

Compaq OpenVMS Alpha용 MQSeries



시스템 관리 안내서

버전 5 릴리스 1

Compaq OpenVMS Alpha용 MQSeries



시스템 관리 안내서

버전 5 릴리스 1

주!

이 정보 및 이 책이 지원하는 제품을 사용하기 전에 409 페이지의 『부록L. 주의사항』의 일반 정보를 읽어보십시오.

초판(2001년 5월)

새 버전에서 별도로 표시하지 않는 한 이 내용은 Compaq OpenVMS Alpha용 MQSeries 버전 5.1 및 이후의 모든 릴리스 및 개정 버전에 적용됩니다.

© Copyright International Business Machines Corporation 1994, 2001. All rights reserved.

목차

그림 ix

표. xi

서적 정보. xiii

이 서적의 대상. xiii

이 서적을 이해하기 위한 선결 사항 xiii

서적 사용 방법. xiii

 부록 사용. xiv

인터넷상의 MQSeries 정보. xiv

Compaq OpenVMS Alpha용 MQSeries V5.1의 새로운 기능 xv

제1부 자세한 내용 1

제1장 MQSeries 소개 7

MQSeries와 메시지 큐잉 7

 시간 독립적 응용프로그램 7

 메시지에 의한 처리 7

메시지와 큐. 8

 메시지의 정의 8

 큐의 정의 8

오브젝트 10

 오브젝트 이름 10

 오브젝트 관리 11

 MQSeries 큐 관리자 11

 MQSeries 큐. 12

 프로세스 정의 15

 채널. 16

 클러스터 16

 이름 목록. 16

시스템 디폴트 오브젝트 17

로컬 관리 및 리모트 관리 17

클라이언트 및 서버. 17

 클라이언트-서버 환경에서의 MQSeries 응용프로그램. 18

 그림. 18

큐 관리자 기능 확장 18

 사용자 종료 19

 설치가능 서비스. 19

보안. 20

 OAM(Object Authority Manager) 기능 20

 DCE 보안. 20

트랜잭션식 지원. 20

제2장 MQSeries 관리 소개. 23

로컬 관리 및 리모트 관리 23

제어 명령을 사용하여 관리 작업 수행 24

MQSC 명령을 사용하여 관리 작업 수행 24

PCF 명령을 사용하여 관리 작업 수행 25

 MQSC 및 PCF의 속성 25

 이스케이프 PCF. 25

MQSeries 파일 이름 이해 25

 큐 관리자 이름 변환 26

 오브젝트 이름 변환. 26

대소문자 구분 이해. 27

 제어 명령의 대소문자 구분 27

 MQSC 명령의 대소문자 구분 28

제3장 제어 명령을 사용한 큐 관리자 관리 29

제어 명령 사용 29

 제어 명령 사용 29

큐 관리자 작성 30

 큐 관리자 작성 지침 30

디폴트 큐 관리자 작성. 34

큐 관리자 시작 34

기존의 큐 관리자를 디폴트로 만들기 35

큐 관리자 정지 35

 정지 종료 35

 즉시 종료 36

 선제 종료 36

 큐 관리자 종료에 문제점이 있을 경우 37

큐 관리자 재시작 37

큐 관리자 삭제 37

제4장 로컬 MQSeries 오브젝트 관리 39

MQI를 사용하는 응용프로그램 지원 39

MQSC 명령을 사용한 로컬 관리 작업의 수행. 40

 시작하기 전에 41

 MQSC 기능의 대화식 사용 41

 MQSC 명령의 피드백. 42

 MQSC의 대화식 입력 종료 43

 큐 관리자 속성 표시 43

 디폴트가 아닌 큐 관리자의 사용 44

 큐 관리자 속성 대체 44

 텍스트 파일에서 MQSC 명령 실행 44

MQSC 명령 파일	45	MVS/ESA에서 큐 관리자에 대한 작업	76
MQSC 보고서	46	리모트로 MQSC를 사용할 때 문제점이 발생할 경우.	77
제공된 MQSC 명령 파일의 실행	47	리모트 큐의 로컬 정의 작성.	78
명령 확인을 위한 runmqsc의 사용.	47	리모트 큐의 로컬 정의가 작동되는 방법에 대한 이해.	78
MQSC 문제점 해결	48	리모트 큐에 메시지를 넣는 다른 방법	79
로컬 큐에 대한 작업	50	리모트 큐에서 다른 명령 사용	80
로컬 큐 정의.	50	트랜스미션 큐 작성.	80
데드-레터 큐의 정의	51	별명으로 리모트 큐 정의 사용	81
디폴트 오브젝트 속성의 표시	51	큐 관리자 별명	81
로컬 큐 정의 복사	52	리플라이-투 큐 별명	81
로컬 큐 속성 변경	53	데이터 변환	82
로컬 큐 지우기	53	큐 관리자 CCSID 변경	84
로컬 큐 삭제.	53		
큐 찾아보기	54		
알리아스 큐에 대한 작업.	57	제7장 MQSeries 오브젝트 보호	85
알리아스 큐 정의	58	MQSeries 자원을 보호해야 하는 이유	85
큐 알리아스가 있는 기타 명령의 사용	59	시작하기 전에	85
모델 큐에 대한 작업	60	자원 ID가 MQM인 Compaq OpenVMS용	
모델 큐의 정의	60	MQSeries에서의 사용자 ID	86
모델 큐에 대한 기타 명령의 사용	60	자세한 정보	86
트리거를 위한 오브젝트 관리	61	오브젝트 권한 관리자 이해	87
트리거를 위한 응용프로그램 큐 정의	61	OAM 작동 방법	87
이니시에이션 큐 정의	62	권한 ID를 통한 액세스 관리	87
프로세스 정의 작성.	62	디폴트 권한 ID	88
프로세스 정의 표시.	63	OAM으로 보호할 수 있는 자원	88
		권한에 대해 권한 ID 사용	89
제5장 관리 작업 자동화	65	오브젝트 권한 관리자 사용 안함	89
PCF 명령	65	오브젝트 권한 관리자 명령 사용	89
MQSC 및 PCF의 속성	66	OAM 명령 사용시 지정해야 할 사항.	90
이스케이프 PCF.	66	setmqaut 명령 사용	90
MQAI를 통한 간편한 PCF 사용	66	액세스 권한	92
리모트 관리를 위한 명령 서버 관리	67	권한 명령 표시	92
명령 서버 시작	67	오브젝트 권한 관리자 지침	92
명령 서버 상태 표시	68	사용자 ID.	92
명령 서버 정지	68	큐 관리자 디렉토리.	93
		큐	93
제6장 리모트 MQSeries 오브젝트 관리	69	대체 사용자 권한	93
채널, 클러스터 및 리모트 큐잉.	69	컨텍스트 권한	94
클러스터를 사용한 리모트 관리.	70	리모트 보안 고려사항	94
MQSC 명령을 사용한 로컬 큐 관리자에서의 리모트 관리.	71	채널 명령 보안	95
리모트 관리에 대한 큐 관리자 준비	71	권한 스펙 테이블에 대한 이해	96
리모트 관리를 위한 채널 및 트랜스미션 큐 준비	72	MQI 권한.	97
채널 및 트랜스미션 큐 정의.	73	관리 권한	100
채널 시작.	74	이스케이프 PCF의 MQSC 명령에 대한 권한	100
채널의 자동 정의	75	권한 파일에 대한 이해	103
리모트로 MQSC 명령 발행.	75	권한 파일 경로.	103

권한 파일의 내용	104	디스크가 가득 찰 경우에 발생하는 사항	153
권한 파일 관리.	106	로그 파일 관리.	153
제8장 MQSeries 데드-레터 큐 핸들러.	109	복구에 로그 사용	155
DLQ 핸들러 실행.	109	문제점에서 복구	155
샘플 DLQ 핸들러, amqsdlq	110	매체 복원	155
DLQ 핸들러 규칙 테이블	111	시동시 손상된 오브젝트 복구	157
제어 데이터.	111	다른 시기에 손상된 오브젝트 복구	157
규칙(패턴 및 조치)	112	MQSeries 로그 파일 보호.	157
규칙 테이블 규정	116	백업 및 복원	158
규칙 테이블 처리 방법	117	MQSeries 백업	158
모든 DLQ 메시지 처리.	118	MQSeries 복원	158
DLQ 핸들러 규칙 테이블 예	119	복구 시나리오	159
제9장 도구화 이벤트.	123	디스크 드라이브 오류	159
도구 이벤트의 개념	123	손상된 큐 관리자 오브젝트.	160
이벤트를 사용하는 이유.	125	손상된 단일 오브젝트	160
이벤트 유형.	125	자동 매체 복원 실패.	161
이벤트 큐를 통한 이벤트 공고.	126	dmpmqlog 명령을 사용하여 로그의 내용 덤프	161
이벤트 사용 및 사용 안함	127	제12장 이름 서비스 사용	181
이벤트 메시지	127	다른 큐 관리자에 있는 큐를 공유하기 위한 DEC	
제10장 트랜잭션식 지원.	129	사용	181
데이터베이스 통합.	130	공유 큐에 대한 구성 작업	181
제한사항	131	DCE 구성	182
데이터베이스 연결.	131	제13장 MQSeries 구성.	183
데이터베이스 관리자 구성	132	MQSeries 구성 파일.	183
Oracle 구성.	134	구성 파일 편집.	183
Oracle에 대한 최소 지원 레벨	134	MQSeries 구성 파일, mqs.ini.	184
환경 변수 설정 점검	134	큐 관리자 구성 파일 qm.ini	185
Oracle XA 지원 사용	134	MQSeries 구성 정보 변경에 대한 속성.	185
Oracle 스위치 로드 파일 작성	134	AllQueueManagers 스탠자.	185
Oracle용 XAResourceManager 구성 정보 추가	137	ClientExitPath 스탠자	187
Oracle 구성 매개변수 변경.	138	DefaultQueueManager 스탠자	187
관리 작업	139	ExitProperties 스탠자	187
인다우트(in-doubt) 작업 단위.	140	LogDefaults 스탠자	188
dspmqtrn 명령 사용	140	QueueManager 스탠자	190
rsvmqtrn 명령 사용	141	큐 관리자 구성 정보 변경	191
혼합 결과 및 오류	142	Service 스탠자.	191
구성 정보 변경.	143	ServiceComponent 스탠자.	192
제11장 복구 및 재시작	145	Log 스탠자.	193
메시지 손실 여부 확인(로그 기록)	145	XAResourceManager 스탠자	195
로그 형식	146	Channels 스탠자	196
로그 기록 유형.	146	LU62 및 TCP 스탠자	198
체크포인팅 - 완전한 복구	148	ExitPath 스탠자	200
로그 크기 계산.	151	mqs.ini 및 qm.ini 파일 예.	200
로그 관리	152	제14장 문제점 판별	205

기본 점검사항	205
이전에 MQSeries가 성공적으로 실행되었습니까?	205
오류 메시지가 있습니까?	206
문제점을 설명하는 리턴 코드가 있습니까?	206
문제점을 다시 만들 수 있습니까?	206
마지막 성공적인 실행 이후로 변경을 수행했습니까?	206
이전에 응용프로그램이 성공적으로 실행되었습니까?	206
문제점이 네트워크의 특정 부분에만 영향을 줍니까?	208
특정 시기에 문제점이 발생했습니까?	208
문제점이 간헐적입니까?	208
서비스 갱신사항을 적용했습니까?	208
갱신사항을 적용해야 합니까?	209
공통 프로그래밍 오류	209
다음 수행 사항	209
올바르지 않은 출력이 확보되었습니까?	210
PCF 명령으로부터 응답을 수신하는 데 실패했습니까?	210
일부 큐가 실패했습니까?	211
문제점이 리모트 큐에만 영향을 줍니까?	212
응용프로그램 설계 고려사항	212
메시지 길이의 효과	212
메시지 지속성의 효과	213
특정 메시지 검색	213
다른 길이의 메시지를 포함하는 큐	213
동기점 빈도	213
MQPUT1 호출의 사용	214
올바르지 않은 출력	214
큐에 표시되지 않는 메시지	214
예상치 못하거나 손상된 정보를 포함하는 메시지 분산 큐를 사용할 경우 올바른 출력에 대한 문제점	216
오류 로그	217
로그 파일	217
이전 오류	218
연산자 메시지	218
오류 로그 예	218
데드-레터 큐	221
구성 파일 및 문제점 판별	221
MQSeries 추적 사용	222
추적 파일 이름	222
샘플 추적 데이터	222
FFST(First Failure Support Technology)	223
FFST를 조사하는 방법	223

클라이언트의 문제점 판별	228
클라이언트 종료	229
클라이언트의 오류 메시지	229

제15장 성능 조정	231
프로세스 고유 매개변수의 값 설정	232

제16장 OpenVMS용 MQSeries 및 클러스터링	235
OpenVMS 클러스터에 MQSeries 설치	235
OpenVMS 클러스터 Failover 세트	236
OpenVMS 클러스터 Failover 세트의 개요	236
OpenVMS 클러스터 Failover 세트 개념	237
OpenVMS 클러스터 Failover 세트 구성 준비	238
OpenVMS 클러스터 Failover 세트 구성	239
OpenVMS 클러스터 Failover 세트 구성 후 작업	239
FAILOVER.INI 구성 파일 편집	240
Failover 세트에서 사용되는 명령 프로시저	241
Failover 세트 관리	242
Failover 모니터 시동	243
Failover 세트 내에서 큐 관리자 시작	243
Failover 세트 내에서 큐 관리자 종료	244
Failover 세트 내에서 큐 관리자 이동	244
Failover 세트 상태 표시	245
DCL 기호를 Failover 세트 상태로 설정	246
Failover 모니터 프로세스 정지	247
갱신이 진행되는 동안 명령 실행	247
Failover 세트 상태 변경	247
ICC 연관에 대한 보안 설정	248
Failover 세트의 문제점 해결	249
Failover 세트와 함께 OpenVMS용 MultiNet 사용	250
Failover 세트 사용 예	250

제2부 참조사항 257

제17장 MQSeries 제어 명령	259
MQSeries 오브젝트 이름 지정 규칙	259
오브젝트 파일 보기	260
구문 다이어그램 읽는 방법	260
구문 도움말	261
예	262
MQSeries 리턴 코드	263
crtmqcvx(데이터 변환)	264
crtmqm(큐 관리자 작성)	266
dlmqm(큐 관리자 삭제)	270
dmpmqlog(로그 덤프)	273

dspmqaout(권한 표시).	275
dspmqcsv(명령 서버 표시).	279
dspmqlfs(MQSeries 파일 표시).	281
dspmqltrc(MQSeries 형식화 추적 출력 표시).	283
dspmqltrn(MQSeries 트랜잭션 표시).	285
endmqcsv(명령 서버 종료).	287
endmqlsr(리스너 종료).	290
endmqm(큐 관리자 종료).	291
endmqtrc(MQSeries 추적 종료).	295
failover(Failover 세트 관리).	296
rcdmqimg(매체 이미지 기록).	300
rcrmqobj(오브젝트 재작성).	303
rsvmqtrn(MQSeries 트랜잭션 해석).	307
runmqchi(채널 시작기 실행).	309
runmqchl(채널 실행).	310
runmqdlq(데드-레터 큐 핸들러 실행).	311
runmqfm(Failover 모니터 시작).	313
runmqlsr(리스너 실행).	314
runmqsc(MQSeries 명령 실행).	316
runmqtmc(클라이언트 트리거 모니터 시작).	319
runmqtrm(트리거 모니터 시작).	320
setmqaut(권한 설정/재설정).	321
strmqcsv(명령 서버 시작).	329
strmqm(큐 관리자 시작).	330
strmqtrc(MQSeries 추적 시작).	332

제3부 부록 335

부록A. Compaq OpenVMS용 MQSeries 요약	337
프로그램 및 부품 번호	337
하드웨어 요구사항.	337
소프트웨어 요구 사항	337
연결	337
보안	338
유지보수 기능	338
호환성.	338
지원 컴파일러	338
언어 선택	338
국제화.	339
부록B. 시스템 디폴트	341
부록C. 디렉토리 구조	343
MQS_ROOT:[MQM]의 디렉토리와 파일	344
MQS_ROOT:[MQM.QMGRS.QMNAME] 서브디렉토리의 디렉토리와 파일	345

부록D. 명령 세트 비교	349
큐 관리자의 관리에 사용하는 명령	349
명령 서버 관리에 사용하는 명령	349
큐 관리에 사용하는 명령	350
프로세스 관리에 사용하는 명령	350
채널 관리에 사용하는 명령.	351
기타 제어 명령.	351

부록E. 샘플 MQI 프로그램 및 MQSC 파일	353
MQSC 명령 파일 샘플	353
C 및 COBOL 프로그램 샘플.	353
기타 도구	354

부록F. OpenVMS 클러스터 Failover 세트 템플리트	355
템플리트 구성 파일 FAILOVER.TEMPLATE	355
템플리트 StartCommand 프로시저 START_QM.TEMPLATE.	357
템플리트 EndCommand 프로시저 END_QM.TEMPLATE.	359
템플리트 TidyCommand 프로시저 TIDY_QM.TEMPLATE	362

부록G. Compaq OpenVMS용 MQSeries에서의 코드 세트 지원.	363
---	-----

부록H. MONMQ 진단 유틸리티	365
개요	365
MONMQ의 변수	366
디폴트 값 지정.	368
추적 섹션 및 연관된 편지함을 열거나 작성	368
논리 단위 정의 표시.	368
LU 닫기 및 삭제.	369
채널 세부사항 표시	369
채널의 현재 추적 마스크 표시.	370
대상 스레드 스택의 내용 표시.	370
활성 MQSeries 관련 프로세스 및 메모리 사용량 표시	371
채널이 보유한 모든 메시지 표시	371
현재 노드에 있는 모든 MQSeries 관련 전역 섹션 표시	372
mutex 테이블을 클라이언트 추적 프로세스로 송신 하라는 신호를 대상 스레드에 보내기	372
내부 이벤트 테이블을 클라이언트 추적 프로세스로 송신하라는 신호를 대상 스레드에 보내기	373
맵핑된 내부 공유 메모리 테이블을 클라이언트 추적 프로세스로 송신하라는 신호를 대상 스레드에 보내기	374

이름 및 16진수 ID별로 활성화 MQSeries 구성요소 표시	375
지정된 구성요소 내의 함수 표시	375
프로세스가 시작된 지점에서 추적 활성화	376
시동 후 즉시 MQSeries 프로세스가 추적되지 않도록 방지	376
대상 스레드를 지정한 채널에 연결	376
지정된 채널에서 대상 스레드 연결 끊기	377
LU 추적 편지함에 기록된 실시간 추적 메시지 표시	377
현재 클라이언트 프로세스 분리 및 종료	377
추적 데이터 지정	377
추적 필터 테이블에서 단일 입력 항목 제거	378
2진 파일에 추적 메시지를 기록하는 클라이언트 프로세스	379
2진 추적 메시지 파일 닫기.	379
텍스트 파일에 추적 메시지를 기록하는 클라이언트 프로세스	379
텍스트 추적 메시지 파일 닫기.	379
시간 소인 메시지	379
메시지 시간 소인 정지	380
추적 사용	380
추적 사용 안함.	380
메시지 실행 기록 저장	380
메시지 실행 기록 사용 안함	380
메시지 실행 기록 삭제	381
실행 기록 용량 설정	381
채널의 스택 및 실행 기록 데이터 재설정	381
마스크 비트 사용/사용 안함	381
채널에 색상 설정	383
출력의 경로를 파일에 재지정	383
추적 2진 파일 분석	384
MQSeries 스레드의 현재 상태 표시.	386
추적 닫기 및 MONMQ 종료.	387
추적을 닫지 않고 MONMQ 종료	387

MONMQ를 사용하여 공유 메모리 관리	387
MONMQ의 스크립트 및 매크로	388
샘플 추적 세션.	390
부록I. 사용자 종료	401
채널 및 워크로드 종료	401
MQSeries 클러스터 워크로드 종료	401
부록J. 트러스트 응용프로그램	403
사용자 응용프로그램	403
트러스트 응용프로그램 설정	403
채널과 리스너를 트러스트 응용프로그램으로 실행	404
빠른 비지속 메시지	404
부록K. 추가 정보	407
응용프로그램 프로그래밍 안내.	407
응용프로그램 트리거	407
부록L. 주의사항	409
등록상표 및 서비스표.	411
참고 문헌	413
MQSeries의 모든 플랫폼에 적용되는 서적	413
MQSeries 플랫폼별 서적	413
소프트키피 서적	415
HTML 형식	415
PDF(Portable Document Format)	415
BookManager 형식	415
포스트스크립트 형식	416
Windows 도움말 형식	416
인터넷에서 사용 가능한 MQSeries 정보	416
관련 서적	416
용어집	417
색인	429

그림

1. 큐, 메시지 및 응용프로그램	39	19. Compaq OpenVMS용 MQSeries 시스템용 MQSeries 구성 파일 예	201
2. DISPLAY QMGR 명령의 일반적인 출력 형태	43	20. Compaq OpenVMS용 MQSeries의 큐 관리자 구성 파일 예	202
3. MQSC 명령 파일 myprog.in의 추출	46	21. 샘플 Compaq OpenVMS용 MQSeries 추적	223
4. MQSC 보고서 파일 myprog.out의 추출	47	22. ICC\$SYSTARTUP.COM에 필요한 샘플 항목	249
5. 큐 브라우저에 표시되는 일반적인 결과	55	23. FAILOVER.INI 구성 파일을 작성하기 위한 failover.template	251
6. 리모트 관리	72	24. start_failover_set 명령 프로시저	253
7. 리모트 관리에 대한 채널 및 큐 설정	73	25. end_failover_set 명령 프로시저	255
8. DLQ 핸들러 규칙 테이블의 규칙 예	112	26. 큐 관리자를 시작한 후의 디폴트 디렉토리 구조	343
9. 도구 이벤트에 대한 이해.	124	27. 템플릿 구성 파일: failover.template	356
10. 단일 노드상의 서로 다른 플랫폼 사이의 큐 관리자 모니터링	125	28. 템플릿 StartCommand 프로시저: Start_QM.template	358
11. Oracle 스위치 로드 파일 oraswit.c에 대한 소스 코드	135	29. 템플릿 EndCommand 프로시저: END_QM.template	361
12. Oracle용 샘플 XAResourceManager 항목	138	30. Template TidyCommand 프로시저: TIDY_QM.template	362
13. Sample dspmqtrn output	141		
14. 오류가 있는 트랜잭션에 대한 dspmqtrn 출력 샘플	143		
15. 주석 처리된 XAResourceManager 스탠자	144		
16. 체크포인팅	149		
17. 장기 실행 트랜잭션에서의 체크포인팅	150		
18. dmpmqlog 출력 예	166		

표

1. 제어 명령의 범주.	29	18. 큐의 시스템 오브젝트 및 디폴트 오브젝트	341
2. MQI 호출에 필요한 보안 권한	97	19. 채널의 시스템 오브젝트 및 디폴트 오브젝트	341
3. 필요한 MQSC 명령 및 보안 권한	101	20. 이름 목록의 시스템 오브젝트 및 디폴트 오브젝트	342
4. PCF 명령과 필요한 보안 권한.	102	21. 프로세스의 시스템 오브젝트 및 디폴트 오브젝트	342
5. XA 준수 관계형 데이터베이스.	131	22. 큐 관리자의 관리에 사용하는 명령	349
6. 로그 오버헤드 크기(모든 값은 근사값임)	151	23. 명령 서버 관리에 사용하는 명령	349
7. 가능한 ISO CCSID 목록	186	24. 큐 관리에 사용하는 명령.	350
8. 디폴트 미해결 연결 요청(TCP).	199	25. 프로세스 관리에 사용하는 명령	350
9. FAILOVER.INI 파일 내의 필드 설명	240	26. 채널 관리에 사용하는 명령	351
10. 명령 프로시저에 전달된 매개변수.	241	27. 기타 제어 명령	351
11. Failover 세트 큐 관리자 상태.	245	28. MQSC 명령 파일.	353
12. Failover 세트 노드 큐 관리자 상태	245	29. 샘플 프로그램 - 소스 파일.	353
13. Failover 세트 노드 모니터 상태	245	30. 기타 파일.	354
14. DCL 기호 및 설명	246	31. 로케일 및 CCSID	363
15. 구문 다이어그램을 읽는 방법	260		
16. dspmqaut 명령의 보안 권한	276		
17. 여러 오브젝트 유형에 대한 권한 지정	324		

서적 정보

Compaq OpenVMS용 MQSeries 또는 MQSeries와 같이 컨텍스트로 참조되는 Compaq OpenVMS Alpha용 MQSeries V5.1는 MQSeries 제품군의 일부입니다. 이러한 제품들은 메시지 큐를 사용하여 응용프로그램간의 통신을 가능하게 하는 응용프로그램 프로그래밍 서비스를 제공합니다. 이러한 통신 형식을 상용 메시징이라고 합니다. 통신에 참여하는 응용프로그램은 다양한 시스템 및 운영 체제 유형의 서로 다른 노드에 있을 수 있습니다. 이러한 경우, 각 시스템과 운영 체제에서 MQI(Message Queue Interface)라는 공통 응용프로그램 프로그래밍 인터페이스를 사용하여 한 플랫폼에서 개발한 프로그램을 다른 플랫폼으로 즉시 전송할 수 있습니다.

이 서적에서는 Compaq OpenVMS Alpha용 MQSeries V5.1의 시스템 관리와 이 프로그램이 OpenVMS 환경에서 상용 메시징을 지원하기 위해 제공하는 서비스에 대해 설명합니다. 여기에는 응용프로그램이 메시지 수신에 사용하는 큐에 대한 관리 및 응용프로그램이 자신에게 필요한 큐에 액세스할 수 있도록 하는 작업 등이 포함됩니다.

이 서적의 대상

기본적으로 이 서적은 시스템 관리자와 MQSeries의 구성 및 관리 작업을 관리하는 시스템 프로그래머를 위한 것입니다. 이 서적은 MQSeries 관리 작업에 대한 지식을 어느 정도 가지고 있는 응용프로그램 프로그래머에게도 유용합니다.

이 서적을 이해하기 위한 선결 사항

이 서적을 사용하려면 OpenVMS 운영 체제 및 이 운영 체제와 연관된 유틸리티를 잘 이해하고 있어야 합니다. 이전에 메시지 큐잉 제품을 사용하여 작업한 경험은 없다고 하더라도 메시징 큐잉의 기본 개념에 대해서는 이해하고 있어야 합니다.

서적 사용 방법

이 서적의 본문 내용은 다음과 같습니다.

- MQSeries 소개
- OpenVMS 시스템의 일상적인 관리 작업 설명, 로컬 MQSeries 오브젝트 및 리모트 MQSeries 오브젝트 관리, 보안, 트랜잭션 지원, 문제점 판단 등과 같은 주제에 대한 설명

서적 정보

부록 사용

부록은 참조 자료를 제공합니다. 몇몇 부록에는 다음에 다른 MQSeries 서적에 통합될 정보가 포함되어 있습니다.

인터넷상의 MQSeries 정보

MQSeries URL

MQSeries 제품군 홈 페이지의 URL은 다음과 같습니다.

<http://www.ibm.com/software/ts/mqseries/>

Compaq OpenVMS Alpha용 MQSeries V5.1의 새로운 기능

이 *Compaq OpenVMS Alpha용 MQSeries V5.1* 시스템 관리 안내서에서는 다음과 같은 새로운 기능들을 설명합니다.

MQSeries 큐 관리자 클러스터

MQSeries 큐 관리자를 연결하여 큐 관리자 클러스터를 구성할 수 있습니다. 클러스터 내에서 큐 관리자는 호스트하는 큐를 다른 모든 큐 관리자가 사용하도록 할 수 있습니다. 클러스터의 모든 큐 관리자는 명확한 채널 정의, 리모트 큐 정의 또는 각 목적지에 대한 트랜스미션 큐 없이 같은 클러스터에 있는 다른 모든 큐 관리자에 메시지를 송신할 수 있습니다. MQSeries 클러스터를 사용하여 얻을 수 있는 주요 이점은 다음과 같습니다.

- 시스템 관리 작업 감소
- 사용 가능성 증가
- 워크로드 밸런스 유지

자세한 정보는 16 페이지의 『클러스터』를 참조하십시오.

주: MQSeries 클러스터는 OpenVMS 클러스터와는 다릅니다. 클러스터라는 용어가 사용되면 이는 MQSeries 큐 관리자 클러스터를 가리킵니다. OpenVMS 클러스터는 언제나 *OpenVMS 클러스터*라고 합니다. OpenVMS 클러스터에 대한 정보는 235 페이지의 『제16장 OpenVMS용 MQSeries 및 클러스터링』을 참조하십시오.

MQAI(MQSeries Application Interface)

Compaq OpenVMS용 MQSeries는 MQSeries 구성시 간단하게 PCF 메시지를 사용할 수 있도록 프로그래밍 인터페이스인 MQAI(MQSeries Application Interface)를 지원합니다. 모든 명령에 대한 설명을 포함하여 MQAI에 대한 자세한 정보는 *MQSeries Administration Interface Programming Guide and Reference*를 참조하십시오.

메시지 큐 크기

메시지 큐 크기는 최대 2 GB까지 가능합니다.

큐 관리자의 제어된 대기 종료

endmqm 명령에 큐 관리자를 제어된 방법으로 동기 종료할 수 있는 새 옵션이 추가되었습니다.

Java 지원

Compaq OpenVMS용 MQSeries는 이제 Java 컴파일러로 작업할 수 있습니다.

웹 관리

Windows NT 시스템에서 Netscape Navigator나 Microsoft Internet Explorer 등의 HTML 브라우저를 사용하여 Compaq OpenVMS용 MQSeries에 대한 다음과 같은 관리 작업을 리모트로 수행할 수 있습니다.

- MQSeries 관리자로 로그인
- 큐 관리자 선택 및 큐 관리자에 MQSC 명령 실행
- MQSC 스크립트 작성, 편집 및 삭제

제1부 자세한 내용

제1장 MQSeries 소개	7	MQSC 및 PCF의 속성	25
MQSeries와 메시지 큐잉	7	이스케이프 PCF.	25
시간 독립적 응용프로그램	7	MQSeries 파일 이름 이해	25
메시지에 의한 처리	7	큐 관리자 이름 변환	26
메시지와 큐.	8	오브젝트 이름 변환.	26
메시지의 정의	8	대소문자 구분 이해.	27
메시지 길이.	8	제어 명령의 대소문자 구분	27
큐의 정의	8	MQSC 명령의 대소문자 구분	28
응용프로그램이 메시지를 송/수신하는 방법	9	제3장 제어 명령을 사용한 큐 관리자 관리	29
사전정의된 큐 및 다이내믹 큐	9	제어 명령 사용	29
큐에 있는 메시지 검색	9	제어 명령 사용	29
오브젝트	10	큐 관리자 작성	30
오브젝트 이름	10	큐 관리자 작성 지침	30
오브젝트 관리	11	고유 큐 관리자 이름의 지정.	31
오브젝트 속성	11	큐 관리자 수 제한	31
MQSeries 큐 관리자	11	디폴트 큐 관리자 지정.	31
MQI 호출.	12	데드-레터 큐 지정	32
MQSeries 큐.	12	디폴트 트랜스미션 큐 지정	33
큐 오브젝트 사용	12	필요한 로그 기록 매개변수 지정	33
MQSeries에서 사용되는 특정 로컬 큐	13	큐 관리자 작성 후 구성 파일 백업	33
프로세스 정의	15	디폴트 큐 관리자 작성.	34
채널.	16	큐 관리자 시작	34
클러스터	16	기존의 큐 관리자를 디폴트로 만들기	35
이름 목록.	16	큐 관리자 정지	35
시스템 디폴트 오브젝트	17	정지 종료	35
로컬 관리 및 리모트 관리	17	즉시 종료	36
클라이언트 및 서버.	17	선제 종료	36
클라이언트-서버 환경에서의 MQSeries 응용프로 그램.	18	큐 관리자 종료에 문제점이 있을 경우	37
큐 관리자 기능 확장	18	큐 관리자 재시작	37
사용자 종료	19	큐 관리자 삭제	37
설치가능 서비스.	19	제4장 로컬 MQSeries 오브젝트 관리	39
보안.	20	MQI를 사용하는 응용프로그램 지원	39
OAM(Object Authority Manager) 기능	20	MQSC 명령을 사용한 로컬 관리 작업의 수행.	40
DCE 보안.	20	시작하기 전에	41
트랜잭션식 지원.	20	MQSeries 오브젝트 이름.	41
제2장 MQSeries 관리 소개.	23	입력 및 출력 경로 재지정	41
로컬 관리 및 리모트 관리	23	MQSC 기능의 대화식 사용.	41
제어 명령을 사용하여 관리 작업 수행	24	MQSC 명령의 피드백.	42
MQSC 명령을 사용하여 관리 작업 수행	24	MQSC의 대화식 입력 종료.	43
PCF 명령을 사용하여 관리 작업 수행	25	큐 관리자 속성 표시	43
		디폴트가 아닌 큐 관리자의 사용	44

큐 관리자 속성 대체	44	채널의 자동 정의	75
텍스트 파일에서 MQSC 명령 실행	44	리모트로 MQSC 명령 발행	75
MQSC 명령 파일	45	MVS/ESA에서 큐 관리자에 대한 작업	76
MQSC 보고서	46	리모트 큐잉에 대한 권장사항	77
제공된 MQSC 명령 파일의 실행	47	리모트로 MQSC를 사용할 때 문제점이 발생할	
명령 확인을 위한 runmqsc의 사용	47	경우	77
MQSC 문제점 해결	48	리모트 큐의 로컬 정의 작성	78
로컬 큐에 대한 작업	50	리모트 큐의 로컬 정의가 작동되는 방법에 대한	
로컬 큐 정의	50	이해	78
데드-레터 큐의 정의	51	예	78
디폴트 오브젝트 속성의 표시	51	리모트 큐에 메시지를 넣는 다른 방법	79
로컬 큐 정의 복사	52	리모트 큐에서 다른 명령 사용	80
로컬 큐 속성 변경	53	트랜스미션 큐 작성	80
로컬 큐 지우기	53	디폴트 트랜스미션 큐	80
로컬 큐 삭제	53	별명으로 리모트 큐 정의 사용	81
큐 찾아보기	54	큐 관리자 별명	81
알리아스 큐에 대한 작업	57	리플라이-투 큐 별명	81
알리아스 큐 정의	58	데이터 변환	82
큐 알리아스가 있는 기타 명령의 사용	59	큐 관리자가 메시지를 내장 형식으로 변환할	
모델 큐에 대한 작업	60	수 없을 경우	82
모델 큐의 정의	60	파일 ccsid.tbl	82
모델 큐에 대한 기타 명령의 사용	60	사용자 정의 형식의 메시지 변환	83
트리거를 위한 오브젝트 관리	61	큐 관리자 CCSID 변경	84
트리거를 위한 응용프로그램 큐 정의	61		
이니시에이션 큐 정의	62	제7장 MQSeries 오브젝트 보호	85
프로세스 정의 작성	62	MQSeries 자원을 보호해야 하는 이유	85
프로세스 정의 표시	63	시작하기 전에	85
제5장 관리 작업 자동화	65	자원 ID가 MQM인 Compaq OpenVMS용	
PCF 명령	65	MQSeries에서의 사용자 ID	86
MQSC 및 PCF의 속성	66	자세한 정보	86
이스케이프 PCF	66	오브젝트 권한 관리자 이해	87
MQAI를 통한 간편한 PCF 사용	66	OAM 작동 방법	87
리모트 관리를 위한 명령 서버 관리	67	권한 ID를 통한 액세스 관리	87
명령 서버 시작	67	권한 ID 및 1차 권한 ID	88
명령 서버 상태 표시	68	기본 원칙이 하나 이상의 권한 ID를 보유할 경	
명령 서버 정지	68	우	88
제6장 리모트 MQSeries 오브젝트 관리	69	디폴트 권한 ID	88
채널, 클러스터 및 리모트 큐잉	69	OAM으로 보호할 수 있는 자원	88
클러스터를 사용한 리모트 관리	70	권한에 대해 권한 ID 사용	89
MQSC 명령을 사용한 로컬 큐 관리자에서의 리모트		오브젝트 권한 관리자 사용 안함	89
관리	71	오브젝트 권한 관리자 명령 사용	89
리모트 관리에 대한 큐 관리자 준비	71	OAM 명령 사용시 지정해야 할 사항	90
리모트 관리를 위한 채널 및 트랜스미션 큐 준비	72	권한 목록	90
채널 및 트랜스미션 큐 정의	73	setmqaut 명령 사용	90
채널 시작	74	권한 명령 및 설치가능 서비스	91
		액세스 권한	92
		권한 명령 표시	92

