT NULL, STATE INTEGER, TSTATEOBJ BLOB(10K), *primaryKey*(TUID))	트랜잭션 테이블을 작성하는 SQL 명령
imq.persist.jdbc.table.IMQTACK35	CREATE TABLE ${*name*} (TUID BIGINT NOT NULL, TXNACK BLOB(10k))	트랜잭션 확인 테이블을 작성하는 SQL 명령

보안 등록 정보

표 14–7은 보안 서비스 관련 브로커 등록 정보를 나열합니다.

표 14–7 브로커 보안 등록 정보

등록 정보	유형	기본값	설명
`imq.accesscontrol.enabled`	부울	`true`	액세스 제어를 사용합니까? `true`이면 시스템에서 액세스 제어 등록 정보 파일을 확인하여 인증된 사용자에게 연결 서비스를 사용하거나 특정 대상과 관련된 특정 작업을 수행할 수 있는 권한이 부여되었는지 확인합니다.
`imq.`*serviceName*`.accesscontrol.enabled`	부울	없음	연결 서비스에 대해 액세스 제어를 사용합니까? 이 등록 정보를 지정할 경우 지정된 연결 서비스의 `imq.accesscontrol.enabled`를 무시합니다. `true`이면 시스템에서 액세스 제어 등록 정보 파일을 확인하여 인증된 사용자에게 지정된 연결 서비스를 사용하거나 특정 대상과 관련된 특정 작업을 수행할 수 있는 권한이 부여되었는지 확인합니다.
`imq.accesscontrol.file.filename`	문자열	`accesscontrol.properties`	액세스 제어 등록 정보 파일 이름 파일 이름은 액세스 제어 디렉토리의 상대 경로를 지정합니다(부록 A 참조).
`imq.`*serviceName* `.accesscontrol.file.filename`	문자열	없음	연결 서비스의 액세스 제어 등록 정보 파일 이름 이 등록 정보를 지정할 경우 지정된 연결 서비스의 `imq.accesscontrol.file.filename`을 무시합니다. 파일 이름은 액세스 제어 디렉토리의 상대 경로를 지정합니다(부록 A 참조).

표 14–7 브로커 보안 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.authentication.type	문자열	digest	비밀번호 인코딩 방법: basic: 기본 64 digest: MD5
imq.*serviceName* .authentication.type	문자열	없음	연결 서비스에 대한 비밀번호 인코딩 방법: basic: 기본 64 digest: MD5 이 등록 정보를 지정한 경우 지정된 연결 서비스에 대해 imq.authentication.type을 무시합니다.
imq.authentication.basic.user_repository	문자열	file	기본 64 인증에 대한 사용자 저장소 유형: file: 파일 기반 ldap: LDAP
imq.authentication.client.response.timeout	정수	180	인증 요청에 대한 클라이언트 응답 대기 시간 간격(초)
imq.passfile.enabled	부울	false	비밀번호 파일에서 비밀번호를 가져옵니까?
imq.passfile.dirpath	문자열	부록 A 참조	비밀번호 파일을 포함하는 디렉토리 경로
imq.passfile.name	문자열	passfile	비밀번호 파일 이름
imq.imqcmd.password	문자열	없음	관리자의 비밀번호 명령 유틸리티(imqcmd)에서는 이 비밀번호를 사용하여 명령을 실행하기 전에 사용자를 인증합니다.
imq.user_repository.ldap.server	문자열	없음	LDAP 서버의 호스트 이름 및 포트 번호 값은 다음 형태로 구성됩니다. *hostName*: *port* 여기서 *hostName*은 LDAP 서버를 실행하는 호스트의 정규화된 DNS 이름이고 *port*는 서버에서 사용하는 포트 번호입니다.

표 14–7 브로커 보안 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
			페일오버 서버 목록을 지정하려면 다음 구문을 사용합니다. *host1*:*port1* `ldap://`*host2*:*port2* `ldap://`*host3*:*port3* … 목록의 항목을 공백으로 구분합니다. 각 페일오버 서버 주소의 접두어는 `ldap://`입니다. SSL을 사용하고 `imq.user_repository.ldap.ssl.enabled` 등록 정보를 `true`로 설정한 경우에도 이 형식을 사용합니다. 주소에 `ldaps`를 지정할 필요는 없습니다.
`imq.user_repository.ldap.principal`	문자열	없음	LDAP 사용자 저장소에 바인드할 때 사용하는 고유 이름 LDAP 서버에서 익명 검색을 허용하는 경우에는 필요하지 않습니다.
`imq.user_repository.ldap.password`[1]	문자열	없음	LDAP 사용자 저장소에 바인드할 때 사용하는 비밀번호 LDAP 서버에서 익명 검색을 허용하는 경우에는 필요하지 않습니다.
`imq.user_repository.ldap.`*propertyName*	제공 예정	제공 예정	제공 예정
`imq.user_repository.ldap.base`	문자열	없음	LDAP 사용자 항목에 사용되는 디렉토리 기반
`imq.user_repository.ldap.uidattr`	문자열	없음	LDAP 사용자 이름에 대한 공급자별 속성 식별자
`imq.user_repository.ldap.usrfilter`	문자열	없음	***(선택 사항)*** LDAP 사용자 검색에 사용할 JNDI 필터
`imq.user_repository.ldap.grpsearch`	부울	`false`	LDAP 그룹 검색을 사용하시겠습니까? **주** – Message Queue에서는 중첩 그룹을 지원하지 않습니다.

[1] 비밀번호 파일에만 사용해야 합니다.

표 14–7 브로커 보안 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.user_repository.ldap.grpbase	문자열	없음	LDAP 그룹 항목에 사용되는 디렉토리 기반
imq.user_repository.ldap.gidattr	문자열	없음	LDAP 그룹 이름에 대한 공급자별 속성 식별자
imq.user_repository.ldap.memattr	문자열	없음	LDAP 그룹의 사용자 이름에 대한 공급자별 속성 식별자
imq.user_repository.ldap.grpfilter	문자열	없음	*(**선택 사항**)* LDAP 그룹 검색에 사용할 JNDI 필터
imq.user_repository.ldap.timeout	정수	280	LDAP 검색 시간 제한(초)
imq.user_repository.ldap.ssl.enabled	부울	false	LDAP 서버와 통신할 때 SSL을 사용합니까?
imq.keystore.file.dirpath	문자열	부록 A 참조	키 저장소 파일을 포함하는 디렉토리 경로
imq.keystore.file.name	문자열	keystore	키 저장소 파일의 이름
imq.keystore.password[1]	문자열	없음	키 저장소 파일의 비밀번호
imq.audit.enabled[2]	부울	false	브로커 로그 파일에 대한 감사 로깅을 시작하시겠습니까?

[1] 비밀번호 파일에만 사용해야 합니다.

[2] Message Queue 엔터프라이즈판에만 적용

모니터링 등록 정보

표 14–8은 모니터링 서비스 관련 브로커 등록 정보를 나열합니다.

표 14–8 브로커 모니터링 등록 정보

등록 정보	유형	기본값	설명

`imq.log.level`[1]	문자열	`INFO`	로깅 수준
			
			출력 채널에 기록할 수 있는 로깅 정보 범주를 지정합니다. 가능한 값은 내림차순으로 다음과 같습니다.
			`ERROR`
			`WARNING`
			`INFO`
			
			각 수준에는 상위 수준이 포함됩니다(예: `WARNING`에는 `ERROR`가 포함됨).

`imq.destination.logDeadMsgs`[1]	부울	`false`	사용 불능 메시지에 대한 정보를 기록하시겠습니까? `true`이면 다음 이벤트가 기록됩니다. ■ 대상이 최대 크기 또는 최대 메시지 수에 도달하여 꽉 찼습니다. ■ 브로커가 관리 명령 또는 전달 확인 이외의 메시지를 제거합니다. ■ 브로커가 메시지를 사용 불능 메시지 대기열로 이동합니다.

`imq.log.console.stream`	문자열	`ERR`	콘솔 출력 대상:
			`OUT`: `stdout`
			`ERR`: `stderr`

`imq.log.console.output`	문자열	`ERROR|WARNING`	콘솔에 기록할 로깅 정보 범주:
			`NONE`
			`ERROR`
			`WARNING`
			`INFO`
			`ALL`
			
			`ERROR`, `WARNING` 및 `INFO` 범주는 상위 범주를 포함하지 **않습니다**. 따라서 필요에 따라 각각의 범주를 명시적으로 지정해야 합니다. 여러 범주를 조합하여 지정할 수 있으며, 이 때 각 범주는 세로 막대(`|`)로 구분합니다.

`imq.log.file.dirpath`	문자열	부록 A 참조	로그 파일을 포함하는 디렉토리 경로
`imq.log.file.filename`	문자열	`log.txt`	로그 파일의 이름

[1] `imqcmd update bkr` 명령과 함께 사용할 수 있습니다.

표 14–8 브로커 모니터링 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.log.file.output	문자열	ALL	로그 파일에 기록할 로깅 정보의 범주: NONE ERROR WARNING INFO ALL ERROR, WARNING 및 INFO 범주는 상위 범주를 포함하지 **않습니다**. 따라서 필요에 따라 각각의 범주를 명시적으로 지정해야 합니다. 여러 범주를 조합하여 지정할 수 있으며, 이 때 각 범주는 세로 막대(|)로 구분합니다.
imq.log.file.rolloverbytes[1]	정수	-1	새 로그 파일로 출력을 롤오버할 때 파일 길이(바이트) 값 -1은 무제한 바이트 수(롤오버 시 파일 길이와 상관 없음)를 의미합니다.
imq.log.file.rolloversecs[1]	정수	604800(1주)	새 로그 파일로 출력을 롤오버할 때 파일의 표시 시간(초) 값 -1은 무제한(롤오버 시 파일 표시 시간과 상관 없음)을 의미합니다.
imq.log.syslog.output[2]	문자열	ERROR	syslogd(1M)에 기록할 로깅 정보 범주: NONE ERROR WARNING INFO ALL ERROR, WARNING 및 INFO 범주는 상위 범주를 포함하지 **않습니다**. 따라서 필요에 따라 각각의 범주를 명시적으로 지정해야 합니다. 여러 범주를 조합하여 지정할 수 있으며, 이 때 각 범주는 세로 막대(|)로 구분합니다.

[1] imqcmd update bkr 명령과 함께 사용할 수 있습니다.

[2] Solaris 플랫폼에만 해당

표 14–8 브로커 모니터링 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.log.syslog.facility[2]	문자열	LOG_DAEMON	메시지 로깅을 위한 syslog 기능 가능한 값은 syslog(3C) 설명서 페이지에 나열된 값을 미러링합니다. Message Queue에서 사용 가능한 값은 다음과 같습니다. LOG_USER LOG_DAEMON LOG_LOCAL0 LOG_LOCAL1 LOG_LOCAL2 LOG_LOCAL3 LOG_LOCAL4 LOG_LOCAL5 LOG_LOCAL6 LOG_LOCAL7
imq.log.syslog.identity[2]	문자열	imqbrokerd_${imq.*instanceName* }	syslog에 기록될 모든 메시지에 접두사로 사용할 아이디 문자열
imq.log.syslog.logpid[2]	부울	true	메시지와 함께 브로커 프로세스 아이디를 기록하시겠습니까?
imq.log.syslog.logconsole[2]	부울	false	메시지를 syslog로 전송할 수 없는 경우 시스템 콘솔에 기록하시겠습니까?
imq.log.timezone	문자열	지역 표준 시간대	로그 타임스탬프의 표준 시간대 가능한 값은 java.util.TimeZone.getTimeZone 메소드에서 사용되는 것과 동일한 값입니다. **예:** GMT GMT-8:00 America/LosAngeles Europe/Rome Asia/Tokyo
imq.metrics.enabled	부울	true	로거에 메트릭 정보를 기록하시겠습니까? 메트릭 메시지 생성(imq.metrics.topic.enabled를 통해 제어됨)에는 영향을 미치지 않습니다.

[2] Solaris 플랫폼에만 해당

표 14–8 브로커 모니터링 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.metrics.interval	정수	-1	메트릭 정보를 로거에 기록하는 시간 간격(초) 메트릭 메시지 생성 간격(imq.metrics.topic.interval를 통해 제어됨)에는 영향을 미치지 않습니다. 값 -1은 무기한 간격(메트릭 정보를 로거에 기록하지 않음)을 의미합니다.
imq.metrics.topic.enabled	부울	true	메트릭 메시지를 메트릭 주제 대상에 생성하시겠습니까? false인 경우 메트릭 주제 대상에 가입하려고 하면 클라이언트측 예외가 발생합니다.
imq.metrics.topic.interval	정수	60	메트릭 메시지를 메트릭 주제 대상에 생성하는 시간 간격(초)
imq.metrics.topic.persist	부울	false	메트릭 주제 대상으로 보낸 메트릭 메시지가 지속적입니까?
imq.metrics.topic.timetolive	정수	300	메트릭 주제 대상으로 보낸 메트릭 메시지의 수명(초)

클러스터 구성 등록 정보

표 14–9는 브로커 클러스터 관련 구성 등록 정보를 나열합니다.

표 14–9 클러스터 구성에 대한 브로커 등록 정보

등록 정보	유형	기본값	설명
imq.cluster.brokerlist[1]	문자열	없음	브로커 주소 목록 이 목록은 쉼표로 구분된 하나 이상의 주소로 구성됩니다. 각 주소는 *hostName*:*portNumber*의 형태로 클러스터 내 브로커의 호스트 이름과 포트 매퍼 포트 번호를 지정합니다. **예:** host1:3000,host2:8000,ctrlhost

[1] 클러스터의 모든 브로커 값이 동일해야 합니다.

표 14–9 클러스터 구성에 대한 브로커 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
imq.cluster.hostname[2]	문자열	없음	cluster 연결 서비스에 대한 호스트 이름 또는 IP 주소 이 등록 정보를 지정할 경우 cluster 연결 서비스의 imq.hostname을 무시합니다(표 14–1 참조).
imq.cluster.port[2]	정수	0	cluster 연결 서비스에 대한 포트 번호 값 0은 포트 번호를 포트 매퍼에서 동적으로 할당하도록 지정합니다.
imq.cluster.transport[1]	문자열	tcp	cluster 연결 서비스에 대한 네트워크 전송 프로토콜 브로커 간의 암호화된 보안 메시지를 전달하려면 이 등록 정보를 ssl로 설정합니다.
imq.cluster.url[1,3]	문자열	없음	cluster 구성 파일의 URL(있는 경우) **예:** http://webserver/imq/cluster.properties (웹 서버의 파일인 경우) file:/net/mfsserver/imq/cluster.properties (공유 드라이브의 파일인 경우)
imq.cluster.masterbroker[1]	문자열	없음	클러스터 마스터 브로커(있는 경우)의 호스트 이름과 포트 번호 값의 형식은 *hostName*: *portNumber*입니다. 여기서 *hostName*은 마스터 브로커의 호스트 이름이고 *portNumber*는 포트 매퍼의 포트 번호입니다. **예:** ctrlhost:7676

[2] 클러스터의 각 브로커에 대해 독립적으로 지정할 수 있습니다.

[1] 클러스터의 모든 브로커 값이 동일해야 합니다.

[3] imqcmd update bkr 명령과 함께 사용할 수 있습니다.

알파벳 순 브로커 등록 정보 목록

276 페이지 “알파벳 순 브로커 등록 정보 목록”은 알파벳 순 브로커 구성 등록 정보 목록이며, 이 장에서 관련 표에 대한 상호 참조를 제공합니다.

표 14–10 알파벳 순 브로커 등록 정보 목록

등록 정보	표
imq.accesscontrol.enabled	표 14–7
imq.accesscontrol.file.filename	표 14–7
imq.audit.enabled	표 14–7
imq.authentication.basic.user_repository	표 14–7
imq.authentication.client.response.timeout	표 14–7
imq.authentication.type	표 14–7
imq.autocreate.destination.isLocalOnly	표 14–3
imq.autocreate.destination.limitBehavior	표 14–3
imq.autocreate.destination.maxBytesPerMsg	표 14–3
imq.autocreate.destination.maxNumMsgs	표 14–3
imq.autocreate.destination.maxNumProducers	표 14–3
imq.autocreate.destination.maxTotalMsgBytes	표 14–3
imq.autocreate.destination.useDMQ	표 14–3
imq.autocreate.queue	표 14–3
imq.autocreate.queue.consumerFlowLimit	표 14–3
imq.autocreate.queue.localDeliveryPreferred	표 14–3
imq.autocreate.queue.maxNumActiveConsumers	표 14–3
imq.autocreate.queue.maxNumBackupConsumers	표 14–3
imq.autocreate.topic	표 14–3
imq.autocreate.topic.consumerFlowLimit	표 14–3
imq.cluster.brokerlist	표 14–9
imq.cluster.hostname	표 14–9
imq.cluster.masterbroker	표 14–9
imq.cluster.port	표 14–9
imq.cluster.transport	표 14–9
imq.cluster.url	표 14–9
imq.destination.DMQ.truncateBody	표 14–2

표 14–10 알파벳 순 브로커 등록 정보 목록 *(계속)*

등록 정보	표
imq.destination.logDeadMsgs	표 14–8
imq.hostname	표 14–1
imq.imqcmd.password	표 14–7
imq.keystore.file.dirpath	표 14–7
imq.keystore.file.name	표 14–7
imq.keystore.password	표 14–7
imq.keystore.*propertyName*	표 14–7
imq.log.console.output	표 14–8
imq.log.console.stream	표 14–8
imq.log.file.dirpath	표 14–8
imq.log.file.filename	표 14–8
imq.log.file.output	표 14–8
imq.log.file.rolloverbytes	표 14–8
imq.log.file.rolloversecs	표 14–8
imq.log.level	표 14–8
imq.log.syslog.facility	표 14–8
imq.log.syslog.identity	표 14–8
imq.log.syslog.logconsole	표 14–8
imq.log.syslog.logpid	표 14–8
imq.log.syslog.output	표 14–8
imq.log.timezone	표 14–8
imq.message.expiration.interval	표 14–2
imq.message.max_size	표 14–2
imq.metrics.enabled	표 14–8
imq.metrics.interval	표 14–8
imq.metrics.topic.enabled	표 14–8
imq.metrics.topic.interval	표 14–8
imq.metrics.topic.persist	표 14–8