오브젝트 권한 관리자 지침	92
사용자 ID.	92
큐 관리자 디렉토리.	93
큐	93
대체 사용자 권한	93
컨텍스트 권한	94
리모트 보안 고려사항	94
채널 명령 보안	95
PCF 명령	95
MQSC 채널 명령	96
채널에 대한 제어 명령	96
권한 스펙 테이블에 대한 이해	96
MQI 권한.	97
관리 권한	100
이스케이프 PCF의 MQSC 명령에 대한 권한	100
PCF 명령에 대한 권한	102
권한 파일에 대한 이해	103
권한 파일 경로.	103
권한 디렉토리	104
권한 파일의 내용	104
클래스 권한 파일	106
모든 클래스 권한 파일	106
권한 파일 관리.	106
권한 파일에 대한 권한	106
제8장 MQSeries 데드-레터 큐 핸들러.	109
DLQ 핸들러 실행.	109
샘플 DLQ 핸들러, amqsdlq	110
DLQ 핸들러 규칙 테이블	111
제어 데이터.	111
규칙(패턴 및 조치)	112
패턴 대조 키워드	113
조치 키워드.	114
규칙 테이블 규정	116
규칙 테이블 처리 방법	117
모든 DLQ 메시지 처리.	118
DLQ 핸들러 규칙 테이블 예	119
제9장 도구화 이벤트.	123
도구 이벤트의 개념	123
이벤트를 사용하는 이유.	125
이벤트 유형.	125
이벤트 큐를 통한 이벤트 공고.	126
트리거된 이벤트 큐 사용	127
이벤트 사용 및 사용 안함	127
이벤트 메시지	127

제10장 트랜잭션식 지원.	129
데이터베이스 통합.	130
제한사항	131
데이터베이스 연결.	131
데이터베이스 관리자 구성	132
스위치 로드 파일 작성	132
데이터베이스 관리자 정의	132
Oracle 구성.	134
Oracle에 대한 최소 지원 레벨	134
환경 변수 설정 점검	134
Oracle XA 지원 사용	134
Oracle 스위치 로드 파일 작성	134
OpenVMS 시스템에서 Oracle 스위치 로드 파일 작성	135
Oracle용 XAResourceManager 구성 정보 추가	137
Oracle 구성 매개변수 변경.	138
관리 작업	139
인다우트(in-doubt) 작업 단위.	140
dspmqtrn 명령 사용	140
rsvmqtrn 명령 사용	141
혼합 결과 및 오류	142
구성 정보 변경.	143
데이터베이스 관리자 인스턴스 제거	144
제11장 복구 및 재시작	145
메시지 손실 여부 확인(로그 기록)	145
로그 형식	146
로그 제어 파일.	146
로그 기록 유형.	146
순환 로깅	146
선형 로깅	147
체크포인팅 - 완전한 복구	148
로그 크기 계산.	151
로그 관리	152
디스크가 가득 찰 경우에 발생하는 사항	153
로그 파일 관리.	153
로그 파일 위치.	154
복구에 로그 사용	155
문제점에서 복구	155
매체 복원	155
매체 이미지 복구	156
시동시 손상된 오브젝트 복구	157
다른 시기에 손상된 오브젝트 복구	157
MQSeries 로그 파일 보호.	157
백업 및 복원	158
MQSeries 백업	158
MQSeries 복원	158

복구 시나리오	159	이전에 응용프로그램이 성공적으로 실행되었습니	
디스크 드라이브 오류	159	까?.	206
손상된 큐 관리자 오브젝트.	160	이전에 응용프로그램이 성공적으로 실행되지	
손상된 단일 오브젝트	160	않은 경우	207
자동 매체 복원 실패.	161	문제점이 네트워크의 특정 부분에만 영향을 줘니	
dmpmqlog 명령을 사용하여 로그의 내용 덤프	161	까?.	208
제12장 이름 서비스 사용	181	특정 시기에 문제점이 발생했습니까?.	208
다른 큐 관리자에 있는 큐를 공유하기 위한 DEC		문제점이 간헐적입니까?	208
사용	181	서비스 갱신사항을 적용했습니까?.	208
공유 큐에 대한 구성 작업	181	갱신사항을 적용해야 합니까?	209
DCE 구성	182	공통 프로그래밍 오류	209
제13장 MQSeries 구성.	183	다음 수행 사항.	209
MQSeries 구성 파일.	183	올바르지 않은 출력이 확보되었습니까?	210
구성 파일 편집.	183	PCF 명령으로부터 응답을 수신하는 데 실패했습	
구성 파일을 편집해야 할 경우.	184	니까?	210
구성 파일 우선순위	184	일부 큐가 실패했습니까?	211
구성 파일에 대한 변경사항 구현	184	문제점이 리모트 큐에만 영향을 줘니까?	212
MQSeries 구성 파일, mqs.ini.	184	응용프로그램 설계 고려사항	212
큐 관리자 구성 파일 qm.ini	185	메시지 길이의 효과	212
MQSeries 구성 정보 변경에 대한 속성.	185	메시지 지속성의 효과.	213
AllQueueManagers 스탠자.	185	특정 메시지 검색	213
ClientExitPath 스탠자	187	다른 길이의 메시지를 포함하는 큐	213
DefaultQueueManager 스탠자	187	동기점 빈도.	213
ExitProperties 스탠자	187	MQPUT1 호출의 사용	214
LogDefaults 스탠자	188	올바르지 않은 출력	214
QueueManager 스탠자	190	큐에 표시되지 않는 메시지.	214
큐 관리자 구성 정보 변경	191	예상치 못하거나 손상된 정보를 포함하는 메시지	215
Service 스탠자.	191	분산 큐를 사용할 경우 올바르게 않은 출력에 대	
ServiceComponent 스탠자.	192	한 문제점	216
Log 스탠자.	193	오류 로그	217
XAResourceManager 스탠자	195	로그 파일	217
Channels 스탠자	196	이전 오류	218
LU62 및 TCP 스탠자	198	연산자 메시지	218
ExitPath 스탠자	200	오류 로그 예	218
mqs.ini 및 qm.ini 파일 예.	200	데드-레터 큐	221
제14장 문제점 판별	205	구성 파일 및 문제점 판별	221
기본 점검사항	205	MQSeries 추적 사용.	222
이전에 MQSeries가 성공적으로 실행되었습니까?	205	추적 파일 이름.	222
오류 메시지가 있습니까?	206	샘플 추적 데이터	222
문제점을 설명하는 리턴 코드가 있습니까?.	206	FFST(First Failure Support Technology)	223
문제점을 다시 만들 수 있습니까?	206	FFST를 조사하는 방법	223
마지막 성공적인 실행 이후로 변경을 수행했습니		클라이언트의 문제점 판별	228
까?.	206	클라이언트 종료	229
		클라이언트의 오류 메시지	229
		OS/2, UNIX 및 OpenVMS 시스템 클라이언	
		트	229

DOS 및 Windows 클라이언트	229
제15장 성능 조정	231
프로세스 고유 매개변수의 값 설정	232
제16장 OpenVMS용 MQSeries 및 클러스터링	235
OpenVMS 클러스터에 MQSeries 설치.	235
OpenVMS 클러스터 Failover 세트	236
OpenVMS 클러스터 Failover 세트의 개요	236
OpenVMS 클러스터 Failover 세트 개념	237
OpenVMS 클러스터 Failover 세트 구성 준비	238
OpenVMS 클러스터 Failover 세트 구성	239
OpenVMS 클러스터 Failover 세트 구성 후 작업	239
FAILOVER.INI 구성 파일 편집	240
Failover 세트에서 사용되는 명령 프로시저	241
Failover 세트 관리	242
Failover 모니터 시동.	243
Failover 세트 내에서 큐 관리자 시작	243
Failover 세트 내에서 큐 관리자 종료	244
Failover 세트 내에서 큐 관리자 이동	244
Failover 세트 상태 표시	245
DCL 기호를 Failover 세트 상태로 설정	246
Failover 모니터 프로세스 정지	247
갱신이 진행되는 동안 명령 실행	247
Failover 세트 상태 변경	247
ICC 연관에 대한 보안 설정	248
Failover 세트의 문제점 해결	249
Failover 세트와 함께 OpenVMS용 MultiNet 사용	250
Failover 세트 사용 예	250
failover.template 사용자 정의.	250
Failover 세트 명령 프로시저의 수정사항	251
Failover 세트 시작 명령 프로시저 start_failover_set.com의 예	251
Failover 세트 종료 명령 프로시저 end_failover_set.com의 예.	254

제1장 MQSeries 소개

이 장에서는 관리자의 관점에서 Compaq OpenVMS용 MQSeries를 소개하고, MQSeries와 메시징의 기본 개념에 대해 설명합니다. 이 장에는 다음과 같은 절이 있습니다.

- 『MQSeries와 메시지 큐잉』
- 8 페이지의 『메시지와 큐』
- 10 페이지의 『오브젝트』
- 17 페이지의 『시스템 디폴트 오브젝트』
- 17 페이지의 『로컬 관리 및 리모트 관리』
- 17 페이지의 『클라이언트 및 서버』
- 18 페이지의 『큐 관리자 기능 확장』
- 20 페이지의 『보안』

MQSeries와 메시지 큐잉

MQSeries가 있으면 응용프로그램은 메시지 큐잉을 사용하여 메시지에 의한 처리를 할 수 있습니다. 응용프로그램은 적절한 메시지 큐잉 소프트웨어 제품을 사용하여 다양한 플랫폼 사이에서 통신할 수 있습니다. 예를 들어 OpenVMS 및 MVS/ESA 응용프로그램은 각각 Compaq OpenVMS용 MQSeries와 OS/390용 MQSeries를 통해 통신할 수 있습니다. 응용프로그램은 기존의 통신 메커니즘과는 상관이 없습니다.

MQSeries 제품은 응용프로그램이 실행되는 플랫폼에 관계없이, 메시지 큐 인터페이스(MQI)라고 알려진 공통 응용프로그램 프로그래밍 인터페이스(API)를 구현합니다. 그러므로 응용프로그램을 다른 플랫폼으로 인식하기가 쉽습니다.

MQI는 *MQSeries Application Programming Reference*에 자세히 설명되어 있습니다.

시간 독립적 응용프로그램

메시지 큐잉을 사용하면 시간에 관계없이 송/수신 프로그램간의 메시지 교환이 이루어 집니다. 이는 송신 응용프로그램과 수신 응용프로그램이 분리되어 있기 때문에 송신자는 수신자의 메시지 수신확인을 기다리지 않고도 처리를 계속할 수 있다는 것을 뜻합니다. 사실, 메시지를 송신할 때 대상 응용프로그램이 실행 중일 필요는 없습니다. 하지만 메시지를 검색하려면 대상 응용프로그램을 시작해야 합니다.

메시지에 의한 처리

메시지는 큐에 도착하는 즉시 트리거라고 하는 메커니즘을 사용하여 자동으로 응용프로그램을 시작할 수 있습니다. 필요하다면 메시지가 처리되었을 때 응용프로그램을 정지할 수 있습니다.

메시지와 큐

메시지와 큐는 메시지 큐잉 시스템의 기본적인 구성요소입니다.

메시지의 정의

메시지란 메시지를 사용하는 응용프로그램에 의미가 있는 바이트 문자열을 말합니다. 메시지는 한 응용프로그램에서 다른 응용프로그램으로(또는 같은 응용프로그램의 다른 부분으로) 정보를 전송하는 데 사용됩니다. 응용프로그램은 같은 플랫폼 또는 서로 다른 플랫폼에서 실행될 수 있습니다.

MQSeries 메시지는 두 부분으로 이루어집니다.

- **응용프로그램 데이터**

응용프로그램 데이터의 내용과 구조는 사용하는 응용프로그램에 의해 정의됩니다.

- **메시지 설명자**

메시지 설명자는 메시지를 식별하며, 메시지 유형 및 송신 응용프로그램에서 메시지에 지정한 우선순위와 같은 제어 정보가 들어 있습니다.

메시지 설명자의 형식은 MQSeries에서 정의합니다. 메시지 설명자에 대한 모든 설명은 *MQSeries Application Programming Reference* 안내서를 참조하십시오.

메시지 길이

MQSeries에서 최대 메시지 길이는 100 MB(여기에서 1 MB는 1 048 576 바이트)입니다. 실제로 사용할 때 메시지의 길이는 다음과 같은 사항에 따라 제한될 수 있습니다.

- 리시브 큐에 정의되어 있는 최대 메시지 길이
- 큐 관리자에 정의되어 있는 최대 메시지 길이
- 송신 또는 수신 응용프로그램에서 정의한 최대 메시지 길이
- 메시지에 사용할 수 있는 저장영역의 크기

응용프로그램에 필요한 정보를 모두 송신하려면 여러 개의 메시지가 필요할 수도 있습니다.

큐의 정의

큐란 메시지 저장에 사용하는 데이터 구조를 말합니다. 응용프로그램이나 큐 관리자의 일반 조작으로 메시지를 큐에 넣습니다(put).

각각의 큐는 큐 관리자가 소유합니다. 큐 관리자는 소유하고 있는 큐를 관리하고, 수신한 모든 메시지를 적절한 큐에 저장합니다.

큐의 최대 크기는 2 GB입니다. 큐에 필요한 저장영역의 크기 계획에 관한 정보는 *MQSeries Planning Guide*를 참조하거나 다음 웹 사이트에서 플랫폼별 성능 보고서를 참조하십시오.

<http://www.ibm.com/software/ts/mqseries/txppacs/txpm1.html>

응용프로그램이 메시지를 송/수신하는 방법

응용프로그램은 *MQI* 호출을 사용하여 메시지를 송/수신합니다. 예를 들어 응용프로그램은 메시지를 큐에 넣기(put) 위해 다음을 수행합니다.

1. MQI MQOPEN 호출을 발행하여 필요한 큐를 엽니다.
2. MQI MQPUT 호출을 발행하여 메시지를 큐에 넣습니다(put).
3. 다른 응용프로그램이 MQI MQGET 호출을 발행하여 동일한 큐에서 메시지를 검색할 수 있습니다.

MQI 호출에 대한 자세한 정보는 *MQSeries Application Programming Reference*를 참조하십시오.

사전정의된 큐 및 다이내믹 큐

큐는 작성된 방식에 따라 특징이 있습니다.

- **사전정의된 큐**는 관리자가 적절한 명령 세트를 사용하여 작성합니다. 예를 들어 MQSC 명령 DEFINE QLOCAL은 사전정의된 로컬 큐를 작성합니다. 사전정의된 큐는 영구적이어서 큐를 사용하는 응용프로그램에 대해 별도로 존재하며, MQSeries를 재시작해도 계속 남아 있습니다.
- **다이내믹 큐**는 응용프로그램이 모델 큐의 이름을 지정하는 OPEN 요청을 발행할 때 작성됩니다. 작성된 큐는 모델 큐인 템플릿 큐 정의를 기본으로 하고 있습니다. 모델 큐는 MQSC 명령 DEFINE QMODEL을 사용하여 작성할 수 있습니다. 예를 들어 저장할 수 있는 최대 메시지 수 등 모델 큐의 속성은 거기서 작성된 다이내믹 큐로 상속됩니다.

모델 큐에는 다이내믹 큐가 영구적인지 일시적인지를 지정하는 속성이 있습니다. 퍼머넌트 큐는 응용프로그램이나 큐 관리자가 재시작되어도 계속 남아 있지만 템포러리 큐는 손실됩니다.

큐에 있는 메시지 검색

MQSeries에서, 권한이 부여된 응용프로그램은 다음과 같은 검색 알고리즘에 따라 큐에 있는 메시지를 검색할 수 있습니다.

- FIFO
- 메시지 설명자에 정의되어 있는 메시지 우선순위. 우선순위가 동일한 메시지는 FIFO를 기준으로 검색합니다.
- 특정 메시지에 대한 프로그램 요청

응용프로그램의 MQGET 요청에 따라 사용할 메소드가 결정됩니다.

오브젝트

이 서적에서 설명하는 작업의 대부분은 MQSeries 오브젝트의 조작을 포함합니다. MQSeries Version 5.1의 오브젝트 유형에는 큐 관리자, 큐, 프로세스 정의, 채널, 클러스터 및 이름 목록이 있습니다.

오브젝트 조작이나 오브젝트 관리에는 다음이 포함됩니다.

- 큐 관리자 시작 및 정지
- 응용프로그램, 특히 큐에 대한 오브젝트 작성
- 다른(리모트) 시스템에 있는 큐 관리자에 대한 통신 경로 작성을 위한 채널 작업. 이에 대해서는 MQSeries 상호통신에 자세히 설명되어 있습니다.
- 큐 관리자의 클러스터를 작성하면 전체 관리 프로세스를 단순화하거나 워크로드 밸런스를 유지할 수 있습니다.

이 서적의 다음 장에는 관리에 대한 자세한 내용이 수록되어 있습니다.

- 23 페이지의 『제2장 MQSeries 관리 소개』
- 29 페이지의 『제3장 제어 명령을 사용한 큐 관리자 관리』
- 39 페이지의 『제4장 로컬 MQSeries 오브젝트 관리』
- 69 페이지의 『제6장 리모트 MQSeries 오브젝트 관리』

오브젝트 이름

MQSeries 오브젝트의 이름 지정 규칙은 오브젝트에 따라 다르게 선택합니다.

큐 관리자의 각 인스턴스는 그 이름으로 알 수 있습니다. 이 이름은 상호 연결된 큐 관리자의 네트워크에서 고유해야 하며, 이를 통해 큐 관리자가 메시지를 수신할 타겟 큐 관리자를 분명히 식별할 수 있어야 합니다.

나머지 오브젝트 유형의 경우, 오브젝트마다 그와 연관된 이름이 있어서 이름별로 오브젝트를 참조할 수 있습니다. 이러한 이름은 하나의 큐 관리자와 오브젝트 유형에 고유해야 합니다. 예를 들어 큐와 프로세스의 이름이 같을 수는 있지만 두 개의 큐가 같은 이름을 가질 수는 없습니다.

MQSeries에서 이름은, 최대 20자로 제한된 채널을 제외하면 최대 48자까지 사용할 수 있습니다. 이름에 대한 자세한 정보는 259 페이지의 『MQSeries 오브젝트 이름 지정 규칙』을 참조하십시오.

오브젝트 관리

다음을 사용하여 오브젝트를 작성, 대체, 표시 및 삭제할 수 있습니다.

- 키보드에서 입력하는 제어 명령
- 키보드에서 입력하거나 파일에서 읽을 수 있는 MQSeries 명령(MQSC)
- 자동 프로그램에 사용할 수 있는 PCF(Programmable Command Format) 명령
- 프로그램에서의 MQAI(MQSeries Administration Interface) 호출

자세한 정보는 23 페이지의 『제2장 MQSeries 관리 소개』를 참조하십시오.

오브젝트 속성

오브젝트의 등록 정보는 속성에 의해 정의됩니다. 일부 속성은 지정할 수 있으며 그 외의 속성은 볼 수만 있습니다. 예를 들어 큐에 허용되는 최대 메시지 길이는 *MaxMsgLength* 속성에 의해 정의되며, 이 속성은 큐를 작성할 때 지정할 수 있습니다. *DefinitionType* 속성은 큐가 작성되는 방법을 지정하며, 이 속성은 표시만 할 수 있습니다.

MQSeries에는 속성을 참조하는 두 가지 방식이 있습니다.

- PCF 이름을 사용하는 경우. 예: *MaxMsgLength*
- MQSC 이름을 사용하는 경우. 예: *MAXMSGL*

속성의 공식적인 이름은 PCF 이름입니다. 이 서적에서는 MQSC 기능의 사용이 중요하므로 주어진 속성의 PCF 이름보다는 MQSC 이름이 예에서 더 많이 사용될 것입니다.

MQSeries 큐 관리자

큐 관리자는 응용프로그램에 큐잉 서비스를 제공하며 큐 관리자에 속해 있는 큐를 관리합니다. 큐 관리자는 다음을 확인합니다.

- 수신된 명령에 따라 오브젝트 속성이 변경되는지 여부.
- 적절한 조건이 충족될 때 트리거 이벤트나 도구 이벤트와 같은 특수 이벤트가 생성되는지 여부.
- MQPUT 호출을 통해 응용프로그램에서 요청한 대로 메시지를 올바른 큐에 넣는지 (put) 여부. 이와 같이 수행할 수 없으면 응용프로그램에 통보되고 적절한 이유 코드가 주어집니다.

각각의 큐는 하나의 큐 관리자에 속하며 그 큐 관리자의 로컬 큐가 됩니다. 또 응용프로그램이 연결되는 큐 관리자는 해당 응용프로그램의 로컬 큐 관리자가 됩니다. 응용프로그램에 있어서, 해당 로컬 큐 관리자에 속하는 큐는 로컬 큐입니다.

리모트 큐는 또다른 큐 관리자에 속하는 큐입니다.

오브젝트

리모트 큐 관리자는 로컬 큐 관리자 이외의 큐 관리자입니다. 리모트 큐 관리자는 네트워크를 건너 리모트 기계에 존재할 수도 있고 로컬 큐 관리자와 동일한 컴퓨터에 존재할 수도 있습니다.

MQSeries는 동일한 시스템에서 여러 큐 관리자를 지원합니다.

MQI 호출

일부 MQI 호출에 큐 관리자 오브젝트를 사용할 수 있습니다. 예를 들어 MQI 호출 MQINQ를 사용하여 큐 관리자 오브젝트의 속성에 관해 조회할 수 있습니다.

주: 메시지는 큐 관리자 오브젝트에 넣을(put) 수 없습니다. 메시지는 언제나 큐 관리자 오브젝트가 아닌 큐 오브젝트에 넣습니다(put).

MQSeries 큐

큐는 다음을 사용하여 MQSeries에 정의됩니다.

- 적절한 MQSC DEFINE 명령
- PCF Create Queue 명령

명령은 큐의 유형과 그 속성을 지정합니다. 예를 들어 로컬 큐 오브젝트에는 MQI 호출 시, 응용프로그램이 해당 큐를 참조할 때 발생하는 사항을 지정하는 속성이 있습니다. 속성의 예는 다음과 같습니다.

- 응용프로그램이 큐에서 메시지를 검색할 수 있는지 여부(GET 가능)
- 응용프로그램이 메시지를 큐에 넣을(put) 수 있는지 여부(PUT 가능)
- 큐에 대한 액세스가 한 응용프로그램 전용인지 또는 응용프로그램 사이에서 공유되는지 여부
- 한번에 큐에 저장할 수 있는 최대 메시지 수(최대 큐 용량)
- 큐에 넣을(put) 수 있는 최대 메시지 길이

큐 오브젝트 정의에 대한 자세한 정보는 *MQSeries 명령 참조서* 또는 *MQSeries Programmable System Management*를 참조하십시오.

큐 오브젝트 사용

MQSeries에는 여러 가지 유형의 큐 오브젝트가 있습니다. 각 유형의 오브젝트는 제품 명령을 통해 조작할 수 있으며 여러 방식으로 실제 큐와 연관됩니다.

- 로컬 큐 오브젝트

로컬 큐 오브젝트는 응용프로그램이 연결된 큐 관리자에 속하는 로컬 큐를 식별합니다. 모든 큐는 각 큐가 큐 관리자에 속한다는 점에서 로컬 큐이며, 해당 큐 관리자의 관점에서도 큐는 로컬 큐입니다.

- 리모트 큐 오브젝트

리모트 큐 오브젝트는 다른 큐 관리자에 속하는 큐를 식별합니다. 이 큐는 해당 큐 관리자의 로컬 큐로 정의해야 합니다. 리모트 큐 오브젝트를 정의할 때 지정한 정보를 사용하면 로컬 큐 관리자가 리모트 큐 관리자를 찾을 수 있기 때문에 리모트 큐로 갈 메시지가 올바른 큐 관리자로 가게 됩니다.

응용프로그램이 다른 큐 관리자의 큐로 메시지를 송신하려면, 여러 큐 관리자를 클러스터로 그룹화하지 않은 한 큐 관리자간에 트랜스미션 큐와 채널을 정의해야 합니다. 클러스터에 대한 자세한 정보는 70 페이지의 『클러스터를 사용한 리모트 관리』를 참조하십시오.

- 알리아스 큐 오브젝트

알리아스 큐 오브젝트를 사용하면 MQI 호출 시 응용프로그램이 오브젝트를 간접적으로 참조하여 큐에 액세스할 수 있습니다. 알리아스 큐 이름이 MQI 호출에 사용될 경우, 그 이름은 런타임 시 로컬 큐나 리모트 큐 이름으로 해석됩니다. 따라서 알리아스 큐 정의만을 변경해서 별명으로 해석된 새 큐 이름이 반영되도록 할 수 있기 때문에 응용프로그램을 변경하지 않아도 응용프로그램에 사용된 큐를 변경할 수 있습니다.

알리아스 큐는 큐가 아니라 다른 큐에 액세스하는 데 사용할 수 있는 오브젝트입니다.

- 모델 큐 오브젝트

모델 큐 오브젝트는 다이내믹 큐를 작성할 때 템플릿으로 사용되는 일련의 큐 속성을 정의합니다. 다이내믹 큐는 응용프로그램이 큐 이름으로 모델 큐의 이름을 지정하여 MQOPEN 요청을 발행할 때 큐 관리자에 의해 작성됩니다. 이런 방식으로 작성된 다이내믹 큐는 그 속성을 모델 큐 정의에서 가져오는 로컬 큐입니다. 다이내믹 큐 이름은 응용프로그램에서 지정하거나, 큐 관리자가 이름을 생성하여 응용프로그램으로 리턴할 수도 있습니다.

이런 방식으로 정의된 다이내믹 큐는 제품을 재시작하면 없어지는 템포러리 큐 또는 재시작해도 계속 남아있는 퍼머넌트 큐입니다.

MQSeries에서 사용되는 특정 로컬 큐

MQSeries에서는 그 조작과 관련된 특정 목적에 일부 로컬 큐를 사용합니다. 그러한 로컬 큐를 MQSeries에서 사용하려면 반드시 정의해야 합니다.

응용프로그램 큐: MQI를 통해 응용프로그램에 사용되는 큐는 응용프로그램 큐라고 합니다. 이 큐는 응용프로그램이 연결되는 큐 관리자의 로컬 큐이거나 다른 큐 관리자가 소유하는 리모트 큐가 될 수 있습니다.

응용프로그램은 로컬 큐나 리모트 큐에 메시지를 넣을(put) 수 있습니다. 그러나 로컬 큐에서만 메시지를 가져올(get) 수 있습니다.

이니시에이션 큐: 이니시에이션 큐는 트리거에 사용되는 큐입니다. 큐 관리자는 트리거 이벤트가 발생할 때 트리거 메시지를 이니시에이션 큐에 넣습니다(put). 트리거 이벤트는 큐 관리자가 감지하는 조건의 논리적 결합입니다. 예를 들어 트리거 이벤트는 큐의 메시지 수가 사전정의된 용량에 도달할 때 생성될 수 있습니다. 이 이벤트가 생성되면 큐 관리자가 트리거 메시지를 지정된 이니시에이션 큐에 넣습니다(put). 이 트리거 메시지는 이니시에이션 큐를 모니터링하는 특수 응용프로그램인 트리거 모니터에 의해 검색됩니다. 그런 다음, 트리거 모니터는 트리거 메시지에 지정되어 있는 응용프로그램을 시작합니다.

큐 관리자가 트리거를 사용하는 경우, 그 큐 관리자에 최소한 하나의 이니시에이션 큐를 정의해야 합니다.

61 페이지의 『트리거를 위한 오브젝트 관리』와 320 페이지의 『runmqtrm(트리거 모니터 시작)』을 참조하십시오. 트리거에 대한 자세한 정보는 *MQSeries Application Programming Guide*를 참조하십시오.

트랜스미션 큐: 트랜스미션 큐는 리모트 큐 관리자로 갈 메시지를 임시 저장합니다. 로컬 큐 관리자로부터 메시지를 직접 수신할 리모트 큐 관리자마다 최소한 하나의 트랜스미션 큐가 정의되어 있어야 합니다. 트랜스미션 큐는 리모트 관리에도 사용됩니다. 71 페이지의 『MQSC 명령을 사용한 로컬 큐 관리자에서의 리모트 관리』를 참조하십시오. 분산 큐잉에서의 트랜스미션 큐 사용에 대한 정보는 *MQSeries 상호통신*을 참조하십시오.

클러스터 트랜스미션 큐: 클러스터 안의 각 큐 관리자에게는 SYSTEM.CLUSTER.TRANSMIT.QUEUE라고 하는 클러스터 트랜스미션 큐가 있습니다. 이 큐에 대한 정의는 Version 5.1의 모든 큐 관리자에서 디폴트로 작성됩니다.

클러스터에 속하는 큐 관리자는 클러스터 트랜스미션 큐의 메시지를 동일한 클러스터에 있는 다른 큐 관리자에 송신할 수 있습니다.

클러스터 큐 관리자는 클러스터에 속하지 않는 큐 관리자와 통신할 수 있습니다. 이를 위해서는 클러스터 안의 큐 관리자부터 클러스터 외부의 큐 관리자까지의 채널과 트랜스미션 큐를 기존의 분산 큐잉 환경에서와 같은 방식으로 정의해야 합니다.

이름을 해석하는 동안에는 클러스터 트랜스미션 큐가 디폴트 트랜스미션 큐보다 우선합니다. 클러스터에 속하지 않는 큐 관리자가 리모트 큐에 메시지를 넣을(put) 경우, 데스티네이션 큐 관리자와 이름이 같은 트랜스미션 큐가 없으면 디폴트 트랜스미션 큐를 사용하는 것이 디폴트 조치입니다.

큐 관리자가 클러스터에 속하는 경우에는, 데스티네이션 큐 관리자가 클러스터의 속하지 않는 경우를 제외하고는 디폴트 조치로 SYSTEM.CLUSTER.TRANSMIT.QUEUE를 사용합니다.

데드-레터 큐: 데드-레터 큐는 올바른 목적지로 라우트될 수 없는 메시지를 저장합니다. 이런 경우는 예를 들어 데스티네이션 큐가 꽉 찼을 때 발생합니다. 제공된 데드-레터 큐는 SYSTEM.DEAD.LETTER.QUEUE라고 합니다. 이 큐는 다른 플랫폼에서 미배달 메시지 큐라고도 합니다.

분산 큐잉에서는 관련된 큐 관리자마다 데드-레터 큐를 정의해야 합니다.

명령 큐: SYSTEM.ADMIN.COMMAND.QUEUE라고 하는 명령 큐는 권한이 있는 응용프로그램이 처리를 위해 MQSeries 명령을 송신할 수 있는 로컬 큐입니다. 이러한 명령은 명령 서버라고 하는 MQSeries 구성요소에서 검색합니다. 명령 서버는 명령을 검증하여 큐 관리자가 처리하도록 올바른 명령을 전달하고, 그 리플라이-투 큐에 응답을 리턴합니다.

명령 큐는 작성되는 큐 관리자마다 자동으로 작성됩니다.

리플라이-투 큐: 응용프로그램이 요청 메시지를 송신하면 메시지를 수신한 응용프로그램은 송신 응용프로그램으로 응답 메시지를 다시 송신할 수 있습니다. 이 메시지는 보통 송신 응용프로그램의 로컬 큐인 리플라이-투 큐라고 하는 큐에 넣습니다(put). 리플라이-투 큐의 이름은 송신 응용프로그램에 의해 메시지 설명자의 일부로 지정됩니다.

이벤트 큐: MQSeries Version 5.1은 MQI 응용프로그램과는 별도로 큐 관리자를 모니터링하는 데 사용되는 도구 이벤트를 지원합니다. 도구 이벤트는 여러 경우에 생성될 수 있는데 그 예는 다음과 같습니다.

- 사용 불가능하거나 존재하지 않는 큐에 응용프로그램이 메시지를 넣고자(put) 할 경우
- 큐가 꽉 찬 경우
- 채널을 시작할 경우

도구 이벤트가 발생하면 큐 관리자는 이벤트 메시지를 이벤트 큐에 넣습니다(put). 이렇게 하면 그 이벤트가 문제를 나타낼 때 모니터링 응용프로그램이 메시지를 읽고 관리자에게 알리거나 수정 조치를 시작할 수 있습니다.

주: 트리거 이벤트는 트리거 이벤트가 같은 조건에서 발생하지 않는다는 점에서 도구 이벤트와 매우 다르며, 이벤트 메시지를 생성하지 않습니다.

도구 이벤트에 대한 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.

프로세스 정의

프로세스 정의 오브젝트는 MQSeries 큐 관리자의 트리거 이벤트에 대한 응답으로 시작되는 응용프로그램을 정의합니다. 자세한 정보는 14 페이지의 『이니시에이션 큐』를 참조하십시오.

오브젝트

프로세스 정의 속성에는 응용프로그램 ID, 응용프로그램 유형 및 응용프로그램 고유 데이터가 있습니다.

프로세스 정의를 작성하려면 MQSC 명령 DEFINE PROCESS나 PCF 명령 Create Process를 사용하십시오.

채널

채널은 하나의 큐 관리자부터 다른 큐 관리자까지의 통신 경로를 제공하는 오브젝트입니다. 채널은 분산 메시지 큐잉에 사용되어 하나의 큐 관리자에서 다른 큐 관리자로 메시지를 이동합니다. 채널은 기저의 통신 프로토콜과는 관련이 없습니다. 큐 관리자는 같은 플랫폼에 있거나 다른 플랫폼에 있을 수 있습니다. 큐 관리자들이 서로 통신하려면 메시지를 송신하는 큐 관리자에 하나의 채널 오브젝트를 정의하고, 메시지를 수신하는 큐 관리자에 또 다른 보완 채널 오브젝트를 정의해야 합니다.

채널과 그 사용 방법에 대한 정보는 *MQSeries 상호통신*과 72 페이지의 『리모트 관리를 위한 채널 및 트랜스미션 큐 준비』를 참조하십시오.

클러스터

분산 큐잉을 사용하는 일반적인 MQSeries 네트워크에서는 모든 큐 관리자가 독립적입니다. 하나의 큐 관리자가 메시지를 다른 큐 관리자에 송신해야 할 경우, 그 큐 관리자에게는 트랜스미션 큐, 리모트 큐 관리자에 대한 채널 및 메시지를 송신할 모든 큐에 대한 리모트 큐 정의가 정의되어 있어야 합니다.

클러스터는 복잡한 트랜스미션 큐, 채널, 큐 정의가 없어도 단일 네트워크에서 큐 관리자들이 서로 직접 통신할 수 있도록 설정한 큐 관리자의 그룹입니다.

주: MQSeries 클러스터는 OpenVMS 클러스터와는 다릅니다. 클러스터라는 용어가 사용되면 이는 MQSeries 큐 관리자 클러스터를 가리킵니다. OpenVMS 클러스터는 언제나 *OpenVMS 클러스터*라고 합니다. OpenVMS 클러스터에 대한 정보는 235 페이지의 『제16장 OpenVMS용 MQSeries 및 클러스터링』을 참조하십시오.

클러스터에 대한 정보는 69 페이지의 『제6장 리모트 MQSeries 오브젝트 관리』와 *MQSeries Queue Manager Clusters*를 참조하십시오.

이름 목록

이름 목록은 다른 MQSeries 오브젝트의 목록이 들어 있는 MQSeries 오브젝트입니다. 일반적으로 이름 목록은 트리거 모니터와 같은 응용프로그램에서 사용되어 큐의 그룹을 식별합니다. 이름 목록을 사용하면 응용프로그램과는 별도로 이름 목록을 관리할 수 있다는 장점이 있습니다. 따라서 이름 목록을 사용하는 응용프로그램을 중지하지 않아도 이름 목록을 갱신할 수 있습니다. 뿐만 아니라 한 응용프로그램이 실패해도 이름 목록에는 아무런 영향이 없으므로 나머지 응용프로그램이 계속 이름 목록을 사용할 수 있습니다.

또한 이름 목록은 큐 관리자 클러스터에 사용되어 둘 이상의 MQSeries 오브젝트가 참조하는 클러스터의 목록을 관리할 수 있도록 합니다.

시스템 디폴트 오브젝트

시스템 디폴트 오브젝트는 큐 관리자가 작성될 때마다 자동으로 작성되는 일련의 오브젝트 정의입니다. 응용프로그램에 사용할 수 있도록 설치 시 이러한 오브젝트 정의를 복사하고 수정할 수 있습니다. 디폴트 오브젝트 이름에는 SYSTEM.DEF라는 접두부가 있습니다. 예를 들어 디폴트 로컬 큐는 SYSTEM.DEFAULT.LOCAL.QUEUE이고, 디폴트 수신자 채널은 SYSTEM.DEF.RECEIVER입니다. 이러한 오브젝트의 이름은 바꿀 수 없으며 이들 이름의 디폴트 오브젝트가 필요합니다.

오브젝트를 정의할 때는 명확하게 지정하지 않은 속성이 해당 디폴트 오브젝트에서 복사됩니다. 예를 들어 로컬 큐를 정의하는 경우, 지정하지 않은 속성은 디폴트 큐 SYSTEM.DEFAULT.LOCAL.QUEUE에서 가져옵니다.