표 14–10 알파벳 순 브로커 등록 정보 목록 (계속)

등록 정보	표
imq.metrics.topic.timetolive	표 14–8
imq.passfile.dirpath	표 14–7
imq.passfile.enabled	표 14–7
imq.passfile.name	표 14–7
imq.persist.file.destination.message.filepool.limit	표 14–5
imq.persist.file.message.cleanup	표 14–5
imq.persist.file.message.filepool.cleanratio	표 14–5
imq.persist.file.message.max_record_size	표 14–5
imq.persist.file.sync.enabled	표 14–5
imq.persist.jdbc.brokerid	표 14–6
imq.persist.jdbc.closedburl	표 14–6
imq.persist.jdbc.createdburl	표 14–6
imq.persist.jdbc.driver	표 14–6
imq.persist.jdbc.needpassword	표 14–6
imq.persist.jdbc.opendburl	표 14–6
imq.persist.jdbc.password	표 14–6
imq.persist.jdbc.table.IMQCCREC35	표 14–6
imq.persist.jdbc.table.IMQDEST35	표 14–6
imq.persist.jdbc.table.IMQILIST35	표 14–6
imq.persist.jdbc.table.IMQINT35	표 14–6
imq.persist.jdbc.table.IMQMSG35	표 14–6
imq.persist.jdbc.table.IMQPROPS35	표 14–6
imq.persist.jdbc.table.IMQSV35	표 14–6
imq.persist.jdbc.table.IMQTACK35	표 14–6
imq.persist.jdbc.table.IMQTXN35	표 14–6
imq.persist.jdbc.user	표 14–6
imq.persist.store	표 14–4
imq.ping.interval	표 14–1

표 14–10 알파벳 순 브로커 등록 정보 목록 *(계속)*

등록 정보	표
imq.portmapper.backlog	표 14–1
imq.portmapper.hostname	표 14–1
imq.portmapper.port	표 14–1
imq.*resourceState* .count	표 14–2
imq.*resourceState* .threshold	표 14–2
imq.service.activelist	표 14–1
imq.*serviceName* .accesscontrol.enabled	표 14–7
imq.*serviceName* .accesscontrol.file.filename	표 14–7
imq.*serviceName* .authentication.type	표 14–7
imq.*serviceName* .max_threads	표 14–1
imq.*serviceName* .min_threads	표 14–1
imq.*serviceName* .*protocolType*.hostname	표 14–1
imq.*serviceName* .*protocolType*.port	표 14–1
imq.*serviceName* .threadpool_model	표 14–1
imq.shared.connectionMonitor_limit	표 14–1
imq.system.max_count	표 14–2
imq.system.max_size	표 14–2
imq.transaction.autorollback	표 14–2
imq.user_repository.ldap.base	표 14–7
imq.user_repository.ldap.gidattr	표 14–7
imq.user_repository.ldap.grpbase	표 14–7
imq.user_repository.ldap.grpfilter	표 14–7
imq.user_repository.ldap.grpsearch	표 14–7
imq.user_repository.ldap.memattr	표 14–7
imq.user_repository.ldap.password	표 14–7
imq.user_repository.ldap.principal	표 14–7
imq.user_repository.ldap.*propertyName*	표 14–7
imq.user_repository.ldap.server	표 14–7

표 14–10 알파벳 순 브로커 등록 정보 목록 (계속)

등록 정보	표
imq.user_repository.ldap.ssl.enabled	표 14–7
imq.user_repository.ldap.timeout	표 14–7
imq.user_repository.ldap.uidattr	표 14–7
imq.user_repository.ldap.usrfilter	표 14–7

◆ ◆ ◆ 15장

15 물리적 대상 등록 정보 참조

이 장에서는 물리적 대상의 구성 등록 정보에 대한 참조 정보를 제공합니다. 이러한 등록 정보는 물리적 대상을 작성하거나 업데이트할 때 설정할 수 있습니다. 자동 생성 대상의 경우 브로커의 인스턴스 구성 파일에 기본값을 설정합니다(표 14–3 참조).

물리적 대상 등록 정보

표 15–1 물리적 대상 등록 정보

등록 정보	유형	기본값	설명
maxNumMsgs[1]	정수	-1	사용되지 않은 메시지의 최대 개수 값 -1은 무제한 메시지 수를 의미합니다. 사용 불능 메시지 대기열의 경우 기본값은 1000입니다.
maxBytesPerMsg	문자열	-1	단일 메시지의 최대 크기(바이트) 지속성 메시지 거부는 예외를 통해 생성 클라이언트에 보고됩니다. 비지속성 메시지에 대해서는 알림이 전송되지 않습니다. 값은 다음 접미사를 사용하여 바이트, 킬로바이트 또는 메가바이트 단위로 표현할 수 있습니다. b: 바이트 k: 킬로바이트(1024바이트) m: 메가바이트(1024 × 1024 = 1,048,576바이트) 접미사가 붙지 않은 값은 바이트로 표현됩니다. 값 -1은 무제한 메시지 크기를 의미합니다.

[1] 클러스터 환경에서는 클러스터의 모든 인스턴스에 집합적으로 적용되기 보다는 대상의 각 개별 인스턴스에 적용됩니다.

표 15–1 물리적 대상 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
			예: 1600: 1600바이트 1600b: 1600바이트 16k: 16킬로바이트(= 16,384바이트) 16m: 16메가바이트(= 16,777,216바이트) -1: 제한 없음
maxTotalMsgBytes[1]	문자열	-1	사용되지 않은 메시지의 최대 총 메모리(바이트) 구문은 maxBytesPerMsg의 구문과 같습니다(위 항목 참조). 사용 불능 메시지 대기열의 경우 기본값은 10m입니다.
limitBehavior	문자열	REJECT_NEWEST	메모리 제한 임계값에 도달했을 때 브로커 동작: FLOW_CONTROL: 생성자 속도 늦추기 REMOVE_OLDEST: 가장 오래된 메시지 제거 REMOVE_LOW_PRIORITY: 메시지의 보존 기간을 기준으로 우선 순위가 가장 낮은 메시지 삭제(생성자 클라이언트에 알리지 않음) REJECT_NEWEST: 최근 메시지 거부(지속성 메시지의 경우에만 생성자 클라이언트에 예외 알림) 값이 REMOVE_OLDEST 또는 REMOVE_LOW_PRIORITY이고 useDMQ 등록 정보가 true인 경우 초과 메시지는 사용 불능 메시지 대기열로 이동됩니다. 사용 불능 메시지 대기열 자체의 경우, 기본 제한 동작은 REMOVE_OLDEST이며 FLOW_CONTROL로 설정할 수 없습니다.
maxNumProducers[2]	정수	-1	대상에 대한 최대 메시지 생성자 수 이 제한에 도달하면 새로운 생성자가 생성되지 않습니다. 값 -1은 무제한 생성자 수를 의미합니다.
maxNumActiveConsumers[3]	정수	1	대기열 대상에서 로드 균형 조정 전달 시 최대 활성 메시지 사용자 수 값 -1은 무제한 사용자 수를 의미합니다. Sun Java System Message Queue™ 플랫폼판의 경우 값은 2로 제한됩니다.
maxNumBackupConsumers[3]	정수	0	대기열 대상에서 로드 균형 조정 전달 시 최대 백업 메시지 사용자 수 값 -1은 무제한 사용자 수를 의미합니다. Sun Java System Message Queue 플랫폼판의 경우 값은 1로 제한됩니다.

[1] 클러스터 환경에서는 클러스터의 모든 인스턴스에 집합적으로 적용되기 보다는 대상의 각 개별 인스턴스에 적용됩니다.

[2] 사용 불능 메시지 대기열에 대해 설정할 수 없습니다.

[3] 대기열 대상에만 해당

표 15–1 물리적 대상 등록 정보 *(계속)*

등록 정보	유형	기본값	설명
consumerFlowLimit	정수	1000	단일 일괄 처리에서 사용자에게 전달되는 최대 메시지 수 로드 균형 조정된 대기열 전달에서 이 수는 로드 균형 조정을 시작하기 전에 활성 사용자에게 라우팅되는 초기 대기 메시지 수입니다. 대상 사용자는 연결에 대해 낮은 값을 지정하여 이 제한을 무시할 수 있습니다. 값 -1은 무제한 메시지 수를 의미합니다.
isLocalOnly[2]	부울	false	로컬 전달만 하시겠습니까? 이 등록 정보는 브로커 클러스터의 대상에만 적용되며 대상을 만들고 나면 변경할 수 없습니다. 값이 true이면 대상을 다른 브로커에 복사하지 않으므로 메시지 전달이 로컬 사용자(대상이 생성된 브로커에 연결된 사용자)에게만 제한됩니다.
localDeliveryPreferred[2,3]	부울	false	로컬 전달을 선호합니까? 이 등록 정보는 브로커 클러스터의 로드 균형 조정된 대기열 전달에만 적용됩니다. 값이 true이면 로컬 브로커에 사용자가 없는 경우에만 메시지가 원격 사용자에게 전달됩니다. 대상이 로컬 전용 전달로 제한되지 않아야 합니다(isLocalOnly가 false여야 함).
useDMQ[2]	부울	true	사용 불능 메시지를 사용 불능 메시지 대기열로 보내겠습니까? 값이 false이면 사용 불능 메시지가 삭제됩니다.

[2] 사용 불능 메시지 대기열에 대해 설정할 수 없습니다.

[3] 대기열 대상에만 해당

◆ ◆ ◆ 16 장

16 관리 객체 속성 참조

이 장에서는 관리 대상 객체의 속성에 대한 참조 정보를 제공합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 287 페이지 “연결 팩토리 속성”
- 294 페이지 “대상 속성”
- 295 페이지 “SOAP 종점 속성”

연결 팩토리 속성

연결 팩토리 객체의 속성은 다음 절에 설명된 범주로 분류됩니다.

- 287 페이지 “연결 처리”
- 291 페이지 “클라이언트 아이디”
- 291 페이지 “안정성 및 흐름 제어”
- 292 페이지 “대기열 브라우저 및 서버 세션”
- 293 페이지 “표준 메시지 등록 정보”
- 294 페이지 “메시지 헤더 대체”

연결 처리

표 16–1에는 연결 처리에 사용되는 연결 팩토리 속성이 나열되어 있습니다.

표 16–1 연결 처리에 대한 연결 팩토리 속성

속성	유형	기본값	설명
imqAddressList	문자열	기존 Message Queue™ 3.0 주소(있는 경우), 없는 경우에는 표 16–2의 첫 번째 항목	브로커 주소 목록 이 목록은 쉼표로 구분된 하나 이상의 브로커 주소로 구성됩니다. 각 주소는 클라이언트가 연결할 수 있는 브로커 인스턴스의 호스트 이름, 포트 번호 및 연결 서비스를 지정하거나 나타냅니다. 주소 구문은 연결 서비스 및 포트 할당 방법에 따라 다릅니다. 자세한 내용은 아래를 참조하십시오.
imqAddressListBehavior	문자열	PRIORITY	브로커 주소에 연결을 시도하는 순서: PRIORITY: 주소 목록에 지정된 순서 RANDOM: 임의의 순서 **주 –** 여러 클라이언트가 동일한 연결 팩토리를 공유하는 경우 임의의 연결 순서를 지정하여 모든 클라이언트가 같은 주소로 연결하는 것을 막을 수 있습니다.
imqAddressListIterations	정수	5	연결 설정 또는 재설정을 시도하며 주소 목록을 반복하는 횟수 값 -1은 무제한 반복을 의미합니다.
imqPingInterval	정수	30	클라이언트와 브로커 간 연결을 테스트하는 간격(초) 값 0 또는 -1을 설정하면 연결 테스트를 주기적으로 실행할 수 없습니다.
imqReconnectEnabled	부울	false	끊어진 연결을 다시 설정하시겠습니까?
imqReconnectAttempts	정수	0	다음 주소로 이동하기 전에 주소 목록의 각 주소에 연결 또는 재연결을 시도하는 횟수 값 -1은 무제한 연결 시도를 의미하며 연결될 때까지 첫 번째 주소에 연결하기 위해 반복 시도합니다.
imqReconnectInterval	긴 정수	3000	재연결 시도 간격(밀리초) 이 값은 지정된 주소에 대한 연속적인 시도와 목록의 연속적인 주소에 대해 모두 적용됩니다. **주 –** 값이 너무 작으면 브로커를 복구하는 시간이 충분하지 못할 수도 있습니다. 값이 너무 크면 연결 시 너무 오래 지연될 수 있습니다.

표 16–1 연결 처리에 대한 연결 팩토리 속성 *(계속)*

속성	유형	기본값	설명
imqSSLIsHostTrusted	부울	true	브로커가 제공하는 모든 인증서를 신뢰합니까? false이면 Message Queue 클라이언트 런타임은 제공되는 모든 인증서를 검증합니다. 인증서 서명자가 클라이언트의 트러스트 저장소에 없는 경우 검증이 실패합니다. true이면 인증서 검증이 생략됩니다. 예를 들어, 자체 서명된 인증서를 사용하는 소프트웨어 테스트 중에 유용할 수 있습니다. **참고:** 인증 기관의 서명된 인증서를 사용하려면 이 속성을 false로 설정합니다.

imqAddressList 속성 값은 연결할 하나 이상의 브로커 주소를 지정하는 쉼표로 구분된 문자열입니다. 각 주소의 일반 구문은 다음과 같습니다.

scheme://*address*

여기서 *scheme*은 표 16–2의 첫 번째 열에 표시되는 주소 지정 체계 중 하나를 식별하며 *address*는 브로커 주소 자체를 나타냅니다. 표의 마지막 열에 표시된 대로, 주소를 지정하는 정확한 구문은 주소 지정 체계에 따라 다릅니다.

표 16–2 메시지 브로커 주소 지정 체계

체계	서비스	구문	설명
mq	jms 또는 ssljms	[*hostName*][:*portNumber*][/*serviceName*]	jms 또는 ssljms 연결 서비스에 대해 동적으로 포트를 할당합니다. 주소 목록 항목은 Message Queue 포트 매퍼의 호스트 이름 및 포트 번호를 지정합니다. 포트 매퍼 자체는 연결에 사용할 포트를 동적으로 할당합니다. **기본값:** *hostName* = localhost *portNumber* = 7676 *serviceName* = jms ssljms 연결 서비스의 경우, 모든 변수를 명시적으로 지정해야 합니다.
mqtcp	jms	*hostName*:*portNumber*/jms	jms 연결 서비스를 사용하여 지정된 포트에 연결합니다. 포트 매퍼를 우회하고 지정된 호스트 이름 및 포트 번호로 TCP 연결을 직접 설정합니다.

표 16–2 메시지 브로커 주소 지정 체계 *(계속)*

체계	서비스	구문	설명
mqssl	ssljms	*hostName*:*portNumber* /ssljms	ssljms 연결 서비스를 사용하여 지정된 포트에 연결합니다. 포트 매퍼를 우회하고 지정된 호스트 이름 및 포트 번호로 보안 SSL 연결을 직접 설정합니다.
http	httpjms	http://*hostName*: *portNumber*/*contextRoot* /tunnel 여러 브로커 인스턴스가 동일한 터널 서블릿을 사용하는 경우 다음 구문을 사용하여 임의로 선택된 인스턴스가 아니라 특정 브로커 인스턴스에 연결합니다. http://*hostName*: *portNumber*/*contextRoot* /tunnel? ServerName=*hostName* :*instanceName*	httpjms 연결 서비스를 사용하여 지정된 포트에 연결합니다. 지정된 URL의 Message Queue 터널 서블릿에 HTTP 연결을 설정합니다. 브로커가 HTTP 터널 서블릿에 액세스하도록 구성해야 합니다.
https	httpsjms	https://*hostName*: *portNumber*/*contextRoot* /tunnel 여러 브로커 인스턴스가 동일한 터널 서블릿을 사용하는 경우 다음 구문을 사용하여 임의로 선택된 인스턴스가 아니라 특정 브로커 인스턴스에 연결합니다. https://*hostName*: *portNumber*/*contextRoot* /tunnel? ServerName=*hostName* :*instanceName*	httpsjms 연결 서비스를 사용하여 지정된 포트에 연결합니다. 지정된 URL의 Message Queue 터널 서블릿에 보안 HTTPS 연결을 설정합니다. 브로커가 HTTPS 터널 서블릿에 액세스하도록 구성해야 합니다.

표 16–3 메시지 브로커 주소 예

서비스	브로커 호스트	포트	주소 예
지정하지 않음	지정하지 않음	지정하지 않음	주소 없음(mq://localHost:7676/jms)
지정하지 않음	지정된 호스트	지정하지 않음	myBkrHost (mq://myBkrHost:7676/jms)
지정하지 않음	지정하지 않음	지정된 포트 매퍼 포트	1012 (mq://localHost:1012/jms)
ssljms	로컬 호스트	표준 포트 매퍼 포트	mq://localHost:7676/ssljms
ssljms	지정된 호스트	표준 포트 매퍼 포트	mq://myBkrHost:7676/ssljms
ssljms	지정된 호스트	지정된 포트 매퍼 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	해당 없음	해당 없음	http://websrvr1:8085/imq/tunnel
httpsjms	해당 없음	해당 없음	https://websrvr2:8090/imq/tunnel

클라이언트 아이디

표 16–4에는 클라이언트 아이디에 대한 연결 팩토리 속성이 나열되어 있습니다.

표 16–4 클라이언트 식별에 대한연결 팩토리 속성

속성	유형	기본값	설명
imqDefaultUsername	문자열	guest	브로커 인증 관련 기본 사용자 이름
imqDefaultPassword	문자열	guest	브로커 인증 관련 기본 비밀번호
imqConfiguredClientID	문자열	null	관리상 구성된 클라이언트 식별자
imqDisableSetClientID	부울	false	setClientID 메소드를 사용하여 클라이언트가 클라이언트 식별자를 변경하지 **못하도록** 방지하시겠습니까?

안정성 및 흐름 제어

표 16–5에는 안정성 및 흐름 제어에 사용되는 연결 팩토리 속성이 나열되어 있습니다.

표 16–5 안정성 및 흐름 제어에 대한 연결 팩토리 속성

속성	유형	기본값	설명
imqAckTimeout	문자열	0	예외 발생 전 브로커 확인 대기 최대 시간(밀리초) 값 0은 시간 제한이 없음(무기한 대기)을 의미합니다. **주 –** 경우에 따라서는 값이 너무 낮을 경우 조기 시간 초과가 발생할 수 있습니다. 예를 들어, 보안(SSL) 연결을 사용하는 LDAP 사용자 저장소에 대한 사용자의 초기 인증에는 30초 이상이 소요될 수 있습니다.
imqConnectionFlowCount	정수	100	측정된 일괄 처리의 페이로드 메시지 수 페이로드 메시지를 클라이언트로 전달하는 작업이 이 메시지 수 이후에 일시적으로 중단되므로 누적된 제어 메시지를 전달할 수 있습니다. 클라이언트 런타임의 알림 메시지가 도착하면 페이로드 메시지 전달이 다시 시작되어 지정된 수에 이를 때까지 계속됩니다. 값 0은 메시지 전달 측정을 사용하지 않도록 설정하므로 페이로드 메시지 트래픽이 많을 경우 Message Queue 제어 메시지가 차단될 수 있습니다.
imqConnectionFlowLimitEnabled	부울	false	연결 수준에서 메시지 흐름을 제한합니까?

표 16–5 안정성 및 흐름 제어에 대한 연결 팩토리 속성 (계속)

속성	유형	기본값	설명
imqConnectionFlowLimit	정수	1000	사용을 위해 전달 및 버퍼링할 연결당 최대 메시지 수 보류 중인 사용하지 않은 페이로드 메시지 수(imqConnectionFlowCount에서 제어하는 흐름 측정에 따름)가 이 제한을 초과하면 연결 시 메시지 전달이 중지됩니다. 보류 중인 메시지 수가 제한보다 적을 때에만 전달이 다시 시작됩니다. 따라서 보류 중인 메시지가 너무 많아서 메모리 부족이 발생하는 것을 방지할 수 있습니다. 이 속성은 imqConnectionFlowLimitEnabled가 false인 경우 무시됩니다.
imqConsumerFlowLimit	정수	100	사용을 위해 전달 및 버퍼링할 사용자당 최대 메시지 수 보류 중인 사용하지 않은 페이로드 메시지 수가 해당 사용자에 대해 이 제한을 초과하면 지정된 사용자로의 메시지 전달이 중지됩니다. 사용자에 대한 보류 중인 메시지 수가 imqConsumerFlowThreshold에 지정된 비율보다 낮을 때에만 전달이 다시 시작됩니다. 따라서 여러 사용자가 동일한 연결에 집중되지 않도록 함으로써 각 사용자 간에 로드 균형 조정을 향상시킬 수 있습니다. 대기열 고유의 consumerFlowLimit 속성에 대해 더 낮은 값을 설정하면 이 제한을 대체할 수 있습니다(15 장 참조). 한 연결을 통한 모든 사용자로의 메시지 전달은 imqConnectionFlowLimit에 지정된 전체 제한에 따라 달라질 수 있습니다.
imqConsumerFlowThreshold	정수	50	메시지 전달을 다시 시작하려면 클라이언트 런타임에 버퍼링된 사용자당 메시지 수를 imqConsumerFlowLimit 비율보다 낮게 지정해야 합니다.

대기열 브라우저 및 서버 세션

표 16–6에는 대기열 찾아보기와 서버 세션에 사용되는 연결 팩토리 속성이 나열되어 있습니다.

표 16–6 대기열 브라우저 및 서버 세션에 대한 연결 팩토리 속성

속성	유형	기본값	설명
imqQueueBrowserMaxMessagesPerRetrieve	정수	1000	대기열 대상의 내용을 찾아볼 때 한 번에 검색할 최대 메시지 수 **주 –** 이 속성은 검색되는 총 메시지 수에는 영향을 주지 않고 클라이언트 런타임에 전달을 위해 청크(적지만 큰 청크 또는 많지만 작은 청크)되는 방법에만 영향을 줍니다. 클라이언트 응용 프로그램은 항상 대기열의 모든 메시지를 받습니다. 속성 값을 변경하면 성능에 영향을 줄 수 있지만, 검색되는 총 데이터 양에는 영향을 주지 않습니다.
imqQueueBrowserRetrieveTimeout	긴 정수	60000	대기열 대상의 내용을 찾아볼 때 예외 발생 전 메시지 검색 대기 최대 시간(밀리초)
imqLoadMaxToServerSession	부울	true	최대 메시지 수를 서버 세션으로 로드합니까? 값이 false이면 클라이언트는 한 번에 하나의 메시지만 로드합니다. 이 속성은 JMS 응용 프로그램 서버 기능에만 적용됩니다.

표준 메시지 등록 정보

표 16–7에 나열된 연결 팩토리 속성은 Message Queue 클라이언트 런타임에서 **JMS(Java Message Service) 사양**에 정의된 특정 표준 메시지 등록 정보를 설정할지 여부를 제어합니다.

표 16–7 표준 메시지 등록 정보에 대한 연결 팩토리 속성

등록 정보	유형	기본값	설명
imqSetJMSXUserID	부울	false	생성된 메시지에 대해 JMSXUserID 등록 정보(사용자가 전송하는 메시지의 아이디)를 설정합니까?
imqSetJMSXAppID	부울	false	생성된 메시지에 대해 JMSXAppID 등록 정보(응용 프로그램에서 전송하는 메시지의 아이디)를 설정합니까?
imqSetJMSXProducerTXID	부울	false	생성된 메시지에 대해 JMSXProducerTXID 등록 정보(메시지가 생성된 트랜잭션의 트랜잭션 아이디)를 설정합니까?
imqSetJMSXConsumerTXID	부울	false	사용한 메시지에 대해 JMSXConsumerTXID 등록 정보(메시지가 사용된 트랜잭션의 트랜잭션 아이디)를 설정합니까?

표 16–7 표준 메시지 등록 정보에 대한 연결 팩토리 속성 *(계속)*

등록 정보	유형	기본값	설명
imqSetJMSXRcvTimestamp	부울	false	사용된 메시지에 대해 JMSXRcvTimestamp 등록 정보(메시지가 사용자에게 전달된 시간)를 설정합니까?

메시지 헤더 대체

표 16–8에는 JMS 메시지 헤더 필드 대체에 사용되는 연결 팩토리 속성이 나열되어 있습니다.

표 16–8 메시지 헤더 대체에 대한 연결 팩토리 속성

속성	유형	기본값	설명
imqOverrideJMSDeliveryMode	부울	false	클라이언트에서 설정한 전달 모드를 대체하시겠습니까?
imqJMSDeliveryMode	정수	2	전달 모드 대체 값: 1 비지속성 2 지속성
imqOverrideJMSExpiration	부울	false	클라이언트에서 설정한 만료 시간을 대체하시겠습니까?
imqJMSExpiration	긴 정수	0	만료 시간 대체 값(밀리초) 값 0은 무제한 만료 시간(메시지 만료 없음)을 의미합니다.
imqOverrideJMSPriority	부울	false	클라이언트에서 설정한 우선 순위 수준을 대체하시겠습니까?
imqJMSPriority	정수	4(일반)	우선 순위 수준 대체 값(0 ~ 9)
imqOverrideJMSHeadersToTemporaryDestinations	부울	false	임시 대상에 대체를 적용합니까?

대상 속성

표 16–9에는 대상 관리 객체에 대해 설정할 수 있는 속성이 나열되어 있습니다.

표 16–9 대상 속성

속성	유형	기본값	설명
imqDestinationName	문자열	Untitled_Destination_Object	물리적 대상 이름 대상 이름에는 영숫자 문자(공백 없음)만 사용할 수 있으며 영문자, 밑줄(_) 또는 달러 기호($) 문자로 시작해야 합니다. 대상 이름은 mq로 시작할 수 없습니다.
imqDestinationDescription	문자열	없음	대상을 설명하는 문자열

SOAP 종점 속성

표 16–10에는 SOAP(Simple Object Access Protocol)에서 사용하는 응용 프로그램의 종점 URL 구성에 사용되는 속성이 나열되어 있습니다. 자세한 내용은 **Java 클라이언트용 Message Queue 개발 안내서**를 참조하십시오.

표 16–10 SOAP 종점 속성

속성	유형	기본값	설명
imqSOAPEndpointList	문자열	없음	공백으로 구분된 메시지를 보낼 SOAP 종점을 나타내는 하나 이상의 URL 목록 각 URL은 서블릿에 연결되어 있어야 SOAP 메시지를 받아서 처리할 수 있습니다. **예:** `http://www.serv1/ http://www.serv2/` 목록에서 두 개 이상의 URL을 지정하는 경우 메시지가 모든 URL로 브로드캐스트됩니다.
imqEndpointName	문자열	Untitled_Endpoint_Object	SOAP 종점 이름
imqEndpointDescription	문자열	없음	SOAP 종점을 설명하는 문자열 **예:** `My endpoints for broadcast`

◆ ◆ ◆ 17장

17 JMS 자원 어댑터 등록 정보 참조

이 장에서는 Message Queue™ JMS 자원 어댑터(JMS RA)의 구성 등록 정보에 대해 설명합니다. JMS 자원 어댑터를 사용하면 표준 J2EE Connector Architecture(JCA)를 통해 Sun Java System™ Message Queue를 J2EE 1.4 Application Server에 통합할 수 있습니다. 응용 프로그램 서버에 연결하면 해당 응용 프로그램 서버에 배포된 응용 프로그램은 자원 어댑터를 통해 Message Queue에서 JMS 메시지를 보내고 받을 수 있습니다.

Message Queue JMS 자원 어댑터는 세 가지 JavaBean 세 구성 요소를 통해서 구성 등록 정보를 제공합니다.

- `ResourceAdapter` JavaBean(297 페이지 "ResourceAdapter JavaBean")은 자원 어댑터의 전체 동작에 영향을 미칩니다.
- `ManagedConnectionFactory` JavaBean(299 페이지 "ManagedConnectionFactory JavaBean")은 message-driven bean(MDB)에서 사용하도록 자원 어댑터에서 생성하는 연결에 영향을 줍니다.
- `ActivationSpec` JavaBean(300 페이지 "ActivationSpec JavaBean")은 메시징 시스템과의 상호 작용에서 MDB를 나타내는 메시지 종점에 영향을 줍니다.

이러한 엔티티에 대한 등록 정보 값을 설정하려면 자원 어댑터 구성/배포 및 MDB 배포를 위해 응용 프로그램 서버에서 제공하는 도구를 사용합니다.

이 장에서는 Message Queue JMS 자원 어댑터의 구성 등록 정보를 나열하고 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

ResourceAdapter JavaBean

`ResourceAdapter` 구성은 기본 JMS 자원 어댑터 동작을 구성합니다. 표 17–1에서는 이 JavaBean을 구성할 때 사용할 수 있는 등록 정보를 나열하고 설명합니다.

표 17–1 자원 어댑터 등록 정보

등록 정보	유형	기본값	설명
`addressList`	문자열	`mq://localhost:7676/jms`	*(**필수**)* Message Queue 서비스에 연결하기 위한 메시지 서비스 주소 `connectionURL`(아래)과 동일하며 둘 중 하나를 선택해야 합니다.
`connectionURL`	문자열	`mq://localhost:7676/jms`	Message Queue 서비스에 연결하기 위한 메시지 서비스 주소 `addressList`(위)와 동일하며 둘 중 하나를 선택해야 합니다.
`userName`	문자열	`guest`	*(**필수**)* Message Queue 서비스에 연결하기 위한 기본 사용자 이름
`password`	문자열	`guest`	*(**필수**)* Message Queue 서비스에 연결하기 위한 기본 비밀번호
`addressListBehavior`	문자열	`PRIORITY`	Message Queue 서비스에 대한 연결 시도 순서: `PRIORITY`: 주소 목록에 지정된 순서 `RANDOM`: 임의의 순서 **주 –** 연결에 실패한 브로커에서는 연결 실패 후에 재연결 시도를 시작하고 이 우선 순위 설정 값에 상관없이 주소 목록을 순차적으로 처리합니다.
`addressListIterations`	정수	`1`	연결 설정 또는 재설정을 시도하며 주소 목록을 반복하는 횟수
`reconnectEnabled`	부울	`false`	손실된 연결을 다시 연결하시겠습니까?
`reconnectAttempts`	정수	`6`	다음 주소로 이동하기 전에 주소 목록의 각 주소에 대한 재연결 시도 횟수
`reconnectInterval`	긴 정수	`30000`	재연결 시도 간격(밀리초)

ManagedConnectionFactory JavaBean

관리 대상 연결 팩토리는 자원 어댑터가 message-driven bean에 제공하는 연결을 정의합니다. 표 17–2에서는 `ManagedConnectionFactory` JavaBean의 등록 정보를 보여 줍니다. 이러한 등록 정보는 설정된 경우 `ResourceAdapter` JavaBean의 해당 등록 정보를 무시합니다.

표 17–2 관리 대상 연결 팩토리 등록 정보

등록 정보	유형	기본값	설명
`addressList`	문자열	`ResourceAdapter` JavaBean에서 상속됨(표 17–1 참조)	Message Queue 서비스에 연결하기 위한 메시지 서비스 주소 목록
`userName`	문자열	`guest`	*(**선택 사항**)* Message Queue 서비스에 연결하기 위한 사용자 이름
`password`	문자열	`guest`	*(**선택 사항**)* Message Queue 서비스에 연결하기 위한 비밀번호
`clientID`	문자열	없음	Message Queue 서비스에 연결하기 위한 클라이언트 식별자
`addressListBehavior`	문자열	`PRIORITY`	Message Queue 서비스에 대한 연결 시도 순서: `PRIORITY`: 주소 목록에 지정된 순서 `RANDOM`: 임의의 순서 **주 –** 연결 실패한 브로커에서는 연결 실패 후에 재연결 시도를 시작하고 이 우선 순위 설정 값에 상관없이 주소 목록을 순차적으로 처리합니다.
`addressListIterations`	정수	`1`	연결 설정 또는 재설정을 시도하며 주소 목록을 반복하는 횟수
`reconnectEnabled`	부울	`false`	손실된 연결을 다시 연결하시겠습니까?
`reconnectAttempts`	정수	`6`	다음 주소로 이동하기 전에 주소 목록의 각 주소에 재연결을 시도하는 횟수
`reconnectInterval`	긴 정수	`30000`	재연결 시도 간격(밀리초)

ActivationSpec JavaBean

표 17–3에서는 ActivationSpec JavaBean의 구성 등록 정보를 보여줍니다. 이러한 등록 정보는 응용 프로그램 서버에서 자원 어댑터에 메시지 종점을 활성화하여 message-driven bean에 연결하도록 지시하는 데 사용됩니다.

표 17–3 활성화 사양 등록 정보

등록 정보	유형	기본값	설명
addressList[1]	문자열	ResourceAdapter JavaBean에서 상속됨	*(**선택 사항**)* Message Queue 서비스에 연결하기 위한 메시지 서비스 주소
destination[2]	문자열	없음	*(**필수**)* 메시지를 사용할 대상의 이름 값은 Message Queue 대상 관리 객체에 대한 destinationName 등록 정보 값이어야 합니다.
destinationType[2]	문자열	없음	*(**필수**)* destination 등록 정보에 의해 지정되는 대상 유형: javax.jms.Queue: 대기열 대상 javax.jms.Topic: 주제 대상
messageSelector[2]	문자열	없음	*(**선택 사항**)* 사용자에게 전달되는 메시지를 필터링하기 위한 메시지 선택기
subscriptionName[2]	문자열	없음	영구 가입 이름 subscriptionDurability가 Durable로 설정되어 있는 경우 이 등록 정보를 설정해야 합니다.
subscriptionDurability[2]	문자열	NonDurable	주제 대상의 사용자 지속성: Durable: 영구 사용자 NonDurable: 비영구 사용자 이 등록 정보는 destinationType이 javax.jms.Topic으로 설정되어 있는 경우에만 유효하며, 비영구 가입의 경우 선택 사항이고 영구 가입의 경우 필수 사항입니다. Durable로 설정된 경우 clientID 및 subscriptionName 등록 정보도 설정해야 합니다.

[1] Message Queue JMS 자원 어댑터에 관련된 등록 정보

[2] 표준 Enterprise JavaBean(EJB) 및 J2EE Connector Architecture(CA) 등록 정보

표 17–3 활성화 사양 등록 정보 *(**계속**)*

등록 정보	유형	기본값	설명
`clientId`[2]	문자열	없음	Message Queue 서비스에 연결하기 위한 클라이언트 ID `subscriptionDurability`가 `Durable`로 설정되어 있는 경우 이 등록 정보를 설정해야 합니다.
`acknowledgeMode`[2]	문자열	`Auto-acknowledge`	*(**선택 사항**)* 확인 모드: `Auto-acknowledge`: 자동 승인 모드 `Dups-ok-acknowledge`: Dups-OK-acknowledge 모드
`customAcknowledgeMode`	문자열	없음	MDB 메시지 사용을 위한 확인 모드 유효한 값은 `No_acknowledge` 또는 null입니다. 트랜잭션되지 않은 비영구 주제 가입에 대해서만 no-acknowledge 모드를 사용할 수 있습니다. 트랜잭션 가입이나 영구 가입에 대해 이 설정을 사용하면 가입이 활성화되지 않습니다.
`endpointExceptionRedeliveryAttempts`	정수	`6`	MDB가 메시지 전달 중에 예외를 발생하는 경우에 메시지를 다시 전달하는 횟수
`sendUndeliverableMsgsToDMQ`	부울	`true`	MDB에서 런타임 예외가 발생하고 재전달 시도 횟수가 `endpointExceptionRedeliveryAttempts` 값을 초과할 경우 메시지를 사용 불능 메시지 대기열에 넣습니까? 값이 `false`이면 Message Queue 브로커는 동일한 MDB를 포함하여 유효한 사용자에게 메시지를 재전송하려고 시도합니다.

[2] 표준 Enterprise JavaBean(EJB) 및 J2EE Connector Architecture(CA) 등록 정보

◆ ◆ ◆ 18장

18 메트릭 참조

이 장에서는 Message Queue™ 메시지 브로커가 모니터링, 조정 및 진단을 위해 제공할 수 있는 메트릭 정보에 대해 설명합니다. 이 정보는 다양한 방식으로 사용할 수 있습니다.