시스템 디폴트에 대한 자세한 정보는 341 페이지의 『부록B. 시스템 디폴트』를 참조하십시오.

로컬 관리 및 리모트 관리

로컬 관리란 로컬 시스템에 정의한 큐 관리자에 대해 관리 작업을 수행하는 것을 뜻합니다. 예를 들어 TCP/IP 터미널 에뮬레이션 프로그램 **Telnet**을 통해 다른 시스템에 액세스한 다음 그 시스템에서 관리를 수행할 수 있습니다. MQSeries에서 이 관리는 채널이 관련되지 않고 운영 체제에 의해 통신이 관리되기 때문에 로컬 관리로 간주될 수 있습니다.

MQSeries는 리모트 관리라고 하는 것을 통해 단일 지점에서의 관리를 지원합니다. 다른 시스템에서 처리되는 로컬 시스템에서 명령을 발행할 수 있습니다. 적절한 채널을 정의해야 하지만 해당 시스템에 로그인할 필요는 없습니다. 대상 서버의 큐 관리자와 명령 서버가 실행 중이어야 합니다. 예를 들어, 리모트 명령을 발행하여 리모트 큐 관리자의 큐 정의를 변경할 수 있습니다.

큐 관리자를 작성하거나 시작하고, 명령 서버를 시작하는 것과 같은 명령은 이 방법으로 발행할 수 없습니다. 이런 유형의 작업을 수행하려면 리모트 시스템에 로그인하여 거기서 명령을 발행하거나 명령을 발행할 수 있는 프로세스를 작성해야 합니다.

클라이언트 및 서버

MQSeries는 MQSeries 응용프로그램에 대한 클라이언트-서버 구성을 지원합니다.

클라이언트 및 서버

MQSeries 클라이언트는 시스템에 설치되는 *MQSeries* 제품의 일부로 응용프로그램의 MQI 호출을 승인하여 *MQI* 서버 시스템으로 전달합니다. 서버에서 해당 큐 관리자에 의해 처리됩니다. 보통은 클라이언트와 서버가 다른 시스템에 상주하지만 같은 시스템에 있을 수도 있습니다.

MQI 서버는 하나 이상의 클라이언트에 큐잉 서비스를 제공하는 큐 관리자입니다. 모든 *MQSeries* 오브젝트(예:큐)는 큐 관리자 시스템, 즉 *MQI* 서버 시스템에만 존재합니다. 서버는 일반 로컬 *MQSeries* 응용프로그램도 지원할 수 있습니다.

MQI 서버와 일반 큐 관리자의 다른 점은 서버에 각 클라이언트와의 전용 통신 링크가 있다는 점입니다. 클라이언트와 서버의 채널 작성에 대한 자세한 정보는 *MQSeries* 상호통신을 참조하십시오.

일반적인 클라이언트 지원에 대한 정보는 *MQSeries* 클라이언트를 참조하십시오.

클라이언트-서버 환경에서의 *MQSeries* 응용프로그램

서버에 연결되면, 클라이언트 *MQSeries* 응용프로그램은 로컬 응용프로그램과 똑같은 방식으로 MQI 호출을 발행할 수 있습니다. 클라이언트 응용프로그램은 MQCONN 호출을 발행하여 지정된 큐 관리자에 연결됩니다. 연결 요청에서 리턴된 연결 핸들을 지정하는 추가 MQI 호출이 이 큐 관리자에 의해 처리됩니다.

응용프로그램을 적절한 클라이언트 라이브러리에 연결해야 합니다. 자세한 정보는 *MQSeries* 클라이언트를 참조하십시오.

큐 관리자 기능 확장

큐 관리자가 제공하는 기능은 다음을 통해 확장될 수 있습니다.

- 사용자 종료
- 설치가능 서비스

사용자 종료

사용자 종료는 사용자가 고유 코드를 큐 관리자 기능에 삽입하는 메커니즘을 제공합니다. 지원되는 사용자 종료는 다음과 같습니다.

- 채널 종료

이 종료는 채널이 작동되는 방식을 변경합니다. 채널 종료는 *MQSeries 상호통신*에 설명되어 있습니다.

- 데이터 변환 종료

이 종료는 응용프로그램에 넣어(put) 데이터의 형식을 변환할 수 있는 소스 코드 단편을 작성합니다. 데이터 변환 종료는 *MQSeries Application Programming Guide*에 설명되어 있습니다.

- 클러스터 워크로드 종료

이 종료에서 수행하는 기능은 종료 제공업체에서 정의합니다. 호출 정의에 대한 정보는 *MQSeries Queue Manager Clusters*를 참조하십시오.

모든 유형의 종료는 분산 큐잉과 관련됩니다. 이러한 종료와 그 사용 방법에 대한 자세한 정보는 *MQSeries 상호통신*을 참조하십시오.

설치가능 서비스

설치 가능 서비스는 다중 시작점을 가진 인터페이스(API)를 형식화한다는 점에서 종료보다 광범위합니다.

설치가능 서비스가 구현된 것을 서비스 구성요소라고 합니다. 제품과 함께 제공된 구성요소를 사용하거나 사용자에게 필요한 기능을 수행하는 구성요소를 직접 작성하여 사용할 수도 있습니다.

현재 다음과 같은 설치가능 서비스가 제공됩니다.

- 권한 서비스

권한 서비스를 사용하면 고유한 보안 기능을 빌드할 수 있습니다.

서비스를 구현하는 디폴트 서비스 구성요소는 제품과 함께 제공되는 OAM(Object Authority Manager)입니다. 디폴트로 OAM은 활성 상태이기 때문에 구성을 위한 어떤 작업도 수행할 필요가 없습니다. 권한 서비스 인터페이스를 사용하여 OAM을 바꾸거나 중대하는 다른 구성요소를 작성할 수 있습니다. OAM에 대한 자세한 정보는 85 페이지의 『제7장 MQSeries 오브젝트 보호』를 참조하십시오.

- 이름 서비스

이름 서비스를 사용하면 응용프로그램이 리모트 큐를 마치 로컬 큐인 것처럼 식별하게 하여 큐를 공유할 수 있습니다. 이름 서비스를 구현하는 디폴트 서비스 구성요소는 MQSeries Version 5.1에서 제공합니다. 이 구성요소는 OSF(Open Software Foundation) DCE(Distributed Computing Environment)를 사용합니다. 예를 들어

기능 확장

DCE를 설치하지 않았을 경우에는 고유한 이름 서비스 구성요소를 쓸 수도 있습니다. 디폴트로, 이름 서비스는 비활성 상태입니다.

181 페이지의 『제12장 이름 서비스 사용』과 *MQSeries Programmable System Management*를 참조하십시오.

보안

MQSeries Version 5 제품에는 보안을 제공하는 두 가지 방법이 있습니다.

- OAM(Object Authority Manager) 기능
- DCE 보안

OAM(Object Authority Manager) 기능

MQI 호출, 명령 및 오브젝트 액세스에 대한 권한 부여는 디폴트로 사용 가능한 OAM(Object Authority Manager)에서 제공합니다. MQSeries 엔티티에 대한 액세스는 MQSeries 사용자 그룹과 OAM을 통해 제어됩니다. 필요한 경우 관리자가 권한을 부여하거나 해지할 수 있는 명령행 인터페이스를 제공합니다.

권한 서비스 구성요소 작성에 대한 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.

DCE 보안

DCE GSS(Generic Security Service)를 사용하는 채널 종료는 MQSeries에서 제공됩니다. 자세한 정보는 *MQSeries 상호통신*을 참조하십시오.

트랜잭션식 지원

응용프로그램은 갱신 세트를 작업 단위로 분류할 수 있습니다. 이들 갱신은 대개는 논리적으로 관련되며, 모두 데이터 무결성을 보존할 수 있어야 합니다. 하나의 갱신이 성공해도 다른 갱신에 실패하면 데이터 무결성이 손실됩니다.

작업 단위는 성공적으로 완료되어야 확약됩니다. 이 때 작업 단위 안에서 갱신된 모든 사항은 영구적이 되거나 되돌릴 수 없습니다. 작업 단위에 실패하면 모든 갱신이 백아웃됩니다. 동기점 통합이 바로 작업 단위가 확약되거나 무결성을 가지고 백아웃되는 프로세스입니다.

로컬 작업 단위에서는 MQSeries 큐 관리자의 자원만 갱신됩니다. 여기에서 1단계 확약 프로세스를 사용하여 큐 관리자 자체에 의해 동기점 통합이 제공됩니다.

글로벌 작업 단위에서는 XA 준수 데이터베이스 등 다른 자원 관리자에 속하는 자원도 갱신됩니다. 여기에서는 2단계 확약 프로시저가 사용되며, 큐 관리자 자체에 의해 작업 단위가 통합될 수 있습니다.

자세한 정보는 129 페이지의 『제10장 트랜잭션식 지원』을 참조하십시오.

제2장 MQSeries 관리 소개

이 장에서는 MQSeries 관리에 대해 소개합니다.

MQSeries 관리에는 큐 관리자, 큐, 프로세스, 이름 목록, 클러스터, 채널과 같은 MQSeries 오브젝트를 작성, 시작, 대체, 보기, 정지 및 삭제하는 작업이 포함됩니다.

이 장에는 다음 절이 포함됩니다.

- 『로컬 관리 및 리모트 관리』
- 24 페이지의 『제어 명령을 사용하여 관리 작업 수행』
- 24 페이지의 『MQSC 명령을 사용하여 관리 작업 수행』
- 25 페이지의 『PCF 명령을 사용하여 관리 작업 수행』
- 25 페이지의 『MQSeries 파일 이름 이해』
- 27 페이지의 『대소문자 구분 이해』

로컬 관리 및 리모트 관리

MQSeries 오브젝트를 로컬이나 리모트로 관리할 수 있습니다.

로컬 관리는 로컬 시스템에 정의한 큐 관리자에서 관리 작업을 수행하는 것을 말합니다. TCP/IP 터미널 에뮬레이션 프로그램 Telnet등을 통해 다른 시스템에 액세스하고 거기에서 관리 작업을 할 수도 있습니다. MQSeries에서는 다른 채널을 사용하지 않기 때문에, 다시 말해 운영 체제에서 통신을 관리하기 때문에 이와 같이 다른 시스템에서 수행하는 관리를 로컬 관리라고 합니다.

MQSeries는 리모트 관리라고 하는 단일 지점에서의 관리를 지원하므로, 다른 시스템에서 처리되는 로컬 시스템에서 명령을 발행할 수 있습니다. 적절한 채널을 정의해야 하지만 해당 시스템에 로그인할 필요는 없습니다. 대상 서버의 큐 관리자와 명령 서버가 실행 중이어야 합니다. 예를 들어, 리모트 명령을 발행하여 리모트 큐 관리자의 큐 정의를 변경할 수 있습니다.

큐 관리자를 작성하거나 시작하고, 명령 서버를 시작하는 것과 같은 명령은 이 방법으로 발행할 수 없습니다. 이런 유형의 작업을 수행하려면 리모트 시스템에 로그인하여 거기에서 명령을 발행하거나 명령을 발행할 수 있는 프로세스를 작성해야 합니다.

리모트 관리에 대한 자세한 설명은 69 페이지의 『제6장 리모트 MQSeries 오브젝트 관리』를 참조하십시오.

제어 명령을 사용하여 관리 작업 수행

제어 명령을 사용하면 큐 관리자 자체에서 관리 작업을 수행할 수 있습니다.

제어 명령에 대한 자세한 정보는 29 페이지의 『제3장 제어 명령을 사용한 큐 관리자 관리』를 참조하십시오.

MQSC 명령을 사용하여 관리 작업 수행

MQSeries 명령(MQSC)을 사용하여 큐 관리자, 채널, 큐, 프로세스 정의를 비롯한 큐 관리자 오브젝트를 관리할 수 있습니다. 예를 들어, 지정한 큐를 정의하고 대체하고 표시하고 삭제하는 데 사용하는 명령들이 있습니다.

MQSC 명령은 명령행에서 **runmqsc** 제어 명령을 호출하여 실행합니다. MQSC 명령을 실행하는 방법은 다음과 같습니다.

- 키보드에서 명령을 입력하여 대화식으로 실행. 41 페이지의 『MQSC 기능의 대화식 사용』을 참조하십시오.
- ASCII 텍스트 파일의 명령 순서로 실행. 44 페이지의 『텍스트 파일에서 MQSC 명령 실행』을 참조하십시오.

명령에 설정한 플래그에 따라 **runmqsc** 명령을 다음 세 모드에서 실행할 수 있습니다.

- 확인 모드: MQSC 명령이 로컬 큐 관리자에서 확인되지만 실제로 실행되지는 않는 모드.
- 직접 모드: MQSC 명령이 로컬 큐 관리자에서 실행되는 모드
- 간접 모드: MQSC 명령이 리모트 큐 관리자에서 실행되는 모드

MQSC 기능 및 텍스트 파일 사용에 대한 자세한 정보는 44 페이지의 『텍스트 파일에서 MQSC 명령 실행』을 참조하십시오. **runmqsc** 명령에 대한 자세한 정보는 316 페이지의 『runmqsc(MQSeries 명령 실행)』를 참조하십시오.

MQSC에 지정되는 오브젝트 속성은 대소문자가 구별되지 않기는 하지만 이 책에서 대문자로 표시되어 있습니다(예: RQMNAME). MQSC 속성 이름은 8자를 초과할 수 없습니다.

MQSC 명령은 AS/400 및 OS/390 등의 다른 플랫폼에서도 사용할 수 있습니다.

MQSC 명령에 대한 요약은 349 페이지의 『부록D. 명령 세트 비교』를 참조하십시오.

각 MQSC 명령 및 구문에 대한 설명은 *MQSeries* 명령 참조서를 참조하십시오.

로컬 관리에서의 MQSC 명령 사용에 대한 자세한 정보는 40 페이지의 『MQSC 명령을 사용한 로컬 관리 작업의 수행』을 참조하십시오.

PCF 명령을 사용하여 관리 작업 수행

MQSeries PCF(Programmable Command Format) 명령은 관리 작업을 관리 프로그램에 프로그래밍할 수 있도록 하기 위해 만들어진 것입니다. 이렇게 해서 프로그램으로 큐와 프로세스 정의를 작성하고 큐 관리자를 변경할 수 있습니다.

PCF 명령은 MQSC 기능에서 제공하는 것과 같은 범위의 기능을 제공합니다.

자세한 정보는 65 페이지의 『PCF 명령』을 참조하십시오.

PCF 데이터 구조와 그 구조를 구현하는 방법에 대한 자세한 설명은 *MQSeries Programmable System Management*를 참조하십시오.

MQAI(MQSeries Administration Interface)를 사용하면 PCF 메시지에 프로그래밍 방식으로 보다 쉽게 액세스할 수 있습니다. 자세한 설명은 66 페이지의 『MQAI를 통한 간편한 PCF 사용』을 참조하십시오.

MQSC 및 PCF의 속성

MQSC에 지정된 오브젝트 속성은 대소문자를 구분하지 않지만 이 서적에서는 대문자로 표시됩니다(예: RQMNAME). 속성 이름은 8자를 초과할 수 없으므로, QDPHIEV와 같은 몇몇 속성 이름은 그 의미를 파악하기가 쉽지 않습니다. PCF의 오브젝트 속성은 기울임꼴로 표시되며, 8자로 제한되지 않으므로 읽기가 쉽습니다. RQMNAME에 해당하는 PCF 명령은 *RemoteQMgrName*이고, QDPHIEV에 해당하는 PCF 명령은 *QDepthHighEvent*입니다.

이스케이프 PCF

이스케이프 PCF는 메시지 텍스트 내에 MQSC 명령을 포함하는 PCF 명령입니다. PCF를 사용하여 명령을 리모트 큐 관리자로 보낼 수 있습니다. 이스케이프 PCF 사용에 대한 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.

MQSeries 파일 이름 이해

파일에는 각각의 MQSeries 큐, 큐 관리자, 이름 목록 및 프로세스 오브젝트가 나타납니다. 오브젝트 이름이 올바르지 않은 파일 이름인 경우도 있으므로, 필요하면 큐 관리자가 오브젝트 이름을 올바른 파일 이름으로 변환합니다.

큐 관리자 디렉토리 경로는 다음 부분으로 이루어집니다.

- 접두부 - 이름의 첫째 부분:

MQS_ROOT:[MQM]

이 접두부는 큐 관리자의 구성 파일에 정의되어 있습니다.

MQSeries 이름 이해

- 리터럴:
QMGRS
- 코드화된 큐 관리자 이름으로, 올바른 디렉토리 이름으로 변환된 큐 관리자 이름입니다. 예를 들어
QUEUE.MANAGER

큐 관리자는 다음과 같이 나타납니다.
QUEUE\$MANAGER

이 프로세스를 이름 변환이라고 합니다.

큐 관리자 이름 변환

MQSeries에서는 큐 관리자 이름을 48자까지 지정할 수 있습니다.

예를 들어 큐 관리자 이름을 다음과 같이 지정할 수 있습니다.

```
QUEUE.MANAGER.ACCOUNTING.SERVICES
```

그러나 각 큐 관리자는 파일로 나타나고 파일 이름의 최대 길이는 제한되므로 이름에 사용되는 글자 수도 제한됩니다. 따라서 오브젝트를 나타내는 파일 이름은 파일 시스템의 요구사항에 맞게 자동으로 변환됩니다.

QUEUE.MANAGER라는 이름의 큐 관리자 예를 사용하여 큐 관리자 이름 변환을 제어하는 규칙은 다음과 같습니다.

1. 개별 문자 변환:
 - becomes \$
 - / becomes _
 - % becomes _
2. 이름이 올바르지 않은 경우:
 - a. 8자로 자릅니다.
 - b. 3자로 된 숫자 접미부를 추가합니다.

예를 들어 디폴트 접두부를 사용할 경우 큐 관리자 이름은 다음과 같습니다.

```
MQS_ROOT:[MQM.QMGRS.QUEUE$MANAGER]
```

또한 변환 알고리즘을 사용하면 대소문자를 구분하지 않는 파일 시스템에서 대소문자에서만 차이가 있는 이름들을 구분할 수 있습니다.

오브젝트 이름 변환

오브젝트 이름이 올바른 파일 시스템 이름이 아닌 경우도 있는데, 이 때는 오브젝트 이름을 변환해야 합니다. 오브젝트 이름을 변환할 때 사용하는 방법은 큐 관리자 이름을 변환할 때 사용하는 방법과는 다릅니다. 이는 시스템마다 큐 관리자 이름은 몇 개 되지

않지만 각 큐 관리자에 다수의 다른 오브젝트가 있을 수 있기 때문입니다. 프로세스 정의, 큐 및 이름 목록만이 파일 시스템에 나타내므로 채널은 영향을 받지 않습니다.

변환 프로세스를 통해 새 이름을 생성하면 원래 오브젝트 이름과의 관계가 복잡해집니다. **dspmqls** 명령을 사용하면 실제 오브젝트 이름과 변환된 오브젝트 이름 사이에서 변환할 수 있습니다.

큐 파일 이름은 “Q” 문자로 시작합니다.

오브젝트 이름 지정에 대한 자세한 정보는 259 페이지의 『MQSeries 오브젝트 이름 지정 규칙』을 참조하십시오.

대소문자 구분 이해

제어 명령의 대소문자 구분

일반적으로 OpenVMS를 대소문자를 구분하지 않는 운영 체제라고 합니다. 따라서 다음 세 명령은 모두 "QUEUEMANAGER"라는 큐 관리자를 작성합니다.

```
$ crtmqm QueueManager
$ crtmqm queuemanager
$ crtmqm QUEUEMANAGER
```

Compaq OpenVMS용 MQSeries를 사용하여 큐 관리자 이름 또는 비슷한 매개변수 이름 양 옆에 큰 따옴표를 넣어 대소문자를 보호할 수 있습니다. 큰 따옴표를 사용하면 다음 세 명령이 서로 다른 큐 관리자를 작성합니다.

```
$ crtmqm "QueueManager"   QueueManager라는 큐 관리자를 작성합니다.
$ crtmqm "queuemanager"   queuemanager라는 큐 관리자를 작성합니다.
$ crtmqm "QUEUEMANAGER"   QUEUEMANAGER라는 큐 관리자를 작성합니다.
```

OpenVMS Version 7.2에는 다음과 같은 새 명령이 도입되었습니다.

```
$ set process /parse_style = ( traditional | extended )
```

이 명령은 OpenVMS가 대문자와 소문자를 처리하는 방법을 변경합니다.

set process /parse_style 명령을 사용하지 않거나, 이 명령을 **traditional** 옵션과 함께 사용하면 OpenVMS가 이전과 마찬가지로 대소문자를 구분하지 않습니다.

이 명령을 **extended** 옵션과 함께 사용하면 검색한 텍스트의 대소문자를 보존할 수 있도록 LIB\$GET_FOREIGN 런타임 라이브러리 루틴의 작동이 변경됩니다. MQSeries에서는 이 루틴을 사용하여 명령행 매개변수를 가져오므로, 매개변수를 큰 따옴표로 묶지 않더라도 매개변수의 대소문자가 보존됩니다.

MQSeries 이름 이해

예를 들어 다음 명령 순서는 세 개의 서로 다른 큐 관리자를 작성합니다. 다음 예에서는 매개변수가 큰 따옴표로 묶여 있지 않습니다.

```
$ set process /parse_style = extended
$ crtmqm QueueManager QueueManager라는 큐 관리자를 작성합니다.
$ crtmqm queuemanager queuemanager라는 큐 관리자를 작성합니다.
$ crtmqm QUEUEMANAGER QUEUEMANAGER라는 큐 관리자를 작성합니다.
```

OpenVMS `set process /parse_style` 명령은 대소문자 구분 외에도 여러 항목을 변경합니다. 시스템에 이 명령을 적용하기 전에 OpenVMS DCL Dictionary를 참조하면 명령에 대한 자세한 정보를 볼 수 있습니다.

MQSC 명령의 대소문자 구분

MQSeries 제어 명령(예: MQSC 기능을 실행하는 `runmqsc`)은 대소문자를 구분하지 않습니다.

MQSC 명령 및 그 속성은 대문자나 소문자로 다시 기록할 수 있습니다. 작은 따옴표로 묶여 있는 경우가 아니면 MQSC 명령의 오브젝트 이름은 대문자로 변환됩니다. 작은 따옴표를 사용하지 않으면 오브젝트가 대문자로 된 이름으로 처리됩니다. 자세한 정보는 *MQSeries* 명령 참조서를 참조하십시오.

제3장 제어 명령을 사용한 큐 관리자 관리

이 장에서는 큐 관리자와 명령 서버에서 조작을 수행하는 방법에 대해 설명합니다. 이 장에는 다음과 같은 절이 있습니다.

- 『제어 명령 사용』
- 30 페이지의 『큐 관리자 작성 지침』
- 34 페이지의 『디폴트 큐 관리자 작성』
- 34 페이지의 『큐 관리자 시작』
- 35 페이지의 『기존의 큐 관리자를 디폴트로 만들기』
- 35 페이지의 『큐 관리자 정지』
- 37 페이지의 『큐 관리자 재시작』
- 37 페이지의 『큐 관리자 삭제』
- 260 페이지의 『오브젝트 파일 보기』

제어 명령 사용

제어 명령을 사용하여 큐 관리자, 명령 서버 및 채널에서 조작을 수행할 수 있습니다. 표1에 나타난 것과 같이 제어 명령은 세 가지 범주로 구분됩니다.

표 1. 제어 명령의 범주

범주	설명
큐 관리자 명령	큐 관리자 제어 명령에는 큐 관리자와 명령 서버의 작성, 시작, 정지 및 삭제에 대한 명령이 있습니다.
채널 명령	채널 명령에는 채널 및 채널 시작기의 시작 및 종료에 대한 명령이 있습니다.
유틸리티 명령	유틸리티 명령에는 다음과 관련된 명령이 있습니다. <ul style="list-style-type: none">• MQSC 명령 실행• 변환 종료• 권한 관리• 큐 관리자 자원의 매체 이미지 레코드 및 복구• 트랜잭션 표시 및 해석• 트리거 모니터• MQSeries 오브젝트의 파일 이름 표시

채널 관리 작업에 대한 정보는 *MQSeries* 상호통신을 참조하십시오.

제어 명령 사용

Compaq OpenVMS용 MQSeries에서, DCL 프롬프트에 제어 명령을 입력합니다. 명령어와 플래그 자체는 대소문자를 구분하지 않지만, 매개변수는 OpenVMS 프로세스 옵션에 따라, 또는 해당 매개변수가 대소문자를 보호하기 위해 큰 따옴표로 묶었는지의

여부에 따라 대문자 변환여부가 결정됩니다. OpenVMS 명령과 큰 따옴표가 대소문자에 주는 영향에 대한 자세한 정보는 27 페이지의 『대소문자 구분 이해』를 참조하십시오.

다음은 일반적인 예입니다.

```
crtmqm -u system.dead.letter.queue "jupiter.queue.manager"
```

- 데드-레터 큐는 소문자로 입력되었더라도 SYSTEM.DEAD.LETTER.QUEUE가 될 수 있습니다. 그 문자가 대문자로 자동 변환되는가의 여부는 OpenVMS 명령인 **set process/parse_style**의 설정에 따라 다릅니다. 27 페이지의 『대소문자 구분 이해』를 참조하십시오.
- 큐 관리자 이름은 큰 따옴표로 묶여 있었기 때문에 "jupiter.queue.manager"로 지정됩니다("JUPITER.queue.manager"와는 다름).

그러므로, 예에서와 같이 명령을 정확히 입력해야 합니다.

큐 관리자 작성

큐 관리자는 자신과 연관된 자원, 특히 자신이 소유하고 있는 큐를 관리합니다. 큐 관리자는 MQSeries 오브젝트를 작성, 수정, 표시 및 삭제하기 위한 MQI(Message Queuing Interface) 호출 및 명령에 대한 큐잉 서비스를 응용프로그램에 제공합니다.

메시지와 큐에 대해 작업하기 전에 최소한 하나의 큐 관리자와 이와 연관된 오브젝트를 작성해야 합니다. 큐 관리자를 작성하려면 MQSeries 제어 명령 **crtmqm**을 사용하십시오. **crtmqm** 명령은 필요한 디폴트 오브젝트와 시스템 오브젝트를 자동으로 작성합니다. 디폴트 오브젝트는 사용자가 작성하는 모든 오브젝트 정의의 기준이 되며, 시스템 오브젝트는 큐 관리자 조작에 필요합니다. 큐 관리자와 해당 큐 관리자의 오브젝트가 작성되면, **strmqm** 명령을 사용하여 큐 관리자를 시작하십시오.

큐 관리자 작성 지침

큐 관리자를 작성하기 전에, 특히 제작 환경에서 고려해야 할 몇 가지 사항이 있습니다. 다음 점검 목록을 사용하십시오.

- 고유 큐 관리자 이름 지정
- 큐 관리자 수 제한
- 디폴트 큐 관리자 지정
- 데드-레터 큐 지정
- 디폴트 트랜스미션 큐 지정
- 필요한 로그 기록 매개변수 지정

- 큐 관리자 작성 후 구성 파일 백업

이 목록의 작업들은 다음 절에서 설명합니다.

고유 큐 관리자 이름의 지정

큐 관리자를 작성할 경우에는 네트워크의 어디에서도 큐 관리자의 이름이 동일하지 않도록 해야 합니다. 큐 관리자 이름은 작성시 점검되지 않으며, 고유한 이름을 사용하지 않으면 분산 큐잉에 채널을 사용할 수 없습니다.

고유한 이름을 사용하는 한 가지 방법은 각 큐 관리자 이름 앞에 고유한 노드 이름으로 접두부를 붙이는 것입니다. 예를 들어 노드가 accounts이면, 큐 관리자의 이름을 accounts.saturn.queue.manager로 지정할 수 있습니다. 여기서, saturn은 특정 큐 관리자를 식별하며 queue.manager는 모든 큐 관리자에 부여할 수 있는 확장자입니다. 확장자를 생략할 수는 있지만, accounts.saturn과 accounts.saturn.queue.manager는 다른 큐 관리자 이름입니다.

다른 엔터프라이즈와 통신하기 위해 MQSeries를 사용할 경우, 사용자 자신의 엔터프라이즈를 접두사로 포함시킬 수도 있습니다. 이해하기가 더 어려워지므로 이 예에서는 하지 않습니다.

주: 제어 명령의 큐 관리자 이름은 OpenVMS 프로세스 옵션에 따라, 또는 대소문자를 보호하기 위해 큐 관리자 이름을 큰 따옴표로 묶었는지의 여부에 따라 대문자 변환 여부가 결정됩니다. 즉, jupiter.queue.manager 및 JUPITER.queue.manager라는 이름을 사용하여 두 개의 큐 관리자를 작성할 수 있습니다. OpenVMS 프로세스 옵션과 큰 따옴표가 대소문자에 주는 영향에 대한 자세한 정보는 27 페이지의 『대소문자 구분 이해』를 참조하십시오.

큐 관리자 수 제한

자원에서 허용하는 수 만큼 큐 관리자를 작성할 수 있습니다. 그러나 각 큐 관리자에는 자체의 고유 자원이 필요하므로, 일반적으로 각각 10개의 큐가 있는 10개의 큐 관리자를 갖는 것보다는 한 노드에서 100개의 큐가 있는 하나의 큐 관리자를 갖는 것이 좋습니다.

제작 시스템에서는 하나의 큐 관리자를 사용하여 여러 노드를 실행하지만, 더 큰 서버 시스템에서는 여러 큐 관리자를 사용하여 실행하는 경우도 있습니다.

디폴트 큐 관리자 지정

디폴트 큐 관리자가 없어도 노드에 MQSeries를 구성할 수는 있지만, 각 노드는 디폴트 큐 관리자를 가지고 있어야 합니다.

큐 관리자를 작성하려면 **crtmqm** 명령을 사용하십시오. 이 명령과 그 매개변수에 대한 상세한 설명은 266 페이지의 『crtmqm(큐 관리자 작성)』을 참조하십시오.

디폴트 큐 관리자의 정의

디폴트 큐 관리자는 응용프로그램이 MQCONN 호출에서 큐 관리자 이름을 지정하지 않을 경우에 그 응용프로그램이 연결하는 큐 관리자입니다. 또한 큐 관리자 이름을 지정하지 않고 **runmqsc** 명령을 실행할 때, MQSC 명령을 처리하는 큐 관리자입니다.

디폴트 큐 관리자를 지정하는 방법

crtmqm 명령에 **-q** 플래그를 포함시키면 작성하는 큐 관리자를 디폴트 큐 관리자로 지정할 수 있습니다. 작성하는 큐 관리자를 디폴트 큐 관리자로 지정하지 않으려면 이 플래그를 생략하십시오.

큐 관리자를 디폴트로 지정하면 노드에 대한 기존의 디폴트 큐 관리자 스펙이 바뀝니다.

디폴트 큐 관리자를 변경할 경우 발생하는 사항

디폴트 큐 관리자를 변경할 경우에는 이로 인해 다른 사용자나 응용프로그램에 영향을 줄 수 있다는 점에 유의하십시오. 현재 연결된 응용프로그램은 추가 MQI 호출이 있을 경우, 원래 연결 호출의 핸들을 사용할 수 있으므로 영향을 받지 않습니다. 이 핸들은 호출이 동일한 큐 관리자로 전달되도록 합니다. 변경이후에 연결하는 응용프로그램은 새 디폴트 큐 관리자에 연결됩니다.

사용자의 의도일 수도 있지만, 디폴트를 변경하기 전에 이 점을 고려해야 합니다.

데드-레터 큐 지정

데드-레터 큐는 메시지가 올바른 목적지로 라우트될 수 없을 경우에 그 메시지를 넣는 (put) 로컬 큐입니다.

주의:

네트워크상의 각 큐 관리자에는 반드시 데드-레터 큐가 있어야 합니다. 큐 관리자에 데드-레터 큐가 없으면 응용프로그램의 오류로 인해 채널이 닫히거나 관리 명령에 대한 응답이 수신되지 않을 수도 있습니다.

예를 들어 응용프로그램이 다른 큐 관리자의 큐에 메시지를 넣으려고(put) 할 경우 잘못된 큐 이름이 제공되면, 채널이 정지되고 메시지는 트랜스미션 큐에 남게 됩니다. 그러면 다른 응용프로그램이 그 메시지들에 대해 이 채널을 사용할 수 없습니다.

큐 관리자에 데드-레터 큐가 있어도 채널에는 영향을 주지 않습니다. 미배달 메시지는 수신 엔드에서 데드-레터 큐에 넣어지고(put) 채널과 트랜스미션 큐는 사용 가능한 상태로 유지됩니다.

따라서 큐 관리자를 작성할 경우에는 -u 플래그를 사용하여 데드-레터 큐의 이름을 지정해야 합니다. 또한 MQSC 명령을 사용하여 큐 관리자의 속성을 대체하고 사용할 데드-레터 큐를 지정할 수도 있습니다. MQSC ALTER 명령의 예로 44 페이지의 『큐 관리자 속성 대체』를 참조하십시오.

데드-레터 큐에서 메시지가 발견되면, MQSeries와 함께 제공된 데드-레터 큐 핸들러를 사용하여 이 메시지들을 처리할 수 있습니다. 데드-레터 큐 핸들러와 데드-레터 큐에 위치하는 메시지 수를 줄이는 방법에 대한 자세한 정보는 109 페이지의 『제8장 MQSeries 데드-레터 큐 핸들러』를 참조하십시오.

디폴트 트랜스미션 큐 지정

트랜스미션 큐는 리모트 큐 관리자로 전송 중인 메시지들이 전송을 보류한 상태로 대기하는 로컬 큐입니다. 디폴트 트랜스미션 큐는 명확히 정의된 트랜스미션 큐가 없을 때 사용되는 큐입니다. 각 큐 관리자에는 디폴트 트랜스미션 큐가 지정될 수 있습니다.

큐 관리자를 작성할 경우 -d 플래그를 사용하여 디폴트 트랜스미션 큐의 이름을 지정해야 합니다. 이 경우에 실제로 큐가 생성되는 것이 아니므로, 나중에 명확하게 이를 수행해야 합니다. 자세한 정보는 50 페이지의 『로컬 큐에 대한 작업』을 참조하십시오.

필요한 로그 기록 매개변수 지정

crtmqm 명령에서 로그 기록 유형과 로그 파일의 경로 및 크기 등의 로그 기록 매개변수를 지정할 수 있습니다. 개발 환경에서는 디폴트 로그 기록 매개변수가 적절합니다. 그러나 다음과 같은 경우에는 디폴트를 변경할 수 있습니다.

- 많은 로그를 지원할 수 없는 저급 시스템 구성을 가지고 있는 경우.
- 동시에 여러 개의 긴 메시지가 큐에 있을 것으로 예상되는 경우

로그 기록 매개변수 지정에 대한 자세한 정보는

- **crtmqm** 명령에서 266 페이지의 『crtmqm(큐 관리자 작성)』을 참조하십시오.
- 구성 파일 사용에서 193 페이지의 『Log 스탠자』를 참조하십시오.

큐 관리자 작성 후 구성 파일 백업

다음과 같은 경우에 고려해야 할 두 개의 구성 파일이 있습니다.

1. 제품을 설치할 때, MQSeries 구성 파일(mqs.ini)이 작성됩니다. 이 파일에는 큐 관리자를 작성하거나 삭제할 때마다 갱신되는 큐 관리자 목록이 있습니다. 노트마다 하나의 mqs.ini 파일이 있습니다.
2. 새 큐 관리자를 작성할 때, 새로운 큐 관리자 구성 파일(qm.ini)이 자동으로 작성됩니다. 이 파일에는 큐 관리자에 대한 구성 매개변수가 있습니다.

이 파일들에 대한 백업을 만들어야 합니다. 나중에 사용자가 작성한 다른 큐 관리자에 문제가 있을 경우, 문제점을 제거하면 백업을 다시 사용할 수 있습니다. 일반적으로, 새로운 큐 관리자를 작성할 때마다 구성 파일을 백업해야 합니다.

큐 관리자 작성

구성 파일에 대한 자세한 정보는 183 페이지의 『제13장 MQSeries 구성』을 참조하십시오.

디폴트 큐 관리자 작성

디폴트 큐 관리자를 작성하려면 **crtmqm** 명령을 사용하십시오. q 플래그로 지정된 **crtmqm** 명령은 다음을 수행합니다.

- saturn.queue.manager라는 디폴트 큐 관리자를 작성합니다.
- 디폴트 오브젝트와 시스템 오브젝트를 작성합니다.
- 해당하는 디폴트 트랜스미션 큐와 데드-레터 큐의 이름들을 지정합니다.

```
crtmqm -q -d MY.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE "saturn.queue.manager"
```

여기서

-q 는 이 큐 관리자가 디폴트 큐 관리자임을 나타냅니다.

-d MY.DEFAULT.XMIT.QUEUE
디폴트 트랜스미션 큐의 이름입니다.

-u SYSTEM.DEAD.LETTER.QUEUE
데드 레터 큐의 이름입니다.