- 로그 파일에서(184 페이지 "로그 파일로 메트릭 데이터 보내기" 참조)
- 명령 유틸리티의 `metrics bkr` 명령을 통해(244 페이지 "브로커 관리" 참조)
- 메트릭 주제 대상에 보낸 메트릭 메시지에서(190 페이지 "브로커를 모니터하는 응용프로그램 작성" 참조)

이 장의 표에는 사용 가능한 다양한 메트릭 정보와 해당 정보가 제공될 수 있는 양식이 나열됩니다. 명령 유틸리티의 `metrics bkr` 명령을 통해 제공되는 메트릭의 경우 요청하는 데 사용할 수 있는 메트릭 유형이 표에 나열되고, 메트릭 메시지로 제공되는 메트릭의 경우 전달되는 메트릭 주제 대상이 표에 나열됩니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 303 페이지 "JVM 메트릭"
- 304 페이지 "브로커 전체 메트릭"
- 305 페이지 "연결 서비스 메트릭"
- 307 페이지 "대상 메트릭"

JVM 메트릭

표 18–1에서는 브로커가 브로커 프로세스JVM(Java Virtual Machine) 힙에 대해 보고하는 메트릭 정보를 보여줍니다.

표 18–1 JVM 메트릭

메트릭 개수	설명	로그 파일 여부	`metrics bkr` 메트릭 유형	메트릭 주제
JVM 힙: 전체 메모리	현재 전체 메모리(바이트)	예	`cxn`	`mq.metrics.jvm`

표 18–1 JVM 메트릭 *(계속)*

메트릭 개수	설명	로그 파일 여부	metrics bkr 메트릭 유형	메트릭 주제
JVM 힙: 사용 가능한 메모리	현재 사용할 수 있는 메모리 양(바이트)	예	cxn	mq.metrics.jvm
JVM 힙: 최대 메모리	허용 가능한 최대 힙 크기(바이트)	예	없음	mq.metrics.jvm

브로커 전체 메트릭

표 18–2에서는 브로커에서 보고하는 브로커 전체 메트릭 정보를 보여줍니다.

표 18–2 브로커 전체 메트릭

메트릭 개수	설명	로그 파일 여부	metrics bkr 메트릭 유형	메트릭 주제
연결				
Num connections	모든 연결 서비스에 대한 총 현재 연결 수	예	cxn	mq.metrics.broker
Num threads	모든 연결 서비스에 대한 총 현재 스레드 수	예	cxn	없음
최소 스레드	모든 연결 서비스에 대한 총 최소 스레드 수	예	cxn	없음
최대 스레드	모든 연결 서비스에 대한 총 최대 스레드 수	예	cxn	없음
저장된 메시지				
Num messages	메모리 및 영구 저장소에 저장된 현재 페이로드 메시지 수	아니요	없음[1]	mq.metrics.broker
총 메시지 바이트 수	메모리 및 영구 저장소에 현재 저장된 총 페이로드 메시지 크기(바이트)	아니요	없음[1]	mq.metrics.broker
메시지 흐름				
Num messages in	브로커가 시작된 이후에 받은 누적 페이로드 메시지 수	예	ttl	mq.metrics.broker
Num messages out	브로커가 시작된 이후에 보낸 누적 페이로드 메시지 수	예	ttl	mq.metrics.broker
Rate messages in	브로커로 페이로드 메시지가 유입되는 현재 속도	예	rts	없음
Rate messages out	브로커에서 페이로드 메시지가 유출되는 현재 속도	예	rts	없음
Message bytes in	브로커가 시작된 이후에 받은 누적 페이로드 메시지 크기(바이트)	예	ttl	mq.metrics.broker

[1] query bkr 명령을 대신 사용

표 18–2 브로커 전체 메트릭 *(계속)*

메트릭 개수	설명	로그 파일 여부	metrics bkr 메트릭 유형	메트릭 주제
Message bytes out	브로커가 시작된 이후에 보낸 누적 페이로드 메시지 크기(바이트)	예	ttl	mq.metrics.broker
Rate message bytes in	브로커로 페이로드 메시지 바이트가 유입되는 현재 속도	예	rts	없음
Rate message bytes out	브로커에서 페이로드 메시지 바이트가 유출되는 현재 속도	예	rts	없음
Num packets in	브로커가 시작된 이후에 받은 누적 페이로드 및 제어 패킷 수	예	ttl	mq.metrics.broker
Num packets out	브로커가 시작된 이후에 보낸 누적 페이로드 및 제어 패킷 수	예	ttl	mq.metrics.broker
Rate packets in	브로커로 유입되는 페이로드 및 제어 패킷의 현재 흐름 속도	예	rts	없음
Rate packets out	브로커에서 유출되는 페이로드 및 제어 패킷의 현재 흐름 속도	예	rts	없음
Packet bytes in	브로커가 시작된 이후에 받은 누적 페이로드 및 제어 패킷 크기(바이트)	예	ttl	mq.metrics.broker
Packet bytes out	브로커가 시작된 이후에 보낸 누적 페이로드 및 제어 패킷 크기(바이트)	예	ttl	mq.metrics.broker
Rate packet bytes in	브로커로 유입되는 페이로드 및 제어 패킷 바이트의 현재 흐름 속도	예	rts	없음
Rate packet bytes out	브로커에서 유출되는 페이로드 및 제어 패킷 바이트의 현재 흐름 속도	예	rts	없음
대상				
Num destinations	현재 물리적 대상 수	아니요	없음	mq.metrics.broker

연결 서비스 메트릭

표 18–3에서는 브로커가 개별 연결 서비스에 대해 보고하는 메트릭 정보를 보여줍니다.

표 18–3 연결 서비스 메트릭

메트릭 개수	설명	로그 파일 여부	metrics svc 메트릭 유형	메트릭 주제
연결				

표 18–3 연결 서비스 메트릭 *(계속)*

메트릭 개수	설명	로그 파일 여부	metrics svc 메트릭 유형	메트릭 주제
Num connections	현재 연결 수	아니요	cxn[1]	없음
Num threads	현재 스레드 수	아니요	cxn[1]	없음
최소 스레드	서비스에 할당된 최소 스레드 수	아니요	cxn	없음
최대 스레드	서비스에 할당된 최대 스레드 수	아니요	cxn	없음
메시지 흐름				
Num messages in	브로커가 시작된 이후에 연결 서비스를 통해 받은 누적 페이로드 메시지 수	아니요	ttl	없음
Num messages out	브로커가 시작된 이후에 연결 서비스를 통해 보낸 현재 페이로드 메시지 수	아니요	ttl	없음
Rate messages in	연결 서비스를 통해 페이로드 메시지가 브로커로 유입되는 현재 속도	아니요	rts	없음
Rate messages out	연결 서비스를 통해 페이로드 메시지가 브로커에서 유출되는 현재 속도	아니요	rts	없음
Message bytes in	브로커가 시작된 이후에 연결 서비스를 통해 받은 누적 페이로드 메시지 크기(바이트)	아니요	ttl	없음
Message bytes out	브로커가 시작된 이후에 연결 서비스를 통해 보낸 누적 페이로드 메시지 크기(바이트)	아니요	ttl	없음
Rate message bytes in	연결 서비스를 통해 페이로드 메시지 바이트가 브로커로 유입되는 현재 속도	아니요	rts	없음
Rate message bytes out	연결 서비스를 통해 페이로드 메시지 바이트가 브로커에서 유출되는 현재 속도	아니요	rts	없음
Num packets in	브로커가 시작된 이후에 연결 서비스를 통해 받은 누적 페이로드 및 제어 패킷 수	아니요	ttl	없음
Num packets out	브로커가 시작된 이후에 연결 서비스를 통해 보낸 누적 페이로드 및 제어 패킷 수	아니요	ttl	없음
Rate packets in	연결 서비스를 통해 페이로드 및 제어 패킷이 브로커로 유입되는 현재 속도	아니요	rts	없음
Rate packets out	연결 서비스를 통해 페이로드 및 제어 패킷이 브로커에서 유출되는 현재 속도	아니요	rts	없음
Packet bytes in	브로커가 시작된 이후에 연결 서비스를 통해 받은 누적 페이로드 및 제어 패킷 크기(바이트)	아니요	ttl	없음

[1] query svc 명령에서도 사용 가능

표 18–3 연결 서비스 메트릭 *(계속)*

메트릭 개수	설명	로그 파일 여부	metrics svc 메트릭 유형	메트릭 주제
Packet bytes out	브로커가 시작된 이후에 연결 서비스를 통해 보낸 누적 페이로드 및 제어 패킷 크기(바이트)	아니요	ttl	없음
Rate packet bytes in	연결 서비스를 통해 페이로드 및 제어 패킷 바이트가 브로커로 유입되는 현재 속도	아니요	rts	없음
Rate packet bytes out	연결 서비스를 통해 페이로드 및 제어 패킷 바이트가 브로커에서 유출되는 현재 속도	아니요	rts	없음

대상 메트릭

표 18–4에서는 브로커가 개별 대상에 대해 보고하는 메트릭 정보를 보여줍니다.

표 18–4 대상 메트릭

메트릭 개수	설명	로그 파일 여부	metrics dst메트릭 유형	메트릭 주제
메시지 사용자				
Num consumers	현재 연결된 메시지 사용자 수 대기열 대상의 경우 이 속성은 활성 사용자와 백업 사용자를 모두 포함합니다. 주제 대상의 경우 비영구 및 (활성 및 비활성) 영구 가입자를 포함하며 "Num active consumers"와 같습니다."	아니요	con	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*

표 18–4 대상 메트릭 *(계속)*

메트릭 개수	설명	로그 파일 여부	metrics dst 메트릭 유형	메트릭 주제
Peak num consumers	브로커가 시작된 이후에 연결된 최대 메시지 사용자 수 대기열 대상의 경우 이 속성은 활성 사용자와 백업 사용자를 모두 포함합니다. 주제 대상의 경우 비영구 및 (활성 및 비활성) 영구 가입자를 포함하며 "Peak num active consumers"와 같습니다."	아니요	con	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Avg num consumers	브로커가 시작된 이후에 연결된 평균 메시지 사용자 수 대기열 대상의 경우 이 속성은 활성 사용자와 백업 사용자를 모두 포함합니다. 주제 대상의 경우 비영구 및 (활성 및 비활성) 영구 가입자를 포함하며 "Avg num active consumers"와 같습니다."	아니요	con	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Num active consumers	현재 연결된 활성 메시지 사용자 수 주제 대상의 경우 이 속성은 비영구 및 (활성 및 비활성) 영구 가입자를 포함하며 "Num consumers"와 같습니다."	아니요	con	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*

표 18–4 대상 메트릭 *(계속)*

메트릭 개수	설명	로그 파일 여부	metrics dst 메트릭 유형	메트릭 주제
Peak num active consumers	브로커가 시작된 이후에 연결된 최대 활성 메시지 사용자 수 주제 대상의 경우 이 속성은 비영구 및 (활성 및 비활성) 영구 가입자를 포함하며 "Peak num consumers"와 같습니다."	아니요	con	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Avg num active consumers	브로커가 시작된 이후에 연결된 평균 활성 메시지 사용자 수 주제 대상의 경우 이 속성은 비영구 및 (활성 및 비활성) 영구 가입자를 포함하며 "Avg num consumers"와 같습니다."	아니요	con	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Num backup consumers[1]	현재 연결된 백업 메시지 사용자 수	아니요	con	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Peak num backup consumers[1]	브로커가 시작된 이후에 연결된 최대 백업 메시지 사용자 수	아니요	con	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Avg num backup consumers[1]	브로커가 시작된 이후에 연결된 평균 백업 메시지 사용자 수	아니요	con	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
저장된 메시지				
Num messages	메모리 및 영구 저장소에 저장된 현재 메시지 수	아니요	conttlrts[2]	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Peak num messages	브로커가 시작된 이후에 메모리 및 영구 저장소에 저장된 최대 메시지 수	아니요	conttlrts	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*

[1] 대기열 대상에만 해당

[2] query dst 명령에서도 사용 가능

표 18–4 대상 메트릭 (계속)

메트릭 개수	설명	로그 파일 여부	metrics dst 메트릭 유형	메트릭 주제
Avg num messages	브로커가 시작된 이후에 메모리 및 영구 저장소에 저장된 평균 메시지 수	아니요	conttlrts	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
총 메시지 바이트 수	메모리 및 영구 저장소에 저장된 현재 메시지의 총 크기(바이트)	아니요	ttlrts[2]	mq.metrics.destination.queue.*queueName* mq.metrics.destination.topic.*topicName*
Peak total message bytes	브로커가 시작된 이후에 메모리 및 영구 저장소에 저장된 최대 총 메시지 크기(바이트)	아니요	ttlrts	mq.metrics.destination.queue.*queueName* mq.metrics.destination.topic.*topicName*
Avg total message bytes	브로커가 시작된 이후에 메모리 및 영구 저장소에 저장된 평균 총 메시지 크기(바이트)	아니요	ttlrts	mq.metrics.destination.queue.*queueName* mq.metrics.destination.topic.*topicName*
메시지 흐름				
Num messages in	브로커가 시작된 이후에 받은 누적 메시지 수	아니요	ttl	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Num messages out	브로커가 시작된 이후에 보낸 누적 메시지 수	아니요	ttl	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Msg bytes in	브로커가 시작된 이후에 받은 누적 메시지 크기(바이트)	아니요	ttl	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Msg bytes out	브로커가 시작된 이후에 보낸 누적 메시지 크기(바이트)	아니요	ttl	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Peak message bytes	브로커가 시작된 이후에 받은 가장 큰 단일 메시지 크기(바이트)	아니요	ttlrts	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Rate num messages in	받은 메시지의 현재 흐름 속도	아니요	rts	없음
Rate num messages out	보낸 메시지의 현재 흐름 속도	아니요	rts	없음

[2] query dst 명령에서도 사용 가능

표 18–4 대상 메트릭 *(계속)*

메트릭 개수	설명	로그 파일 여부	metrics dst 메트릭 유형	메트릭 주제
Rate msg bytes in	받은 메시지 바이트의 현재 흐름 속도	아니요	rts	없음
Rate msg bytes out	보낸 메시지 바이트의 현재 흐름 속도	아니요	rts	없음
디스크 사용률				
Disk reserved[3]	대상에 대해 예약된 디스크 공간의 양(바이트)	아니요	dsk	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Disk used[3]	대상이 현재 사용 중인 디스크 공간의 양(바이트)	아니요	dsk	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*
Disk utilization ratio[3]	대상에 대해 예약된 디스크 공간을 기준으로 한 사용 중인 디스크 공간 비율	아니요	dsk	mq.metrics.destination.queue.*queueName*mq.metrics.destination.topic.*topicName*

[3] 파일 기반 지속성에만 해당

4

부록

- 부록 A
- 부록 B
- 부록 C
- 부록 D

◆ ◆ ◆ 부 록 A

플랫폼별 Message Queue™ 데이터 위치

Sun Java System™ Message Queue 데이터는 운영 체제 플랫폼별로 다른 위치에 저장됩니다. 다음 표에서는 다음과 같은 플랫폼에서 다양한 유형의 Message Queue 데이터의 위치를 나타냅니다.

- 315 페이지 “Solaris”
- 316 페이지 “Linux”
- 317 페이지 “Windows”

다음 표에서 *instanceName*은 데이터와 관련된 브로커 인스턴스의 이름을 나타냅니다.

Solaris

표 A–1에서는 Solaris 운영 체제에서 Message Queue 데이터의 위치를 보여 줍니다. Message Queue를 Solaris에서 표준 버전의 Sun Java System Application Server와 함께 사용하는 경우 디렉토리 구조는 317 페이지 “Windows”에서 설명한 구조와 같습니다.

표 A–1 Solaris 플랫폼의 Message Queue 데이터 위치

데이터 범주	위치
브로커 인스턴스 구성 등록 정보	`/var/imq/instances/`*instanceName*`/props/config.properties`
브로커 구성 파일 템플리트	`/usr/share/lib/imq/props/broker/`
영구 저장소(메시지, 대상, 영구 가입, 트랜잭션)	`/var/imq/instances/`*instanceName* `/fs350` 또는 JDBC 액세스 가능 데이터 저장소
브로커 인스턴스 로그 파일 디렉토리(기본 위치)	`/var/imq/instances/`*instanceName* `/log/`
관리 대상 객체(객체 저장소)	선택한 로컬 디렉토리 또는 LDAP 서버

표 A–1 Solaris 플랫폼의 Message Queue 데이터 위치 *(계속)*

데이터 범주	위치
보안: 사용자 저장소	`/var/imq/instances/`*instanceName* `/etc/passwd` 또는 LDAP 서버
보안: 액세스 제어 파일(기본 위치)	`/var/imq/instances/`*instanceName* `/etc/accesscontrol.properties`
보안: 비밀번호 파일 디렉토리(기본 위치)	`/var/imq/instances/`*instanceName* `/etc/`
보안: 예제 비밀번호 파일	`/etc/imq/passfile.sample`
보안: 브로커의 키 저장소 파일 위치	`/etc/imq/`
JavaDoc API 설명서	`/usr/share/javadoc/imq/index.html`
예제 응용 프로그램 및 구성	`/usr/demo/imq/`
Java 아카이브(`.jar`), 웹 아카이브(`.war`) 및 자원 어댑터 아카이브(`.rar`) 파일	`/usr/share/lib/`

Linux

표 A–2에서는 Linux 운영 체제에서 Message Queue 데이터의 위치를 보여 줍니다.