“saturn.queue.manager”

이 큐 관리자의 이름입니다. **crtmqm** 명령에서는, 이 이름이 마지막 매개변수여야 합니다.

디폴트 큐 관리자를 작성하면 큐 관리자 이름을 지정하지 않고 이에 대해 일부 명령을 발행할 수 있습니다(**strmqm** 및 **runmqsc** 등). **endmqm**이나 **dltmqm**과 같은 다른 명령에서는 큐 관리자 이름을 지정해야 합니다.

이 예에서, 큐 관리자 이름은 소문자로 되어 있으며 소문자는 큰 따옴표로 묶여 있다는 점에 유의하십시오. 매개변수의 대소문자를 처리하는 방법에 대한 자세한 정보는 27 페이지의 『대소문자 구분 이해』를 참조하십시오.

큐 관리자 시작

큐 관리자를 작성했다라도 시작하지 않으면 명령이나 MQI 호출을 처리할 수 없습니다. 다음 명령을 입력하여 큐 관리자를 시작하십시오.

```
strmqm "saturn.queue.manager"
```

strmqm 명령은 큐 관리자가 시작하여 연결 요청을 받아들일 준비가 될 때까지는 제어를 리턴하지 않습니다.

기존의 큐 관리자를 디폴트로 만들기

디폴트 큐 관리자를 작성하면, 디폴트 큐 관리자의 이름이 MQSeries 구성 파일(mqs.ini)에 있는 *DefaultQueueManager* 스탠자에 삽입됩니다. 만약 스탠자와 그 내용이 존재하지 않을 경우에는 자동으로 작성됩니다.

다음과 같은 경우에는 이 스탠자를 편집해야 합니다.

- 기존의 큐 관리자를 디폴트로 만들려는 경우. 이를 수행하려면 이 스탠자의 큐 관리자 이름을 새로운 디폴트 큐 관리자의 이름으로 변경해야 합니다. 텍스트 편집기를 사용하여 반드시 수동으로 변경해야 합니다.
- 노드에 디폴트 큐 관리자가 없고, 기존의 큐 관리자를 디폴트로 만들려는 경우. 이를 수행하려면 필요한 이름이 있는 *DefaultQueueManager* 스탠자를 직접 작성해야 합니다.
- 다른 큐 관리자를 실수로 디폴트로 지정했다가 원래의 디폴트 큐 관리자로 되돌아가려 할 경우. 이를 수행하려면, MQSeries 구성 파일의 *DefaultQueueManager* 스탠자에서, 원하지 않는 디폴트 큐 관리자 이름을 원하는 큐 관리자 이름으로 대체하여 편집하십시오.

구성 파일에 대한 정보는 183 페이지의 『제13장 MQSeries 구성』을 참조하십시오.

스탠자에 필요한 정보가 들어있을 경우 큐 관리자를 정지한 다음 재시작하십시오.

큐 관리자 정지

endmqm 명령을 사용하여 큐 관리자를 정지시키십시오. 예를 들어 saturn.queue.manager라는 큐 관리자를 정지시키려면, 다음을 입력하십시오.

```
endmqm "saturn.queue.manager"
```

정지 종료

디폴트로, **endmqm** 명령은 지정된 큐 관리자의 제어 종료나 정지 종료를 수행합니다. 이를 완료하는 데는 약간의 시간이 걸립니다. 제어 종료의 경우 연결된 모든 응용프로그램의 연결이 끊어질 때까지 기다리기 때문입니다.

큐 관리자 작성

응용프로그램에 정지할 것을 알려려면 이 종료 유형을 사용하십시오. 다음과 같이 입력할 경우,

```
endmqm -c "saturn.queue.manager"
```

모든 응용프로그램이 정지되는 시기를 알 수 없습니다(endmqm -c "saturn.queue.manager" 명령은 endmqm "saturn.queue.manager" 명령과 같습니다).

즉시 종료

즉시 종료의 경우, 현재의 모든 MQI 호출은 완료되지만 새로운 호출은 실패합니다. 이 유형의 종료는 큐 관리자에서 응용프로그램의 연결이 끊어지기를 기다리지 않습니다.

이 유형의 종료를 일반적인 큐 관리자 정지 방법으로 사용하십시오. 정지 기간 이후에 선택적으로 사용할 수도 있습니다. 즉시 종료를 수행하려면 다음을 입력하십시오.

```
endmqm -i "saturn.queue.manager"
```

선제 종료

주의: endmqm 명령을 사용하는 기타 모든 큐 관리자 정지 시도가 실패하지 않는 이상 이 방법을 사용하지 마십시오. 이 방법은 연결된 응용프로그램에 대해 예측할 수 없는 결과를 가져올 수 있습니다.

즉시 종료가 작동하지 않으면, -p 플래그를 지정하여 선제 종료를 시도해야 합니다. 예를 들면 다음과 같습니다.

```
endmqm -p "saturn.queue.manager"
```

이를 수행하면 모든 큐 관리자 코드가 즉시 정지됩니다.

주: 강제 또는 선제 종료를 수행한 후나 큐 관리자가 실패할 경우에는 큐 관리자가 소유하고 있는 공유 메모리를 클린업하지 않은 상태로 종료될 수도 있습니다. 이것은 재시작시 문제를 초래할 수 있습니다. 이러한 유형의 갑작스러운 종료 후에 MONMQ 유틸리티를 사용하여 클린업하는 방법에 대한 정보는 387 페이지의 『MONMQ를 사용하여 공유 메모리 관리』를 참조하십시오.

큐 관리자 종료에 문제점이 있을 경우

큐 관리자 종료시의 문제점은 종종 응용프로그램에 의해 발생합니다. 예를 들어 응용프로그램이 다음과 같은 경우 문제가 발생합니다.

- MQI 리턴 코드를 적절하게 점검하지 않는 경우
- 정지 알림을 요청하지 않는 경우
- MQDISC 호출을 발행하여 큐 관리자와의 연결을 끊지 않고 종료하는 경우

큐 관리자의 종료 속도가 너무 느리거나 큐 관리자가 정지할 수 없을 것으로 생각되면, Ctrl-Y를 사용하여 **endmqm** 명령을 중단할 수 있습니다. 그런 이후에 다른 **endmqm** 명령을 발행할 수 있는데, 이 때는 즉시 종료나 선제 종료를 지정하는 플래그를 사용하십시오.

endmqm 명령과 그 옵션에 대한 자세한 설명은 291 페이지의 『endmqm(큐 관리자 종료)』을 참조하십시오.

큐 관리자 재시작

큐 관리자를 재시작하려면 다음 명령을 사용하십시오.

```
strmqm "saturn.queue.manager"
```

큐 관리자 삭제

큐 관리자를 삭제하려면, 먼저 큐 관리자를 정지한 후 다음 명령을 사용하십시오.

```
dltmqm "saturn.queue.manager"
```

주의: 큐 관리자를 삭제하면 연관된 모든 자원도 삭제되므로 신중히 결정해야 합니다. 여기에는 모든 큐와 해당 큐의 메시지 뿐만 아니라 모든 오브젝트 정의가 포함됩니다.

dltmqm 명령과 그 옵션에 대한 설명은 270 페이지의 『dltmqm(큐 관리자 삭제)』을 참조하십시오. 신뢰할 수 있는 관리자만 이 명령을 사용할 수 있도록 해야 합니다.

제4장 로컬 MQSeries 오브젝트 관리

이 장에서는 MQI(Message Queing Interface)를 사용하는 응용프로그램을 지원하기 위한 MQSeries 오브젝트 관리 방법을 설명합니다. 이 컨텍스트에서 로컬 관리란 QSeries 오브젝트의 작성, 표시, 변경, 복사 및 삭제를 의미합니다.

이 장은 다음과 같은 절로 구성됩니다.

- 『MQI를 사용하는 응용프로그램 지원』
- 40 페이지의 『MQSC 명령을 사용한 로컬 관리 작업의 수행』
- 44 페이지의 『텍스트 파일에서 MQSC 명령 실행』
- 48 페이지의 『MQSC 문제점 해결』
- 50 페이지의 『로컬 큐에 대한 작업』
- 57 페이지의 『알리아스 큐에 대한 작업』
- 60 페이지의 『모델 큐에 대한 작업』
- 61 페이지의 『트리거를 위한 오브젝트 관리』

MQI를 사용하는 응용프로그램 지원

MQSeries 응용프로그램들을 실행하기 위해서는 오브젝트가 필요합니다. 예를 들면, 그림 1은 큐에서 메시지를 제거하고 처리한 다음, 일부 결과를 동일한 큐 관리자의 다른 큐로 송신하는 응용프로그램을 보여줍니다.

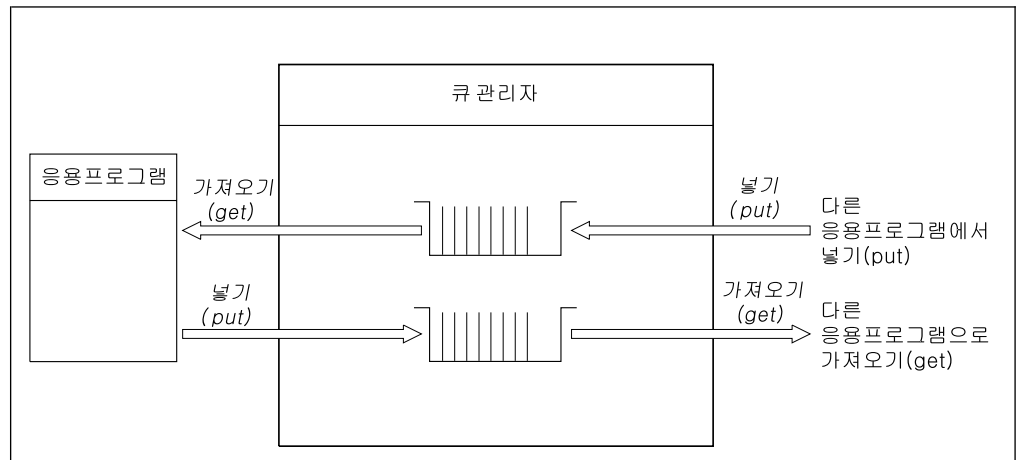


그림 1. 큐, 메시지 및 응용프로그램

응용프로그램은 MQPUT을 사용하여 로컬 또는 리모트 큐에 메시지를 넣을(put) 수 있지만, 메시지를 가져올 때는 MQGET을 사용하여 로컬 큐에서만 직접 가져올(get) 수 있습니다.

이 응용프로그램이 실행되려면 다음과 같은 조건들이 충족되어야 합니다.

- 큐 관리자가 존재해야 하며 실행 중이어야 합니다.
- 메시지가 제거될 첫 번째 응용프로그램 큐가 정의되어야 합니다.
- 응용프로그램이 메시지를 넣을 두 번째 큐도 정의되어야 합니다.
- 응용프로그램이 큐 관리자에 연결될 수 있어야 합니다. 이를 위해서는 응용프로그램이 제품 코드에 링크되어야 합니다. 자세한 정보는 *MQSeries Application Programming Guide*를 참조하십시오.
- 첫 번째 큐에 메시지를 넣을 응용프로그램 또한 큐 관리자에 연결되어야 합니다. 응용프로그램이 리모트일 경우, 그 응용프로그램 또한 트랜스미션 큐와 채널로 설정되어야 합니다. 이러한 시스템에 대해서는 39 페이지의 그림1에 표시되어 있지 않습니다.

MQSC 명령을 사용한 로컬 관리 작업의 수행

이 절에서는 **runmqsc** 명령을 사용하여 명령을 발행한 것으로 가정합니다. 명령은 키보드에서 대화식으로 입력하여 발행하거나 `SYSS$INPUT`의 경로를 재지정하여 ASCII 텍스트 파일에서 명령 순서를 실행할 수 있습니다. 어느 경우나 명령의 형식은 동일합니다.

MQSeries 명령 참조서에서는 각 MQSC 명령과 그 구문을 설명합니다.

MQSeries 스크립트 명령(MQSC)을 사용하여 큐 관리자 자체, 클러스터, 채널, 이름 목록 및 프로세스 정의를 포함하는 큐 관리자 오브젝트를 관리할 수 있습니다. 이 절에서는 큐 관리자, 큐 그리고 프로세스 정의에 대해 설명합니다. 채널 오브젝트 관리에 대한 정보는 *MQSeries* 상호통신에서 DQM 구현을 참조하십시오.

runmqsc 명령을 사용하여 큐 관리자에게 MQSC 명령을 발행합니다. 이 명령은 키보드에서 대화식으로 입력하여 발행하거나 표준 입력의 경로를 재지정해서 ASCII 텍스트 파일에서 명령 순서를 실행할 수 있습니다. 어느 경우나 명령의 형식은 동일합니다.

명령에 설정된 플래그에 따라 세가지 모드로 **runmqsc** 명령을 실행할 수 있습니다.

- **확인 모드:** MQSC 명령이 로컬 큐 관리자에서 확인되지만 실제로 실행되지 않는 모드
- **직접 모드:** MQSC 명령이 로컬 큐 관리자에서 실행되는 모드
- **간접 모드:** MQSC 명령이 리모트 큐 관리자에서 실행되는 모드

MQSC에 지정된 오브젝트 속성은 대소문자를 구분하지 않지만 이 서적에서는 대문자(예: `RQMNAME`)로 표시합니다(대소문자 구분에 대한 자세한 정보는 28 페이지의 『MQSC 명령의 대소문자 구분』을 참조하십시오). MQSC 속성 이름은 8자를 초과할 수 없습니다.

시작하기 전에

MQSC 명령을 실행하려면, 명령을 실행할 큐 관리자를 작성하여 시작해야 합니다. 34 페이지의 『디폴트 큐 관리자 작성』을 참조하십시오.

MQSeries 오브젝트 이름

예에서는 긴 오브젝트 이름을 사용합니다. 이는 사용자가 다루고 있는 오브젝트의 유형을 식별하는 데 도움을 주기 위한 것입니다.

MQSC 명령을 발행할 때는 큐의 로컬 이름만을 지정해야 합니다. 예에서는 다음과 같은 큐 이름들을 사용합니다.

```
ORANGE.LOCAL.QUEUE
```

위 이름에서 LOCAL.QUEUE 부분은 이 큐가 로컬 큐라는 것만을 설명합니다. 이 부분은 일반적으로 로컬 큐 이름에 필요하지 않습니다.

또한 큐 관리자 이름으로 saturn.queue.manager라는 이름을 사용합니다.

위에서 queue.manager 부분은 이 오브젝트가 큐 관리자라는 것만을 설명합니다. 이 부분은 일반적으로 큐 관리자 이름에 필요하지 않습니다.

이 이름들을 사용할 필요는 없습니다. 그러나 사용하지 않을 경우에는, 예에서 해당 이름들을 지정하는 모든 명령을 수정해야 합니다.

입력 및 출력 경로 재지정

다른 운영 체제에서 OpenVMS로 더욱 쉽게 이주할 수 있도록, MQSeries에서는 sys\$input, sys\$output 및 sys\$error에 대해 다음과 같은 UNIX 양식의 경로 재지정 표시기를 지원합니다.

<는 SYSS\$INPUT에 대한 소스를 지정합니다.

>는 SYSS\$OUTPUT에 대한 소스를 지정합니다.

2>는 SYSS\$ERROR에 대한 소스를 지정합니다.

이 기능은 샘플 프로그램의 실행 가능 버전에도 포함됩니다. 그러나, 샘플의 소스에는 포함되지 않기 때문에 소스 코드에서 샘플을 다시 빌드하는 경우에는 사용할 수 없습니다.

MQSC 기능의 대화식 사용

DCL 프롬프트 유형에서 명령을 대화식으로 입력하려면:

```
runmqsc
```

MQSC 명령의 발행

이 명령에서는 큐 관리자 이름이 지정되지 않았습니 다. 따라서 MQSC 명령이 디폴트 큐 관리자에 의해 처리됩니다. 필요하다면 이제 모든 MQSC 명령을 입력할 수 있습니다. 예를 들어 다음을 시도해 보십시오.

```
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE)
```

명령이 다음 줄에서 계속된다는 것을 나타내려면 연결 문자를 사용해야 합니다.

- 마이너스 기호(-)는 명령이 다음 줄의 처음부터 계속됨을 나타냅니다.
- 플러스 기호(+)는 명령이 다음 줄의 첫 번째 비공백 문자에서 계속됨을 나타냅니다.

연속 문자가 아닌 비공백 줄의 마지막 문자로 명령 입력이 종료됩니다. 세미콜론(;)을 입력함으로써 명시적으로 명령 입력을 종료할 수도 있습니다(명령 입력의 마지막 줄 끝에 실수로 연속 문자를 입력한 경우에 특히 유용합니다).

MQSC 명령의 피드백

사용자가 MQSC 기능에서 명령을 발행하면, 큐 관리자는 사용자의 조치를 확인하거나 사용자가 만든 오류를 표시하는 연산자 메시지를 리턴합니다. 예를 들면 다음과 같습니다.

```
AMQ8006: MQSeries queue created
.
.
.
AMQ8405: Syntax error detected at or near end of command segment below:-
z
AMQ8426: Valid MQSC commands are:
    ALTER
    CLEAR
    DEFINE
    DELETE
    DISPLAY
END
    PING
    REFRESH
    RESET
    RESOLVE
    RESUME
    START
    STOP
    SUSPEND
```

첫 번째 메시지는 큐가 작성되었음을 확인합니다. 두 번째 메시지는 구문 오류가 발생했음을 나타냅니다. 이들 메시지는 표준 출력 장치로 보내집니다. 명령을 바르게 입력하지 않았다면 *MQSeries* 명령 참조서에서 올바른 구문을 참조하십시오.

MQSC의 대화식 입력 종료

MQSC 명령의 대화식 입력을 끝내려면 MQSC END 명령을 입력하십시오.

```
END
```

또는 EOF 문자 <CTRL Z>를 입력하여 종료할 수 있습니다.

텍스트 파일 등 다른 소스로부터 입력을 경로 재지정하고 있다면, 이렇게 할 필요가 없습니다.

큐 관리자 속성 표시

runmqsc 명령에 지정된 큐 관리자의 속성을 표시하려면, 다음 MQSC 명령을 사용하십시오.

```
DISPLAY QMGR ALL
```

그림2에서는 일반적인 출력 형태를 보여줍니다.

```
1 : display qmgr all
AMQ8408: Display Queue Manager details.
DESCR( )                DEADQ( )
DEFXMITQ( )             CHADEXIT( )
CLWLXIT( )              CLWLDATA( )
REPOS( )                REPOSNL( )
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) QMNAME(saturn.queue.manager)
CRDATE(2001-01-16)      CRTIME(11.13.56)
ALTDATE(2001-01-16)     ALTTIME(11.13.56)
QMID(saturn.queue.manager_2001-01-16_11.13.56)
TRIGINT(999999999)     MAXHANDS(256)
MAXUMSGS(10000)        AUTHOREV(DISABLED)
INHIBTEV(DISABLED)    LOCALEV(DISABLED)
REMOETEVEV(DISABLED)  PERFMEV(DISABLED)
STRSTPEV(ENABLED)     CHAD(DISABLED)
CHADEV(DISABLED)      CLWLEN(100)
MAXMSGL(4194304)      CCSID(819)
MAXPRTY(9)            CMDLEVEL(510)
PLATFORM(OpenVMS)     SYNCPT
DISTL(YES)
```

그림 2. DISPLAY QMGR 명령의 일반적인 출력 형태

DISPLAY QMGR 명령의 ALL 매개변수는 모든 큐 관리자 속성을 표시합니다. 특히, 명령 실행시 어떤 큐 관리자 이름도 지정되지 않았기 때문에 출력은 디폴트 큐 관리자 이름(saturn.queue.manager), 데드-레터 큐 이름(SYSTEM.DEAD.LETTER.QUEUE) 및 명령 큐(SYSTEM.ADMIN.COMMAND.QUEUE)를 알려줍니다.

MQSC 명령의 발행

계속하기 전에, 이 큐들이 다음 명령을 입력하여 작성되었는지 확인하십시오.

```
DISPLAY QUEUE (SYSTEM.*)
```

이 명령은 'sYSTEM.*' 접두부에 일치하는 큐 목록을 표시합니다. 괄호를 사용해야 합니다.

디폴트가 아닌 큐 관리자의 사용

runmqsc 명령에서 큐 관리자 이름을 지정하여 디폴트가 아닌 로컬 큐 관리자에서 MQSC 명령을 실행할 수 있습니다. 예를 들어 큐 관리자 `jupiter.queue.manager`에서 MQSC 명령을 실행하려면 다음 명령을 사용하십시오.

```
runmqsc "jupiter.queue.manager"
```

이 명령을 사용하면 입력하는 모든 MQSC 명령이 이 큐 관리자에 의해 처리됩니다. 즉, 같은 노드에 있으며 이미 실행중인 것으로 가정합니다.

또한 리모트 큐 관리자에서도 MQSC 명령을 실행할 수 있습니다. 75 페이지의 『리모트 MQSC 명령 발행』을 참조하십시오.

큐 관리자 속성 대체

runmqsc 명령에 지정된 큐 관리자의 속성을 대체하려면, 변경하려는 속성과 값을 지정하는 MQSC 명령 **ALTER QMGR**를 사용하십시오. 예를 들어 `jupiter.queue.manager`의 속성을 대체하려면 다음 명령을 사용하십시오.

```
runmqsc "jupiter.queue.manager"
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

ALTER QMGR 명령은 사용된 데드-레터 큐를 변경하고 **inhibit** 이벤트를 사용 가능하게 합니다.

텍스트 파일에서 MQSC 명령 실행

MQSC 명령의 대화식 실행은 빠른 테스트에는 적당하지만, 긴 명령이 있거나 특정 명령 순서를 반복하고 싶은 경우에는 텍스트 파일에서 입력을 수행해야 합니다(경로 재지정 표시기에 대한 정보는 41 페이지의 『입력 및 출력 경로 재지정』을 참조하십시오). 이를 수행하려면, 먼저 사용자에게 친숙한 텍스트 편집기를 사용하여 MQSC 명령이 있

는 텍스트 파일을 작성하십시오. 예를 들어 다음과 같은 명령은 텍스트 파일 myprog.in에 있는 명령 순서를 실행합니다.

```
runmqsc < myprog.in
```

또한 출력의 경로를 파일에 재지정할 수 있습니다. 입력을 위한 MQSC 명령을 포함하고 있는 파일을 *MQSC 명령 파일*이라고 합니다. 큐 관리자의 응답을 포함하고 있는 출력 파일을 *보고서 파일*이라고 합니다.

runmqsc 명령에서 SYS\$INPUT과 SYS\$OUTPUT의 경로를 재지정하려면, 다음과 같은 형식의 명령을 사용하십시오.

```
runmqsc < myprog.in > myprog.out
```

이 명령은 MQSC 명령 파일 myprog.in에 있는 MQSC 명령을 실행합니다. 큐 관리자 이름을 지정하지 않았기 때문에 MQSC 명령은 디폴트 큐 관리자와 대응하여 실행됩니다. 출력은 보고서 파일 myprog.out로 송신됩니다. 46 페이지의 그림3은 MQSC 명령 파일 myprog.in의 추출을 보여주고 47 페이지의 그림4는 보고서 파일 myprog.out의 해당 추출을 보여줍니다.

runmqsc 명령에서 SYS\$INPUT과 SYS\$OUTPUT의 경로를 재지정하려면, 디폴트가 아닌 큐 관리자 (saturn.queue.manager)에 대하여 다음과 같은 형식의 명령을 사용하십시오.

```
runmqsc "saturn.queue.manager" < myprog.in > myprog.out
```

MQSC 명령 파일

MQSC 명령은 사람이 읽을 수 있는 형식, 즉 ASCII 텍스트로 기록됩니다. 46 페이지의 그림3은 MQSC 명령(DEFINE QLOCAL)과 그 속성을 표시하는 MQSC 명령의 추출입니다. *MQSeries* 명령 참조서에서는 각 MQSC 명령과 그 구문을 설명합니다.

```

.
.
.
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +
DESCR(' ') +
PUT(ENABLED) +
DEFPRTY(0) +
DEFPSIST(NO) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(1024) +
DEFSOPT(SHARED) +
NOHARDENBO +
USAGE(NORMAL) +
NOTRIGGER
.
.
.

```

그림 3. MQSC 명령 파일 myprog.in의 추출

MQSeries 환경간의 이동을 위해 MQSC 명령 파일의 줄 길이를 72자로 제한하는 것이 좋습니다. 플러스 기호는 이 명령이 다음 줄에서 계속된다는 것을 나타냅니다.

Compaq OpenVMS용 MQSeries에서는 연결 문자를 포함하여 줄을 최대 80자로 제한해야 합니다. 플러스 기호는 이 명령이 다음 줄에서 계속된다는 것을 나타냅니다.

MQSC 보고서

runmqsc 명령은 SYS\$OUTPUT으로 송신된 보고서를 리턴합니다. 보고서에는 다음 사항이 포함됩니다.

- 보고서의 소스로서 MQSC를 식별하는 헤더
Starting MQSeries Commands.
- 발행된 MQSC 명령에 선택적으로 번호를 붙인 목록. 디폴트로, 입력의 텍스트가 출력에 반향됩니다. 이 출력에는 47 페이지의 그림4에서 보는 바와 같이 각 출력의 접두부에 순번이 붙어 있습니다. 그러나 **runmqsc** 명령에 -e 플래그를 사용하여 출력을 억제할 수 있습니다.
- 모든 오류 명령에 대한 구문 오류 메시지.
- 각 명령의 실행 결과를 표시하는 연산자 메시지. 예를 들어 DEFINE QLOCAL 명령이 성공적으로 완료되었음을 표시하는 연산자 메시지는 다음과 같습니다.
AMQ8006: MQSeries queue created.
- 스크립트 파일의 실행시 일반적인 오류의 결과로 나타나는 기타 메시지들.
- 읽은 명령의 수, 구문 오류가 있는 명령의 수 및 처리 불가능한 명령의 수를 표시하는 보고서의 통계적 요약.

주: 큐 관리자는 구문 오류가 없는 명령만 처리합니다.


```

Starting MQSeries Commands.
.
.
12:      DEFINE QLOCAL('RED.LOCAL.QUEUE') REPLACE +
:        DESCR(' ') +
:        PUT(ENABLED) +
:        DEFPRTY(0) +
:        DEFPSIST(NO) +
:        GET(ENABLED) +
:        MAXDEPTH(5000) +
:        MAXMSGL(1024) +
:        DEFSOPT(SHARED) +
:        USAGE(NORMAL) +
:        NOTRIGGER
AMQ8006: MQSeries queue created.
:
.
.
15 MQSC commands read.
0 commands have a syntax error.
0 commands cannot be processed.

```

그림 4. MQSC 보고서 파일 *myprog.out*의 추출

제공된 MQSC 명령 파일의 실행

Compaq OpenVMS용 MQSeries 설치 시, 다음과 같은 MQSC 명령 파일이 제공됩니다.

amqscos0.tst

샘플 프로그램이 사용하는 오브젝트의 정의

파일은 MQS_EXAMPLES: 디렉토리에 위치합니다.

명령 확인을 위한 **runmqsc**의 사용

실제로 실행하지는 않고 로컬 큐 관리자의 MQSC 명령을 확인하기 위해 **runmqsc** 명령을 사용할 수 있습니다. 이를 수행하려면, **runmqsc** 명령에 **-v** 플래그를 설정하십시오. 예를 들면 다음과 같습니다.

```
runmqsc -v < myprog.in > myprog.out
```

MQSC 명령 파일에 대하여 **runmqsc**를 실행하면, 큐 관리자는 실제로 MQSC 명령을 실행하지 않고 각 명령을 확인하여 보고서를 리턴합니다. 이렇게 하면 명령 파일에 있는 모든 명령의 구문을 점검할 수 있습니다. 이것은 명령 파일에서 많은 명령들을 실행할 때 특히 중요합니다.

MQSC 명령 실행

이 보고서는 47 페이지의 그림4에 나타난 것과 유사합니다.

MQSC 명령을 리모트로 확인하기 위해 이 방법을 사용할 수는 없습니다. 예를 들어 다음과 같은 명령을 시도하는 경우,

```
runmqsc -w 30 -v "jupiter.queue.manager" < myprog.in > myprog.out
```

큐 관리자가 리모트라는 것을 표시하기 위해 사용하는 -w 플래그는 무시되고, 명령이 확인 모드에서 로컬로 실행됩니다.

MQSC 문제점 해결

MQSC 명령을 실행할 수 없는 경우, 다음 점검 목록을 사용하여 목록에 표시된 일반적인 문제점들 중에 해당 사항이 있는지 점검하십시오. 오류가 생성되었다더라도 문제점이 무엇인지 항상 분명하지는 않습니다.

runmqsc 명령을 사용할 때는 다음 사항을 기억하십시오.

- 파일에서 입력의 경로를 재조정할 때는 간접 연산자 <를 사용하십시오. 간접 연산자를 생략하면, 큐 관리자는 파일 이름을 큐 관리자 이름으로 해석하여 다음과 같은 오류 메시지를 발행합니다.

```
AMQ8118: MQSeries queue manager does not exist.
```

- 출력의 경로를 파일로 재조정할 경우, > 간접 연산자를 사용하십시오. 디폴트로, 출력은 **runmqsc** 명령을 실행한 디렉토리로 찾아 갑니다. 출력을 특정 파일과 디렉토리로 송신하기 위해 완전한 파일 이름을 지정하십시오.
- 명령을 실행할 큐 관리자를 작성했는지 확인하십시오.
이를 수행하려면, 디폴트로 MQS_ROOT:[MQM] 디렉토리에 위치하는 구성 파일인 mqs.ini를 찾으십시오. 이 파일에는 큐 관리자의 이름과 디폴트 큐 관리자(있는 경우)의 이름이 있습니다.
- 큐 관리자가 실행 중이어야 합니다. 그렇지 않다면 실행시키십시오(34 페이지의 『큐 관리자 시작』을 참조하십시오). 이미 시작된 경우라면 오류 메시지가 표시됩니다.
- 디폴트 큐 관리자를 정의하지 않았다면 **runmqsc** 명령에서 큐 관리자 이름을 지정하십시오. 그렇지 않으면 다음과 같은 오류가 발생합니다.

```
AMQ8146: MQSeries queue manager not available.
```

이러한 유형의 문제점을 수정하려면 35 페이지의 『기존의 큐 관리자를 디폴트로 만들기』를 참조하십시오.

- MQSC 명령을 **runmqsc** 매개변수로 지정할 수 없습니다. 예를 들면, 다음은 올바르지 않습니다.

```
runmqsc DEFINE QLOCAL(FRED)
```

- **runmqsc** 명령을 발행하기 전에 DCL에서 MQSC 명령을 입력할 수 없습니다. 예를 들면 다음과 같습니다.

```
DEFINE QLOCAL(Queue1)
%DCL-W-PARMDEL, invalid parameter delimiter - check use of special characters
```

- **runmqsc**에서 제어 명령을 실행할 수 없습니다. 예를 들어 일단 MQSC를 대화식으로 실행하고 있다면 큐 관리자를 시작할 수 없습니다.

```
$ runmqsc
0790997, 5724-A38 (C) Copyright IBM Corp. 1996, 2001 ALL RIGHTS RESERVED.
Starting MQSeries Commands.
strmqm saturn.queue.manager
    1 : strmqm saturn.queue.manager
AMQ8405: Syntax error detected at or near end of command segment below:-
s
AMQ8426: Valid MQSC commands are:
    ALTER
    CLEAR
    DEFINE
    DELETE
    DISPLAY
END
    PING
    REFRESH
    RESET
    RESOLVE
    RESUME
    START
    STOP
    SUSPEND
*CANCEL*
One MQSC command read.
One command has a syntax error.
All valid MQSC commands were processed.
$
```

77 페이지의 『리모트로 MQSC를 사용할 때 문제점이 발생할 경우』를 참조하십시오.

로컬 큐에 대한 작업

이 절에는 사용 가능한 일부 MQSC 명령에 대한 예가 있습니다. 이 명령들에 대한 자세한 설명을 보려면 *MQSeries* 명령 참조서를 참조하십시오.

로컬 큐 정의

응용 프로그램에서, 로컬 큐 관리자는 응용 프로그램을 연결하는 큐 관리자입니다. 로컬 큐 관리자가 관리하는 큐는 그 큐 관리자에 대해 로컬 상태입니다.

MQSC 명령인 DEFINE QLOCAL을 사용하여 로컬 큐의 정의를 작성하고 큐라고 불리는 데이터 구조를 작성하십시오. 또한, 기본 로컬 큐의 특성에서 큐의 특성을 수정할 수 있습니다.

이 예에서 정의된 큐인 ORANGE.LOCAL.QUEUE는 다음과 같은 특성들을 갖도록 지정되었습니다.

- 이것은 GET에서 사용 가능하고 PUT에서 사용 불가능하며, FIFO(first-in-first-out)로 운영됩니다.
- 이것은 이니시에이션 큐나 트랜스미션 큐와 같은 ‘보통’ 큐가 아니며 트리거 메시지를 생성하지 않습니다.
- 큐의 최대 깊이는 1,000 메시지이고, 메시지의 최대 길이는 2,000 바이트입니다.

다음 MQSC 명령은 다음 작업을 수행합니다.

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL)
```

주:

1. 이들 속성의 대부분은 제품과 함께 제공된 기본값입니다. 여기서는 해당 속성을 설명하기 위한 목적으로 기록하였습니다. 사용자가 기본값을 원하거나 속성이 변경되지 않은 경우, 속성은 생략될 수 있습니다. 51 페이지의 『디폴트 오브젝트 속성의 표시』를 참조하십시오.
2. USAGE(NORMSL)는 이 큐가 트랜스미션 큐가 아니라는 것을 표시합니다.
3. 같은 큐 관리자에 ORANGE.LOCAL.QUEUE라는 이름의 로컬 큐가 이미 존재할 경우 이 명령은 실패합니다. 큐의 기존 정의에 겹쳐쓰려면 REPLACE 속성을 사용하십시오. 53 페이지의 『로컬 큐 속성 변경』을 참조하십시오.

데드-레터 큐의 정의

각각의 큐 관리자는 올바른 목적지로 전달되지 못한 메시지를 저장했다가 나중에 검색할 수 있도록 로컬 큐를 데드-레터큐로 사용해야 합니다. 큐 관리자에게 데드-레터 큐에 대한 정보를 명확히 알려야 합니다. **crtmqm** 명령에서 데드-레터 큐를 지정하여 이를 수행하거나 또는 ALTER QMGR 명령을 사용하여 데드-레터 큐를 나중에 지정할 수 있습니다. 또한 데드-레터 큐를 사용하려면 먼저 그것을 정의해야 합니다.

SYSTEM.DEAD.LETTER.QUEUE라는 샘플 데드-레터 큐는 제품과 함께 제공됩니다. 이 큐는 샘플을 실행할 때 자동으로 작성됩니다. 필요하다면 이 정의를 수정할 수 있습니다. 이름을 바꿀 필요는 없습니다.

데드-레터 큐는 다음 사항 이외의 특별한 요구사항을 갖지 않습니다.

- 데드-레터 큐는 로컬 큐여야 합니다.
- 데드-레터 큐는 그 MAXMSGL(Maximum Message Length) 속성을 사용하여 큐 관리자가 핸들링해야 하는 가장 큰 메시지 및 데드-레터 헤더(MQDLH) 크기를 수용할 수 있어야 합니다.

MQSeries는 데드-레터 큐에서 발견된 메시지의 처리 또는 제거 방법을 지정하는 데드-레터 큐 핸들러를 제공합니다. 자세한 정보는 109 페이지의 『제8장 MQSeries 데드-레터 큐 핸들러』를 참조하십시오.

디폴트 오브젝트 속성의 표시

MQSeries 오브젝트를 정의할 때, 그것은 기본 오브젝트에서 지정하지 않은 모든 속성을 갖습니다. 예를 들어, 로컬 큐를 정의할 때, 큐는 SYSTEM.DEFAULT.LOCAL.QUEUE 라고 불리는 기본 로컬 큐의 정의에서 생략된 모든 속성을 물려받습니다. 이 속성들이 무엇인지 정확히 알려면 다음 명령을 사용하십시오.

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE) ALL
```

주: 이 명령의 구문은 해당 DEFINE 명령의 구문과 다릅니다.

이들을 개별적으로 지정함으로써 속성들을 선택적으로 표시할 수 있습니다. 예를 들면 다음과 같습니다.

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +
      MAXDEPTH +
      MAXMSGL +
      CURDEPTH
```

로컬 큐에 대한 작업

이 명령은 다음과 같이 세 개의 지정된 속성을 표시합니다.

```
AMQ8409: Display Queue details.  
QUEUE(ORANGE.LOCAL.QUEUE)  
MAXDEPTH(1000)  
MAXMSGL(2000)  
CURDEPTH(0)
```

CURDEPTH 는 현재의 큐 깊이, 즉, 큐의 메시지 수입니다. 큐 용량을 모니터링하여 그 큐가 다 차지 않았음을 확인할 수 있기 때문에 이는 유용한 속성입니다.

로컬 큐 정의 복사

DEFINE 명령에서 LIKE 속성을 사용하여 큐 정의를 복사할 수 있습니다. 예를 들면 다음과 같습니다.

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE)
```

이 명령은 시스템 디폴트 로컬 큐의 속성이 아니라 원본 큐인 ORANGE.LOCAL.QUEUE와 동일한 속성을 가진 큐를 작성합니다.

또한 이러한 DEFINE 명령 형식을 사용하여 큐 정의를 복사할 수 있으며 원본의 속성을 하나 이상 변경하여 그것으로 원본 속성을 대체할 수 있습니다. 예를 들면 다음과 같습니다.

```
DEFINE QLOCAL (THIRD.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE) +  
MAXMSGL(1024)
```

이 명령은 ORANGE.LOCAL.QUEUE 큐의 속성을 THIRD.QUEUE 큐로 복사하지만, 새로운 큐의 최대 메시지 길이를 2000 바이트가 아닌 1024 바이트로 지정합니다.

주:

1. DEFINE 명령에서 LIKE 속성을 사용할 때 큐 속성만을 복사합니다. 큐에 있는 메시지는 복사하지 않습니다.
2. LIKE를 지정하지 않고 로컬 큐를 정의하면, DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE)와 같습니다.

로컬 큐 속성 변경

큐 속성은 ALTER QLOCAL 명령을 사용하거나 DEFINE QLOCAL 명령에서 REPLACE 속성을 사용하는 두 가지 방법으로 변경할 수 있습니다. 50 페이지의 『로컬 큐 정의』에서, ORANGE.LOCAL.QUEUE 큐를 정의했습니다. 예를 들어 이 큐의 최대 메시지 길이를 10 000 바이트로 늘리고 싶다고 가정합니다.

- ALTER 명령을 사용할 경우

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

이 명령은 하나의 속성, 즉 최대 메시지 길이의 속성을 바꿉니다. 다른 모든 속성들은 그대로 남습니다.

- DEFINE 명령에서 REPLACE 옵션을 사용할 경우, 예를 들면 다음과 같습니다.

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

이 명령은 최대 메시지 길이를 변경할 뿐만 아니라, 디폴트 값으로 제공된 다른 모든 속성들도 변경합니다. 이 큐는 전에는 넣기(put) 금지였으나 이제는 넣기(put)가 가능합니다. 변경하지 않았다면, SYSTEM.DEFAULT.LOCAL.QUEUE 큐에서 지정된 대로 PUT 사용 기능이 디폴트입니다.

기존 큐의 최대 메시지 길이를 줄여도, 기존 메시지는 영향을 받지 않습니다. 그러나 새 메시지는 새 기준에 충족되어야 합니다.

로컬 큐 지우기

로컬 큐 MAGENTA.QUEUE에서 모든 메시지를 삭제하려면 다음 명령을 사용하십시오.

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

다음과 같은 경우에는 큐를 지울 수 없습니다.

- 동기점이하에서 큐에 넣어진(put) 미확약 메시지가 있는 경우
- 응용프로그램이 현재 그 큐를 열고 있는 경우

로컬 큐 삭제

로컬 큐를 삭제하려면 MQSC 명령 DELETE QLOCAL을 사용하십시오. 큐에 미확약된 메시지가 있을 경우 그 큐는 삭제되지 않습니다. 그러나 큐에 하나 이상의 확약 메

로컬 큐에 대한 작업

시지가 있고 미확약 메시지가 없다면, 그 큐는 PURGE 옵션을 지정해야만 삭제할 수 있습니다. 예를 들면 다음과 같습니다.

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

PURGE 대신에 NOPURGE를 지정하는 경우, 큐에 확약 메시지가 있으면 큐가 삭제되지 않습니다.

큐 찾아보기

큐에 있는 메시지의 내용을 보는 데 사용할 수 있도록, OpenVMS용 MQSeries는 샘플 큐 브라우저를 제공합니다. 브라우저는 소스와 실행 가능한 모듈로서 제공됩니다. 디폴트로 파일 이름과 경로는 다음과 같습니다.

소스 MQS_EXAMPLES:AMQSBCG0.C

실행 파일

[.BIN]AMQSBCG.EXE

MQS_EXAMPLES:

.

샘플은 다음과 같은 두 개의 매개변수를 갖습니다.

- 큐 이름(예: SYSTEM.ADMIN.RESPQ.TEST)
- 큐 관리자 이름(예: JJJH)

위의 예가 다음 명령에 표시되어 있습니다.

```
amqsbcg "SYSTEM.ADMIN.RESPQ.TEST" "JJJH"
```

디폴트는 없으며 두 개의 매개변수가 모두 필요합니다. 이 명령의 일반적인 결과는 55 페이지의 그림5에서 볼 수 있습니다.

로컬 큐에 대한 작업

```
MQGET of message number 2
****Message descriptor****
StrucId : 'MD ' Version : 2
Report : 0 MsgType : 8
Expiry : -1 Feedback : 0
Encoding : 546 CodedCharSetId : 819
Format : 'MQSTR '
Priority : 0 Persistence : 0
MsgId : X'414D51204A4A4A4820202020202020206EC8753A23200000'
CorrelId : X'0000000000000000000000000000000000000000000000000000000000000000'
BackoutCount : 0
ReplyToQ : '
ReplyToQMgr : 'JJJH '
** Identity Context
UserIdentifier:'SYSTEM'
AccountingToken :
X'05363535343000000000000000000000000000000000000000000000000000000000000006'
ApplIdentityData : '
** Origin Context
PutApplType : '12'
PutApplName : 'AMQSPUT.EXE '
PutDate : '20010129' PutTime : '19484323'
ApplOriginData : '
GroupId : X'0000000000000000000000000000000000000000000000000000000000000000'
MsgSeqNumber : '1'
Offset : '0'
MsgFlags : '0'
OriginalLength : '14'

**** Message ****

Length - 14 bytes

00000000: 6D65 7373 6167 6520 3220 4441 5441 'message 2 DATA '

MQGET of message number 3
****Message descriptor****
StrucId : 'MD ' Version : 2
Report : 0 MsgType : 8
Expiry : -1 Feedback : 0
Encoding : 546 CodedCharSetId : 819
Format : 'MQSTR '
Priority : 0 Persistence : 0
MsgId : X'414D51204A4A4A4820202020202020206EC8753A33200000'
CorrelId : X'0000000000000000000000000000000000000000000000000000000000000000'
BackoutCount : 0
ReplyToQ : '
ReplyToQMgr : 'JJJH '
** Identity Context
UserIdentifier:'SYSTEM'
AccountingToken :
X'05363535343000000000000000000000000000000000000000000000000000000000000006'
ApplIdentityData : '
** Origin Context
PutApplType : '12'
PutApplName : 'AMQSPUT.EXE '
PutDate : '20010129' PutTime : '19491145'
ApplOriginData : ' '

```

그림 5. 큐 브라우저에 표시되는 일반적인 결과 (2/3)

알리아스 큐에 대한 작업

리아스 큐라는 것을 알지 못합니다. 이 알리아스를 사용하는 각 MQI 호출에 대하여 큐 관리자는 실제 큐 이름을 해석하는데, 이 큐 이름은 해당 큐 관리자에서 정의된 리모트 큐 또는 로컬 큐가 될 수 있습니다.

TARGQ 속성의 값을 변경함으로써, MQI 호출의 경로를 다른 큐 관리자에게 속한 다른 큐에 재지정할 수도 있습니다. 이것은 유지보수, 이주 및 로드 밸런스 유지에 유용합니다.

알리아스 큐 정의

다음 명령은 알리아스 큐를 작성합니다.

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGQ (YELLOW.QUEUE)
```

이 명령은 MY.ALIAS.QUEUE를 지정하는 MQI 호출의 경로를 YELLOW.QUEUE에 재지정합니다. 이 명령은 대상 큐를 작성하지 않으며, 큐 YELLOW.QUEUE가 런타임 시 존재하지 않으면 MQI 호출이 실패합니다.

알리아스 정의를 변경하는 경우, MQI 호출의 경로를 다른 큐에 재지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGQ (MAGENTA.QUEUE) REPLACE
```

이 명령은 MQI 호출의 경로를 다른 큐인 MAGENTA.QUEUE에 재지정합니다.

또한 알리아스 큐를 사용하여 단일 큐(대상 큐)가 서로 다른 응용프로그램들에 대해 각기 다른 속성을 갖는 것처럼 보이도록 할 수 있습니다. 이 조치는 두 개의 알리아스를 각 응용프로그램에 하나씩 정의함으로써 수행됩니다. 다음과 같은 두 개의 응용프로그램이 있다고 가정합니다.

- 응용프로그램 ALPHA는 YELLOW.QUEUE에 메시지를 넣을(put) 수 있지만, 그 큐에서 메시지를 가져올(get) 수는 없습니다.
- 응용프로그램 BETA는 YELLOW.QUEUE에서 메시지를 가져올(get) 수 있지만, 그 큐에 메시지를 넣을(put) 수는 없습니다.

다음의 명령을 사용하여 이를 수행할 수 있습니다.

* This alias is put enabled and get disabled for application ALPHA

```
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +
    TARGQ (YELLOW.QUEUE) +
    PUT(ENABLED) +
    GET (DISABLED)
```

* This alias is put disabled and get enabled for application BETA

```
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +
    TARGQ (YELLOW.QUEUE) +
    PUT (DISABLED) +
    GET (ENABLED)
```

ALPHA는 그 MQI 호출에서 ALPHAS.ALIAS.QUEUE라는 큐 이름을 사용하고 BETA는 BETAS.ALIAS.QUEUE라는 큐 이름을 사용합니다. 둘 다 동일한 큐에 액세스하지만 방법은 다릅니다.

큐 알리아스를 정의할 때 로컬 큐에서 사용하는 것과 같은 방식으로 LIKE와 REPLACE 속성을 사용할 수 있습니다.

큐 알리아스가 있는 기타 명령의 사용

적절한 MQSC 명령을 사용하여 큐 알리아스 속성을 표시하거나 대체할 수 있으며, 큐 알리아스 오브젝트를 삭제할 수도 있습니다. 예를 들면 다음과 같습니다.

* Display the queue alias' attributes
* ALL = Display all attributes

```
DISPLAY QUEUE (ALPHAS.ALIAS.QUEUE) ALL
```

* ALTER the base queue name, to which the alias resolves.
* FORCE = Force the change even if the queue is open.

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGQ(ORANGE.LOCAL.QUEUE) FORCE
```

* Delete this queue alias, if you can.

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

예를 들어 현재 응용프로그램이 큐를 열어 놓았거나 이 큐로 해석되는 어떤 큐를 열어 놓은 경우에는 큐 알리아스를 삭제할 수 없습니다. 이러한 사항과 기타 큐 알리아스 명령에 대한 더 자세한 정보는 *MQSeries* 명령 참조서를 참조하십시오.

모델 큐에 대한 작업

큐 관리자는 모델 큐로서 정의된 큐 이름을 지정하는 응용프로그램에서 MQI 호출을 수신하는 경우 *다이나믹 큐*를 작성합니다. 큐 관리자는 큐가 작성될 때 새로운 다이나믹 큐의 이름을 생성합니다. 모델 큐는 그 이름으로 작성된 모든 다이나믹 큐의 속성들을 지정하는 템플릿입니다.

모델 큐는 필요한 경우, 응용프로그램이 큐를 작성할 수 있도록 하는 편리한 방법을 제공합니다.

모델 큐의 정의

로컬 큐를 정의할 때와 같은 방법으로 일련의 속성을 가진 모델 큐를 정의합니다. 모델 큐와 로컬 큐는 일련의 같은 속성들을 갖습니다. 한 가지 다른 점은 모델 큐에서는 작성된 다이나믹 큐를 *템포러리*나 *퍼머넌트*로 지정할 수 있다는 것입니다(퍼머넌트 큐는 큐 관리자가 재시작해도 계속 유지되지만 템포러리 큐는 유지되지 않습니다). 예를 들면 다음과 같습니다.

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +
  DESCR('Queue for messages from application X') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE(NORMAL) +
  DEFTYPE (PERDYN)
```

이 명령은 모델 큐 정의를 작성합니다. DEFTYPE 속성에서, 이 템플릿으로 작성된 실제의 큐는 퍼머넌트 다이나믹 큐입니다.

주: 지정되지 않은 속성은 SYSYTEM.DEFAULT.MODEL.QUEUE 디폴트 큐에서 자동으로 복사됩니다.

모델 큐를 정의할 때, 로컬 큐에서 사용하는 것과 같은 방식으로 LIKE와 REPLACE 속성들을 사용할 수 있습니다.

모델 큐에 대한 기타 명령의 사용

적절한 MQSC 명령을 사용하여 모델 큐의 속성을 표시하거나 대체할 수 있으며, 모델 큐 오브젝트를 삭제할 수도 있습니다. 예를 들면 다음과 같습니다.

```

* Display the model queue's attributes
* ALL = Display all attributes

DISPLAY QUEUE (GREEN.MODEL.QUEUE) ALL

* ALTER the model to enable puts on any
* dynamic queue created from this model.

ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)

* Delete this model queue:

DELETE QMODEL (RED.MODEL.QUEUE)
    
```

트리거를 위한 오브젝트 관리

MQSeries는 큐에서 특정 조건들이 충족될 때 자동으로 응용프로그램을 시작하는 기능을 제공합니다. 그 조건들 중 한 예로 큐에서 메시지가 지정된 수에 도달하는 경우를 들 수 있습니다. 이 기능을 트리거라고 하며 *MQSeries Application Programming Guide*에 자세히 설명되어 있습니다. 이 절은 Compaq OpenVMS용 MQSeries에서 트리거를 지원하기 위한 필수 오브젝트의 설정 방법을 설명합니다.

트리거를 위한 응용프로그램 큐 정의

응용프로그램 큐는 MQI를 통한 메시지를 위해 응용프로그램에서 사용하는 로컬 큐입니다. 트리거가 기능하기 위해서는 많은 큐 속성들이 응용프로그램 큐에서 정의되어야 합니다. 트리거는 *Trigger* 속성(MQSC의 TRIGGER)에 의해 사용 가능하게 됩니다.

이 예에서와 같이, 로컬 큐 MOTOR.INS.QUEUE에 5나 그 이상의 우선순위를 가진 메시지 100개가 있을 때 트리거 이벤트가 생성됩니다.

```

DEFINE QLOCAL (MOTOR.INS.QUEUE) +
  PROCESS (MOTOR.INS.PROC) +
  MAXMSGL (2000) +
  DEFPSIST (YES) +
  INITQ (MOTOR.INS.INT.Q) +
  TRIGGER +
  TRIGTYPE (DEPTH) +
  TRIGDPH (100)+
  TRIGPRI (5)
    
```

설명:

QLOCAL (MOTOR.INS.QUEUE)

정의되고 있는 응용프로그램 큐의 이름을 지정합니다.

트리거를 위한 오브젝트 관리

PROCESS (MOTOR.INS.PROC)

트리거 모니터 프로그램이 시작할 응용프로그램의 이름을 지정합니다.

MAXMSGL (2000)

큐의 메시지 최대 길이를 지정합니다.

DEFPSIST (YES)

메시지가 이 큐에서 지속적이라는 것을 지정합니다.

INITQ (MOTOR.INS.INT.Q)

큐 관리자가 트리거 메시지를 넣는(put) 이니시에이션 큐의 이름입니다.

TRIGGER

트리거 속성 값입니다.

TRIGTYPE (DEPTH)

필요한 우선순위(TRIMPRI)를 가진 메시지 수가 TRIGDPTH에 지정된 수에 이를 때 트리거 이벤트가 생성된다는 것을 지정합니다.

TRIGDPTH (100)

트리거 이벤트를 생성하는데 필요한 메시지의 수를 지정합니다.

TRIMPRI (5)

트리거 이벤트의 생성 여부를 결정하는데 있어 큐 관리자가 그 수를 세는 메시지의 우선순위입니다. 5나 그 이상의 우선순위를 가진 메시지만이 계산됩니다.

이니시에이션 큐 정의

트리거 이벤트가 발생할 때, 큐 관리자는 응용프로그램 큐 정의에 지정된 이니시에이션 큐에 트리거 메시지를 넣습니다(put). 이니시에이션 큐에는 특별한 설정이 없지만 다음 로컬 큐 MOTOR.INS.INT.Q의 정의를 참고로 사용할 수 있습니다.

```
DEFINE QLOCAL(MOTOR.INS.INT.Q) +
  GET (ENABLED) +
  NOSHARE +
  NOTRIGGER +
  MAXMSGL (2000) +
  MAXDEPTH (10)
```

프로세스 정의 작성

DEFINE PROCESS 명령을 사용하여 프로세스 정의를 작성하십시오. 프로세스 정의는 응용프로그램 큐를 해당 큐에서 메시지를 처리하는 응용프로그램과 연관시킵니다. 이 작업은 응용프로그램 큐 MOTOR.INS.QUEUE의 PROCESS 속성을 통해 수행됩니다. 다

다음 MQSC 명령은 다음 예에서 식별된, 필요한 프로세스, MOTOR.INS.PROC를 정의합니다.

```
DEFINE PROCESS (MOTOR.INS.PROC) +
  DESCR ('Insurance request message processing') +
  APPLTYPE (OPENVMS) +
  APPLICID ('DKA0:[MQM.ADMIN.TEST]IRMP01.EXE') +
  USERDATA ('open, close, 235')
```

설명:

MOTOR.INS.PROC

프로세스 정의의 이름입니다. 15자를 초과해서는 안됩니다.

DESCR ('Insurance request message processing')

정의에 관련되는 응용프로그램을 설명하는 텍스트이며 키워드 다음에 표시됩니다. 이 텍스트는 DISPLAY PROCESS 명령을 사용할 때 표시됩니다. 이것은 프로세스 되는 것을 식별하는 데 도움을 줍니다. 문자열에 공백을 사용하는 경우, 문자열을 작은 따옴표로 묶어야 합니다.

APPLTYPE (OPENVMS)

OpenVMS에서 실행되는 응용프로그램의 유형입니다.

APPLICID ('DKA0:[MQM.ADMIN.TEST]IRMP01.EXE')

응용프로그램에서 실행 가능한 프로그램의 이름입니다.

USERDATA ('open, close, 235')

응용프로그램에서 사용할 수 있는 사용자 정의 데이터입니다.

프로세스 정의 표시

ALL 키워드와 함께 DISPLAY PROCESS 명령을 사용하여 정의 결과를 조사하십시오. 예를 들면 다음과 같습니다.

```
DISPLAY PROCESS (MOTOR.INS.PROC) ALL

24 : DISPLAY PROCESS (MOTOR.INS.PROC) ALL
AMQ8407: Display Process details.
DESCR ('Insurance request message processing') +
APPLICID ('DKA0:[MQM.ADMIN.TEST]IRMP01.EXE') +
USERDATA (open, close, 235) +
PROCESS (MOTOR.INS.PROC) +
APPLTYPE (OPENVMS)
```

트리거를 위한 오브젝트 관리

또한 MQSC 명령 ALTER PROCESS를 사용하여 기존의 프로세스 정의를 대체할 수 있고, DELETE PROCESS를 사용하여 프로세스 정의를 삭제할 수 있습니다.

제5장 관리 작업 자동화

이 장에서는 사용자가 MQSeries 오브젝트를 관리한 경험이 있는 것으로 가정합니다.

일부 관리 및 모니터링 작업의 자동화가 설치 시스템에 유용할 지의 여부를 결정해야 할 때가 있습니다. 프로그래밍 가능 명령 형식(PCF) 명령을 사용하여 로컬 및 리모트 큐 관리자에 대해 관리 작업을 자동화할 수 있습니다.

이 장에서는 다음에 대해 설명합니다.

- 『PCF 명령』에서, 관리 작업을 자동화하기 위해 프로그래밍 가능 명령 형식을 사용하는 방법
- 67 페이지의 『리모트 관리를 위한 명령 서버 관리』에서, 명령 서버 사용 방법

PCF 명령

MQSeries 프로그래밍 가능 명령 형식(PCF) 명령은 관리 작업이 관리 프로그램 내에 프로그램되도록 합니다. 이러한 방법으로 프로그램에서 큐, 프로세스 정의, 채널 및 이름 목록을 작성하고 큐 관리자를 변경할 수 있습니다.

PCF 명령에서는 MQSC 기능이 제공하는 동일한 범위의 기능을 다룹니다.

그러므로, 단일 노드에서 네트워크에 있는 큐 관리자로 PCF 명령을 발행하도록 프로그램을 작성할 수 있습니다. 이러한 방식으로 관리 태스크를 중앙 집중화하고 자동화할 수 있습니다.

각 PCF 명령은 MQSeries 메시지의 응용프로그램 데이터 부분에 삽입되는 데이터 구조입니다. 각 명령은 다른 메시지와 같은 방법으로 MQI 함수 MQPUT을 사용하여 타겟 큐 관리자로 전송됩니다. 메시지를 수신하는 큐 관리자의 명령 서버는 그 메시지를 명령 메시지로 해석하고 그 명령을 실행합니다. 응답을 받기 위해 응용프로그램은 MQGET 호출을 발행하고 응답 데이터는 다른 데이터 구조에 리턴됩니다. 그러면 응용프로그램은 응답을 처리하고 그에 따라 조치를 취할 수 있습니다.

주: MQSC 명령과는 달리, PCF 명령 및 응답은 읽을 수 있는 텍스트 형식이 아닙니다.

다음은 응용프로그램 프로그래머가 PCF 명령 메시지를 작성하기 위해 지정해야 하는 몇 가지 사항입니다.

메시지 설명자

표준 MQSeries 메시지 설명자입니다. 여기서,

메시지 유형(*MsgType*)은 MQMT_REQUEST입니다.

메시지 형식(*Format*)은 MQFMT_ADMIN입니다.

응용프로그램 데이터

PCF 헤더를 포함하는 PCF 메시지가 들어 있습니다. 여기서,

PCF 메시지 유형(*Type*)은 MQCFT_COMMAND를 지정합니다.

명령 ID는 명령을 식별합니다. 예를 들면 다음과 같습니다. *Change Queue* (MQCMD_CHANGE_Q).

PCF 데이터 구조와 그 구조를 구현하는 방법에 대한 자세한 설명은 *MQSeries Programmable System Management*를 참조하십시오.

MQSC 및 PCF의 속성

MQSC에 지정되는 오브젝트 속성은 대소문자를 구별하지 않지만 이 서적에서는 대문자로 표시되어 있습니다(예: RQMNAME). MQSC 속성 이름은 8자를 초과할 수 없습니다.

8자로 제한되지 않는 PCF의 오브젝트 속성은 이 서적에서 기울임체로 표시됩니다. 예를 들어, RQMNAME에 해당되는 PCF는 *RemoteQMgrName*입니다.

이스케이프 PCF

이스케이프 PCF는 메시지 텍스트 내에 MQSC 명령을 포함하는 PCF 명령입니다. PCF를 사용하여 명령을 리모트 큐 관리자로 보낼 수 있습니다. 이스케이프 PCF 사용에 대한 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.

MQAI를 통한 간편한 PCF 사용

MQAI는 OpenVMS 플랫폼에서 사용할 수 있는 MQSeries의 관리 인터페이스입니다.

이 MQAI는 데이터 백을 통해 큐 관리자에 대한 관리 작업을 수행합니다. 데이터 백을 사용하면 PCF를 사용할 때 보다 더 쉬운 방식으로 오브젝트의 등록 정보(또는 매개변수)를 핸들링할 수 있습니다.

다음과 같은 경우에 MQAI를 사용할 수 있습니다.

- **PCF 메시지의 사용을 간편하게 할 경우**

MQAI를 사용하면 MQSeries를 쉽게 관리할 수 있습니다. 예를 들어 고유한 PCF 메시지를 작성하지 않아도 되기 때문에 복잡한 데이터 구조와 연관되는 문제점을 피할 수 있습니다.

MQI 호출을 사용하여 작성한 프로그램 내의 매개변수를 전달하려면, PCF 메시지에 명령과 문자열 또는 정수 데이터의 세부사항이 포함되어야 합니다. 이렇게 하려면, 프로그램의 각 구조에 대해 몇 개의 명령문이 필요하며 또한 메모리 공간을 할당해야 합니다. 이것은 시간이 오래 걸리며 복잡한 작업입니다.

반면에, MQAI를 사용하여 작성된 프로그램은 매개변수를 적절한 데이터 백에 전달하므로 각 구조에 대해 하나의 명령문만 있으면 됩니다. MQAI 데이터 백을 사용하면 배열을 핸들링하고 저장영역을 할당할 필요가 없기 때문에 PCF의 세부적인 사항을 신경쓰지 않아도 됩니다.

- 자체 관리 응용프로그램과 관리 도구를 구현하려 할 경우

예를 들어, Windows NT 및 Windows 2000용 MQSeries 버전 5.2에서 제공되는 Active Directory Services는 MQAI를 사용합니다(현재 OpenVMS 플랫폼에서는 이 사용법에 대한 예가 없습니다).

- 오류 상태를 더욱 쉽게 핸들링하려는 경우

MQSC 명령으로부터 다시 리턴 코드를 받는 것은 쉽지 않지만, MQAI를 사용하면 프로그램이 쉽게 오류 상태를 핸들링할 수 있습니다.

데이터 백을 작성하여 상주시키고 나면, 응답 메시지를 대기하는 mqExecute 호출을 사용하여 큐 관리자의 명령 서버로 관리 명령 메시지를 보낼 수 있습니다. mqExecute 호출은 명령 서버와의 교환을 핸들링하고 응답 백에 응답을 리턴합니다.

MQAI 사용에 대한 자세한 정보는 *MQSeries Administration Interface Programming Guide and Reference*를 참조하십시오.

PCF에 대한 일반적인 내용에 대한 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.

리모트 관리를 위한 명령 서버 관리

각 큐 관리자는 연관된 명령 서버를 가질 수 있습니다. 명령 서버에서는 리모트 큐 관리자에서 수신되는 명령이나, 응용프로그램에서 수신되는 PCF 명령을 처리합니다. 명령 서버는 큐 관리자로 명령을 보내 명령이 처리되도록 하고 명령의 출처에 따라 완료 코드나 연산자 메시지를 리턴합니다.

명령 서버는 PCF, MQAI를 포함하는 모든 관리와, 리모트 관리에 대해서도 필수입니다.

주: 리모트 관리의 경우에는 타겟 큐 관리자가 실행 중인지 확인해야 합니다. 그렇지 않으면, 명령을 포함하는 메시지가 해당 메시지를 발행한 큐 관리자에서 나갈 수 없습니다. 이런 경우 이 메시지들은 리모트 큐 관리자를 지원하는 로컬 트랜스미션 큐에서 대기합니다. 가능하면 이러한 상황을 피해야 합니다.

명령 서버 시작

이 명령을 사용하여 명령 서버를 시작하려면 다음을 입력하십시오.

명령 서버 리모트 관리

```
strmqcsv "saturn.queue.manager"
```

여기서 saturn.queue.manager는 명령 서버가 시작되는 큐 관리자입니다.

명령 서버 상태 표시

리모트 관리의 경우, 타겟 큐 관리자의 명령 서버가 실행 중인지 확인하십시오. 실행되고 있지 않으면, 리모트 명령을 처리할 수 없습니다. 명령을 포함하는 모든 메시지는 타겟 큐 관리자의 명령 큐에서 대기합니다.

여기서 saturn.queue.manager라는 큐 관리자의 명령 서버 상태를 표시하려면, 다음 명령을 입력하십시오.

```
dspmqcsv "saturn.queue.manager"
```

대상 시스템에서 이 명령을 발행해야 합니다. 명령 서버가 실행 중이면, 다음 메시지가 리턴됩니다.

```
AMQ8027    MQSeries Command Server Status ...: Running
```

명령 서버 정지

명령 서버를 종료하려면, 다음 명령을 입력하십시오(이전 예를 사용할 경우).

```
endmqcsv "saturn.queue.manager"
```

다음과 같은 두 가지 방법으로 명령 서버를 정지할 수 있습니다.

- 제어 정지의 경우, -c 플래그와 함께 **endmqcsv** 명령을 사용하십시오. 이것이 디폴트입니다.
- 즉시 정지의 경우, -i 플래그와 함께 **endmqcsv** 명령을 사용하십시오.

주: 큐 관리자를 정지하면 그 큐 관리자와 연관되는 명령 서버도 종료됩니다(명령 서버가 이미 시작된 경우).

제6장 리모트 MQSeries 오브젝트 관리

이 장에서는 다른 큐 관리자의 MQSeries 오브젝트를 관리하는 방법에 대해 설명합니다. 또한 리모트 큐 오브젝트를 사용하여 메시지의 목적지와 응답 메시지를 제어하는 방법에 대해서도 설명합니다.

이 장에는 다음과 같은 절이 있습니다.

- 『채널, 클러스터 및 리모트 큐잉』
- 71 페이지의 『MQSC 명령을 사용한 로컬 큐 관리자에서의 리모트 관리』
- 78 페이지의 『리모트 큐의 로컬 정의 작성』
- 81 페이지의 『별명으로 리모트 큐 정의 사용』

채널과 그 속성 그리고 이를 설정하는 방법에 대한 자세한 정보는 *MQSeries 상호통신* 을 참조하십시오.

채널, 클러스터 및 리모트 큐잉

큐 관리자는 메시지를 송신하고 필요에 따라 응답을 다시 수신하여 다른 큐 관리자와 통신합니다. 리시브 큐 관리자는,

- 같은 시스템에 있을 수 있습니다.
- 같은 위치의 다른 시스템 또는 다른 국가에 있을 수 있습니다.
- 로컬 큐 관리자와 같은 플랫폼에서 실행할 수 있습니다.
- MQSeries에서 지원하는 다른 플랫폼에서 실행할 수 있습니다.

이 메시지들은 다음에서 생성됩니다.

- 노드 간에 데이터를 전송하는 사용자 작성 응용프로그램.
- PCF, MQAI 또는 ADSI를 사용하는 사용자 작성 관리 응용프로그램.
- 다음을 송신하는 큐 관리자
 - 다른 큐 관리자에 대한 도구 이벤트 메시지.
 - 간접 모드에서 **runmqsc** 명령으로 발행된 MQSC(여기서 명령은 다른 큐 관리자에서 실행됩니다).

메시지를 다른 리모트 큐 관리자에 송신하려면, 로컬 큐 관리자에 메시지 도착을 감지하고, 다음을 구성하는 메시지를 전송하기 위한 메커니즘이 있어야 합니다.

- 최소한 하나의 채널
- 트랜스미션 큐
- 메시지 채널 에이전트(MCA)
- 채널 리스너

리모트 오브젝트 관리

- 채널 시작기

채널은 두 개의 큐 관리자 사이의 단방향 통신 링크로, 리모트 큐 관리자에 있는 큐로 목적지가 지정된 메시지를 전달할 수 있습니다.

채널의 각 끝에는 분리된 정의가 있습니다. 예를 들어, 한 쪽 끝이 송신자나 서버이면, 다른 쪽 끝은 수신자나 요청자여야 합니다. 단순한 채널은 로컬 큐 관리자에 있는 송신자 채널 정의와 리모트 큐 관리자에 있는 수신자 채널 정의로 구성됩니다. 두 개의 정의는 이름이 같아야 하며 함께 단일 채널을 구성해야 합니다.

로컬 큐 관리자가 송신한 메시지에 리모트 큐 관리자가 응답할 것으로 예상되면, 로컬 큐 관리자에 다시 응답을 보내기 위해 두 번째 채널이 설정되어야 합니다.

채널은 MQSC DEFINE CHANNEL 명령을 사용하여 정의됩니다. 이 장에서는 달리 명시하지 않는 한 채널에 관련되는 예에서 디폴트 채널 속성을 사용합니다.

메시지 송수신을 제어하는 채널의 각 끝에는 메시지 채널 에이전트(MCA)가 있습니다. MCA는 트랜스미션 큐에서 메시지를 취해 큐 관리자 사이의 통신 링크에 그 메시지를 넣는(put) 작업을 수행합니다.

트랜스미션 큐는 메시지가 MCA에 의해 선택되어 리모트 큐 관리자에 송신되기 전에 임시로 메시지를 보유하는 특수화된 로컬 큐입니다. 리모트 큐 정의에서 트랜스미션 큐의 이름을 지정할 수 있습니다.

72 페이지의 『리모트 관리를 위한 채널 및 트랜스미션 큐 준비』에서는 이러한 정의를 사용하여 리모트 관리를 설정하는 방법을 보여줍니다.

일반적으로 분산 큐잉을 설정하는 방법에 대한 자세한 정보는 *MQSeries* 상호통신을 참조하십시오.

클러스터를 사용한 리모트 관리

분산 큐잉을 사용하는 일반적인 MQSeries 네트워크에서는 모든 큐 관리자가 독립적입니다. 하나의 큐 관리자가 메시지를 다른 큐 관리자에 송신해야 할 경우, 그 큐 관리자에는 트랜스미션 큐, 리모트 큐 관리자에 대한 채널, 그리고 메시지를 송신할 모든 큐에 대한 리모트 큐 정의가 정의되어 있어야 합니다.

클러스터는 큐 관리자들이 복잡한 트랜스미션 큐, 채널 및 큐 정의 없이 단일 네트워크를 통해 서로 직접 통신할 수 있는 방식으로 설정된 일련의 큐 관리자입니다. 클러스터는 쉽게 설정할 수 있으며, 보통 어떤 방식에 의해 논리적으로 관련되는 큐 관리자를 포함하고 있고 데이터나 응용프로그램을 공유해야 합니다.

클러스터를 작성하고 나면, 그 안에 있는 큐 관리자는 복잡한 채널이나 리모트 큐 정의 없이 서로 통신할 수 있습니다. 클러스터가 아무리 작은 경우에도 시스템 관리 오버헤드가 감소됩니다.

클러스터에서의 큐 관리자 네트워크 설정은 일반적인 분산 큐잉 환경을 설정하는 것보다 더 적은 정의를 포함합니다. 작성할 정의가 적어지면, 더 빠르고 쉽게 네트워크를 설정하거나 변경할 수 있으며, 정의에서의 오류 발생 위험도 줄어 들 수 있습니다.

클러스터를 설정하려면, 보통 큐 관리자당 하나의 클러스터 송신자(CLUSSDR) 정의와 하나의 클러스터 수신자(CLUSRCVR) 정의가 필요합니다. 트랜스미션 큐 정의나 리모트 큐 정의는 필요하지 않습니다. 리모트 관리의 원리는 클러스터 내에서 사용될 경우와 같지만, 정의 자체는 아주 간단합니다.

클러스터와 그 속성, 그리고 이를 설정하는 방법에 대한 자세한 정보는 *MQSeries Queue Manager Clusters*를 참조하십시오.

MQSC 명령을 사용한 로컬 큐 관리자에서의 리모트 관리

이 절에서는 로컬 큐 관리자에서 리모트 큐 관리자를 관리하는 방법을 알려줍니다. 다음을 사용하여 로컬 노드에서 리모트 관리를 구현할 수 있습니다.

- MQSC 명령
- PCF 명령

큐와 채널에 대한 준비는 본질적으로 두 가지 방법에서 동일합니다. 이 서적에서 사용되는 예에서는 MQSC 명령이 이해하기가 쉬우므로 이 명령을 보여주고 있습니다. 그러나, 원한다면 예를 PCF로 변환할 수 있습니다. PCF를 사용하여 관리 프로그램을 작성하는 방법에 대한 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.

리모트 관리에서는 대화식으로 또는 명령을 포함하는 텍스트 파일에서 MQSC 명령을 리모트 큐 관리자에 송신합니다. 리모트 큐 관리자는 같은 시스템에 있거나 다른 시스템에 있을 수 있는데, 일반적으로 다른 시스템에 있습니다. 다른 MQSeries 환경(예: AIX, AS/400, MVS/ESA, OS/2)에서 리모트로 큐 관리자를 관리할 수 있습니다.

리모트 관리를 구현하려면, 특정 오브젝트를 작성해야 합니다. 요구사항을 특수화하지 않았을 경우에는 디폴트 값(예를 들어, 메시지의 길이에 대한 디폴트 값)으로 충분한지 확인해야 합니다.

리모트 관리에 대한 큐 관리자 준비

72 페이지의 그림6에서는 `runmqsc` 명령을 사용한 리모트 관리에 필요한 채널과 큐 관리자의 구성을 보여줍니다. `source.queue.manager`는 MQSC 명령을 발행할 수 있고 이 명령의 결과(연산자 메시지)가 리턴될(가능할 경우) 소스 큐 관리자입니다. `target.queue.manager`는 명령을 처리하고 모든 연산자 메시지를 생성하는 데스티네이션 큐 관리자입니다.

주: source.queue.manager는 반드시 디폴트 큐 관리자여야 합니다. 큐 관리자 작성에 대한 자세한 정보는 266 페이지의 『crtmqm(큐 관리자 작성)』을 참조하십시오.

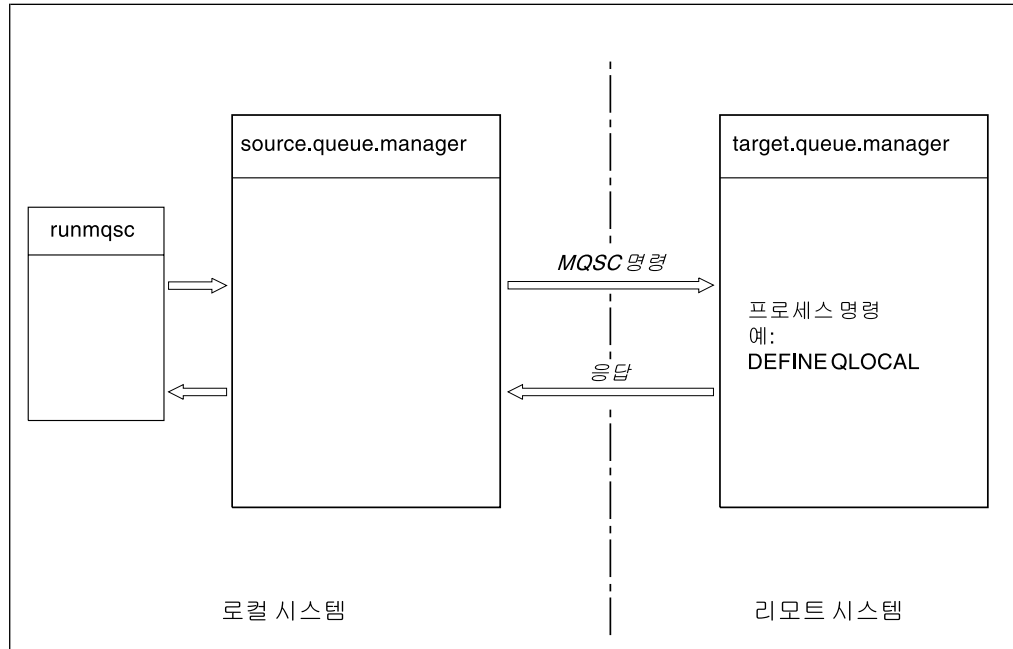


그림 6. 리모트 관리

두 시스템 모두에서 아직 그렇게 수행하지 않았다면, 다음을 수행해야 합니다.

- **crtmqm** 명령을 사용하여 큐 관리자를 작성하십시오.
- **strmqm** 명령을 사용하여 큐 관리자를 시작하십시오.

이 명령들은 로컬이나 네트워크 기능(예: Telnet)을 통해 실행해야 합니다.

데스티네이션 큐 관리자에서는,

- 명령 큐인 SYSTEM.ADMIN.COMMAND.QUEUE가 있어야 합니다. 이 큐는 큐 관리자가 적성될 때 디폴트로 작성합니다.
- 명령 서버는 **strmqcsv** 명령을 사용하여 시작해야 합니다.

리모트 관리를 위한 채널 및 트랜스미션 큐 준비

MQSC 명령을 리모트로 실행하려면, 두 개의 채널(각 방향마다 하나씩)과 이에 연관되는 트랜스미션 큐를 설정해야 합니다. 이 예에서는 TCP/IP가 전송 유형으로 사용되며 사용자가 관련된 TCP/IP 주소를 알고 있는 것으로 가정합니다.

채널 source.to.target은 소스 큐 관리자에서 목적지로 MQSC 명령을 송신합니다. 그 송신자는 source.queue.manager에 있고 그 수신자는 큐 관리자 target.queue.manager에 있습니다. 채널 target.to.source는 소스 큐 관리자에 대해 생성되는 연산자 메시지와 명령으로부터의 출력을 리턴합니다. 각 송신자에 대해 트랜스미션 큐도 정의해야 합니다. 이 큐는 수신하는 큐 관리자의 이름이 부여되는 로컬 큐입니다. XMITQ 이름은 큐 관리자 별명을 사용하지 않는 한 리모트 관리에 대한 리

모트 큐 관리자 이름과 일치해야 합니다. 그림7에서는 이 구성에 대한 요약을 보여줍니다.

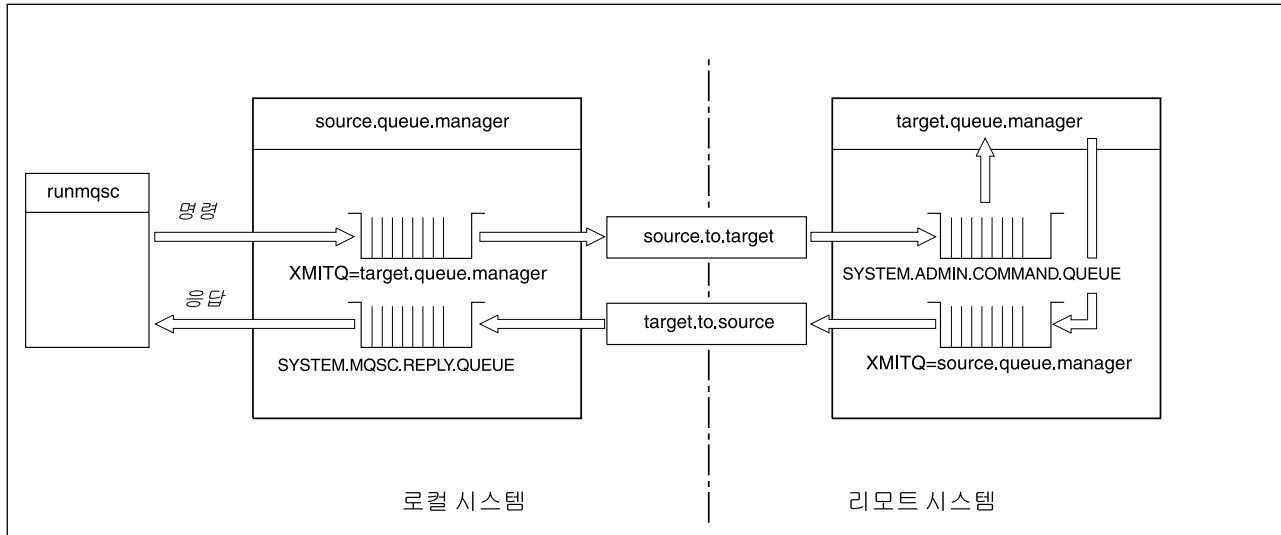


그림7. 리모트 관리에 대한 채널 및 큐 설정

리모트 채널 설정에 대한 자세한 정보는 *MQSeries 상호통신*을 참조하십시오.

채널 및 트랜스미션 큐 정의

소스 큐 관리자에서, 다음의 MQSC 명령을 발행하여 채널과 트랜스미션 큐를 정의하십시오.