표 A–2 Linux 플랫폼의 Message Queue 데이터 위치

데이터 범주	위치
브로커 인스턴스 구성 등록 정보	`/var/opt/sun/mq/instances/`*instanceName* `/props/config.properties`
브로커 구성 파일 템플리트	`/opt/sun/mq/private/share/lib/props/`
영구 저장소(메시지, 대상, 영구 가입, 트랜잭션)	`/var/opt/sun/mq/instances/`*instanceName* `/fs350/` 또는 JDBC 액세스 가능 데이터 저장소
브로커 인스턴스 로그 파일 디렉토리(기본 위치)	`/var/opt/sun/mq/instances/`*instanceName* `/log/`
관리 대상 객체(객체 저장소)	선택한 로컬 디렉토리 또는 LDAP 서버
보안: 사용자 저장소	`/var/opt/sun/mq/instances/`*instanceName* `/etc/passwd` 또는 LDAP 서버
보안: 액세스 제어 파일(기본 위치)	`/var/opt/sun/mq/instances/`*instanceName* `/etc/accesscontrol.properties`
보안: 비밀번호 파일 디렉토리(기본 위치)	`/var/opt/sun/mq/instances/`*instanceName* `/etc/`

표 A–2 Linux 플랫폼의 Message Queue 데이터 위치 *(계속)*

데이터 범주	위치
보안: 예제 비밀번호 파일	`/etc/opt/sun/mq/passfile.sample`
보안: 브로커의 키 저장소 파일 위치	`/etc/opt/sun/mq/`
JavaDoc API 설명서	`/opt/sun/mq/javadoc/index.html`
예제 응용 프로그램 및 구성	`/opt/sun/mq/examples/`
Java 아카이브(`.jar`), 웹 아카이브(`.war`) 및 자원 어댑터 아카이브(`.rar`) 파일	`/opt/sun/mq/share/lib/`
공유 라이브러리(`.so`) 파일	`/opt/sun/mq/lib/`

Windows

표 A–3에서는 Windows 운영 체제에서 Message Queue 데이터의 위치를 보여 줍니다. 또한 이 표는 Message Queue가 독립 실행형 버전의 Sun Java System Application Server에 번들된 경우 Solaris 플랫폼에 적용됩니다. 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 변경하십시오. `IMQ_HOME` 및 `IMQ_VARHOME` 디렉토리 변수에 대한 자세한 내용은 25 페이지 “디렉토리 변수 규칙”을 참조하십시오.

표 A–3 Windows 플랫폼의 Message Queue 데이터 위치

데이터 범주	위치
브로커 인스턴스 구성 등록 정보	`IMQ_VARHOME\instances\`*instanceName*`\props\config.properties`
브로커 구성 파일 템플리트	`IMQ_HOME\lib\props\broker\`
영구 저장소(메시지, 대상, 영구 가입, 트랜잭션)	`IMQ_VARHOME\instances\`*instanceName* `\fs350\` 또는 JDBC 액세스 가능 데이터 저장소
브로커 인스턴스 로그 파일 디렉토리(기본 위치)	`IMQ_VARHOME\instances\`*instanceName* `\log\`
관리 대상 객체(객체 저장소)	선택한 로컬 디렉토리 또는 LDAP 서버
보안: 사용자 저장소	`IMQ_VARHOME\instances\`*instanceName* `\etc\passwd` 또는 LDAP 서버
보안: 액세스 제어 파일(기본 위치)	`IMQ_VARHOME\instances\`*instanceName*`\etc\accesscontrol.properties`

표 A–3 Windows 플랫폼의 Message Queue 데이터 위치 *(계속)*

데이터 범주	위치
보안: 비밀번호 파일 디렉토리(기본 위치)	`IMQ_HOME\etc\`
보안: 예제 비밀번호 파일	`IMQ_HOME\etc\passfile.sample`
보안: 브로커의 키 저장소 파일 위치	`IMQ_HOME\etc\`
JavaDoc API 설명서	`IMQ_HOME\javadoc\index.html`
예제 응용 프로그램 및 구성	`IMQ_HOME\demo\`
Java 아카이브(`.jar`), 웹 아카이브(`.war`) 및 자원 어댑터 아카이브(`.rar`) 파일	`IMQ_HOME\lib\`

◆ ◆ ◆ 부 록 B

Message Queue™ 인터페이스의 안정성

Sun Java System™ Message Queue에서는 관리자가 작업을 자동화할 때 도움이 되는 여러 인터페이스를 사용합니다. 이 부록에서는 인터페이스의 안정성에 따라 인터페이스를 분류합니다. 인터페이스가 안정적일수록 제품의 향후 버전에서 변경될 가능성이 적어집니다.

이 부록에 나열되지 않은 인터페이스는 고객용이 아니라 개인 인터페이스입니다.

분류 체계

부록 B에서는 안정성 분류 체계에 대해 설명합니다.

표 B–1 인터페이스 안정성 분류 체계

분류	설명
개인	고객이 직접 사용하지 않음. 임의의 릴리스에서 변경 또는 제거될 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. 천천히 신중하게 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.

표 B–1 인터페이스 안정성 분류 체계 (계속)

분류	설명
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 제거되거나 호환되지 않는 방식으로 또는 상당히 많이 변경될 수 있음을 고객에게 알립니다. 고객은 불안정한 인터페이스에 대해 명시적인 종속성을 생성하지 않는 것이 바람직합니다.

인터페이스 안정성

부록 B에 인터페이스와 해당 분류가 나열되어 있습니다.

표 B–2 Message Queue 인터페이스의 안정성

인터페이스	분류
명령줄 인터페이스	
`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`, `imqusermgr`의 출력	불안정
명령	
`imqobjmgr` 명령 파일	변화
`imqbrokerd` 명령	안정
`imqadmin` 명령	불안정
`imqcmd` 명령	안정
`imqdbmgr` 명령	불안정
`imqkeytool` 명령	안정
`imqobjmgr` 명령	안정

표 B–2 Message Queue 인터페이스의 안정성 *(계속)*

인터페이스	분류
`imqusermgr` 명령	불안정
API	
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
.jar 및 .war 파일	
`imq.jar` 위치 및 이름	안정
`jms.jar` 위치 및 이름	변화
`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
파일	
브로커 로그 파일 위치 및 내용 형식	불안정

표 B–2 Message Queue 인터페이스의 안정성 (계속)

인터페이스	분류
비밀번호 파일	불안정
`accesscontrol.properties` 파일	불안정
시스템 대상	
`mq.sys.dmq` 대상	안정
`mq.metrics.*` 대상	변화
구성 등록 정보	
Message Queue JMS 자원 어댑터 구성 등록 정보	변화
Message Queue JMS 자원 어댑터 JavaBean 및 ActivationSpec 구성 등록 정보	변화
메시지 등록 정보 및 형식	
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
기타	
Message Queue JMS 자원 어댑터 패키지, `com.sun.messaging.jms.ra`	변화
지속성 메시지 저장소의 JDBC 스키마	변화

◆ ◆ ◆ 부록 C

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 클라이언트에서는 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.

- 323 페이지 "HTTP/HTTPS 지원 구조"
- 325 페이지 "HTTP 지원 활성화"
- 332 페이지 "HTTPS 지원 활성화"
- 341 페이지 "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.

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 서버/응용 프로그램 서버에 전송되는지 확인합니다.
- 클라이언트는 필요한 경우 HTTP 프록시 서버를 사용해서 브로커와 통신할 수 있습니다. 프록시의 주소는 클라이언트를 시작할 때 명령줄 옵션을 사용하여 지정합니다. 자세한 내용은 332 페이지 "HTTP 프록시 사용"을 참조하십시오.
- HTTP 또는 HTTPS 터널 서블릿(모두 Message Queue와 함께 제공)이 웹 서버/응용 프로그램 서버에 로드되고 페이로드 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 사용하여 여러 브로커에 액세스할 수 있습니다.

- 브로커 측에서는 `httpjms` 또는 `httpsjms` 연결 서비스가 해당 터널 서블릿에서 들어오는 메시지를 분해하고 다중화를 해제합니다.
- 웹 서버/응용 프로그램 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 서버/응용 프로그램 서버와 브로커에 모두 오류가 발생하고 브로커가 다시 시작되지 않은 경우에는 웹 서버/응용 프로그램 서버가 클라이언트 연결을 복원한 후 클라이언트에 알리지 않고 브로커 연결을 계속 기다릴 수 있습니다. 이런 상황을 방지하려면 항상 브로커를 다시 시작해야 합니다.

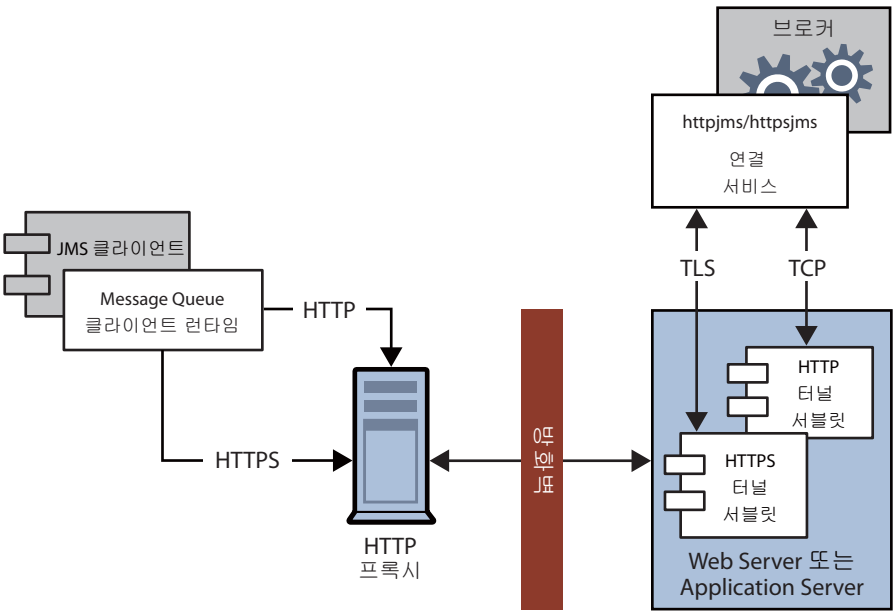


그림 C–1 HTTP/HTTPS 지원 구조

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 브로커에 자체 서명된 인증서를 전달합니다. 인증서는 브로커가 HTTPS 터널 서블릿에 대해 암호화된 연결을 설정할 때 사용합니다. 이 연결이 설정되고 나면 클라이언트 응용 프로그램과 웹 서버/응용 프로그램 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 배포합니다. 다음 서버에 HTTP 터널 서블릿을 배포할 수 있습니다.**

- Sun Java System™ Web Server
- Sun Java System Application Server

2 **브로커의 httpjms 연결 서비스를 구성하고 브로커를 시작합니다.**

3 **HTTP 연결을 구성합니다.**

1단계. HTTP 터널 서블릿 배포

HTTP 터널 서블릿을 Sun Java System Web Server 또는 Sun Java System Application Server에 웹 아카이브(.war) 파일로 배포할 수 있습니다.

HTTP 터널 서블릿을 .war 파일로 배포하는 작업은 웹 서버/응용 프로그램 서버에서 제공하는 배포 기법을 사용하여 수행합니다. HTTP 터널 서블릿 .war 파일(imqhttp.war)은 .jar, .war 및 .rar 파일이 포함된 디렉토리에 있으며 운영 체제에 따라 다릅니다(부록 A 참조).

.war 파일에는 웹 서버/응용 프로그램 서버에서 서블릿을 로드하여 실행하는 데 필요한 기본 구성 정보가 들어 있는 배포 설명자가 포함되어 있습니다. 웹 서버/응용 프로그램 서버에 따라 서블릿 URL의 컨텍스트 루트 부분을 지정해야 하는 경우도 있습니다.

웹 아카이브 파일로 배포

Sun Java System Web Server 상의 배포는 325 페이지 “Sun Java System Web Server에 HTTP 터널 서블릿 배포”를 참조하십시오.

Sun Java System Application Server 상의 배포는 327 페이지 “Sun Java System Application Server에 HTTP 터널 서블릿 배포”를 참조하십시오.

Sun Java System Web Server에 HTTP 터널 서블릿 배포

다음은 Sun Java System Web Server에 배포 시의 참조 지침입니다. 웹 브라우저를 사용하여 서블릿 URL에 액세스함으로써 HTTP 터널 서블릿 배포 성공 여부를 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 위치와 WAR 파일 경로 필드에서 `imqhttp.war` 파일을 가리키는 적절한 값을 선택합니다. 이 파일이 있는 디렉토리는 운영 체제에 따라 다릅니다(부록 A 참조).**
8. **응용 프로그램 URI 필드에 경로를 입력합니다.**
   응용 프로그램 URI 필드 값은 다음 터널 서블릿 URL의 */contextRoot* 부분입니다.

   `http://`*hostName* `:`*portNumber* `/` *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.

    `http://`*hostName*`:`*portNumber*`/`
    contextRoot`/tunnel`

    이제 클라이언트에서 이 URL을 통해 HTTP 연결을 사용하는 메시지 서비스에 연결할 수 있습니다.

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.**
로그 클라이언트 액세스 제어를 사용하여 로깅을 비활성화합니다.

Sun Java System Application Server에 HTTP 터널 서블릿 배포

이 절에서는 HTTP 터널 서블릿을 Sun Java System Application Server에 `.war` 파일로 배포한 다음 Sun Java System 브로커에서 연결을 수신하도록 터널 서블릿을 구성하는 방법에 대해 설명합니다.

다음 두 단계를 거쳐야 합니다.

- Application Server 배포 도구를 사용하여 HTTP 터널 서블릿 배포
- 응용 프로그램 서버 인스턴스의 `server.policy` 파일 수정

배포 도구 사용

▼ Application Server 환경에서 HTTP 터널 서블릿을 배포하는 방법

1 **웹 기반 관리 GUI에서 다음을 선택합니다.**
App 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 입력한 다음 확인을 누릅니다.**
`imqhttp.war` 파일의 위치는 운영 체제에 따라 다릅니다(부록 A 참조).

4 **컨텍스트 루트 필드의 값을 설정하고 확인을 누릅니다.**
컨텍스트 루트 필드 값은 터널 서블릿 URL의 */contextRoot* 부분입니다.

`http://` *hostName* `:` *portNumber* `/` *contextRoot*`/tunnel`

예를 들어, 컨텍스트 루트 필드를 `/imq`로 설정할 수 있습니다.

터널 서블릿이 성공적으로 배포되었고 기본적으로 사용되며 다음 위치(이 예의 경우)에 있음을 나타내는 확인 화면이 표시됩니다.

```
/var/opt/SUNWappserver8/domains/domain1/server1/applications/
j2ee-modules/imqhttp_1
```

이제 서블릿을 다음 URL에서 사용할 수 있습니다.

```
http://hostName:portNumber/
contextRoot/tunnel
```

이제 클라이언트에서 이 URL을 통해 HTTP 연결을 사용하는 메시지 서비스에 연결할 수 있습니다.

server.policy 파일 수정

Application Server에서 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/SUNWappserver8/domains/domain1/server1/config/
server.policy
```

터널 서블릿이 Message Queue 브로커에서 연결을 수신하도록 구성하려면 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/SUNWappserver8/domains/domain1/server1/
                applications/j2ee-modules/imqhttp_1/-
{
        permission java.net.SocketPermission "*",
                "connect,accept,resolve";
};
```

2단계. httpjms 연결 서비스 구성

기본적으로 브로커에 대해 HTTP 지원이 활성화되어 있지 않으므로 httpjms 연결 서비스가 활성화되도록 브로커를 재구성해야 합니다. 재구성한 경우 68 페이지 “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 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 서버/응용 프로그램 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 내용은 328 페이지 “2단계. httpjms 연결 서비스 구성”을 참조하십시오.

등록 정보	설명
`imq.httpjms.http.servletHost`	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`.
`imq.httpjms.http. servletPort`	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.

등록 정보	설명
imq.httpjms.http. pullPeriod	브로커에서 메시지를 가져오기 위해 각 클라이언트가 런타임에 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/응용 프로그램 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/응용 프로그램 서버 자원이 절약됩니다. 기본값: -1.
imq.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 객체를 사용해서 브로커에 대한 HTTP 연결을 설정해야 합니다. 이 절에서는 HTTP 연결 구성 문제에 대해 논의합니다.

연결 팩토리 구성

HTTP 지원을 활성화하려면 연결 팩토리의 imqAddressList 속성을 HTTP 터널 서블릿 URL로 설정해야 합니다. HTTP 터널 서블릿 URL의 일반 구문은 다음과 같습니다.

http://*hostName*:*portNumber*

/*contextRoot*/tunnel

여기서 *hostName*:*portNumber*는 HTTP 터널 서블릿을 호스트하는 웹 서버/응용 프로그램 서버의 이름과 포트이며, *contextRoot*는 웹 서버/응용 프로그램 서버에서 터널 서블릿을 배포할 때 설정된 경로입니다.

일반적인 연결 팩토리 속성, 특히 `imqAddressList` 속성에 대한 자세한 내용은 **Java 클라이언트용 Message Queue 개발 안내서**를 참조하십시오.

다음 방법 중 하나를 사용하여 연결 팩토리 속성을 설정할 수 있습니다.

- 연결 팩토리 관리 대상 객체를 만드는 `imqobjmgr` 명령에 `-o` 옵션을 사용하거나(164 페이지 "연결 팩토리 추가" 참조) 관리 콘솔(`imqadmin`)을 사용하여 연결 팩토리 관리 대상 객체를 만들 때 속성을 설정합니다.
- 클라이언트를 시작하는 명령에 `-D` 옵션을 사용합니다(**Java 클라이언트용 Message Queue 개발 안내서** 참조).
- 클라이언트 코드에서 프로그래밍 방식으로 연결 팩토리를 만든 후 API 호출을 사용하여 해당 연결 팩토리의 속성을 설정합니다(**Java 클라이언트용 Message Queue 개발 안내서** 참조).

단일 서블릿을 사용하여 다중 브로커에 액세스

다중 브로커를 실행할 경우 다중 웹 서버/응용 프로그램 서버 및 서블릿 인스턴스를 구성하지 않아도 됩니다. 동시에 실행 중인 브로커 간에 단일 웹 서버/응용 프로그램 서버와 HTTP 터널 서블릿 인스턴스를 공유할 수 있습니다. 다중 브로커 인스턴스가 단일 터널 서블릿을 공유하는 경우 `imqAddressList` 연결 팩토리 속성을 다음과 같이 구성해야 합니다.

`http://`*hostName*`:`*portNumber*

`/`*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 2005
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.

HTTPS 지원 활성화

다음 절에서는 HTTPS 지원을 활성화하는 단계에 대해 설명합니다. 325 페이지 "HTTP 지원 활성화"와 비슷한 내용이며 SSL 인증서 생성 및 액세스에 필요한 단계가 추가되어 있습니다.