```

* Define the sender channel at the source queue manager
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME (RHX5498) +
  XMITQ ('target.queue.manager') +
  TRPTYPE(TCP)

* Define the receiver channel at the source queue manager

DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)

* Define the transmission queue on the source

DEFINE QLOCAL ('target.queue.manager') +
  USAGE (XMITQ)
    
```

채널과 트랜스미션 큐를 작성하려면 데스티네이션 큐 관리자(target.queue.manager)에서 다음 명령을 발행하십시오.

```
* Define the sender channel on the destination queue manager

DEFINE CHANNEL ('target.to.source') +
    CHLTYPE(SDR) +
    CONNAME (RHX7721) +
    XMITQ ('source.queue.manager') +
    TRPTYPE(TCP)

* Define the receiver channel on the destination queue manager

DEFINE CHANNEL ('source.to.target') +
    CHLTYPE(RCVR) +
    TRPTYPE(TCP)

* Define the transmission queue on the destination queue manager

DEFINE QLOCAL ('source.queue.manager') +
    USAGE (XMITQ)
```

주: 송신자 채널 정의에서 CONNAME 속성에 대해 지정된 TCP/IP 연결 이름은 단지 설명을 위한 것입니다. 이것은 연결의 다른 끝에 있는 시스템의 네트워크 이름입니다. 사용자 네트워크에 맞는 값을 사용하십시오.

채널 시작

다음 설명에서는 채널의 양쪽 끝이 Compaq OpenVMS용 MQSeries에서 실행되고 있음을 가정합니다. 그러한 경우가 아니면, OpenVMS 이외 채널 끝에 대한 관련 문서를 참조하십시오.

두 개의 채널을 시작하려면, 먼저 두 노드에서 MQSeries에 대해 TCP/IP 서비스가 구성되어 있고 연결의 양쪽 끝에서 그 서비스가 실행되고 있는지 확인하십시오.

각 채널의 수신자 끝에서 리스너를 시작하십시오.

- 소스 큐 관리자에서 다음을 입력하십시오.

```
runmq1sr -m "source.queue.manager" -t tcp
```

- 테스트네이션 큐 관리자에서 다음을 입력하십시오.

```
runmq1sr -m "target.queue.manager" -t tcp
```

소스 큐 관리자에 대한 **runmqchl** 명령에서 큐 관리자 이름은 필수가 아닙니다. 소스 큐 관리자는 디폴트 큐 관리자이기 때문입니다. 테스트네이션 큐 관리자가 해당 노드에서 디폴트 큐 관리자가 아니면 테스트네이션 큐 관리자에 대해 **runmqchl** 명령에서 큐 관리자 이름이 필요합니다.

그런 다음, 채널을 시작하십시오.

- 소스 큐 관리자에서, 다음을 입력하십시오.

```
runmqchl -m "source.queue.manager" -c "source.to.target"
```

- 테스트네이션 큐 관리자에서, 다음을 입력하십시오.

```
runmqchl -m "target.queue.manager" -c "target.to.source"
```

runmqslr 및 **runmqchl** 명령은 MQSeries 제어 명령입니다. 그 명령들은 **runmqsc** 를 사용하여 발행할 수 없습니다. 리스너와 채널은 **runmqsc** 명령이나 스크립트를 사용하여 시작할 수 있습니다(시작 채널 및 시작 리스너).

채널의 자동 정의

채널의 자동 정의는 테스트네이션 큐 관리자가 MQSeries 버전 5.1 이상의 제품에서 실행 중인 경우에만 적용됩니다. 인바운드 첨부 요청이 수신되고 적절한 수신자 또는 서버 연결 목적지를 채널 정의 파일(CDF)에서 찾을 수 없으면, MQSeries는 자동으로 정의를 작성하여 이를 CDF에 추가합니다. 자동 정의는 MQSeries: SYSTEM.AUTO.RECEIVER 및 SYSTEM.AUTO.SVRCONN과 함께 제공되는 두 개의 디폴트 정의를 기초로 합니다.

MQSC 명령인 ALTER QMGR(또는 PCF 명령인 Change Queue Manager)로 큐 관리자 오브젝트를 갱신하여 수신자 및 송신자 연결 정의에 대한 자동 정의가 사용 가능하도록 할 수 있습니다.

채널 정의의 자동 작성에 대한 자세한 정보는 *MQSeries 상호통신*을 참조하십시오.

클러스터에 대한 채널의 자동 정의에 대해서는 *MQSeries Queue Manager Clusters*를 참조하십시오.

리모트로 MQSC 명령 발행

명령 서버는 MQSC 명령을 원격으로 처리할 경우 반드시 테스트네이션 큐 관리자에서 실행 중 상태여야 합니다(이것은 소스 큐 관리자에서 필요하지 않습니다).

리모트 관리

- 데스티네이션 큐 관리자에서 다음을 입력하십시오.

```
strmqcsv "target.queue.manager"
```

- 소스 큐 관리자에서는 다음을 입력하여 큐잉 모드에서 대화식으로 MQSC를 실행할 수 있습니다.

```
runmqsc -w 30 "target.queue.manager"
```

runmqsc 명령 형식(-w 플래그가 있는)은 큐잉 모드에서 MQSC 명령을 실행합니다. 여기서, 명령은 수정된 형식으로 명령 서버 입력 큐에 넣어져서(put) 순서대로 실행됩니다.

MQSC 명령을 입력할 때, 이 명령은 리모트 큐 관리자(이 경우, target.queue.manager)로 경로가 재지정됩니다. 시간 종료는 30초로 설정됩니다. 응답이 30초 내에 수신되지 않으면, 로컬(소스) 큐 관리자에서 다음 메시지가 생성됩니다.

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

MQSC 세션 끝에서, 로컬 큐 관리자는 도착한 시간 종료 응답을 표시합니다. MQSC 세션이 완료되면, 추가 응답은 제거됩니다.

간접 모드에서는 리모트 큐 관리자에서 MQSC 명령 파일을 실행할 수도 있습니다. 예를 들면 다음과 같습니다.

```
runmqsc -w 60 "target.queue.manager" < mycomds.in > report.out
```

여기서 mycomds.in은 MQSC 명령을 포함하는 파일이고 report.out은 보고서 파일입니다.

MVS/ESA에서 큐 관리자에 대한 작업

Compaq OpenVMS용 MQSeries 큐 관리자에서 MVS/ESA 큐 관리자에 대해 MQSC 명령을 발행할 수 있습니다. 그러나, 이렇게 하려면 송신자에서 **runmqsc** 명령과 채널 정의를 수정해야 합니다.

특히, OpenVMS 노드에서 -x 플래그를 **runmqsc** 명령에 추가해야 합니다.

```
runmqsc -w 30 -x "target.queue.manager"
```

송신자 채널에서, CONVERT 속성을 YES로 설정하십시오. 이것은 시스템 사이에 필요한 데이터 변환이 OpenVMS 측에서 수행됨을 지정합니다. 채널 정의 명령은 이제 다음과 같이 됩니다.

```
* Define the sender channel at the source queue manager on OpenVMS

DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME (RHX5498) +
  XMITQ ('target.queue.manager') +
  TRPTYPE(TCP) +
  CONVERT (YES)
```

이전처럼 소스 큐 관리자에서 수신자 채널과 트랜스미션 큐도 정의해야 합니다. 다시, 이 예에서는 전송 프로토콜로 TCP/IP가 사용된다고 가정합니다.

리모트 큐잉에 대한 권장사항

리모트 큐잉을 구현할 경우, 다음을 수행하십시오.

1. 리모트 시스템에서 실행될 MQSC 명령을 명령 파일에 넣으십시오(put).
2. **runmqsc** 명령에서 -v 플래그를 지정하여 MQSC 명령을 로컬로 확인하십시오. 다른 큐 관리자에서 MQSC 명령을 확인할 경우에는 **runmqsc**를 사용할 수 없습니다.
3. 가능하면 명령 파일이 오류 없이 로컬로 실행되는지 점검하십시오.
4. 마지막으로, 리모트 시스템에 대해 명령 파일을 실행하십시오.

리모트로 MQSC를 사용할 때 문제점이 발생할 경우

MQSC 명령을 리모트로 실행하는 데 어려움이 있을 경우에는, 다음 점검 목록을 사용하여 다음 사항들을 수행했는지 점검하십시오.

- 테스트네이션 큐 관리자에서 명령 서버를 시작하였는지.
- 올바른 트랜스미션 큐를 정의하였는지.
- 다음 두 가지에 대해 메시지 채널의 양 끝을 정의하였는지.
 - 명령이 송신되는 채널.
 - 응답이 리턴되는 채널.
- 채널 정의에서 올바른 연결 이름(CONNAME)을 지정하였는지.
- 메시지 채널을 시작하기 전에 리스너를 시작하였는지.

리모트 관리

- 예를 들어 채널이 시작되었지만 잠시 후 종료된 경우, 연결 끊기 간격이 만기되지 않았는가를 확인했는지. 이것은 특히 수동으로 채널을 시작할 경우에 중요합니다.
- 데스티네이션 큐 관리자에 대해 감지하지 못하는 큐 관리자로부터의 요청(예를 들어, 새로운 매개변수를 포함하는 요청)을 송신하지 않았는가를 확인했는지.

또한 48 페이지의 『MQSC 문제점 해결』을 참조하십시오.

리모트 큐의 로컬 정의 작성

리모트 큐의 로컬 정의로 리모트 큐 정의를 사용할 수 있습니다. 다른 큐 관리자에서 로컬 큐를 식별하려면 로컬 큐 관리자에서 리모트 큐 오브젝트를 작성하십시오.

리모트 큐의 로컬 정의가 작동되는 방법에 대한 이해

응용프로그램은 로컬 큐 관리자에 연결하여 MQOPEN 호출을 발행합니다. 열린 호출에서 지정된 큐 이름은 로컬 큐 관리자의 리모트 큐 정의 이름입니다. 리모트 큐 정의는 데스티네이션 큐, 데스티네이션 큐 관리자 및 선택적으로 트랜스미션 큐의 이름을 제 공합니다. 메시지를 리모트 큐에 넣으려면(put), 응용프로그램은 MQOPEN 호출에서 리턴된 핸들을 지정하여 MQPUT 호출을 발행합니다. 큐 관리자는 리모트 큐 이름과 리모트 큐 관리자 이름을 메시지에서 리모트 큐 관리자 헤더에 추가합니다. 이 정보는 네트워크에서 메시지를 해당되는 올바른 목적지로 라우트하는데 사용됩니다.

사용자가 관리자인 경우 리모트 큐 정의를 변경하여 메시지의 목적지를 제어할 수 있습니다.

예

목적: 응용프로그램은 리모트 큐 관리자가 소유하는 큐에 메시지를 넣도록(put) 요구합니다.

작동 원리: 응용프로그램은 saturn.queue.manager와 같은 큐 관리자에 연결합니다. 데스티네이션 큐는 다른 큐 관리자가 소유합니다.

MQOPEN 호출에서 응용프로그램은 다음 필드들을 지정합니다.

필드 값	설명
<i>ObjectName</i> CYAN.REMOTE.QUEUE	리모트 큐 오브젝트의 로컬 이름을 지정합니다. 이것은 데스티네이션 큐와 데스티네이션 큐 관리자를 정의합니다.
<i>ObjectType</i> (Queue)	이 오브젝트를 큐로 식별합니다.
<i>ObjectQmgrName</i> Blank 또는 saturn.queue.manager	이 필드는 선택적입니다. 공백일 경우, 로컬 큐 관리자의 이름이 사용됩니다(이것은 리모트 큐 정의가 수행되고 응용프로그램이 연결된 큐 관리자입니다). 공백일 경우 로컬 큐 관리자의 이름을 지정해야 합니다.

이후에 응용프로그램은 MQPUT 호출을 발행하여 메시지를 이 큐에 넣습니다(put).

로컬 큐 관리자에서 다음 MQSC 명령을 사용하여 리모트 큐의 로컬 정의를 작성할 수 있습니다.

```
DEFINE QREMOTE (CYAN.REMOTE.QUEUE) +
  DESCR ('Queue for auto insurance requests from the branches') +
  RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE) +
  RQMNAME ('jupiter.queue.manager') +
  XMITQ (INQUOTE.XMIT.QUEUE)
```

설명:

QREMOTE (CYAN.REMOTE.QUEUE)

리모트 큐 오브젝트의 로컬 이름을 지정합니다. 이것은 리모트 큐 관리자 jupiter.queue.manager에서 AUTOMOBILE.INSURANCE.QUOTE.QUEUE 큐를 열기 위해 MQOPEN 호출에서 큐 관리자에 연결된 응용프로그램이 지정해야 하는 이름입니다.

DESCR ('Queue for auto insurance requests from the branches')

큐 사용을 설명하는 추가 텍스트입니다.

RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE)

리모트 큐 관리자에서 테스트네이션 큐의 이름을 지정합니다. 이것은 큐 이름 CYAN.REMOTE.QUEUE를 지정하는 응용프로그램에 의해 송신되는 메시지들에 대한 실제 테스트네이션 큐입니다. 큐 AUTOMOBILE.INSURANCE.QUOTE.QUEUE는 리모트 큐 관리자에서 로컬 큐로 정의해야 합니다.

RQMNAME ('jupiter.queue.manager')

테스트네이션 큐 AUTOMOBILE.INSURANCE.QUOTE.QUEUE를 소유하는 리모트 큐 관리자의 이름을 지정합니다. 이 이름은 작은 따옴표로 묶어야 합니다.

XMITQ (INQUOTE.XMIT.QUEUE)

트랜스미션 큐의 이름을 지정합니다. 이것은 선택적입니다. 지정하지 않을 경우 리모트 큐 관리자와 이름이 같은 큐가 사용됩니다.

어느 경우에서든지 트랜스미션 큐임을 지정하는 *Usage* 속성을 사용하여(MQSC에서 USAGE(XMIT)) 적절한 트랜스미션 큐를 로컬 큐로 정의해야 합니다.

리모트 큐에 메시지를 넣는 다른 방법

리모트 큐의 로컬 정의를 사용하는 것이 메시지를 리모트 큐에 넣는(put) 유일한 방법은 아닙니다. 응용프로그램은 MQOPEN 호출의 일부로 리모트 큐 관리자 이름을 포함

리모트 큐의 로컬 정의

하는 전체 큐 이름을 지정할 수 있습니다. 이 경우 리모트 큐의 로컬 정의는 필요하지 않습니다. 그러나 이 방법을 사용하려면 응용프로그램이 런타임시 리모트 큐 관리자의 이름을 알고 있거나 이 이름에 액세스할 수 있어야 합니다.

리모트 큐에서 다른 명령 사용

적절한 MQSC 명령을 사용하여 리모트 큐 오브젝트의 속성을 표시 또는 변경하거나, 리모트 큐 오브젝트를 삭제할 수 있습니다. 예를 들면 다음과 같습니다.

```
* Display the remote queue's attributes.
* ALL = Display all attributes

DISPLAY QUEUE (CYAN.REMOTE.QUEUE) ALL

* ALTER the remote queue to enable puts.
* This does not affect the destination queue,
* only applications that specify this remote queue.

ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)

* Delete this remote queue
* This does not affect the destination queue
* only its local definition

DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

주: 리모트 큐를 삭제할 경우 리모트 큐의 로컬 표현만 삭제하십시오. 리모트 큐 자체나 그 큐의 메시지들은 삭제하지 마십시오.

트랜스미션 큐 작성

트랜스미션 큐는 큐 관리자가 메시지 채널을 통해 메시지를 리모트 큐 관리자에 전달할 때 사용되는 로컬 큐입니다. 채널은 리모트 큐 관리자에 대한 단방향 링크를 제공합니다. 메시지는 채널에서 승인할 때까지 트랜스미션 큐에 넣어집니다. 채널을 정의할 때 메시지 채널의 송신 엔드에서 트랜스미션 큐 이름을 지정해야 합니다.

Usage 속성(MQSC에서의 USAGE)은 큐가 트랜스미션 큐인지 아니면 일반 큐인지를 지정합니다.

디폴트 트랜스미션 큐

선택적으로 *XmitQName* 속성(MQSC에서 XMITQ)을 사용하여 리모트 큐 오브젝트에서 트랜스미션 큐를 지정할 수 있습니다. 정의된 트랜스미션 큐가 없으면 디폴트가 사용됩니다. 응용프로그램이 메시지를 리모트 큐에 넣을(put) 경우, 데스티네이션 큐 관리

자와 이름이 같은 트랜스미션 큐가 존재하면, 그 큐가 사용됩니다. 이 큐가 존재하지 않으면 로컬 큐 관리자에서 *DefaultXmitQ* 속성(MQSC에서 DEFXMITQ)에 지정된 큐가 사용됩니다.

예를 들어, 다음 MQSC 명령은 `target.queue.manager`로 가는 메시지에 대해 `source.queue.manager`에서 디폴트 트랜스미션 큐를 작성합니다.

```
DEFINE QLOCAL ('target.queue.manager') +
DESCR ('Default transmission queue for target qm') +
USAGE (XMITQ)
```

응용프로그램은 적절한 헤더와 함께 트랜스미션 큐에 직접 메시지를 넣거나(put), 간접적으로(예를 들어, 리모트 큐 정의를 통해) 넣을(put) 수 있습니다. 78 페이지의 『리모트 큐의 로컬 정의 작성』을 참조하십시오.

별명으로 리모트 큐 정의 사용

다른 큐 관리자에서 큐를 찾는 것 외에도 다음 리모트 큐의 로컬 정의를 사용할 수 있습니다.

- 큐 관리자 별명
- 리플라이-투 큐 별명

두 유형의 별명 모두 리모트 큐의 로컬 정의를 통해 해석합니다.

일반적으로 리모트 큐잉에서는 메시지가 해당되는 목적지에 도착하려면 적절한 채널이 설정되어야 합니다.

큐 관리자 별명

별명은 데스티네이션 큐 관리자의 이름(메시지에 지정된)이 메시지 라우트에서 큐 관리자에 의해 수정되는 프로세스입니다. 큐 관리자 별명은 큐 관리자 네트워크 내에서 메시지 목적지를 제어하는 데 사용할 수 있기 때문에 중요합니다.

제어 지점에서 큐 관리자의 리모트 큐 정의를 변경하여 이를 수행할 수 있습니다. 송신 응용프로그램은 지정된 큐 관리자 이름이 별명이라는 것을 인식하지 못합니다.

큐 관리자 별명에 대한 자세한 정보는 *MQSeries* 상호통신을 참조하십시오.

리플라이-투 큐 별명

선택적으로 응용프로그램은 요청 메시지를 큐에 넣을 때 리플라이-투 큐의 이름을 지정할 수 있습니다. 메시지를 처리하는 응용프로그램이 리플라이-투 큐의 이름을 추출할 경우, 필요하면 응답 메시지를 송신할 곳을 인식합니다.

리플라이-투 큐 별명은 리플라이-투 큐(요청 메시지에 지정된)가 메시지 라우트에서 큐 관리자에 의해 변경되는 프로세스입니다. 송신 응용프로그램은 지정된 리플라이-투 큐 이름이 별명이라는 것을 인식하지 못합니다.

리플라이-투 큐 별명을 사용하면 리플라이-투 큐의 이름과 선택적으로 큐 관리자의 이름을 변경할 수 있습니다. 그러면, 응답 메시지에 사용되는 라우트를 제어할 수 있습니다.

요청 메시지, 응답 메시지 및 리플라이-투 큐에 대한 자세한 정보는 *MQSeries Application Programming Reference*를 참조하십시오. 리플라이-투 큐 별명에 대한 자세한 정보는 *MQSeries* 상호통신을 참조하십시오.

데이터 변환

MQSeries 정의 형식(내장 형식이라고도 함)의 메시지 데이터는 큐 관리자에 의해 하나의 코드화 문자 세트에서 다른 코드화 문자 세트로 변환될 수 있습니다. 이 때, 두 문자 세트는 단일 언어나 일련의 유사한 언어들과 관련됩니다.

예를 들어, ID(CCSIDs)가 850 및 500인 코드화 문자 세트 사이의 변환이 지원됩니다. 두 문자 세트 모두 서유럽 언어에 적용되기 때문입니다.

EBCDIC 줄바꾸기(NL) 문자를 ASCII로 변환하는 것에 대한 정보는 185 페이지의 『AllQueueManagers 스탠자』를 참조하십시오.

지원되는 변환은 *MQSeries Application Programming Reference*의 코드 페이지 변환 테이블에 정의되어 있습니다.

큐 관리자가 메시지를 내장 형식으로 변환할 수 없을 경우

큐 관리자는 CCSID가 서로 다른 자국어 그룹을 나타낼 경우에 메시지를 내장 형식으로 자동 변환할 수 없습니다. 예를 들어, CCSID 850 및 CCSID 1025(Cyrillic 스크립트를 사용하는 언어에 대한 EBCDIC 코드화 문자 세트인) 사이의 변환은 하나의 코드화 문자 세트에 있는 많은 문자가 다른 코드화 문자 세트로 표시될 수 없으므로 지원되지 않습니다. 서로 다른 자국어로 작업 중인 큐 관리자 네트워크가 있고, 일부 코드화 문자 세트 사이의 데이터 변환이 지원되지 않으면, 디폴트 변환을 사용할 수 있습니다. 디폴트 데이터 변환은 83 페이지의 『디폴트 데이터 변환』에 설명되어 있습니다.

파일 **ccsid.tbl**

ccsid.tbl 파일은 다음을 지정합니다.

- 추가 코드 세트. 추가 코드 세트를 지정하려면, ccsid.tbl을 편집해야 합니다(이를 수행하는 방법에 대한 지침은 파일에 제공됩니다).
- 디폴트 데이터 변환

ccsid.tbl에 기록된 정보를 갱신할 수 있습니다. 예를 들어, 운영 체제의 차후 릴리스에서 추가 코드화 문자 세트를 지원하면 이를 수행할 수 있습니다.

Compaq OpenVMS용 MQSeries에서는 샘플 ccsid.tbl 파일이 다음과 같이 제공됩니다.

```
MQS_EXAMPLES:CCSID.TBL
```

그리고, 활성화된 ccsid.tbl 파일이 다음 디렉토리에 위치합니다.

```
MQS_ROOT:[MQM.CONV.TABLE]
```

디폴트 데이터 변환: 디폴트 데이터 변환을 구현하려면, ccsid.tbl을 편집하여 디폴트 EBCDIC CCSID와 디폴트 ASCII CCSID를 지정하고 디폴트 CCSID를 지정하십시오. 이를 수행하는 방법에 대한 지시사항은 파일에 포함되어 있습니다.

ccsid.tbl을 갱신하여 디폴트 데이터 변환을 구현할 경우, 변경이 적용되기 전에 큐 관리자를 재시작해야 합니다.

디폴트 데이터 변환 프로세스는 다음과 같습니다.

- 소스와 대상 CCSID 사이의 변환이 지원되지 않지만, 소스 및 대상 환경의 CCSID가 둘 다 EBCDIC이거나 둘 다 ASCII인 경우, 문자 데이터는 변환 없이 대상 응용프로그램에 전달됩니다.
- 하나의 CCSID가 ASCII 코드화 문자 세트를 나타내고, 다른 것이 EBCDIC 코드화 문자 세트를 나타낼 경우, MQSeries는 ccsid.tbl에 정의된 디폴트 데이터 변환 CCSID를 사용하여 데이터를 변환합니다.

주: 메시지에 대해 지정된 코드화 문자 세트와 디폴트 코드화 문자 세트에서 같은 값을 가지고 있는 문자로 변환되도록 문자들을 제한해야 합니다. MQSeries 오브젝트 이름에 대해 올바른 문자 세트만을 사용할 경우, 일반적으로 이 요구사항에 만족하게 됩니다. 예외는 일본어에서 사용되는 EBCDIC CCSID 290, 930, 1279, 5026에서 발생합니다. 이 언어에서는 소문자에 기타 EBCDIC CCSID에서 사용된 것과는 다른 코드가 사용됩니다.

사용자 정의 형식의 메시지 변환

사용자 정의 형식의 메시지들은 큐 관리자에 의해 하나의 코드화 문자 세트에서 다른 코드화 문자 세트로 변환될 수 없습니다. 사용자 정의 형식으로 된 데이터를 변환해야 할 경우, 그러한 각 형식에 대해 데이터 변환 종료표를 제공해야 합니다. 사용자 정의 형식으로 된 문자 데이터의 변환을 위해 디폴트 CCSID를 사용하는 것은 가능하지만 권장사항이 아닙니다. 사용자 정의 형식에서의 데이터 변환과 데이터 변환 종료표 작성에 대한 자세한 정보는 *MQSeries Application Programming Guide*를 참조하십시오.

큐 관리자 CCSID 변경

ALTER QMGR 명령의 CCSID 속성을 사용하여 큐 관리자의 CCSID를 변경할 경우에는 큐 관리자를 정지하였다가 재시작하는 것이 좋습니다.

그러면 명령 서버와 채널 프로그램을 포함하여 모든 실행 중인 응용프로그램이 정지한 다음 재시작합니다.

이는 큐 관리자 CCSID가 변경될 경우 실행 중인 응용프로그램이 계속 기존의 CCSID를 사용하려고 하기 때문에 반드시 필요한 작업입니다.

제7장 MQSeries 오브젝트 보호

이 장에서는 Compaq OpenVMS용 MQSeries에서의 보안 제어 기능과 이 제어를 구현할 수 있는 방법에 대해 설명합니다.

이 장에는 다음과 같은 절이 있습니다.

- 『MQSeries 자원을 보호해야 하는 이유』
- 『시작하기 전에』
- 87 페이지의 『오브젝트 권한 관리자 이해』
- 89 페이지의 『오브젝트 권한 관리자 명령 사용』
- 92 페이지의 『오브젝트 권한 관리자 지침』
- 96 페이지의 『권한 스펙 테이블에 대한 이해』
- 103 페이지의 『권한 파일에 대한 이해』

MQSeries 자원을 보호해야 하는 이유

MQSeries 큐 관리자는 잠재적인 가치가 있는 정보의 전송을 핸들링하므로, 권한 시스템을 보호할 필요가 있습니다. 이렇게 하면, 권한이 없는 사용자가 액세스하는 데서 발생할 수 있는 정보 유실이나 유출로부터 큐 관리자가 소유하고 관리하는 자원을 보호할 수 있습니다. 보안 시스템에서, 다음 중 어떤 것도 권한이 없는 사용자나 응용프로그램이 액세스하거나 변경할 수 없다는 것은 아주 중요합니다.

- 큐 관리자에 연결.
- 큐, 클러스트, 채널 및 프로세스와 같은 MQSeries 오브젝트 액세스.
- MQSC 명령과 PCF 명령과 같은, 큐 관리자 관리에 대한 명령.
- MQSeries 메시지에 액세스
- 메시지와 연관되는 컨텍스트 정보.

자원에 액세스할 수 있는 사용자에 대한 고유 방침을 정해야 합니다.

시작하기 전에

모든 큐 관리자 자원은 다음의 VMS 권한 ID를 사용하여 실행됩니다.

MQM

이 권한 ID는 MQSeries 설치 동안 작성되며 MQSeries 자원을 제어해야 하는 모든 사용자에게 이 자원 속성을 반드시 부여해야 합니다.

자원 ID가 MQM인 Compaq OpenVMS용 MQSeries에서의 사용자 ID

사용자 ID가 MQM OpenVMS 권한 ID를 보유한 경우, 모든 MQSeries 자원에 대한 권한을 모두 가지고 있는 것입니다. 사용자 ID는 **crtmqcvx**를 제외하고 모든 Compaq OpenVMS용 MQSeries 제어 명령을 사용할 수 있도록 반드시 OpenVMS MQM 권한 ID를 보유해야 합니다. 특히, 다음을 수행하려면 이 권한이 필요합니다.

- **runmqsc** 명령을 사용하여 MQSC 명령을 실행합니다.
- Compaq OpenVMS용 MQSeries에서 **setmqaut** 명령을 사용하여 권한을 관리합니다.

채널 명령을 리모트 시스템의 큐 관리자에 송신할 경우, 사용자 ID가 대상 시스템에서 OpenVMS 권한 ID MQM을 보유하고 있는지 확인해야 합니다. PCF 및 MQSC 채널 명령 목록에 대해서는 95 페이지의 『채널 명령 보안』을 참조하십시오.

또한 MQSeries를 설치하면 ID MQS_SERVER가 작성됩니다. 여기에는 VMS가 MQSeries에 대한 잠금 정보를 보존하는 자원 도메인 소유권이 부여됩니다. 디폴트로, 이 ID에 대한 액세스 권한은 다음과 같은 사용자에게 부여됩니다.

- 사용자 MQM과 같은 사용자 그룹에 있는 사용자.
- 시스템 사용자.
- SYSPRV, SYSLCK 또는 BYPASS 특권이 설정된 사용자.

다른 사용자가 MQ 자원에 액세스할 수 있도록 하려면, 다음 명령을 실행하여 MQS_SERVER ID가 적절한 WORLD 특권을 갖도록 해야 합니다.

```
SET SECURITY/CLASS=RESOURCE [MQS_SERVER] /PROTECTION=(W:RWL)
```

주: 사용자 ID가 다음을 발행할 경우에는 권한 ID MQM을 보유해야 하는 것이 중요하지 않습니다.

- 관리 프로그램에서 이스케이프 PCF를 포함하는 PCF 명령
- 응용프로그램에서 MQI 호출

자세한 정보

자세한 정보는 다음을 참조하십시오.

- Compaq OpenVMS용 MQSeries 명령 세트에 대해서는 23 페이지의 『제2장 MQSeries 관리 소개』를 참조하십시오.
- Compaq OpenVMS용 MQSeries 제어 명령에 대해서는 259 페이지의 『제17장 MQSeries 제어 명령』을 참조하십시오.
- PCF 명령 및 이스케이프 PCF에 대해서는 *MQSeries Programmable System Management*를 참조하십시오.

- MQI 호출에 대해서는 *MQSeries Application Programming Guide*와 *MQSeries Application Programming Reference*를 참조하십시오.

오브젝트 권한 관리자 이해

디폴트로, 큐 관리자 자원에 대한 액세스는 권한 서비스 설치 가능 구성요소를 통해 제어됩니다. 이 구성요소는 형식적으로 Compaq OpenVMS용 MQSeries용 OAM(Object Authority Manager)이라고 합니다. 이것은 Compaq OpenVMS용 MQSeries와 함께 제공되며, 자동으로 설치되고 달리 지정하지 않는 한, 사용자가 작성하는 각 큐 관리자에 대해 사용 가능하게 됩니다. 이 장에서, 용어 QAM은 이 제품과 함께 제공되는 Object Authority Manager를 말합니다.

OAM은 권한 서비스의 설치가능 구성요소입니다. 설치가능 서비스로 OAM을 제공하면 다음을 수행할 수 있는 유연성이 부여됩니다.

- 제공된 인터페이스를 사용하여 제공된 OAM을 고유한 권한 서비스 구성요소로 대체.
- 제공된 인터페이스를 다시 사용하여 사용자 고유의 권한 서비스 구성요소의 기능과 함께 OAM이 제공한 기능 증대.
- OAM 제거하거나 사용하지 않고 권한 서비스 없이 실행.

설치 가능 서비스에 대한 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.

OAM은 사용자의 권한을 관리하여 큐와 프로세스 정의 및 채널을 포함한 MQSeries 오브젝트를 조작합니다. 또한 특정 사용자 그룹의 오브젝트에 대한 액세스 권한을 부여하거나 취소할 수 있는 명령 인터페이스를 제공합니다. 자원에 대한 액세스를 허용하는 결정은 OAM에 의해 이루어지며, 큐 관리자는 그 결정을 따릅니다. OAM이 결정할 수 없는 경우, 큐 관리자는 그 자원에 대한 액세스를 금지합니다.

OAM 작동 방법

OAM은 근본적인 OpenVMS 운영 체제의 보안 기능을 사용하여 작동합니다. 특히, OAM은 OpenVMS 사용자와 그룹 ID 및 권한 ID를 사용합니다. 사용자는 필수 권한을 가지고 있는 경우에만 큐 관리자 오브젝트에 액세스할 수 있습니다.

권한 ID를 통한 액세스 관리

이 명령 인터페이스에서는 사용자 ID 대신 기본 원칙이라는 용어를 사용합니다. 그 이유는 사용자 ID에 부여된 권한은 다른 엔티티, 예를 들어 MQI 호출을 발행하는 응용 프로그램이나 PCF 명령을 발행하는 관리 프로그램에도 부여될 수 있기 때문입니다. 이러한 경우, 프로그램과 연관되는 기본 원칙은 프로그램이 시작될 때 사용한 사용자 ID가 아니어도 됩니다. 그러나 이 설명에서 기본 원칙과 사용자 ID는 항상 OpenVMS 사용자 ID입니다.

권한 ID 및 1차 권한 ID

MQSeries 자원에 대한 액세스 권한 관리는 OpenVMS 권한 ID, 즉 기본 원칙에서 보유하고 있는 ID를 기초로 합니다. 기본 원칙은 하나 이상의 OpenVMS 권한 ID를 보유할 수 있습니다. 그룹은 특정 권한 ID가 부여된 모든 기본 원칙 세트에 정의됩니다.

OAM은 개인 기본 원칙보다 권한 ID 레벨에서 권한을 유지합니다. ID명과의 기본 원칙 맵핑은 OAM 내에서 수행되며 조작성 권한 ID 레벨에서 수행됩니다. 그러나 개인 기본 원칙 권한을 표시할 수 있습니다.

기본 원칙이 하나 이상의 권한 ID를 보유할 경우

기본 원칙이 가지고 있는 권한은 보유하는 모든 권한 ID, 즉 해당되는 프로세스 권한의 합집합입니다. 기본 원칙이 자원에 대한 액세스를 요청할 때마다 OAM은 이 합집합을 계산하고 다시 이 합집합에 대해 권한을 점검합니다. 제어 명령 `setmqaut`를 사용하여 특정 기본 원칙이나 ID에 대한 권한을 설정할 수 있습니다.

주: `setmqaut` 명령을 사용한 변경사항은 오브젝트가 사용 중이 아니면 즉시 적용됩니다. 이러한 경우, 그 변경사항은 오브젝트가 다음 번에 열릴 때 강제로 적용됩니다. 그러나 기본 원칙의 권한 ID 목록에 대한 변경사항은 큐 관리자가 재설정될 때까지, 즉 정지된 후 재시작될 때까지 적용되지 않습니다.

기본 원칙과 연관된 권한은 OAM이 계산할 때 캐시됩니다. ID가 캐시된 후 변경되는 ID 권한은 큐 관리자가 재시작될 때까지 인식되지 않습니다. 큐 관리자가 실행 중일 때 권한을 변경하지 않도록 하십시오.

디폴트 권한 ID

OAM은 모든 사용자에게 이름으로 지정되는 디폴트를 인식합니다. 이 그룹은 'NOBODY' 의사 권한 ID에 의해 정의됩니다. 'NOBODY'는 MQSeries 명령을 사용하여 권한을 지정하기 위한 올바른 권한 ID처럼 사용할 수 있습니다. 디폴트로, 어떤 권한도 이 ID에 제공되지 않습니다. 특정 권한이 없는 사용자는 권한 ID를 통해 MQSeries 자원에 대한 액세스가 부여될 수 있습니다.

OAM으로 보호할 수 있는 자원

OAM을 통해 다음을 제어할 수 있습니다.

- MQI를 통한 MQSeries 오브젝트에 액세스. 응용프로그램이 오브젝트에 액세스하려고 할 때, OAM은 요청하는 사용자 ID가 요청한 조작성에 대한 권한을 가지고 있는지 보유한 ID를 통해 점검합니다.

특히, 이것은 큐와 그 큐의 메시지들이 권한이 없는 액세스로부터 보호될 수 있음을 의미합니다.

- MQSC 명령을 사용하기 위한 사용권한. 권한 ID MQM을 보유한 기본 원칙만 큐 관리자 관리 명령, 예를 들어, 큐를 작성하기 위한 명령을 실행할 수 있습니다.

- 제어 명령을 사용하기 위한 사용권한. 권한 ID MQM을 보유한 기본 원칙만 제어 명령, 예를 들어, 큐 관리자 작성, 명령 서버 시작 또는 **runmqsc** 사용을 실행할 수 있습니다.
- PCF 명령을 사용하기 위한 사용권한

여러 사용자들에게 동일 오브젝트에 대해 다른 종류의 액세스 권한이 부여될 수 있습니다. 예를 들어, 특정 큐에 대해 하나의 ID를 보유하고 있는 사용자들이 넣기(put) 및 가져오기(get) 조작을 모두 수행할 수 있고, 다른 ID를 보유하고 있는 사용자들은 그 큐를 찾아 볼 수만 있습니다(찾아보기 옵션이 있는 MQGET). 마찬가지로 ID를 갖고 있는 사용자들이 큐에 대해 가져오기(get) 및 넣기(put) 권한을 가질 수 있지만, 그 큐를 변경하거나 삭제할 수는 없습니다.

권한에 대해 권한 ID 사용

권한에 대해 개인 기본 원칙보다 ID를 사용하게 되면 필요한 관리 규모가 줄어듭니다. 일반적으로, 여러 기본 원칙에 대해 특정 종류의 액세스가 필요합니다. 예를 들어, 특정 응용프로그램을 실행하려는 일반 사용자들로 구성되는 ID를 정의할 수도 있습니다. 새로운 사용자에게는 해당되는 OpenVMS 사용자 ID에 적절한 ID를 부여하여 간단히 액세스 권한을 부여할 수 있습니다.

가능하면 ID 수를 적게 유지하십시오. 예를 들어, 기본 원칙을 특정 사용자를 위한 그룹과 관리자를 위한 그룹으로 나누어서 시작하면 좋습니다.

오브젝트 권한 관리자 사용 안함

디폴트로, OAM을 사용할 수 있습니다. 다음과 같이 큐 관리자를 작성하기 전에 논리 이름 MQSNOAUT를 설정하면 이를 사용하지 않을 수도 있습니다.

```
$ DEFINE/SYSTEM MQSNOAUT TRUE
```

그러나 이렇게 하면 일반적으로 나중에 OAM을 재시작할 수 없습니다. 더 나은 접근 방식은 OAM을 사용하고 모든 사용자와 응용프로그램이 적절한 사용자 ID를 통해 액세스하는지 확인하는 것입니다.

큐 관리자 구성 파일(qm.ini)에서 권한 서비스 스탠자를 제거하여 테스트 목적으로만 OAM을 사용하지 않을 수도 있습니다.

오브젝트 권한 관리자 명령 사용

OAM은 권한을 부여하고 취소하기 위한 명령 인터페이스를 제공합니다. 이러한 명령을 사용하려면 먼저 적절한 권한, 즉 사용자 ID는 OpenVMS 권한 ID MQM을 보유해야 합니다. 이 ID는 제품을 설치할 때 설정되어 있어야 합니다.

OAM 명령 사용

사용자 ID가 ID MQM을 보유하면, 큐 관리자에 대한 관리 권한을 갖습니다. 이제, 사용자 ID를 사용하여 MQI 요청이나 명령을 발행할 수 있는 권한이 부여된 것입니다.

OAM은 사용자 권한을 관리하기 위한 두 가지의 명령을 제공합니다. 이 명령들은 사용자의 OpenVMS DCL에서 호출할 수 있습니다. 이 명령은 다음과 같습니다.

- **setmqaut**(권한 설정 또는 재설정)
- **dspmqaut**(권한 표시)

권한 점검은 MQCONN, MQOPEN, MQPUT1 및 MQCLOSE 호출에서 발생합니다.

권한 점검은 이러한 호출의 첫번째 인스턴스에서만 수행되며 권한은 오브젝트를 재설정할 때까지, 즉 닫은 후 다시 열때까지 수정되지 않습니다.

그러므로 **setmqaut**를 사용한 오브젝트 권한 변경은 오브젝트를 재설정할 때까지 적용되지 않습니다.

OAM 명령 사용시 지정해야 할 사항

권한 명령은 지정된 큐 관리자에게 적용됩니다. 큐 관리자를 지정하지 않으면, 디폴트 큐 관리자가 사용됩니다. 이 명령에서 고유한 오브젝트, 즉 오브젝트 이름과 그 유형을 지정해야 합니다. 권한이 적용될 ID명이나 기본 원칙도 지정해야 합니다.

권한 목록

setmqaut 명령에서 권한 목록을 지정합니다. 이것은 권한이 부여되는지, 아니면 취소되는지 여부와, 권한이 적용되는 자원을 지정하는 간단한 방법입니다. 목록에서 각 권한은 앞에 + 또는 - 부호가 붙은 소문자 키워드로 지정됩니다. 지정된 권한을 추가하려면 + 부호를, 권한을 제거하려면 - 부호를 사용하십시오. 단일 명령에 몇 개의 권한도 지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
+browse -get +put
```

setmqaut 명령 사용

필수 권한을 가지고 있으면, **setmqaut** 명령을 사용하여 특정 오브젝트에 액세스하기 위한 기본 원칙 또는 권한 ID의 권한을 부여하거나 취소할 수 있습니다. 다음 예는 **setmqaut** 명령을 사용하는 방법을 보여줍니다.

```
setmqaut -m "saturn.queue.manager" -t queue -n RED.LOCAL.QUEUE -g GROUPA +browse -get +put
```

이 예의 내용은 다음과 같습니다.

용어...	지정하는 내용...
saturn.queue.manager	큐 관리자 이름.
queue	오브젝트 유형.
RED.LOCAL.QUEUE	오브젝트 이름.
GROUPA	권한이 제공되는 그룹의 ID.
+browse -get +put	지정된 큐에 대한 권한 목록. '+'나 '-' 부호와 키워드 사이에 공백이 있어서는 안됩니다.

권한 목록은 제공되는 권한의 위치를 지정합니다.

용어...	수행 사항...
+browse	큐에서 메시지를 찾을 수 있는(찾아보기 옵션이 있는 MQGET) 권한 추가.
-get	큐에서 메시지를 가져올 수 있는(MQGET) 권한 제거.
+put	큐에 메시지를 넣을 수 있는(MQPUT) 권한 추가.

이는 OpenVMS ID GROUPA를 소유하는 사용자 ID로 시작된 응용프로그램이 이러한 권한들을 갖는 것을 의미합니다.

하나 이상의 기본 원칙과 하나 이상의 ID를 동시에 지정할 수 있습니다. 예를 들어, 다음 명령은 큐 MyQueue에서 기본 원칙 FVUSER와 ID GROUPA 및 GROUPB에 대해 넣기 (put) 권한을 취소합니다.

```
setmqaut -m "saturn.queue.manager" -t queue -n "MyQueue" -p FVUSER -g GROUPA -g GROUPB -put
```

주: 이 명령은 또한 FVUSER가 소유하는 모든 권한 ID, 즉 FVUSER가 속하는 모든 그룹에 대해 넣기(put) 권한을 취소합니다.

명령 및 구문에 대한 형식적 정의에 대해서는 321 페이지의 『setmqaut(권한 설정/재설정)』를 참조하십시오.

권한 명령 및 설치가능 서비스

setmqaut 명령은 갱신이 적용되는 설치 가능 서비스 구성요소의 이름을 지정하는 추가 매개변수를 가지고 있습니다. 동시에 여러 개의 설치가능 구성요소를 실행할 경우에는 이 매개변수를 지정해야 합니다. 디폴트로, 이 매개변수는 대소문자가 구별되지 않습니다. 매개변수를 생략하면 해당 유형의 첫번째 설치 가능 서비스가 있는 경우에만 갱신됩니다. 디폴트로, 이것은 OAM로 제공되었습니다.

OAM 명령 사용

액세스 권한

setmqaut 명령과 연관되는 권한 목록에 정의된 권한들은 다음과 같은 범주로 나눌 수 있습니다.

- MQI 호출에 관련되는 권한
- 관리 명령에 관련되는 권한
- 컨텍스트 권한
- 일반 권한, 즉 MQI 호출이나 명령, 또는 모두에 대한 권한

각 권한은 **setmqaut** 및 **dspmqaut** 명령에서 사용되는 키워드로 지정됩니다. 이에 대해서는 321 페이지의 『setmqaut(권한 설정/재설정)』에 설명되어 있습니다.

권한 명령 표시

dspmqaut 명령을 사용하면 특정 오브젝트에 대해 특정의 기본 원칙이나 ID가 가지고 있는 권한을 볼 수 있습니다. 플래그의 의미는 **setmqaut** 명령에서와 같습니다. 권한은 한번에 하나의 ID나 기본 원칙에 대해서만 표시할 수 있습니다. 이 명령의 형식적 스펙에 대해서는 275 페이지의 『dspmqaut(권한 표시)』를 참조하십시오.

예를 들어, 다음 명령은 그룹 GpAdmin이 큐 관리자 QueueMan1에서 이름이 Annuities 인 프로세스 정의에 대해 가지고 있는 권한을 표시합니다.

```
dspmqaut -m "QueueMan1" -t process -n "Annuities" -g "GpAdmin"
```

이 명령의 결과로 표시되는 키워드들은 활성화된 권한들을 식별합니다.

오브젝트 권한 관리자 지침

일부 조작용은 특히 민감하므로 특권이 있는 사용자들로 제한해야 합니다. 예를 들면, 다음과 같습니다.

- 큐 관리자 시작 및 정지
- 트랜스미션 큐나 명령 큐 SYSTEM.ADMIN.COMMAND.QUEUE와 같은 특정의 특수 큐에 대한 액세스
- 전체 MQI 컨텍스트 옵션을 사용하는 프로그램
- 일반적으로 응용프로그램 큐 작성 및 복사

사용자 ID

제품 설치시 작성한 특수 사용자 ID MQM은 제품 전용으로 사용하기 위한 것이므로, 특권이 없는 사용자는 절대로 사용할 수 없습니다.

MQ 프로세스와 연관된 권한 점검에 사용되는 사용자 ID는 OpenVMS 사용자 ID입니다.

큐 관리자 디렉토리

큐와 다른 큐 관리자 데이터를 포함하는 디렉토리는 제품과는 별개입니다. 이 디렉토리의 오브젝트들은 해당 OAM 권한에 관련되는 OpenVMS 사용자 권한을 갖습니다. 그러나 MQI 자원에 대한 권한을 부여하거나 취소할 경우에는 다음과 같은 이유로 인해 표준 OpenVMS 명령을 사용해서는 안됩니다.

- MQSeries 오브젝트가 항상 해당 시스템 오브젝트 이름과 같은 것은 아닙니다. 이에 대한 자세한 정보는 25 페이지의 『MQSeries 파일 이름 이해』를 참조하십시오.
- 모든 오브젝트는 자원 ID MQM에서 소유합니다.

큐

다이나믹 큐에 대한 권한은 도출된 모델 큐의 권한을 기초로 하지만, 반드시 같지는 않습니다. 자세한 정보는 참고 100 페이지의 1을 참조하십시오.

알리아스 큐와 리모트 큐의 경우, 권한은 알리아스 또는 리모트 큐가 분석하는 큐가 아니라, 오브젝트 자체의 권한입니다. 그러므로 사용자 ID가 액세스 권한을 가지고 있지 않은 로컬 큐에 대해 분석하는 알리아스 큐에 액세스할 수 있는 권한을 사용자 ID에 부여할 수 있습니다.

큐를 작성하려면 특권이 있는 사용자들로 권한을 제한해야 합니다. 그렇게 하지 않으면, 일부 사용자들이 별명을 작성하여 정상적인 액세스 제어를 생략할 수 있습니다.

대체 사용자 권한

대체 사용자 권한은 하나의 사용자 ID가 MQSeries 오브젝트에 액세스할 때 다른 사용자 ID의 권한을 사용할 수 있는지 제어합니다. 이것은 서버가 프로그램에서 요청으로 수신하고 그 프로그램이 요청에 대해 필요한 권한을 가지고 있는지 확인할 때 중요합니다. 서버는 필수 권한을 가지고 있을 수도 있지만, 프로그램에서 요청한 조치에 대한 권한을 가지고 있는지 알아야 합니다.

예를 들면 다음과 같습니다.

- 사용자 ID PAYSERV 사에서 실행되는 서버 프로그램은 사용자 ID USER1가 큐에 넣은 요청 메시지를 큐에서 검색합니다.
- 서버 프로그램이 요청 메시지를 가져오면, 그 요청을 처리하여 요청 메시지에 지정된 리플라이-투 큐에 다시 응답을 넣습니다.
- 고유한 사용자 ID(PAYSERV)를 사용하여 리플라이-투 큐 열기 권한을 부여하는 대신, 서버는 다른 사용자 ID(이 경우, USER1)를 지정할 수 있습니다. 이 예에서는 대체 사용자 권한을 사용하여, PAYSERV가 리플라이-투 큐를 열 때 대체 사용자 ID로 USER1의 지정 여부를 제어할 수 있습니다.

대체 사용자 ID는 오브젝트 설명자의 *AlternateUserId* 필드에 지정됩니다.

주: 모든 MQSeries 오브젝트에서 대체 사용자 ID를 사용할 수 있습니다. 대체 사용자 ID를 사용해도 다른 자원 관리자에서 사용되는 사용자 ID에는 영향이 미치지 않습니다.

컨텍스트 권한

컨텍스트는 특정 메시지에 적용되는 정보로, 메시지의 일부인 MQMD 메시지 설명자에 포함됩니다. 컨텍스트 정보는 두 개의 섹션으로 제공됩니다.

ID 섹션

이 부분은 메시지를 제공하는 사람을 지정합니다. 다음 필드들로 구성됩니다.

- *UserIdentifier*
- *AccountingToken*
- *ApplIdentityData*

원본 섹션

이 섹션은 메시지를 제공한 장소와 큐에 넣은 시기를 지정합니다. 다음 필드들로 구성됩니다.

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

응용프로그램은 MQOPEN 또는 MQPUT 호출이 될 때 컨텍스트 데이터를 지정할 수 있습니다. 이 데이터는 응용프로그램에 의해 생성되고 다른 메시지에서 전달되거나 디폴트로 큐 관리자에 의해 생성될 수 있습니다. 예를 들어, 서버 프로그램에서 컨텍스트 데이터를 사용하여 요청자의 ID를 점검할 수 있습니다. 이 때, 메시지가 권한이 부여된 사용자 ID로 실행되는 응용프로그램에서 제공되는지 테스트합니다.

서버 프로그램은 *UserIdentifier*를 사용하여 대체 사용자의 사용자 ID를 판별할 수 있습니다.

컨텍스트 권한을 사용하여 MQOPEN 또는 MQPUT1 호출에서 컨텍스트 옵션 중 하나의 지정 여부를 제어할 수 있습니다. 컨텍스트 옵션에 대한 정보는 *MQSeries Application Programming Guide*를 참조하십시오. 컨텍스트에 관련되는 메시지 설명자 필드의 설명에 대해서는 *MQSeries Application Programming Reference*를 참조하십시오.

리모트 보안 고려사항

리모트 보안의 경우, 다음을 고려해야 합니다.

넣기 권한

큐 관리자 사이의 보안을 위해, 채널이 다른 큐 관리자로부터 송신된 메시지를 수신할 때 사용하는 넣기 권한을 지정할 수 있습니다.

다음과 같이 채널 속성 PUTAUT를 지정하십시오.

DEF 디폴트 사용자 ID. 이것은 메시지 채널 에이전트가 실행되는 사용자 ID입니다.

CTX 메시지 컨텍스트의 사용자 ID.

트랜스미션 큐

큐 관리자는 자동으로 트랜스미션 큐에 리모트 메시지를 넣습니다. 이를 위해 필요한 특수 권한은 없습니다. 그러나 메시지를 직접 트랜스미션 큐에 넣으려면 특수 권한이 필요합니다. 97 페이지의 표2를 참조하십시오.

채널 엑시트

추가된 보안에 대해 채널 엑시트를 사용할 수 있습니다.

자세한 정보는 *MQSeries 상호통신*을 참조하십시오.

채널 명령 보안

채널은 MQAI, MQSC 명령 및 제어 명령을 통해 PCF 명령으로 발행할 수 있습니다.

PCF 명령

리모트 OpenVMS 시스템에서 PCF 메시지를 SYSTEM.ADMIN.COMMAND.QUEUE에 송신하여 PCF 채널 명령을 발행할 수 있습니다. PCF 메시지의 메시지 설명자에 지정된 사용자 ID는 대상 시스템에 대한 권한 ID MQM을 보유해야 합니다. 다음 명령들이 여기에 속합니다.

- *ChangeChannel*
- *CopyChannel*
- *CreateChannel*
- *DeleteChannel*
- *PingChannel*
- *ResetChannel*
- *StartChannel*
- *StartChannelInitiator*
- *StartChannelListener*
- *StopChannel*
- *ResolveChannel*

PCF 보안 요구사항에 대해서는 *MQSeries Programmable System Management*를 참조하십시오.

MQSC 채널 명령

PCF 이스케이프 메시지로 직접 명령을 송신하거나 간접 모드에서 **runmqsc**를 사용한 명령을 발행하여 리모트 OpenVMS 시스템에 MQSC 채널 명령을 발행할 수 있습니다. 연관된 PCF 메시지의 메시지 설명자에 지정된 사용자 ID는 대상 시스템에 대한 권한 ID MQM을 보유해야 합니다 (PCF 명령은 간접 모드에서 **runmqsc**에서 발행된 MQSC 명령에 포함됩니다). 다음 명령들이 여기에 속합니다.

- ALTER CHANNEL
- DEFINE CHANNEL
- DELETE CHANNEL
- PING CHANNEL
- RESET CHANNEL
- START CHANNEL
- START CHINIT
- START LISTENER
- STOP CHANNEL
- RESOLVE CHANNEL

runmqsc 명령에서 발행된 MQSC 명령의 경우, PCF 메시지의 사용자 ID는 보통 현재 사용자의 ID입니다.

채널에 대한 제어 명령

채널의 제어 명령에 대해, 이 명령을 발행하는 사용자 ID는 권한 ID MQM을 보유하고 있어야 합니다. 다음 명령들이 여기에 속합니다.

- **runmqchi**(채널 시작기 실행)
- **runmqchl**(채널 실행)

권한 스펙 테이블에 대한 이해

97 페이지에서 시작하는 권한 스펙 테이블은 권한 작동 방법과 적용하는 제한사항을 명확하게 정의합니다. 테이블은 다음 상황에 적용됩니다.

- MQI 호출을 발행하는 응용프로그램
- 이스케이프 PCF로 MQSC 명령을 발행하는 관리 프로그램
- PCF 명령을 발행하는 관리 프로그램

이 절에서, 정보는 다음을 지정하는 일련의 테이블로 표시됩니다.

수행되는 조치

MQI 옵션, MQSC 명령 또는 PCF 명령

액세스 제어 오브젝트

큐, 프로세스 또는 큐 관리자

권한 필요 여부

‘MQZAO_’ 상수로 표시됩니다.

테이블에서, 앞에 MQZAO_가 붙은 상수는 특정 엔티티의 **setmqaut** 명령에 대한 권한 목록 내의 키워드에 해당됩니다. 예를 들어, MQZAO_BROWSE는 키워드 +browse에 해당되고 마찬가지로, 키워드 MQZAO_SET_ALL_CONTEXT는 키워드 +setall 등에 해당됩니다. 이러한 상수는 제품과 함께 제공되는 헤더 파일 cmqzc.h에 정의됩니다. 자세한 정보는 104 페이지의 『권한 파일의 내용』을 참조하십시오.

MQI 권한

응용프로그램은 실행될 때(또는 권한이 있다고 여겨질 때) 사용자 ID에 관련 권한이 부여된 경우, 특정 MQI 호출과 옵션만 발행할 수 있습니다.

네 가지의 MQI 호출인 MQCONN, MQOPEN, MQPUT1 및 MQCLOSE에서 권한 점검을 해야 합니다.

MQOPEN 및 MQPUT1의 경우, 권한 점검은 분석된 후의 이름(또는 이름들)이 아니라, 열릴 때의 오브젝트 이름에 대해 수행됩니다. 예를 들어, 응용프로그램에 별명이 해석되는 기본 큐를 열수 있는 권한이 없어도 알리아스 큐를 열 수 있는 권한이 부여될 수도 있습니다. 큐 관리자 별명 정의를 직접 열지 않고 해당되는 이름이 오브젝트 설명자의 *ObjectName* 필드에 표시되는 경우에는 큐 관리자 별명이 아닌, 이름 분석 프로세스 동안 발견되는 첫번째 정의를 점검하는 것이 규칙입니다. 열리는 특정 오브젝트마다 항상 권한이 필요합니다. 어떤 경우에는 큐 관리자 오브젝트의 권한을 통해 확보되는 추가의 큐 독립적 권한이 필요합니다.

표2에는 각 호출에 필요한 권한이 요약되어 있습니다.

표 2. MQI 호출에 필요한 보안 권한

권한 필요 대상:	큐 오브젝트(1)	프로세스 오브젝트	큐 관리자 오브젝트	이름 목록
MQCONN 옵션	N/A	N/A	MQZAO_CONNECT	N/A
MQOPEN 옵션				
MQOO_INQUIRE	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)	MQZAO_INQUIRE (2)
MQOO_BROWSE	MQZAO_BROWSE	N/A	점검 안함	N/A
MQOO_INPUT_*	MQZAO_INPUT	N/A	점검 안함	N/A
MQOO_SAVE_ALL_CONTEXT (3)	MQZAO_INPUT	N/A	점검 안함	N/A
MQOO_OUTPUT(정상 큐) (4)	MQZAO_OUTPUT	N/A	점검 안함	N/A
MQOO_PASS_IDENTITY_CONTEXT (5)	MQZAO_PASS_IDENTITY_CONTEXT	N/A	점검 안함	N/A

권한 스펙 테이블

표 2. MQI 호출에 필요한 보안 권한 (계속)

권한 필요 대상:	큐 오브젝트(1)	프로세스 오브젝트	큐 관리자 오브젝트	이름 목록
MQOO_PASS_ ALL_CONTEXT (5, 6)	MQZAO_PASS_ ALL_CONTEXT	N/A	점검 안함	N/A
MQOO_SET_ IDENTITY_CONTEXT (5, 6)	MQZAO_SET_ IDENTITY_CONTEXT	N/A	MQZAO_SET_ IDENTITY_CONTEXT (7)	N/A
MQOO_SET_ ALL_CONTEXT (5, 8)	MQZAO_SET_ ALL_CONTEXT	N/A	MQZAO_SET_ ALL_CONTEXT (7)	N/A
MQOO_OUTPUT(트랜스미션 큐) (9)	MQZAO_SET_ ALL_CONTEXT	N/A	MQZAO_SET_ ALL_CONTEXT (7)	N/A
MQOO_SET	MQZAO_SET	N/A	점검 안함	N/A
MQOO_ALTERNATE_ USER_AUTHORITY	(10)	(10)	MQZAO_ ALTERNATE_ USER_AUTHORITY (10, 11)	(10)
MQPUT1 옵션				
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT (12)	N/A	점검 안함	N/A
MQPMO_PASS_ ALL_CONTEXT	MQZAO_PASS_ ALL_CONTEXT (12)	N/A	점검 안함	N/A
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT (12)	N/A	MQZAO_SET_ IDENTITY_CONTEXT (7)	N/A
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT (12)	N/A	MQZAO_SET_ ALL_CONTEXT (7)	N/A
(트랜스미션 큐) (9)	MQZAO_SET_ ALL_CONTEXT	N/A	MQZAO_SET_ ALL_CONTEXT (7)	N/A
MQPMO_ALTERNATE_ USER_AUTHORITY	(13)	N/A	MQZAO_ ALTERNATE_ USER_AUTHORITY (11)	N/A
MQCLOSE 옵션				
MQCO_DELETE	MQZAO_DELETE (14)	N/A	N/A	N/A
MQCO_DELETE_PURGE	MQZAO_DELETE (14)	N/A	N/A	N/A

특정 참고 사항:

1. 모델 큐가 열릴 경우:

- 지정된 열기 옵션에 대해 필요한 다른 권한 이외에도 모델 큐와 모델에 대해서도 MQZAO_DISPLAY 권한이 필요합니다.

- MQZAO_CREATE 권한은 다이내믹 큐 작성에는 필요하지 않습니다.
 - 모델 큐를 여는 데 사용되는 사용자 ID는 작성된 다이내믹 큐에 대해 자동으로 큐 고유의 모든 권한(MQZAO_ALL과 같음)을 가집니다.
2. 열리는 오브젝트 유형에 따라 큐, 프로세스, 이름 목록 또는 큐 관리자 오브젝트가 점검됩니다.
 3. MQOO_INPUT_*도 지정해야 합니다. 이것은 로컬, 모델 또는 알리아스 큐에 대해 유효합니다.
 4. 이 점검은 참고 9에 지정된 경우를 제외하고, 모든 출력 클래스에 대해 수행됩니다.
 5. MQOO_OUTPUT도 지정해야 합니다.
 6. MQOO_PASS_IDENTITY_CONTEXT도 이 옵션으로 알 수 있습니다.
 7. 이 권한은 큐 관리자 오브젝트와 특정 큐 모두에게 필요합니다.
 8. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT, MQOO_SET_IDENTITY_CONTEXT도 이 옵션으로 알 수 있습니다.
 9. 이 점검은 MQUS_TRANSMISSION의 Usage 큐 속성을 가지고 있고, 출력할 때 직접 열리는 로컬 또는 모델 큐에 대해 수행됩니다. 이는 리모트 큐가 열릴 경우(리모트 큐 관리자와 리모트 큐의 이름을 지정하거나 리모트 큐의 로컬 정의 이름을 지정하는 경우)에는 적용되지 않습니다.
 10. MQOO_INQUIRE(임의의 오브젝트 유형에 대해)나, (큐에 대해) MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT 또는 MQOO_SET도 지정해야 합니다. 수행되는 점검은 지정된 다른 옵션의 경우처럼, 특정 이름의 오브젝트 권한에 대해 제공된 대체 사용자 ID를 사용하고, MQZAO_ALTERNATE_USER_IDENTIFIER 점검에 대해 현재 응용프로그램 권한을 사용합니다.
 11. 이 권한은 지정되는 모든 AlternateUserId를 허용합니다.
 12. 큐가 MQUS_TRANSMISSION의 Usage 큐 속성을 가지고 있지 않아도 MQZAO_OUTPUT 점검이 수행됩니다.
 13. 수행되는 점검은 지정된 다른 옵션의 경우처럼, 특정 이름의 큐 권한에 대해 제공된 대체 사용자 ID를 사용하고,

MQZAO_ALTERNATE_USER_IDENTIFIER 점검에 대해 현재 응용프로그램 권한을 사용합니다.

14. 점검은 다음 두 가지 모두 충족되는 경우에만 수행됩니다.

- 퍼머넌트 다이내믹 큐가 닫히고 삭제됩니다.
- 사용 중인 오브젝트 핸들을 리턴한 MQOPEN이 큐를 작성한 것이 아닙니다.

그렇지 않은 경우, 점검은 수행되지 않습니다.

일반 참고 사항:

1. 특수 권한 MQZAO_ALL_MQI에는 오브젝트 유형에 관련되는 다음의 모든 것이 포함됩니다.
 - MQZAO_CONNECT
 - MQZAO_INQUIRE
 - MQZAO_SET
 - MQZAO_BROWSE
 - MQZAO_INPUT
 - MQZAO_OUTPUT
 - MQZAO_PASS_IDENTITY_CONTEXT
 - MQZAO_PASS_ALL_CONTEXT
 - MQZAO_SET_IDENTITY_CONTEXT
 - MQZAO_SET_ALL_CONTEXT
 - MQZAO_ALTERNATE_USER_AUTHORITY
2. MQZAO_DELETE(참고 14 참조)와 MQZAO_DISPLAY는 관리 권한으로 클래스화됩니다. 그러므로 이들은 MQZAO_ALL_MQI에 포함되지 않습니다.
3. '점검 안함'은 권한 점검이 수행되지 않음을 의미합니다.
4. 'N/A'는 권한 점검이 해당 조작에 관련되지 않음을 의미합니다. 예를 들어, 프로세스 오브젝트에 대해 MQPUT 호출을 발행할 수 없습니다.

관리 권한

이 권한은 사용자가 관리 명령을 발행할 수 있게 하며, 이스케이프 PCF 메시지나 PCF 명령 자체가 MQSC 명령이 될 수 있습니다. 이러한 방법을 사용하면 프로그램은 해당 사용자 대신 실행시 큐 관리자에게 메시지로 관리 명령을 송신할 수 있습니다.

이스케이프 PCF의 MQSC 명령에 대한 권한

101 페이지의 표3에는 이스케이프 PCF에 포함된 각 MQSC 명령에 필요한 권한이 요약되어 있습니다.

표 3. 필요한 MQSC 명령 및 보안 권한

(2)권한 필요 대상:	큐 오브젝트	프로세스 오브젝트	큐 관리자 오브젝트	이름 목록
MQSC 명령				
ALTER 오브젝트	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
CLEAR QLOCAL	MQZAO_CLEAR	N/A	N/A	N/A
DEFINE 오브젝트 NOREPLACE (3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	N/A	MQZAO_CREATE (4)
DEFINE 오브젝트 REPLACE (3, 5)	MQZAO_CHANGE	MQZAO_CHANGE	N/A	MQZAO_CHANGE
DELETE 오브젝트	MQZAO_DELETE	MQZAO_DELETE	N/A	MQZAO_DELETE
DISPLAY 오브젝트	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY

특정 참고 사항:

1. 명령을 제출하는 프로그램(예: **runmqsc**)이 실행 중이면 사용자 ID 는 큐 관리자에 대해 MQZAO_CONNECT 권한도 가지고 있어야 합니다.
2. 오브젝트 유형에 따라 큐, 프로세스, 이름 목록 또는 큐 관리자 오브젝트가 점검됩니다.
3. DEFINE 명령의 경우, LIKE 오브젝트가 지정되어 있으면 이에 대해 MQZAO_DISPLAY 권한도 필요한데, LIKE가 생략되어 있으면 해당되는 SYSTEM.DEFAULT.xxx 오브젝트에 대해 이 권한이 필요합니다.
4. MQZAO_CREATE 권한은 특정 오브젝트나 오브젝트 유형에 대해 고유하지 않습니다. SETMQAUT 명령에서 QMGR 오브젝트 유형을 지정하여, 지정된 큐 관리자에 대해 모든 오브젝트에 대한 작성 권한이 부여됩니다.
5. 이것은 바꾸려는 오브젝트가 실제로 이미 존재하고 있는 경우에 적용됩니다. 그렇지 않은 경우, DEFINE 오브젝트 NOREPLACE처럼 점검합니다.

일반 참고 사항:

1. PCF 명령을 수행하려면, 큐 관리자에서 DISPLAY 권한을 가지고 있어야 합니다.
2. 이스케이프 PCF를 실행할 수 있는 권한은 이스케이프 PCF 메시지의 텍스트 내에 있는 MQSC 명령에 따라 다릅니다.
3. 'N/A'는 권한 점검이 해당 조작에 관련되지 않음을 의미합니다. 예를 들어, 큐 관리자 오브젝트에 대해 CLEAR QLOCAL을 발행할 수 없습니다.

권한 스펙 테이블

PCF 명령에 대한 권한

표4에는 각 PCF 명령에 필요한 권한이 요약되어 있습니다.

표 4. PCF 명령과 필요한 보안 권한

(2) 권한 필요 대상:	큐 오브젝트	프로세스 오브젝트	큐 관리자 오브젝트	이름 목록
PCF 명령				
오브젝트 변경	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE	MQZAO_CHANGE
큐 지우기	MQZAO_CLEAR	N/A	N/A	N/A
오브젝트 복사(바꾸지 않 고) (3)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	N/A	MQZAO_CREATE (4)
오브젝트 복사(바꾸면서) (3, 6)	MQZAO_CHANGE	MQZAO_CHANGE	N/A	MQZAO_CHANGE
오브젝트 작성(바꾸지 않 고) (5)	MQZAO_CREATE (4)	MQZAO_CREATE (4)	N/A	MQZAO_CREATE (4)
오브젝트 작성(바꾸면서) (5, 6)	MQZAO_CHANGE	MQZAO_CHANGE	N/A	MQZAO_CHANGE
오브젝트 삭제	MQZAO_DELETE	MQZAO_DELETE	N/A	MQZAO_DELETE
오브젝트 조회	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY	MQZAO_DISPLAY
오브젝트 이름 조회	점검 안함	점검 안함	점검 안함	점검 안함
큐 통계 재설정	MQZAO_DISPLAY 및 MQZAO_CHANGE	N/A	N/A	N/A

특정 참고 사항:

1. 프로그램이 명령을 제출할 때의 사용자 ID는 로컬 큐 관리자에 연결할 수 있는 권한과 출력을 위해 명령 관리 큐를 열수 있는 권한도 가지고 있어야 합니다.
2. 오브젝트 유형에 따라 큐, 프로세스, 이름 목록 또는 큐 관리자 오브젝트가 점검됩니다.
3. Copy 명령의 경우, From 오브젝트에 대해서도 MQZAO_DISPLAY 권한이 필요합니다.
4. MQZAO_CREATE 권한은 특정 오브젝트나 오브젝트 유형에 대해 고유하지 않습니다. SETMQAUT 명령에서 QMGR 오브젝트 유형을 지정하여, 지정된 큐 관리자에 대해 모든 오브젝트에 대한 작성 권한이 부여됩니다.
5. Create 명령의 경우, 해당되는 SYSTEM.DEFAULT.* 오브젝트에 대해서도 MQZAO_DISPLAY 권한이 필요합니다.
6. 이것은 바꾸려는 오브젝트가 이미 존재하고 있는 경우에 적용됩니다. 그렇지 않은 경우, Copy나 Create처럼 오브젝트를 바꾸지 않고 점검합니다.

일반 참고 사항:

1. PCF 명령을 수행하려면, 큐 관리자에서 DISPLAY 권한을 가지고 있어야 합니다.
2. 특수 권한 MQZAO_ALL_ADMIN에는 오브젝트 유형에 관련되는 다음의 모든 것이 포함됩니다.
 - MQZAO_CHANGE
 - MQZAO_CLEAR
 - MQZAO_DELETE
 - MQZAO_DISPLAY

MQZAO_CREATE는 특정 오브젝트나 오브젝트 유형에 고유하지 않으므로 포함되지 않습니다.
3. '점검 안함'은 권한 점검이 수행되지 않음을 의미합니다.
4. 'N/A'는 권한 점검이 해당 조작에 관련되지 않음을 의미합니다. 예를 들어, 프로세스 오브젝트에서 **Clear Queue** 명령을 사용할 수 없습니다.

권한 파일에 대한 이해

주: 이 절에 있는 정보는 문제점 판별을 위해 제공됩니다. 정상적인 상황에서는 권한 명령을 사용하여 권한 정보를 보고 변경하십시오.

Compaq OpenVMS용 MQSeries는 특정 파일 구조를 사용하여 보안을 구현합니다. 모든 권한 파일이 자체적으로 보안이 되는지 확인하는 것을 제외하면 이 파일들에 대해 어떤 것도 수행하지 않습니다.

보안은 권한 파일에 의해 구현됩니다. 이러한 관점에서 볼 때, 권한에는 세 가지 유형이 있습니다.

- 예를 들어, 큐에 메시지를 넣기 위한 권한과 같이 단일 오브젝트에 적용되는 권한
- 예를 들어, 큐를 작성할 수 있는 권한과 같이 오브젝트 클래스에 적용되는 권한
- 예를 들어, 다른 사용자 대신 조작을 수행할 수 있는 권한과 같이 모든 오브젝트 클래스에 적용되는 권한

권한 파일 경로

권한 파일에 대한 경로는 유형에 따라 다릅니다. 예를 들어, 오브젝트에 대한 권한을 지정할 경우, 큐 관리자는 해당 권한 파일을 작성합니다. 이 파일들을 서브디렉토리에 넣고, 경로는 큐 관리자 이름, 권한 유형, 그리고 적절할 경우 오브젝트 이름으로 정의됩니다.

모든 권한이 직접 오브젝트 인스턴스에 적용되는 것은 아닙니다. 예를 들어, 오브젝트를 작성할 수 있는 권한은 개인 인스턴스보다 오브젝트 클래스에 적용됩니다. 또한 일

권한 파일

부 권한은 전체 큐 관리자에게 적용됩니다. 예를 들어, 대체 사용자 권한은 그 사용자가 다른 사용자와 연관되는 권한을 가정할 수 있음을 의미합니다.

권한 디렉토리

디폴트로, saturn 큐 관리자에 대한 권한 디렉토리는 다음과 같습니다.

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH.QUEUES]

큐에 대한 권한 파일.

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH.PROCDEF]

프로세스 정의에 대한 권한 파일.

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH.QMANAGER]

큐 관리자에 대한 권한 파일.

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH]\$ACCLASS

모든 클래스에 적용되는 권한.

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH.NAMELIST]

모든 이름 목록에 적용되는 권한.

오브젝트 디렉토리에서, \$CLASS 파일은 전체 클래스에 관련되는 권한을 보유하고 있습니다.

주: \$CLASS(특정 클래스에 대한 권한을 지정하는 권한 파일)와 \$ACCLASS(모든 클래스에 대한 권한을 지정하는 권한 파일) 사이에는 차이가 있습니다.

오브젝트 권한 파일의 경로는 오브젝트 자체의 경로를 기초로 합니다. AUTH가 오브젝트 유형 디렉토리 앞에 삽입됩니다. **dspmqls** 명령을 사용하여 지정된 오브젝트에 대한 경로를 표시할 수 있습니다.

예를 들어, SYSTEM.DEFAULT.LOCAL.QUEUE의 이름과 경로가 다음과 같을 경우

MQS_ROOT: [MQM.QMGRS.SATURN.QUEUES.SYSTEM\$DEFAULT\$LOCAL\$QUEUE]

해당되는 권한 파일의 이름과 경로는 다음과 같습니다.

MQS_ROOT: [MQM.QMGRS.SATURN.AUTH.QUEUES.SYSTEM\$DEFAULT\$LOCAL\$QUEUE]

주: 이 경우, 큐와 연관되는 파일의 실제 이름은 큐 자체의 이름과 같지 않습니다. 자세한 내용은 25 페이지의 『MQSeries 파일 이름 이해』를 참조하십시오.

권한 파일의 내용

특정 ID의 권한은 권한 파일에서 스탠자 세트에 정의됩니다. 자세한 정보는 103 페이지의 『권한 파일에 대한 이해』를 참조하십시오. 권한은 이 파일과 연관되는 오브젝트에 적용됩니다. 예를 들면 다음과 같습니다.