▼ HTTPS 지원을 활성화하는 방법

1. **HTTPS 터널 서블릿에 대해 자체 서명된 인증서를 생성합니다.**
2. **HTTP 터널 서블릿 `.war` 파일 배포 설명자를 다음과 같이 수정합니다.**
   - 인증서 키 저장소가 있는 위치를 가리킵니다.
     - 인증서 키 저장소 비밀번호를 지정합니다.
3. **HTTP 터널 서블릿을 배포합니다. 다음 서버에 HTTP 터널 서블릿을 배포할 수 있습니다.**
   - Sun Java System Web Server
     - Sun Java System Application Server
4. **브로커의 httpsjms 연결 서비스를 구성하고 브로커를 시작합니다.**
5. **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.

keytool 유틸리티를 실행하여 터널 서블릿에 대해 자체 서명된 인증서를 생성합니다. 명령 프롬프트에서 다음을 입력합니다.

```
JRE_HOME/bin/keytool -servlet keyStoreLocation
```

유틸리티는 필요한 정보를 묻는 메시지를 표시합니다Unix 시스템에서 키 저장소를 만들 권한을 가지려면 keytool을 수퍼유저(루트)로 실행해야 할 수도 있습니다.

먼저 keytool은 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 잠글 비밀번호(키 비밀번호)를 묻습니다. 이 프롬프트에 대한 응답으로 Return 키를 눌러야 합니다. 그러면 키 비밀번호가 키 저장소 비밀번호와 동일하게 지정됩니다.

주 – 입력한 비밀번호를 기억해 두십시오. 나중에 이 비밀번호를 제공해야 터널 서블릿이 키 저장소를 열 수 있습니다.

JDK keytool 유틸리티는 자체 서명된 인증서를 생성하여 *keyStoreLocation* 인수에서 지정한 위치의 Message Queue 키 저장소 파일에 넣을 수 있습니다.

주 – HTTPS 터널 서블릿에서 키 저장소를 볼 수 있어야 합니다. *keyStoreLocation* 에 있는 생성된 키 저장소를 HTTPS 터널 서블릿에서 액세스할 수 있는 위치로 이동/복사해야 합니다(334 페이지 “3단계. HTTP 터널 서블릿 배포” 참조).

2단계. HTTP 터널 서블릿 .war 파일의 설명자 파일 수정

HTTP 터널 서블릿의 .war 파일에는 웹 서버/응용 프로그램 서버에서 서블릿을 로드하여 실행하는 데 필요한 기본 구성 정보가 들어 있는 배포 설명자가 포함되어 있습니다.

하지만 imqhttps.war 파일의 배포 설명자에서는 터널 서블릿에 필요한 키 저장소 파일이 있는 위치를 알 수 없습니다. 따라서 imqhttps.war 파일을 배포하기 전에 터널 서블릿의 배포 설명자(XML 파일)를 편집하여 키 저장소 위치 및 비밀번호를 지정해야 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 **keystoreLocation 및 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 있음).

3단계. HTTP 터널 서블릿 배포

HTTP 터널 서블릿을 Sun Java System Web Server 또는 Sun Java System Application Server에 웹 아카이브(WAR) 파일로 배포할 수 있습니다.

HTTPS 터널 서블릿을 .war 파일로 배포하는 작업은 웹 서버/응용 프로그램 서버에서 제공하는 배포 기법을 사용하여 수행합니다. HTTPS 터널 서블릿 .war 파일(imqhttps.war)은 운영 체제에 따라 다른 디렉토리에 있습니다(부록 A 참조).

클라이언트와 브로커 사이에서 종단간 보안 통신이 이루어질 수 있도록 웹 서버에 암호화가 활성화되었는지 확인해야 합니다.

웹 아카이브 파일로 배포

Sun Java System Web Server 상의 배포는 334 페이지 “Sun Java System Web Server에 HTTPS 터널 서블릿 배포”를 참조하십시오.

Sun Java System Application Server 상의 배포는 336 페이지 “Sun Java System Application Server에 HTTPS 터널 서블릿 배포”를 참조하십시오.

Sun Java System Web Server에 HTTPS 터널 서블릿 배포

이 절에서는 Sun Java System Web Server에서 HTTPS 터널 서블릿을 .war 파일로 배포하는 방법에 대해 설명합니다. 웹 브라우저를 사용하여 서블릿 URL에 액세스함으로써 HTTPS 터널 서블릿 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 WebServer_TOPDIR/bin/https/jre/lib/ext로 복사하는 것입니다.

▼ 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(333 페이지 "2단계. HTTP 터널 서블릿 .war 파일의 설명자 파일 수정" 참조)을 가리키는 적절한 값을 선택합니다.**
8. **응용 프로그램 URI 필드에 경로를 입력합니다.**

   응용 프로그램 URI 필드 값은 다음 터널 서블릿 URL의 */contextRoot* 부분입니다.

   https://*hostName* :*portNumber* / *contextRoot*/tunnel

   예를 들어, *contextRoot*를 imq로 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 URL에서 사용할 수 있습니다.

    https://*hostName*:*portNumber*/imq/tunnel

    이제 클라이언트에서 이 URL을 통해 보안 HTTPS 연결을 사용하는 메시지 서비스에 연결할 수 있습니다.

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.**
   로그 클라이언트 액세스 제어를 사용하여 로깅을 비활성화합니다.

Sun Java System Application Server에 HTTPS 터널 서블릿 배포

이 절에서는 Sun Java System Application Server에서 HTTPS 터널 서블릿을 `.war` 파일로 배포하는 방법에 대해 설명합니다.

다음 두 단계를 거쳐야 합니다.

- Application Server 배포 도구를 사용하여 HTTP 터널 서블릿 배포
- 응용 프로그램 서버 인스턴스의 `server.policy` 파일 수정

배포 도구 사용

다음 절차에서는 Application Server 환경에서 HTTPS 터널 서블릿을 배포하는 방법을 보여 줍니다.

▼ Application Server 환경에서 HTTPS 터널 서블릿을 배포하는 방법

1. **웹 기반 관리 GUI에서 다음을 선택합니다.**
   App Server > 인스턴스 > server1 > 응용 프로그램 > 웹 응용 프로그램
2. **배포 버튼을 누릅니다.**
3. **파일 경로: 텍스트 필드에 HTTPS 터널 서블릿 `.war` 파일(`imqhttps.war`)의 위치를 입력한 다음 확인을 누릅니다.**
   `imqhttps.war` 파일의 위치는 운영 체제에 따라 다릅니다(부록 A 참조).
4. **컨텍스트 루트 필드의 값을 설정하고 확인을 누릅니다.**
   컨텍스트 루트 필드 값은 터널 서블릿 URL의 */contextRoot* 부분입니다.

   `https://`*hostName* `:`*portNumber* `/` *contextRoot*`/tunnel`

   예를 들어, 컨텍스트 루트 필드를 다음과 같이 설정할 수 있습니다.

   `/imq`

다음 화면에 터널 서블릿이 성공적으로 배포되었고 기본적으로 사용되며 다음 위치(이 예의 경우)에 있다고 표시됩니다.

```
/var/opt/SUNWappserver8/domains/domain1/server1/applications/
j2ee-modules/imqhttps_1
```

이제 서블릿을 다음 URL에서 사용할 수 있습니다.

```
https://hostName:portNumber/
contextRoot/tunnel
```

이제 클라이언트에서 이 URL을 통해 HTTPS 연결을 사용하는 메시지 서비스에 연결할 수 있습니다.

server.policy 파일 수정

Application Server에서 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/SUNWappserver8/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/SUNWappserver8/domains/domain1/server1/
                applications/j2ee-modules/imqhttps_1/-
{
        permission java.net.SocketPermission "*",
                "connect,accept,resolve";
};
```

4단계. httpsjms 연결 서비스 구성

기본적으로 브로커에 대해 HTTPS 지원이 활성화되어 있지 않으므로 httpsjms 연결 서비스가 활성화되도록 브로커를 재구성해야 합니다. 재구성한 경우 68 페이지 “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 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 내용은 337 페이지 "4단계. httpsjms 연결 서비스 구성"을 참조하십시오.

등록 정보	설명
imq.httpsjms.https.servletHost	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 .
imq.httpsjms.https. servletPort	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.
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.

등록 정보	설명
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.

5단계. HTTPS 연결 구성

클라이언트 응용 프로그램에서는 제대로 구성된 연결 팩토리 관리 객체를 사용해서 브로커에 대한 HTTPS 연결을 설정해야 합니다.

하지만 클라이언트는 JSSE(Java Secure Socket Extension)에서 제공하는 SSL 라이브러리에도 액세스해야 하며 루트 인증서도 있어야 합니다. SSL 라이브러리는 JDK 1.4와 함께 제공됩니다. 이전 버전의 JDK가 있는 경우에는 339 페이지 "JSSE 구성"을 참조하십시오. 그렇지 않으면 340 페이지 "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/응용 프로그램 서버 인증서를 사용하는 경우에는 인증서를 트러스트 데이터베이스에 추가해야 합니다. 여기에 해당되면 다음 지침에 따르고, 그렇지 않으면 340 페이지 "연결 팩토리 구성"으로 이동합니다.

인증서가 *certFile*에 저장되어 있고 *trustStoreFile*이 키 저장소라면 다음 명령을 실행합니다.

```
JRE_HOME/bin/keytool -import -trustcacerts
 -alias aliasForCertificate -file certFile

-keystore trustStoreFile
```

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=trustStoreFile
javax.net.ssl.trustStorePassword=trustStorePasswd
```

연결 팩토리 구성

HTTPS 지원을 활성화하려면 연결 팩토리의 `imqAddressList` 속성을 HTTPS 터널 서블릿 URL로 설정해야 합니다. HTTPS 터널 서블릿 URL의 일반 구문은 다음과 같습니다.

```
https://hostName:portNumber

/contextRoot/tunnel
```

여기서 *hostName*:*portNumber*는 HTTPS 터널 서블릿을 호스트하는 웹 서버의 이름과 포트이며, *contextRoot*는 웹 서버에서 터널 서블릿을 배포할 때 설정된 경로입니다.

일반적인 연결 팩토리 속성, 특히 `imqAddressList` 속성에 대한 자세한 내용은 **Java 클라이언트용 Message Queue 개발 안내서**를 참조하십시오.

다음 방법 중 하나를 사용하여 연결 팩토리 속성을 설정할 수 있습니다.

- 연결 팩토리 관리 대상 객체를 만드는 `imqobjmgr` 명령에 `-o` 옵션을 사용하거나(164 페이지 "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(**Java 클라이언트용 Message Queue 개발 안내서** 참조).
- 클라이언트 응용 프로그램 코드에서 프로그래밍 방식으로 연결 팩토리를 만든 후 API 호출을 사용하여 해당 연결 팩토리의 속성을 설정합니다(**Java 클라이언트용 Message Queue 개발 안내서** 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 서블릿 인스턴스를 공유할 수 있습니다. 다중 브로커 인스턴스가 단일 터널 서블릿을 공유하는 경우 `imqAddressList` 연결 팩토리 속성을 다음과 같이 구성해야 합니다.

`https://`*`hostName`*`:`*`portNumber`*

`/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 secured 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.

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.

▼ 클라이언트가 연결되지 않는 경우

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.

◆ ◆ ◆ 부 록 D

자주 사용하는 명령 유틸리티 명령

이 부록에서는 자주 사용하는 Message Queue™ 명령 유틸리티(imqcmd) 명령을 나열합니다. 명령줄에서 사용할 수 있는 명령 옵션 및 속성의 전체 목록은 242 페이지 "명령 유틸리티"의 242 페이지 "명령 유틸리티"를 참조하십시오.

구문

```
imqcmd subcommand argument [
options]
imqcmd -h|H
imqcmd -v
```

-H 또는 -h는 전체 도움말을 제공합니다. -v 하위 명령은 버전 정보를 제공합니다.

imqcmd를 사용하면 명령 유틸리티에서 비밀번호를 입력하라는 메시지를 표시합니다. 프롬프트 메시지가 표시되지 않도록 하여 보안을 향상시키려면 -passfile *pathToPassfile* 옵션을 사용하여 이 유틸리티를 관리자 이름 및 비밀번호를 포함하는 비밀번호 파일로 나타냅니다.

예: `imqcmd query bkr -u` *adminUserName* `-passfile` *pathToPassfile* `-b` *myServer*`:7676`

브로커 및 클러스터 관리

```
imqcmd query bkr
imqcmd pause bkr
imqcmd restart bkr
imqcmd resume bkr
imqcmd shutdown bkr -b myBroker:7676
imqcmd update bkr -o "imq.system.max_count=1000"
imqcmd reload cls
```

브로커 구성 등록 정보(-o 옵션)

344 페이지 “브로커 구성 등록 정보(-o 옵션)”는 자주 사용하는 브로커 구성 등록 정보를 나열합니다. 브로커 구성 등록 정보 및 설명의 전체 목록은 14 장을 참조하십시오.

표 D–1 브로커 구성 등록 정보(-o 옵션)

등록 정보	주
imq.autocreate.queue	
imq.autocreate.queue.maxNumActiveConsumers	무제한으로 설정하려면 -1 지정
imq.autocreate.queue.maxNumBackupConsumers	무제한으로 설정하려면 -1 지정
imq.autocreate.topic	
imq.cluster.url	
imq.destination.DMQ.truncateBody	
imq.destination.logDeadMessages	
imq.log.file.rolloverbytes	무제한으로 설정하려면 -1 지정
imq.log.file.rolloversecs	무제한으로 설정하려면 -1 지정
imq.log.level	NONEERRORWARNINGINFO
imq.message.max_size	무제한으로 설정하려면 -1 지정
imq.portmapper.port	
imq.system.max_count	무제한으로 설정하려면 -1 지정
imq.system.max_size	무제한으로 설정하려면 -1 지정

서비스 및 연결 관리

```
imqcmd list svc
imqcmd query svc
imqcmd update svc -n jms -o "minThreads=200" -o "maxThreads=400" -o "port=8995"
imqcmd pause svc -n jms
imqcmd resume svc -n jms
imqcmd list cxn -svn jms
imqcmd query cxn -n 1234567890
```

영구 가입자 관리

```
imqcmd list dur -d MyTopic
imqcmd destroy dur -n myDurSub -c "clientID-111.222.333.444"
imqcmd purge dur -n myDurSub -c "clientID-111.222.333.444"
```

트랜잭션 관리

```
imqcmd list txn
imqcmd commit txn -n 1234567890
imqcmd query txn -n 1234567890
imqcmd rollback txn -n 1234567890
```

대상 관리

```
imqcmd create dst -n MyQueue -t q -o "maxNumMsgs=1000" -o "maxNumProducers=5"
imqcmd update dst -n MyTopic -t t -o "limitBehavior=FLOW_CONTROL|

REMOVE_OLDEST|REJECT_NEWEST|REMOVE_LOW_PRIORITY"
imqcmd compact dst -n MyQueue -t q
imqcmd purge dst -n MyQueue -t q
imqcmd pause dst -n MyQueue -t q -pst PRODUCERS|CONSUMERS|ALL
imqcmd resume dst -n MyQueue -t q
imqcmd destroy dst -n MyQueue -t q
imqcmd query dst -n MyQueue -t q
imqcmd list dst -tmp
```

대상 구성 등록 정보(-o 옵션)

345 페이지 “대상 구성 등록 정보(-o 옵션)”는 자주 사용하는 대상 구성 등록 정보를 나열합니다. 대상 구성 등록 정보 및 설명의 전체 목록은 15 장을 참조하십시오.

표 D–2 대상 구성 등록 정보(-o 옵션)

등록 정보	주
consumerFlowLimit	무제한으로 설정하려면 -1 지정
isLocalOnly(생성 전용)	
limitBehavior	FLOW_CONTROLREMOVE_OLDESTREJECT_NEWESTREMOVE_LOW_PRIORITY

표 D–2 대상 구성 등록 정보(-o 옵션) *(계속)*

등록 정보	주
localDeliveryPreferred(대기열 전용)	
maxNumActiveConsumers(대기열 전용)	무제한으로 설정하려면 -1 지정
maxNumBackupConsumers(대기열 전용)	무제한으로 설정하려면 -1 지정
maxBytesPerMsg	무제한으로 설정하려면 -1 지정
maxNumMsgs	무제한으로 설정하려면 -1 지정
maxNumProducers	무제한으로 설정하려면 -1 지정
maxTotalMsgBytes	무제한으로 설정하려면 -1 지정
useDMQ	

메트릭

```
imqcmd metrics bkr -m cxn|rts|ttl -int 5 -msp 20
imqcmd metrics svc -m cxn|rts|ttl
imqcmd metrics dst -m con|dsk|rts|ttl
```

색인

A

acknowledgeMode 활성화 사양 속성, 301
ActivationSpec JavaBean, 300
addressList 관리 대상 연결 팩토리 속성, 299
addressList 자원 어댑터 속성, 298
addressList 활성화 사양 속성, 300
addressListBehavior 관리 대상 연결 팩토리 속성, 299
addressListBehavior 자원 어댑터 속성, 298
addressListIterations 관리 대상 연결 팩토리 속성, 299
addressListIterations 자원 어댑터 속성, 298
admin 그룹, 130
admin 사용자, 128, 132, 134-135
ADMIN 서비스 유형, 76
admin 연결 서비스, 76, 103
anonymous 그룹, 130
API 설명서, 316, 317, 318
AUTOSTART 등록 정보, 70

C

clientID 관리 대상 연결 팩토리 속성, 299
clientID 활성화 사양 속성, 300, 301
config.properties 파일, 89, 174, 175, 176
connectionURL 자원 어댑터 속성, 298
customAcknowledgeMode 활성화 사양 속성, 301

D

default.properties 파일, 88
destination 활성화 사양 속성, 300
destinationType 활성화 사양 속성, 300
durable subscriptions, destroying, 109

E

endpointExceptionRedeliveryAttempts 활성화 사양 속성, 301
/etc/hosts 파일(Linux), 174

G

guest 사용자, 128

H

hosts 파일(Linux), 174
HTTP
  연결 서비스
    참조 httpjms 연결 서비스
  전송 드라이버, 323
  지원 구조, 323-324
  프록시, 323
HTTP 연결
  다중 브로커,, 331
  요청 간격, 330
  지원, 323

HTTP 연결 (**계속**)
터널 서블릿
참조 HTTP 터널 서블릿
HTTP 터널 서블릿
배포, 325-328
정보, 324
httpjms 연결 서비스
구성, 328-330
설정, 325-332
정보, 76, 103
HTTPS
연결 서비스
참조 httpsjms 연결 서비스
지원 구조, 323-324
HTTPS 연결
다중 브로커,, 341
요청 간격, 338
지원, 323
터널 서블릿
참조 HTTPS 터널 서블릿
HTTPS 터널 서블릿
배포, 334-337
정보, 324
httpsjms 연결 서비스, 141
구성, 337-339
설정, 332-341
정보, 76, 103

I

imq.accesscontrol.enabled 등록 정보, 82, 268, 277
imq.accesscontrol.file.filename 등록 정보, 83, 268, 277
imq.audit.enabled 등록 정보, 85, 152, 271, 277
imq.authentication.basic.user_repository 등록 정보, 83, 269, 277
imq.authentication.client.response.timeout 등록 정보, 84, 269, 277
imq.authentication.type 등록 정보, 84, 269, 277
imq.autocreate.destination.isLocalOnly 등록 정보, 262, 277
imq.autocreate.destination.limitBehavior 등록 정보, 261, 277
imq.autocreate.destination.maxBytesPerMsg 등록 정보, 261, 277
imq.autocreate.destination.maxCount 등록 정보, 261, 277
imq.autocreate.destination.maxNumMsgs 등록 정보, 261, 277
imq.autocreate.destination.maxNumProducers 등록 정보, 262, 277
imq.autocreate.destination.maxTotalMsgBytes 등록 정보, 261, 263, 277
imq.autocreate.destination.useDMQ 등록 정보, 123
imq.autocreate.queue.consumerFlowLimit 등록 정보, 262, 277
imq.autocreate.queue.localDeliveryPreferred 등록 정보, 263, 277
imq.autocreate.queue.maxNumActiveConsumers 등록 정보, 99, 262, 277
imq.autocreate.queue.maxNumBackupConsumers 등록 정보, 99, 262, 277
imq.autocreate.queue 등록 정보, 99, 260, 277
imq.autocreate.topic 등록 정보, 99, 260, 277
imq.cluster.brokerlist 등록 정보, 171, 173, 174, 175, 176, 275, 277
imq.cluster.masterbroker 등록 정보, 171, 175, 176, 276, 277
imq.cluster.port 등록 정보, 172, 276, 277
imq.cluster.transport 등록 정보, 172, 174, 175, 276, 277
imq.cluster.url 등록 정보, 99, 171, 172, 173, 174, 175, 176, 276, 277
imq.destination.DMQ.truncateBody 등록 정보, 79, 99, 260, 277
imq.destination.logDeadMsgs 등록 정보, 86, 99, 272, 278
imq.hostname 등록 정보, 77, 257, 278
imq.httpjms.http.servletHost property, 329
imq.imqcmd.password 등록 정보, 84, 269, 278
imq.keystore.file.dirpath 등록 정보, 143, 271, 278
imq.keystore.file.name 등록 정보, 144, 278
imq.keystore.password 등록 정보, 85, 144, 151, 278
imq.keystore.*property_name* 등록 정보, 278
imq.log.console.output 등록 정보, 86, 272, 278
imq.log.console.stream 등록 정보, 86, 278
imq.log.file.dirpath 등록 정보, 86, 272, 278
imq.log.file.filename 등록 정보, 86, 272, 278
imq.log.file.output 등록 정보, 86, 273, 278

imq.log.file.rolloverbytes 등록 정보, 86, 99, 273, 278
imq.log.file.rolloversecs 등록 정보, 86, 99, 273, 278
imq.log.level 등록 정보, 86, 99, 272, 278
imq.log.syslog.facility 등록 정보, 274, 278
imq.log.syslog.identity 등록 정보, 274, 278
imq.log.syslog.logconsole 등록 정보, 274, 278
imq.log.syslog.logpid 등록 정보, 274, 278
imq.log.syslog.output 등록 정보, 86, 273, 278
imq.log.timezone 등록 정보, 274, 278
imq.message.expiration.interval 등록 정보, 78, 260, 278
imq.message.max_size 등록 정보, 78, 99, 260, 278
imq.metrics.enabled 등록 정보, 86, 274, 278
imq.metrics.interval 등록 정보, 86, 275, 278
imq.metrics.topic.enabled 등록 정보, 87, 275, 278
imq.metrics.topic.interval 등록 정보, 87, 275, 278
imq.metrics.topic.persist 등록 정보, 87, 275, 278
imq.metrics.topic.timetolive 등록 정보, 87, 275, 279
imq.passfile.dirpath 등록 정보, 84, 269, 279
imq.passfile.enabled 등록 정보, 84, 269, 279
imq.passfile.name 등록 정보, 84, 269, 279
imq.persist.file.destination.message.filepool.limit 등록 정보, 81, 264, 279
imq.persist.file.message.cleanup 등록 정보, 81, 264, 279
imq.persist.file.message.filepool.cleanratio 등록 정보, 81, 264, 279
imq.persist.file.message.max_record_size 등록 정보, 264, 279
imq.persist.file.message.vrfile.max_record_size 등록 정보, 80
imq.persist.file.sync.enabled 등록 정보, 81, 265, 279
  Sun Cluster 요구 사항, 265, 279
imq.persist.file.sync 등록 정보, 91
imq.persist.jdbc.brokerid 등록 정보, 82, 265, 279
imq.persist.jdbc.closedburl 등록 정보, 81, 266, 279
imq.persist.jdbc.createdburl 등록 정보, 81, 266, 279
imq.persist.jdbc.driver 등록 정보, 81, 265, 279
imq.persist.jdbc.needpassword 등록 정보, 82, 266, 279
imq.persist.jdbc.opendburl 등록 정보, 81, 266, 279
imq.persist.jdbc.password 등록 정보, 82, 151, 266, 279
imq.persist.jdbc.table.IMQCCREC35 등록 정보, 82, 266, 279
imq.persist.jdbc.table.IMQDEST35 등록 정보, 82, 267, 279
imq.persist.jdbc.table.IMQINT35 등록 정보, 267, 279
imq.persist.jdbc.table.IMQLIST35 등록 정보, 267, 279
imq.persist.jdbc.table.IMQMSG35 등록 정보, 267, 279
imq.persist.jdbc.table.IMQPROPS35 등록 정보, 267, 279
imq.persist.jdbc.table.IMQSV35 등록 정보, 82, 266, 279
imq.persist.jdbc.table.IMQTACK35 등록 정보, 267, 279
imq.persist.jdbc.table.IMQTXN35 등록 정보, 267, 279
imq.persist.jdbc.user 등록 정보, 82, 266, 279
imq.persist.store 등록 정보, 80, 92, 263
imq.ping.interval 등록 정보, 78, 259, 279
imq.portmapper.backlog 등록 정보, 77, 258, 280
imq.portmapper.hostname 등록 정보, 77, 257, 280
imq.portmapper.port 등록 정보, 76, 99, 258, 280
imq.protocol.*protocolType*.inbufsz, 207
imq.protocol.*protocolType*.nodelay, 207
imq.protocol.*protocolType*.outbufsz, 207
imq.*resource_state*.count 등록 정보, 260, 280
imq.*resource_state*.threshold 등록 정보, 260, 280
imq.*resourceState*.count 등록 정보, 79
imq.service.activelist 등록 정보, 76, 257, 280
imq.*service_name*.accesscontrol.enabled 등록 정보, 280
imq.*service_name*.accesscontrol.file.filename 등록 정보, 268, 280
imq.*service_name*.authentication.type 등록 정보, 269, 280
imq.*service_name*.max_threads 등록 정보, 258, 280
imq.*service_name*.min_threads 등록 정보, 258, 280
imq.*service_name.protocol_type*.hostname 등록 정보, 172, 258, 276, 277, 280
imq.*service_name.protocol_type*.port 등록 정보, 258, 280
imq.*service_name*.threadpool_model 등록 정보, 258, 280
imq.*serviceName*.accesscontrol.enabled 등록 정보, 82
imq.*serviceName*.accesscontrol.file.filename 등록 정보, 83
imq.*serviceName*.authentication.type 등록 정보, 84
imq.*serviceName*.max_threads 등록 정보, 77
imq.*serviceName*.min_threads 등록 정보, 77

imq.*serviceName.protocolType*.hostname 등록 정보, 77
imq.*serviceName.protocolType*.port 등록 정보, 77
imq.*serviceName*.threadpool_model 등록 정보, 77
imq.shared.connectionMonitor_limit 등록 정보, 78, 259, 280
imq.system.max_count 등록 정보, 78, 99, 259, 280
imq.system.max_size 등록 정보, 78, 99, 259, 280
imq.transaction.autorollback 등록 정보, 111, 260, 280
imq.user_repository.ldap.base 등록 정보, 270, 280
imq.user_repository.ldap.gidattr 등록 정보, 271, 280
imq.user_repository.ldap.grpbase 등록 정보, 271, 280
imq.user_repository.ldap.grpfilter 등록 정보, 271, 280
imq.user_repository.ldap.grpsearch 등록 정보, 270, 280
imq.user_repository.ldap.memattr 등록 정보, 271, 280
imq.user_repository.ldap.password 등록 정보, 84, 151, 270, 280
imq.user_repository.ldap.principal 등록 정보, 84, 270, 280
imq.user_repository.ldap.*property_name* 등록 정보, 270, 280
imq.user_repository.ldap.server 등록 정보, 84, 269, 280
imq.user_repository.ldap.ssl.enabled 등록 정보, 271, 281
imq.user_repository.ldap.timeout 등록 정보, 271, 281
imq.user_repository.ldap.uidattr 등록 정보, 270, 281
imq.user_repository.ldap.usrfilter 등록 정보, 270, 281
imqAckTimeout 속성, 291
imqAddressList 속성, 288
imqAddressListBehavior 속성, 288
imqAddressListIterations 속성, 288
imqbrokerd command, restoring configuration change record, 177
imqbrokerd.conf 파일, 69, 73
imqbrokerd 명령, 68
  구성 변경 기록 백업, 177
  구성 파일(Solaris, Linux), 69, 73
  데이터 저장소 지우기, 91, 120
  로깅 등록 정보 설정, 182
  브로커 연결, 173
  브로커 제거, 72
imqbrokerd 명령 *(**계속**)*
  비밀번호 파일, 150
  옵션, 238-242
  인수 전달, 90
  정보, 35
  참조, 238
  클러스터에 브로커 추가, 175
  클러스터에서 브로커 제거, 175
imqcmd 명령
  마스터 브로커에 종속, 177
  메트릭 모니터링, 186-190
  물리적 대상 관리, 113
  물리적 대상 하위 명령(표), 114
  브로커에 대한 보안 연결, 146
  브로커에 보안 연결, 249
  비밀번호 파일, 150
  영구 가입 하위 명령, 108
  일반 옵션, 249, 254
  정보, 35
  참조, 242
  트랜잭션 관리, 109
imqConfiguredClientID 속성, 291
imqConnectionFlowCount 속성, 291
imqConnectionFlowLimit 속성, 292
imqConnectionFlowLimitEnabled 속성, 291
imqConsumerFlowLimit 속성, 292
imqConsumerFlowThreshold 속성, 292
imqdbmgr 명령
  비밀번호 파일, 150
  옵션, 252-253
  정보, 36
  참조, 251
imqDefaultPassword 속성, 291
imqDefaultUsername 속성, 291
imqDestinationDescription 속성, 295
imqDestinationName 속성, 295
imqDisableSetClientID 속성, 291
imqFlowControlLimit 속성, 292
imqJMSDeliveryMode 속성, 294
imqJMSExpiration 속성, 294
imqJMSPriority 속성, 162, 294
imqkeytool 명령
  명령 구문, 142, 333
  사용, 142

imqkeytool 명령 *(계속)*
정보, 36
참조, 255
imqLoadMaxToServerSession 속성, 161, 293
imqobjmgr 명령
옵션, 250
정보, 36
참조, 250
하위 명령, 250
imqOverrideJMSDeliveryMode 속성, 294
imqOverrideJMSExpiration 속성, 294
imqOverrideJMSHeadersToTemporaryDestinations 속성, 162, 294
imqOverrideJMSPriority 속성, 162, 294
imqQueueBrowserMax MessagesPerRetrieve 속성, 161, 293
imqQueueBrowserRetrieveTimeout 속성, 161, 293
imqReconnectAttempts 속성, 288
imqReconnectEnabled 속성, 288
imqReconnectInterval 속성, 288
imqSetJMSXAppID 속성, 293
imqSetJMSXConsumerTXID 속성, 293
imqSetJMSXProducerTXID 속성, 293
imqSetJMSXRcvTimestamp 속성, 294
imqSetJMSXUserID 속성, 293
imqSSLIsHostTrusted 속성, 289
imqsvcadmin 명령
옵션, 254
정보, 36
참조, 254
하위 명령, 254
imqusermgr 명령
비밀번호, 131
사용자 이름, 131
옵션, 253
용도, 129
정보, 36
참조, 253
하위 명령, 253
install.properties 파일, 88

J

J2EE Connector Architecture(JCA), 297
java.naming.factory.initial 속성, 154, 155
java.naming.provider.url 속성, 154, 156
java.naming.security.authentication 속성, 155
java.naming.security.credentials 속성, 155
java.naming.security.principal 속성, 154
Java Runtime
Windows 서비스용, 71
경로 지정, 240, 250, 251, 254
Java 가상 머신, **참조** JVM
javahome 옵션, 71
JCA(J2EE Connector Architecture), 297
JDBC 기반 영구, 성능 조정, 209
JDBC 기반 지속성
설정, 91-93
정보, 81-82
JDBC 지원
구성, 91
드라이버, 265
정보, 81-82
jms 연결 서비스, 76, 103
JMSDeliveryMode 메시지 헤더 필드, 162
JMSExpiration 메시지 헤더 필드, 162
JMSPriority 메시지 헤더 필드, 162
JNDI
객체 저장소, 36, 153
객체 저장소 속성, 154-155, 163
조회, 53
조회 이름, 163, 164
jrehome 옵션, 71
JVM
메트릭
참조 JVM 메트릭
성능 조정, 206-207
성능 효과, 203
JVM 메트릭
imqcmd metrics 사용, 187
메시지 기반 모니터링 사용, 190
메트릭 개수, 303-304
브로커 로그 파일 사용, 185

L

LDAP 서버
객체 저장소 속성, 154

LDAP 서버 *(**계속**)*
사용자 저장소, 133
사용자 저장소 액세스, 133
인증 페일오버, 134

M
ManagedConnectionFactory JavaBean, 299
MDB, **참조** message-driven beans
message-driven beans
자원 어댑터 구성, 297, 300
messageSelector 활성화 사양 속성, 300

N
NORMAL 서비스 유형, 76
nsswitch.conf 파일(Linux), 174

O
Oracle, 92, 94

P
password 관리 대상 연결 팩토리 속성, 299
password 자원 어댑터 속성, 298
PointBase, 92

R
reconnectAttempts 관리 대상 연결 팩토리 속성, 299
reconnectAttempts 자원 어댑터 속성, 298
reconnectEnabled 관리 대상 연결 팩토리 속성, 299
reconnectEnabled 자원 어댑터 속성, 298
reconnectInterval 관리 대상 연결 팩토리 속성, 299
reconnectInterval 자원 어댑터 속성, 298
ResourceAdapter JavaBean, 297
RESTART 등록 정보, 70

S
Secure Socket Layer 표준, **참조** SSL
sendUndeliverableMsgsToDMQ 활성화 사양 속성, 301
Simple Network Time Protocol, 67
SNTP, 67
SSL
암호화 및, 141-149
연결 서비스
참조 SSL 기반 연결 서비스
정보, 85
활성화, 144
SSL 기반 연결 서비스
설정, 141, 142
시작, 145
ssladmin 연결 서비스, 141
정보, 76, 103
ssljms 연결 서비스, 141
정보, 76, 103
subscriptionDurability 활성화 사양 속성, 300
subscriptionName 활성화 사양 속성, 300
Sun Cluster, 구성, 265
syslog, 86, 183

T
TCP, 76, 103
TLS, 76, 103

U
ulimit 명령, 68
update dst 하위 명령, 제한, 118
user 그룹, 130
할당 삭제, 130
userName 관리 대상 연결 팩토리 속성, 299
userName 자원 어댑터 속성, 298

W
W32Time 서비스, 67
Windows 서비스, **참조** 서비스(Windows)

X
xntpd 데몬, 67

감
감사 기록, 152

개
개발 환경 관리 작업, 33-34

객
객체 저장소, 153-156
  LDAP 서버, 153-155
  LDAP 서버 속성, 154
  위치, 315, 316, 317
  파일 시스템, 155-156
  파일 시스템 저장소 속성, 155
객체 저장소의 위치, 154

관
관리 대상 객체
  XA 연결 팩토리
    참조 연결 팩토리 관리 대상 객체
  객체 저장소
    참조 객체 저장소
  관리, 153
  나열, 166
  대기열
    참조 대기열
  삭제, 165
  속성(참조), 287
  업데이트, 167
  조회, 166
  주제
    참조 주제
  필수 정보, 163
관리 도구, 35-37
  관리 콘솔, 36
관리 도구 (**계속**)
  명령줄 유틸리티, 35
관리 작업
  개발 환경, 33-34
  작업 환경, 34-35
관리 콘솔
  시작, 40
  자습서, 39
관리자 비밀번호, 132

구
구성 등록 정보의 우선 순위, 88
구성 변경 기록, 176
  백업, 177
  복원, 177
구성 파일, 88
  기본값, 88
  브로커(그림), 88
  설치, 88
  위치, 315, 316, 317
  인스턴스, 89, 172, 315, 316, 317
  클러스터, 171, 172, 173, 276
  템플리트, 315, 316, 317
  템플리트 위치, 315, 316, 317
  편집, 89-90

권
권한
  계산, 137-138
  관리 서비스, 84
  내장 데이터베이스, 93
  데이터 저장소, 80
  사용자 저장소, 129, 253
  액세스 제어 등록 정보 파일, 84
  키 저장소, 333
권한 부여
  참조 액세스 제어
  관리, 135-141
  사용자 그룹, 84
  정보, 84-85

능
능 요소, JVM, 203

다
다시 시작
  물리적 대상, 119
  브로커, 100-101, 101, 244
  연결 서비스, 107, 245

대
대기열
  관리 대상 객체 추가, 165
  자동 생성, 260, 277
대기열 로드 균형 조정 전달
  등록 정보, 116, 262, 284
대상 관리 객체, 속성, 162
대상 메트릭
  imqcmd metrics 사용, 187, 189
  imqcmd query 사용, 190
  메시지 기반 모니터링 사용, 190
  메트릭 개수, 307-311
대상 삭제, 120-121
대체
  메시지 헤더, 162
  명령줄, 73

데
데이터 저장소
  JDBC 호환, 81-82
  구성, 91
  내용, 91
  디스크에 동기화, 91
  성능 효과, 205
  압축, 80
  위치, 315, 316, 317
  재설정, 239
  정보, 79
  플랫 파일, 80-81

도
도구, 관리, **참조** 관리 도구
도움말(명령줄), 250

동
동기화, 클럭, 67

등
등록 정보
  JDBC 관련, 89, 265-267
  구문, 90
  로거, 272-275
  루틴 서비스, 259-260
  메모리 관리, 115
  물리적 대상
    참조 물리적 대상, 등록 정보
  보안, 268-271
  브로커 모니터링 서비스, 272-275
  브로커 인스턴스 구성, 89
  연결 서비스, 257-259
  자동 생성, 260-263
  지속성, 264-265
  클러스터 구성, 275-276

디
디스크 공간
  물리적 대상 사용률, 121
  확보, 122

라
라우팅 서비스, 등록 정보, 259-260
라우팅 서비스, 브로커, 75, 78

로

로거
  구성 변경, 182
  등록 정보 설정, 182
  로그 메시지 리디렉션, 183
  롤오버 기준, 184
  메시지 형식, 182
  메트릭 정보, 274
  범주, 181
  사용 불능 메시지 형식, 185
  수준, 86, 181, 241, 272, 278
  정보, 86-87
  출력 채널, 86, 181, 183-184
  콘솔에 기록, 242
  콘솔에 쓰기, 86, 272, 278
로그 파일
  기본 위치, 315, 316, 317
  기본 위치 변경, 181
  기본 이름 변경, 181
  등록 정보 설정, 182
  롤오버 기준, 86, 184, 273, 278
  롤오버 빈도, 181
  메트릭 보고, 184-185
  사용 불능 메시지 기록, 185-186
  이름, 181
로깅, **참조** 로거
로드 균형 조정 대기열 전달, 성능 조정, 210

루

루프백 주소, 174

마

마스터 브로커
  구성 변경 기록, 176
  사용 불가, 177
  지정, 171, 172

메

메모리 관리
  물리적 대상 등록 정보 사용, 115
  브로커용, 78
  성능 조정, 209
메시지
  단편화, 80
  대상 제한, 261, 284
  만료 재생 이용, 78, 260, 278
  메트릭 메시지
    참조 메트릭 메시지
  물리적 대상 제한, 115
  물리적 대상에서 제거, 120, 247
  본문 유형 및 메시지, 202
  브로커 제한, 78, 99, 259, 280
  안정적인 전달, 160-161
  지속성, 79
  크기 및 성능, 201-202
  흐름 일시 중지, 119
  흐름 제어
    참조 메시지 흐름 제어
메시지 단편화, 80
메시지 만료, 클럭 동기화 및, 67
메시지 서비스 구조, 204-205
메시지 서비스 성능, 202-205
메시지 재설정 옵션, 120
메시지 헤더 대체, 162
메시지 흐름 제어
  브로커, 78, 115
  성능 조정, 210-211
  성능 효과, 210
  속성, 160-161
  제한, 211-212
  측정, 210-211
메트릭
  데이터
    참조 메트릭 데이터
  메시지
    참조 메트릭 메시지
  정보, 86
  주제 대상, 87, 190-191
메트릭 데이터
  imqcmd metrics 사용, 188
  메시지 기반 모니터링 API 사용, 191-192

메트릭 데이터 *(**계속**)*
물리적 대상
참조 물리적 대상 메트릭
브로커
참조 브로커 메트릭
브로커 로그 파일 사용, 184-185
연결 서비스
참조 연결 서비스 메트릭
메트릭 메시지
유형, 87, 191
정보, 190
메트릭 모니터링 도구
Message Queue 로그 파일, 184-185
Message Queue 명령 유틸리티(imqcmd), 186-190
메시지 기반 모니터링 API, 190-192
비교, 179-180

명

명령 옵션, 구성 무시, 73
명령 파일, 167
명령줄 구문, 237
명령줄 유틸리티
imqbrokerd, **참조**, imqbrokerd 명령, 35
imqcmd, **참조**, imqcmd 명령, 35
imqdbmgr **참조**, imqdbmgr 명령, 36
imqkeytool, **참조**, imqkeytool 명령, 36
imqobjmgr, **참조**, imqobjmgr 명령, 36
imqsvcadmin, **참조**, imqsvcadmin 명령, 36
imqusermgr, **참조**, imqusermgr 명령, 36
기본 구문, 237
도움말, 250
버전 표시, 250
정보, 35

모

모니터링, **참조** 성능 모니터링
모니터링 서비스, 브로커, 75, 85-87
모든 명령 구문, 237-238

문

문제 해결, 213
Windows 서비스 시작, 72

물

물리적 대상
관리, 113
나열, 116-117
다시 시작, 119
등록 정보, 283-285
등록 정보 값, 117
등록 정보 값 표시, 117-118
등록 정보 업데이트, 118
디스크 공간 확보, 122
디스크 사용률, 121-123
만들기, 115-116
메시지 일시 중지, 120
메시지 전달 일괄 처리, 116, 262, 285
메시지 제거, 120, 247
메트릭
참조 물리적 대상 메트릭
사용 불능 메시지 대기열, 123-125
사용 불능 메시지 대기열 사용, 123
삭제, 120-121
속성 업데이트, 247
압축, 121-123
유형, 116, 246
일시 중지, 119, 247
임시, 116
자동 생성, 140
정보, 117-118
정보 가져오기, 248
정보 얻기, 117
제한 동작, 115, 284
클러스터의 제한된 범위, 116, 262, 285
파일 기반 데이터 저장소 압축, 122, 247
물리적 대상 속성, 283-285
물리적 대상 완전 삭제, 120-121
물리적 대상 자동 생성
등록 정보(표), 260-263
액세스 제어, 85

방
방화벽, 323

버
버전, 250

벤
벤치마크, 성능, 194-195

병
병목 현상, 성능, 198

보
보안
  객체 저장소,, 154
  관리자
    참조 보안 관리자
  권한 부여
    참조 권한 부여
  암호화
    참조 암호화
  인증
    참조 인증
보안 관리자
  등록 정보, 268-271
  정보, 82
보안 서비스, 브로커, 75, 82-85

분
분산 트랜잭션, XA 자원 관리자, 109

브
브로커
  HTTP 지원, 325
  HTTPS 지원, 332
  SSL을 사용하여 시작, 145
  Windows 서비스로 실행, 70
  관리, 95
  구성 파일
    참조 구성 파일
  다시 시작, 79, 100-101, 101, 244
  등록 정보 업데이트, 99-100
  등록 정보(참조), 257, 283
  등록 정보 표시, 98-99
  로깅
    참조 로거
  메모리 관리, 78, 115, 205
  메시지 용량, 78, 99, 259, 280
  메시지 흐름 제어
    참조 메시지 흐름 제어
  메트릭
    참조 브로커 메트릭
  물리적 대상 자동 생성 등록 정보, 260-263
  사용 불능 메시지 대기열, 123
  상호 연결
    참조 브로커 클러스터
  시작하는 데 필요한 사용 권한, 68
  액세스 제어
    참조 권한 부여
  연결, 173
  연결 서비스 나열, 103
  오류 복구, 79
  인스턴스 구성 등록 정보, 89
  인스턴스 이름, 238
  일시 중지, 100-101, 244
  자동 다시 시작, 70
  제거, 72
  제한 동작, 78, 205
  종료, 101
  쿼리, 98
  클럭 동기화, 67
브로커 Windows 서비스의 시작 매개 변수, 71
브로커 구성 요소
  라우팅 서비스, 75, 78
  모니터링 서비스, 75, 85-87

브로커 구성 요소 *(계속)*
보안 서비스, 75, 82-85
연결 서비스, 75, 76
지속성 서비스, 75, 79-82
브로커 다시 시작, 101, 244
브로커 메트릭
imqcmd 사용, 102, 188, 190
로거 등록 정보, 86, 184, 274
메시지 기반 모니터링 사용, 190
메트릭 개수(표), 304-305
메트릭 메시지, 87
보고 간격, 로거, 241
브로커 로그 파일 사용, 185
브로커 모니터링 서비스, 등록 정보, 272-275
브로커 연결, 173
브로커 오류 및 보안 연결, 342
브로커 응답, 클라이언트를 위한 대기 시간, 291
브로커 종료, 101, 244
Windows 서비스, 73
브로커 종류, 101
브로커 클러스터
구성 등록 정보, 171, 275-276
구성 변경 기록, 176
구성 파일, 171, 172, 173, 276
구조, 204-205
물리적 대상 복제, 116
물리적 대상 일시 중지, 119
브로커 간 보안 연결, 174
브로커 연결, 173
브로커 추가, 174
사용 이유, 204
성능 효과, 205

비

비동기, 디스크에 메모리, 91
비밀번호
JDBC, 151
LDAP, 151
SSL 키 저장소, 144, 151, 240
관리자, 132
기본값, 291
비밀번호 파일
참조 비밀번호 파일
비밀번호 *(계속)*
이름 지정 규약, 131
인코딩, 269
비밀번호 파일
명령줄 옵션, 240
브로커 구성 등록 정보, 84, 269
사용, 150-151
위치, 151, 316, 317, 318

사

사용 권한
비밀번호 파일, 150
액세스 제어 등록 정보 파일, 136
사용 도움말, 250
사용 불능 메시지
참조 사용 불능 메시지 대기열
로깅, 86
사용 불능 메시지 대기열
maxNumMsgs 값, 124
maxTotalMsgBytes 값, 124
구성, 123-125
로깅, 86, 125
제한 동작, 124
사용권, 시작 옵션, 241
사용률, 122
사용자 그룹
기본값, 84
사전 정의됨, 130
사용자 이름, 291
기본값, 128
형식, 131
사용자 저장소
LDAP, 133
LDAP 서버, 133
관리, 131
사용자 그룹, 131
사용자 상태, 131
위치, 316, 317
정보, 82
채우기, 131
초기 항목, 128
플랫 파일, 128
플랫폼 종속성, 129, 253

삭
삭제, 브로커 인스턴스, 72

생
생성자
  대상 제한, 262, 284
  물리적 대상 제한, 115

서
서버 오류 및 보안 연결, 342
서비스(Windows)
  Java Runtime, 71
  브로커 실행, 70
  브로커 제거, 73
  시작 매개 변수, 71
  시작 문제 해결, 72
  재구성, 70
서비스 유형
  ADMIN, 76
  NORMAL, 76

선
선택기
  성능 효과, 201
  정보, 201

성
성능
  기본 패턴, 195-196
  모니터링
    참조 성능 모니터링
  문제 해결, 213
  벤치마크, 194-195
  병목 현상, 198
  안정성 균형, 198
  영향을 미치는 요소
    참조 성능 요소
성능 (**계속**)
  정보, 193-196
  조정
    참조 성능 조정
  최적화
    참조 성능 조정
  측정, 194
  표시기, 194
성능 모니터링
  도구
    참조 메트릭 모니터링 도구
  메트릭 데이터
    참조 메트릭 데이터
성능 요소
  데이터 저장소, 205
  메시지 본문 유형, 202
  메시지 서비스 구조, 205
  메시지 크기, 201-202
  메시지 흐름 제어, 210
  브로커 제한 동작, 205
  선택기, 201
  연결, 203-204
  영구 가입, 201
  운영 체제, 203
  전달 모드, 199
  전송 프로토콜, 203-204
  트랜잭션, 199-200
  파일 동기화, 265, 279
  하드웨어, 202
  확인 모드, 200
성능 조정
  브로커 조정, 209-210
  시스템 조정, 206-209
  클라이언트 런타임 조정, 210-212
  프로세스 개요, 193-194

수
수명, **참조** 메시지 만료

스
스레드 풀 관리
  공유 스레드, 77
  전용 스레드, 77
  정보, 77

시
시간 동기화 서비스, 67
시스템 클럭 동기화, 67
시작
  SSL 기반 연결 서비스, 145
  클라이언트, 73

안
안정적인 전달, 160-161
  성능 균형, 198

암
암호화
  SSL 기반 서비스, 141-149
  정보, 82, 85
  키 도구 및, 85

압
압축
  물리적 대상, 121-123
  파일 기반 데이터 저장소, 80

액
액세스 규칙, 137
액세스 제어 파일
  버전, 136
  액세스 규칙, 137
  용도, 135
  위치, 316, 317
액세스 제어 파일 *(**계속**)*
  형식, 136

업
업데이트
  브로커, 99-100
  연결 서비스, 105, 108, 245

연
연결
  나열, 107, 246
  서버 또는 브로커 오류, 342
  성능 효과, 203-204
  자동 재연결
    참조 자동 재연결
  쿼리, 108, 246
  파일 설명자 제한에 의해 제한됨, 68
  페일오버
    참조 자동 재연결
연결 서비스
  admin, 76, 103
  HTTP
    참조 HTTP 연결
  httpjms, 76, 103
  HTTPS
    참조 HTTPS 연결
  httpsjms, 76, 103
    참조 httpsjms 연결 서비스
  jms, 76, 103
  SSL 기반, 144
  ssladmin
    참조 ssladmin 연결 서비스
  ssljms
    참조 ssljms 연결 서비스
  다시 시작, 107, 245
  등록 정보, 105, 257-259
  등록 정보 표시, 104
  메트릭 데이터
    참조 연결 서비스 메트릭
  서비스 유형, 76
  스레드 풀 관리, 77

연결 서비스 *(계속)*
스레드 할당, 105
시작 시 활성화, 257
액세스 제어, 82, 268
업데이트, 105, 108, 245
영향을 주는 명령, 245-246
일시 중지, 106, 245
큐리, 104, 108
클러스터, 174
참조 클러스터 연결 서비스
포트 매퍼
참조 포트 매퍼
프로토콜 유형, 76
연결 서비스, 브로커, 75, 76
연결 서비스 메트릭
imqcmd metrics 사용, 188-189
imqcmd query 사용, 190
imqcmd 메트릭 사용, 105-106
메트릭 개수, 305-307
연결 팩토리 관리 대상 객체
JMS 등록 정보 지원 속성, 161-162
대기열 브라우저 동작 속성, 161, 292-293
메시지 헤더 필드 대체, 162
속성, 156-162
안정성 및 흐름 제어 속성, 160-161
연결 처리 속성, 157-159
응용 프로그램 서버 지원 속성, 293
클라이언트 아이디 속성, 159-160
표준 메시지 등록 정보, 293-294

영
영구
데이터 저장소
참조 데이터 저장소
영구 가입
관리, 108
나열, 108, 248
메시지 제거, 248
성능 효과, 201
완전 삭제, 109, 248

예
예제 응용 프로그램, 316, 317, 318

운
운영 체제
Solaris 성능 조정, 206
성능 효과, 203

인
인스턴스 구성 파일, **참조** 구성 파일
인스턴스 디렉토리
및 인스턴스 구성 파일, 133
및 파일 기반 데이터 저장소, 91
제거, 72
인증
참조 액세스 제어
관리, 127-135
정보, 84
인증서, 141-146, 332

일
일시 중지
물리적 대상, 119, 247
브로커, 100-101, 244
연결 서비스, 106, 245

임
임시 물리적 대상, 116

자
자동 생성 물리적 대상, 액세스 제어, 140-141
자동 재연결
속성, 158
제한, 158
자습서, 39

자원 어댑터, 297
  재연결, 298, 299
자체 서명된 인증서, 141-146, 332

작
작업 환경
  관리 작업, 34-35
  설정, 34-35
  유지 관리, 35

재
재연결, 자동, **참조** 자동 재연결

전
전달 모드, 성능 효과, 199
전송 프로토콜
  상대 속도, 203
  성능 조정, 207-209
  성능 효과, 203-204
  프로토콜 유형
    참조 프로토콜 유형

제
제거
  물리적 대상, 120-121
  브로커, 72
제거, 물리적 대상에서 메시지, 120
제품 버전 표시, 250
제한 동작
  물리적 대상, 115, 284
  브로커, 78

주
주제
  관리 대상 객체 추가, 164
주제 *(**계속**)*
  자동 생성, 260, 277

지
지속성
  JDBC
    참조 JDBC 지속성
  JDBC 기반
    참조 JDBC 기반 지속성
  등록 정보, 264-265
  보안, 93
  옵션(그림), 80
  정보, 79
  파일 기반, 80-81
지속성 서비스, 브로커, 75, 79-82

쿼
쿼리
  브로커, 98-99
  연결 서비스, 104, 108

클
클라이언트
  시작, 73
  클럭 동기화, 67
클라이언트 런타임
  구성, 205
  메시지 흐름 조정, 210-211
클라이언트 식별자(ClientID), 159-160
  영구 가입 완전 삭제, 109
클라이언트 응용 프로그램
  성능에 영향을 미치는 요소, 198-202
  예제, 316, 317, 318
클러스터
  참조 브로커 클러스터
  클러스터
    참조 브로커 클러스터
클러스터 구성 등록 정보, 171, 275
클러스터 구성 파일, 171, 172, 173, 276

클러스터 연결 서비스, 141, 174
  네트워크 전송, 172, 276
  포트 번호, 172, 276
  호스트 이름 또는 IP 주소, 172, 276
클러스터용 디렉토리 조회(Linux), 174
클럭 동기화, 67

키
키 도구, 85
키 쌍
  생성, 143
  재생성, 144
키 저장소
  파일, 143, 144, 333

터
터널 서블릿 연결, 342

트
트랜잭션
  관리, 109-111
  롤백, 110, 249
  성능 효과, 199-200
  완결, 111, 249
  정보, 249

파
파일 기반 저장소에 대한 쓰기 작업, 91
파일 기반 지속성, 80-81
파일 동기화
  imq.persist.file.sync.enabled 옵션, 265, 279
  Sun Cluster와, 265, 279
파일 설명자 제한, 68
  연결 제한 및, 68

포
포트 매퍼
  정보, 76
  포트 지정, 238

프
프로토콜
  참조 전송 프로토콜
프로토콜 유형
  HTTP, 76, 103
  TCP, 76, 103
  TLS, 76, 103

하
하드웨어, 성능 효과, 202

흐
흐름 제어, **참조** 메시지 흐름 제어