```
groupb:
  Authority=0x0040007
```

이 스탠자는 GROUPB ID에 대한 권한을 정의합니다. 권한 스펙은 다음과 같은 지정 사항을 기초로 개인 비트 패턴을 합친 것입니다.

권한 키워드	형식적 이름	16진값
connect	MQZAO_CONNECT	0x00000001
browse	MQZAO_BROWSE	0x00000002
get	MQZAO_INPUT	0x00000004
put	MQZAO_OUTPUT	0x00000008
inq	MQZAO_INQUIRE	0x00000010
set	MQZAO_SET	0x00000020
passid	MQZAO_PASS_IDENTITY_CONTEXT	0x00000040
passall	MQZAO_PASS_ALL_CONTEXT	0x00000080
setid	MQZAO_SET_IDENTITY_CONTEXT	0x00000100
setall	MQZAO_SET_ALL_CONTEXT	0x00000200
altusr	MQZAO_ALTERNATE_USER_AUTHORITY	0x00000400
allmqi	MQZAO_ALL_MQI	0x000007FF
crt	MQZAO_CREATE	0x00010000
dlt	MQZAO_DELETE	0x00020000
dsp	MQZAO_DISPLAY	0x00040000
chg	MQZAO_CHANGE	0x00080000
clr	MQZAO_CLEAR	0x00100000
chgaut	MQZAO_AUTHORIZE	0x00800000
alladm	MQZAO_ALL_ADMIN	0x009E0000
none	MQZAO_NONE	0x00000000
all	MQZAO_ALL	0x009E07FF

이 정의들은 헤더 파일 cmqzc.h에서 이루어 집니다. 다음 예에서, GROUPB에는 16진수 0x40007을 기초로 권한이 부여되었습니다. 이것은 다음에 해당됩니다.

MQZAO_CONNECT	0x00000001
MQZAO_BROWSE	0x00000002
MQZAO_INPUT	0x00000004
MQZAO_DISPLAY	0x00040000

권한:	0x00040007

이러한 액세스 권한은 GROUPB 내의 모든 사람이 다음과 같은 MQI 호출을 발행할 수 있음을 의미합니다.

```
MQCONN
MQGET(찾아보기 사용)
```

이들은 또한 이 권한 파일과 연관되는 오브젝트에 대해 DISPLAY 권한도 갖습니다.

클래스 권한 파일

클래스 권한 파일은 전체 클래스에 관련되는 권한을 보유합니다. 이 파일은 『\$CLASS』로서, 특정 오브젝트의 파일과 같은 디렉토리에 있습니다. \$CLASS 파일에서 MQZAO_CRT 항목은 클래스에서 오브젝트를 작성할 수 있는 권한을 제공합니다. 이것은 클래스의 유일한 권한입니다.

모든 클래스 권한 파일

모든 클래스 권한 파일은 전체 큐 관리자에 적용되는 권한을 보유합니다. 이 파일은 \$ACCLASS로, 큐 관리자의 auth 서브디렉토리에 있습니다.

다음 권한은 전체 큐 관리자에 적용되며 모든 클래스 권한 파일이 보유합니다.

항목... 제공되는 권한...

MQZAO_ALTERNATE_USER_AUTHORITY

MQSeries 오브젝트와 상호작용할 때 다른 사용자의 ID를 가정할 수 있습니다.

MQZAO_SET_ALL_CONTEXT

MQPUT를 발행할 때 메시지의 컨텍스트를 설정할 수 있습니다.

MQZAO_SET_IDENTITY_CONTEXT

MQPUT을 발행할 때 메시지의 ID 컨텍스트를 설정할 수 있습니다.

권한 파일 관리

권한 파일을 관리할 때는 몇 가지 사항을 고려해야 합니다.

1. 권한 파일이 보안되고 신뢰할 수 없는 일반 사용자가 쓰기 액세스할 수 없도록 되어 있는지 확인해야 합니다. 『권한 파일에 대한 권한』을 참조하십시오.
2. 파일 권한을 다시 생성하려면, 최소한 다음 중 하나를 수행해야 합니다.
 - 중요한 사항을 갱신한 후에 AUTH 서브디렉토리를 백업합니다.
 - 사용되는 명령을 포함하는 DCL 명령 파일을 보유합니다.
3. 권한 파일을 복사하거나 편집할 수 있습니다. 그러나 보통 이 파일을 수동으로 작성하거나 복구할 필요는 없습니다. 긴급할 상황이 발생하는 경우, 여기에 제공된 정보를 사용하여 유실되거나 손상된 권한 파일을 복구할 수 있습니다.

권한 파일에 대한 권한

권한 파일은 기본 원칙으로 읽을 수 있어야 합니다. 그러나 ID가 MQM인 사용자와 시스템 관리자는 이 파일을 갱신할 수 있어야 합니다.

QAM이 작성하는 권한 파일 내의 권한들은 다음과 같습니다.

S:RWD, O:RWD, G:RWD, W:R (ID=MQM, ACCESS=R+W+E+D+C)

보안 노출이 있는지 주의깊게 검토하기 전에는 이 권한을 변경하지 마십시오.

Compaq OpenVMS용 MQSeries와 함께 제공되는 명령을 사용하여 권한을 변경하려면, MQM 권한 ID를 사용하여 프로세스를 처리해야 합니다.

제8장 MQSeries 데드-레터 큐 핸들러

데드-레터 큐(DLQ, 미배달 메시지 큐라고도 함)는 데스티네이션 큐에 배달되지 못하는 메시지를 가지고 있는 큐입니다. 네트워크의 모든 큐 관리자는 연관된 DLQ를 가지고 있어야 합니다.

메시지를 큐 관리자, 메시지 채널 에이전트(MCA) 및 응용프로그램 별로 DLQ에 넣을 (put) 수 있습니다. DLQ의 모든 메시지에는 접두부로 데드-레터 헤더 구조, MQDLH가 붙어야 합니다. 큐 관리자나 메시지 채널 에이전트가 DLQ에 넣은 메시지들은 항상 MQDLH를 수반하며 DLQ에 메시지를 넣는 응용 프로그램은 MQDLH를 제공해야 합니다. MQDLH 구조의 이유 필드에는 메시지가 DLQ에 있는 이유를 식별하는 이유 코드가 있습니다.

모든 MQSeries 환경에는 DLQ에서 메시지를 처리하기 위해 정기적으로 실행되는 루틴이 있어야 합니다. MQSeries는 데드-레터 큐 핸들러(DLQ 핸들러)라고 하는 디폴트 루틴을 제공하며 이 루틴은 **runmqdlq** 명령을 사용하여 실행합니다.

DLQ의 메시지 처리에 대한 지시사항은 사용자가 작성하는 규칙 테이블에 의해 DLQ 핸들러에 제공됩니다. 즉, DLQ 핸들러는 규칙 테이블의 항목들에 대해 DLQ의 메시지들을 대조합니다. DLQ 메시지가 규칙 테이블의 한 항목과 일치하면, DLQ 핸들러는 그 항목과 연관되는 조치를 수행합니다.

이 장에는 다음 절이 포함됩니다.

- 『DLQ 핸들러 실행』
- 111 페이지의 『DLQ 핸들러 규칙 테이블』
- 117 페이지의 『규칙 테이블 처리 방법』
- 119 페이지의 『DLQ 핸들러 규칙 테이블 예』

DLQ 핸들러 실행

runmqdlq 명령을 사용하여 DLQ 핸들러를 실행합니다. 다음 두 가지 방법으로 처리하려는 DLQ와 사용하려는 큐 관리자의 이름을 지정할 수 있습니다.

- 명령 프롬프트에서 **runmqdlq**에 대한 매개변수로. 예를 들면 다음과 같습니다.

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER < qrule.ru1
```

- 규칙 테이블에서. 예를 들면 다음과 같습니다.

DLQ 핸들러

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

위의 예는 큐 관리자 ABC1.QUEUE.MANAGER가 소유하고 있는 ABC1.DEAD.LETTER.QUEUE라고 하는 DLQ에 적용됩니다.

위에 표시된 대로 DLQ나 큐 관리자를 지정하지 않으면, 설치에 대한 디폴트 큐 관리자가 해당되는 큐 관리자에 속하는 DLQ와 함께 사용됩니다.

runmqdlq 명령은 SYS\$INPUT에서 입력하고 규칙 테이블에서 SYS\$INPUT의 경로를 재지정하여 그 규칙 테이블을 **runmqdlq**와 연관시킬 수 있습니다.

주의: SYS\$INPUT 경로를 규칙 파일로 재지정하지 않고 DLQ 핸들러를 실행하면 DLQ 핸들러가 루프됩니다.

DLQ 핸들러를 실행하려면, DLQ 자체와 DLQ의 메시지가 전달되는 메시지 큐에 액세스할 수 있는 권한이 있어야 합니다. 또한 DLQ 핸들러가 메시지 컨텍스트의 사용자 ID 권한을 가지고 있는 큐에 메시지를 넣을 수 있으면, 다른 사용자의 ID를 취할 수 있는 권한을 가지고 있어야 합니다.

runmqdlq 명령에 대한 자세한 정보는 311 페이지의 『runmqdlq(데드-레터 큐 핸들러 실행)』를 참조하십시오.

샘플 DLQ 핸들러, amqsdlq

runmqdlq 명령을 사용하여 DLQ 핸들러가 실행될 뿐만 아니라 MQSeries는 **runmqdlq**를 통해 제공되는 것과 유사한 기능을 가지고 있는 샘플 DLQ 핸들러 **amqsdlq** 소스를 제공합니다. 소스는 서식 파일로만 제공되므로, 특정의 로컬 요구사항을 충족시키는 DLQ 핸들러를 제공하도록 사용자 정의해야 합니다. 예를들어, 데드-레터 헤더 없이 메시지를 처리할 수 있는 DLQ 핸들러를 사용할 수 있습니다. (디폴트 DLQ 핸들러와 샘플 **amqsdlq**는 모두 데드-레터 헤더 MQDLH로 시작하는 메시지들만 DLQ에서 처리합니다. MQDLH로 시작하지 않는 메시지들은 오류로 식별되어 DLQ에 영구히 남아 있습니다.)

amqsdlq의 소스는 다음 디렉토리에 제공됩니다.

[.DLQ], under MQS_EXAMPLES

컴파일된 버전은 다음 디렉토리에 제공됩니다.

[.BIN], under MQS_EXAMPLES

DLQ 핸들러 규칙 테이블

DLQ 핸들러 규칙 테이블은 DLQ 핸들러가 DLQ에 도착하는 메시지를 처리하는 방법을 정의합니다. 규칙 테이블에는 두 가지 유형의 항목이 있습니다.

- 테이블의 첫번째 항목(선택적)에는 제어 데이터가 있습니다.
- 테이블에 있는 다른 모든 항목들은 준수해야 하는 DLQ 핸들러의 규칙입니다. 각 규칙은 메시지가 대조되는 패턴(일련의 메시지)과 DLQ의 메시지가 지정된 패턴과 대조될 때 취할 조치로 구성됩니다. 규칙 테이블에는 최소한 하나의 규칙이 있어야 합니다.

규칙 테이블의 각 항목은 하나 이상의 키워드로 구성됩니다.

제어 데이터

이 절에서는 DLQ 핸들러 규칙 테이블의 제어 데이터 항목에 포함시킬 수 있는 키워드에 대해 설명합니다. 다음을 주의하십시오.

- 키워드의 디폴트 값(있는 경우)에는 밑줄이 그어져 있습니다.
- 수직선(|)은 지정할 수 있는 대체값만 분리합니다.
- 모든 키워드는 선택적입니다.

INPUTQ (QueueName|_')

처리하려는 DLQ에 이름을 지정할 수 있습니다.

1. **runmqdlq** 명령에 대한 매개변수로 INPUTQ 값을 지정할 경우, 이는 규칙 테이블에서 INPUTQ 값을 대체합니다.
2. **runmqdlq** 명령에 대한 매개변수로 INPUTQ 값을 지정하지 않고 규칙 테이블의 값을 지정할 경우, 규칙 테이블의 INPUTQ 값이 사용됩니다.
3. DLQ를 지정하지 않거나 규칙 테이블에서 INPUTQ(' ')를 지정할 경우, **runmqdlq** 명령에 대한 매개변수로 제공되는 큐 관리자에 속하는 DLQ의 이름이 사용됩니다.
4. **runmqdlq** 명령에 대한 매개변수나 규칙 테이블의 값으로 INPUTQ 값을 지정하지 않을 경우, 규칙 테이블의 INPUTQM 키워드에 이름이 지정된 큐 관리자에 속하는 DLQ가 사용됩니다.

INPUTQM (QueueManagerName|_')

INPUTQ 키워드에 이름이 지정된 DLQ를 소유하는 큐 관리자의 이름을 지정할 수 있습니다.

1. **runmqdlq** 명령에 대한 매개변수로 INPUTQM 값을 지정할 경우, 이것은 규칙 테이블에서 INPUTQM 값을 대체합니다.
2. **runmqdlq** 명령에 대한 매개변수로 INPUTQM 값을 지정하지 않을 경우, 규칙 테이블의 INPUTQM 값이 사용됩니다.

DLQ 핸들러

3. 큐 관리자를 지정하지 않거나 규칙 테이블에 INPUTQM(' ')을 지정할 경우, 설치시 디폴트 큐 관리자가 사용됩니다.

RETRYINT (Interval|60)

DLQ 핸들러가 첫번째 시도에서 처리할 수 없어서 반복해서 시도를 요청한 DLQ의 메시지들을 다시 처리해야 하는 초 단위의 간격입니다. 디폴트로, 재시도 간격은 60초입니다.

WAIT (YES|NO|nnn)

DLQ 핸들러가 처리할 수 있는 추가 메시지가 없음을 감지할 때 DLQ에 추가 메시지가 도착하기를 기다려야 하는지 여부를 나타냅니다.

YES DLQ 핸들러가 영구적으로 기다리도록 합니다.

NO DLQ가 비어 있거나 처리할 수 있는 메시지가 없음을 감지할 때 DLQ 핸들러를 종료시킵니다.

nnn 큐가 비어 있거나 처리할 수 있는 메시지가 없음을 감지한 후, 종료 전에 DLQ 핸들러가 새 작업이 도착할 때까지 *nnn*초 동안 기다리게 합니다.

자주 사용되는 DLQ에 대해서는 WAIT(YES)을 지정하고, 활동 레벨이 낮은 DLQ에 대해서는 WAIT(NO) 또는 WAIT(*nnn*)을 지정하는 것이 바람직합니다. DLQ 핸들러 종료에 허용되면, 트리거링으로 재실행하는 것이 더 좋습니다.

제어 데이터를 규칙 테이블에 포함시키는 방법의 대안으로, **runmqdlq** 명령의 입력 매개변수로 DLQ와 큐 관리자의 이름을 제공할 수 있습니다. 어떤 값이 규칙 테이블과 **runmqdlq** 명령 모두에 지정된 경우, **runmqdlq** 명령에 지정된 값이 우선합니다.

주: 제어 데이터 항목이 규칙 테이블에 포함된 경우, 그 항목은 반드시 테이블에서 첫 번째 항목이어야 합니다.

규칙(패턴 및 조치)

그림8에 DLQ 핸들러 규칙 테이블의 규칙 예가 나와 있습니다.

```
PERSIST(MQPER_PERSISTENT) REASON(MQRC_PUT_INHIBITED) +  
ACTION(RETRY) RETRY(3)
```

그림 8. DLQ 핸들러 규칙 테이블의 규칙 예. 이 규칙은 MQPUT 및 MQPUT1이 금지되어 DLQ에 넣은(put) 지속 메시지를 DLQ 핸들러가 데스티네이션 큐로 세 번 전달하도록 지시합니다.

규칙에서 사용할 수 있는 모든 키워드는 이 절의 나머지 부분에서 설명합니다. 다음을 주의하십시오.

- 키워드의 디폴트 값(있는 경우)에는 밑줄이 그어져 있습니다. 대부분의 키워드에 대해, 디폴트 값은 값을 대조하는 *(별표)입니다.

- 수직선(|)은 지정할 수 있는 대체값만 분리합니다.
- ACTION을 제외한 모든 키워드는 선택적입니다.

이 절에서는 패턴 대조 키워드(DLQ의 메시지가 대조되는 키워드)에 대한 설명을 먼저 하고, 그 다음으로 조치 키워드(DLQ 핸들러가 대조하는 메시지를 처리하는 방법을 결정하는 키워드)에 대해 설명합니다.

패턴 대조 키워드

다음은 DLQ의 메시지가 대조되는 값을 지정하기 위해 사용하는 패턴 대조 키워드에 대한 설명입니다. 모든 패턴 대조 키워드는 선택적입니다.

APPLIDAT (*ApplIdentityData*|*)

DLQ의 메시지에 대해 메시지 설명자 MQMD에 지정된 *ApplIdentityData* 값입니다.

APPLNAME (*PutApplName*|*)

DLQ의 메시지에 대해 메시지 설명자 MQMD의 *PutApplName* 필드에 지정된, MQPUT 또는 MQPUT1 호출을 발행한 응용프로그램의 이름입니다.

APPLTYPE (*PutApplType*|*)

DLQ의 메시지에 대해 메시지 설명자 MQMD에 지정된 *PutApplType* 값입니다.

DESTQ (*QueueName*|*)

메시지의 목적지인 메시지 큐의 이름입니다.

DESTQM (*QueueManagerName*|*)

메시지의 목적지인 메시지 큐의 큐 관리자 이름입니다.

FEEDBACK (*Feedback*|*)

MsgType 값이 MQFB_REPORT이면, *Feedback*은 보고서의 성격에 대해 설명합니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQFB_COA를 사용하면 DLQ에서 테스트네이션 큐에 도착하는 것을 파악해야 하는 메시지들을 식별할 수 있습니다.

FORMAT (*Format*|*)

메시지 데이터의 형식을 설명하기 위해 메시지 송신자가 사용하는 이름입니다.

MSGTYPE (*MsgType*|*)

DLQ에 있는 메시지의 메시지 유형입니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQMT_REQUEST를 사용하면 DLQ에서 응답을 요구하는 메시지들을 식별할 수 있습니다.

PERSIST (*Persistence*|*)

메시지의 지속성 값입니다(메시지의 지속성은 큐 관리자가 재시작을 할 수 있는지 여부를 판별합니다).

DLQ 핸들러

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQPER_PERSISTENT 를 사용하면 DLQ에서 지속되는 메시지들을 식별할 수 있습니다.

REASON (*ReasonCode**)

메시지를 DLQ에 넣은(put) 이유를 설명하는 이유 코드입니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQRC_Q_FULL을 사용하면 데스티네이션 큐가 꽉 차서 DLQ에 넣은 메시지들을 식별할 수 있습니다.

REPLYQ (*QueueName**)

DLQ의 메시지에 대해 메시지 설명자 MQMD에 지정된 리플라이-투 큐의 이름입니다.

REPLYQM (*QueueManagerName**)

DLQ에 있는 메시지의 메시지 설명자 MQMD에 지정된, 리플라이-투 큐의 큐 관리자 이름입니다.

USERID (*UserIdentifier**)

메시지 설명자 MQMD에 지정된, DLQ에서 메시지를 생성한 사용자의 사용자 ID입니다.

조치 키워드

조치 키워드는 일치하는 메시지가 처리되는 방법에 대해 설명하는 키워드로, 아래에 설명되어 있습니다.

ACTION (DISCARD|IGNORE|RETRY|FWD)

이 규칙에 정의된 패턴을 대조하는 DLQ의 메시지에 대해 취하는 조치입니다.

DISCARD

DLQ에서 메시지가 삭제되도록 합니다.

IGNORE

DLQ에 메시지가 남아 있도록 합니다.

RETRY

DLQ 핸들러가 데스티네이션 큐에 다시 메시지를 넣도록 합니다.

FWD DLQ 핸들러가 FWDQ 키워드에 이름이 지정된 큐로 메시지를 전달하도록 합니다.

ACTION 키워드는 반드시 지정해야 합니다. 조치 구현 시도 횟수는 RETRY 키워드가 관리합니다. 시도간 간격은 제어 데이터의 RETRYINT 키워드가 제어합니다.

FWDQ (*QueueName*|&DESTQ|&REPLYQ)

ACTION (FWD)이 요청될 때 메시지가 전달되어야 하는 메시지 큐의 이름입니다.

QueueName

메시지 큐의 이름입니다. FWDQ(' ')는 유효하지 않습니다.

&DESTQ

큐 이름을 MQDLH 구조의 *DestQName* 필드에서 취하도록 합니다.

&REPLYQ

큐 이름을 메시지 설명자 MQMD의 *ReplyToQ* 필드에서 취하도록 합니다.

FWDQ (&REPLYQ)를 지정하는 규칙이 *ReplyToQ* 필드가 공백인 메시지와 대조될 때 오류 메시지가 발생하지 않도록 하려면, 메시지 패턴에서 REPLYQ (?*)를 지정하면 됩니다.

FWDQM (QueueManagerName|&DESTQM|&REPLYQM|'_')

메시지가 전달될 큐의 큐 관리자를 식별합니다.

QueueManagerName

ACTION (FWD)이 요청될 때 메시지가 전달될 큐의 큐 관리자 이름입니다.

&DESTQM

큐 관리자 이름을 MQDLH 구조의 *DestQMgrName* 필드에서 취하도록 합니다.

&REPLYQM

이름을 메시지 설명자 MQMD의 *ReplyToQMgr* 필드에서 취하도록 합니다.

' ' 디폴트 값인 FWDQM(' ')은 로컬 큐 관리자를 식별합니다.

HEADER (YES|NO)

ACTION (FWD)이 요청되는 메시지에 대해 MQDLH가 남아 있어야 하는지를 지정합니다. 디폴트로, MQDLH는 메시지에 남아 있습니다. HEADER 키워드는 FWD 이외의 다른 조치에 대해 올바르지 않습니다.

PUTAUT (DEF|CTX)

DLQ 핸들러가 메시지가 넣을 때 사용해야 하는 권한을 정의합니다.

DEF 메시지를 DLQ 핸들러 자체의 권한을 사용하여 넣도록 합니다.

CTX 메시지를 메시지 컨텍스트의 사용자 ID 권한을 사용하여 넣도록 합니다. PUTAUT (CTX)를 지정할 경우, 다른 사용자의 ID를 알 수 있는 권한을 가지고 있어야 합니다.

RETRY (RetryCount|1)

조치를 시도해야 하는 1-999,999,999 범위의 횟수(제어 데이터의 RETRYINT 키워드에 지정된 간격으로).

주: DLQ 핸들러가 특정 규칙을 시도하는 횟수는 DLQ 핸들러의 현재 인스턴스에 대해 고유합니다. 횟수는 재시작되면 지속되지 않습니다. DLQ 핸들러가 재시작될 경우, 규칙 적용 횟수는 0으로 재설정됩니다.

규칙 테이블 규정

규칙 테이블은 구문, 구조 및 내용에 대해서 다음 규정을 따라야 합니다.

- 규칙 테이블에는 최소한 하나의 규칙이 있어야 합니다.
- 키워드는 모든 순서가 가능합니다.
- 키워드는 어떤 규칙에서든지 한번만 포함될 수 있습니다.
- 키워드는 대소문자가 구분되지 않습니다.
- 키워드와 매개변수 값은 최소한 하나의 공백이나 쉼표로 다른 키워드와 분리되어야 합니다.
- 규칙의 맨 앞과 맨 끝, 그리고 키워드, 구두점 및 값들 사이에는 여러 공백이 있을 수 있습니다.
- 각 규칙은 줄바꾸기에서 시작해야 합니다.
- 이식성으로 인해, 행의 유효 길이는 72자를 초과할 수 없습니다.
- 규칙이 다음 행의 첫번째 비공백 문자에서 계속됨을 나타내려면 행에 더하기 부호(+)를 마지막 비공백 문자로 사용하십시오. 규칙이 다음 행의 맨 앞에서부터 계속됨을 나타내려면 행에 빼기 부호(-)를 마지막 비공백 문자로 사용하십시오. 연속 문자는 키워드와 매개변수 내에서 발생할 수 있습니다.
- 별표(*)로 시작하는 주석 행은 규칙 테이블 내의 모든 곳에서 발생할 수 있습니다.
- 빈 줄은 무시합니다.
- DLQ 핸들러의 각 항목은 하나 이상의 키워드와 이에 연관되는 매개변수들로 구성됩니다. 매개변수는 다음의 구문 규칙을 따라야 합니다.
 - 각 매개변수 값에는 최소한 하나의 유효 문자가 있어야 합니다. 따옴표로 묶어진 값 내의 구분 따옴표는 유효 문자로 간주되지 않습니다. 예를 들어, 올바른 매개변수들은 다음과 같습니다.

FORMAT ('ABC')

세 개의 유효 문자

FORMAT (ABC) 세 개의 유효 문자

FORMAT ('A') 하나의 유효 문자

FORMAT (A) 하나의 유효 문자

FORMAT (' ') 하나의 유효 문자

다음 매개변수들은 유효 문자를 포함하고 있지 않기 때문에 올바르지 않습니다.

FORMAT ('')

FORMAT ()

FORMAT ()

FORMAT

DLQ 핸들러

- 와일드 카드 문자가 지원됩니다. 후미 공백을 제외하고, 단일 문자 대신 물음표 (?)를 사용할 수 있고, 인접 문자 대신(0개 이상) 별표(*)를 사용할 수 있습니다. 별표(*) 및 물음표(?)는 매개변수 값에서 항상 와일드카드 문자로 해석됩니다.
- 와일드카드 문자는 ACTION, HEADER, RETRY, FWDQ, FWDQM, PUTAUT 키워드의 매개변수 내에 포함될 수 없습니다.
- 매개변수 값과 DLQ의 메시지에 있는 해당되는 필드들의 후미 공백은 와일드카드와 대조할 때 유효하지 않습니다. 그러나 따옴표로 묶여 있는 문자열 내에 임베드된 공백과 선행 공백은 와일드카드와 대조할 때 유효합니다.
- 숫자 매개변수에는 물음표(?) 와일드 카드가 포함될 수 없습니다. 별표(*)는 전체 숫자 매개변수 대신 사용할 수 있지만, 숫자 매개변수의 일부로 사용할 수는 없습니다. 예를 들어, 올바른 숫자 매개변수들은 다음과 같습니다.

MSGTYPE(2) 응답 메시지만 적합합니다.

MSGTYPE(*) 모든 메시지 유형이 적합합니다.

MSGTYPE('*') 모든 메시지 유형이 적합합니다.

그러나, MSGTYPE('2*')은 숫자 매개변수의 일부로 별표(*)를 포함하고 있으므로 올바르지 않습니다.

- 숫자 매개변수는 범위 0-999,999,999 내에 있어야 합니다. 매개변수 값이 이 범위에 있을 경우, 매개변수 값은 현재 키워드가 관련된 필드에서 올바르지 않아도 승인됩니다. 기호 이름은 숫자 매개변수에 사용할 수 있습니다.
- 문자열 값이 키워드가 관련된 MQDLH 또는 MQMD의 필드보다 짧은 경우, 값은 필드 길이에 맞도록 공백이 채워집니다. 별표를 제외하고, 값이 필드보다 길면 오류가 진단됩니다. 예를 들어, 다음 값들은 8자 필드에 대해 모두 올바른 문자열 값입니다.

'ABCDEFGH'	8자
'A*C*E*G*I'	별표를 제외하고 5자
'*A*C*E*G*I*K*M*O*'	별표를 제외하고 8자

- 공백, 수문자 또는 마침표(.), 정방향 슬래시 (/), 밑줄(_) 및 퍼센트 부호(%)를 제외한 특수 문자가 있는 문자열은 작은 따옴표로 묶어야 합니다. 따옴표로 묶지 않은 소문자는 대문자로 변환됩니다. 문자열에 따옴표가 있으면, 두 개의 작은 따옴표를 사용하여 따옴표의 시작과 끝을 표시해야 합니다. 문자열의 길이를 계산할 때, 큰 따옴표가 나올 때마다 작은 따옴표로 계산됩니다.

규칙 테이블 처리 방법

DLQ 핸들러는 패턴이 DLQ의 메시지와 일치하는 규칙에 대한 규칙 테이블을 검색합니다. 검색은 테이블에서 첫번째 규칙을 사용하여 시작되며, 테이블 전체에서 순차적으로 계속됩니다. 일치하는 패턴을 사용하는 규칙이 발견되면, 해당 규칙에서 조치를 시도합니다. DLQ 핸들러는 해당 규칙을 적용하려고 할 때마다 규칙에 대한 재시도 횟수

DLQ 핸들러

를 1씩 증가시킵니다. 첫번째 재시도가 실패하면, 시도 횟수가 RETRY 키워드에 지정된 숫자와 일치할 때까지 반복됩니다. 모든 시도가 실패하면, DLQ 핸들러는 테이블에서 다음 일치 규칙을 검색합니다.

이 프로세스는 조치가 성공할 때까지 후속 대조 규칙에 대해 반복됩니다. 각 대조 규칙을 RETRY 키워드에 지정된 횟수만큼 시도했는데 모든 시도가 실패하면 ACTION(IGNORE)으로 간주합니다. 일치하는 규칙이 없을 경우에도 ACTION(IGNORE)으로 간주합니다.

주:

1. MQDLH로 시작하는 DLQ의 메시지들에 대해서만 일치하는 규칙 패턴들을 찾을 수 있습니다. MQDLH로 시작하지 않는 메시지들은 정기적으로 오류가 있는 시작으로 보고되어, DLQ에 영구히 남아 있습니다.
2. 모든 패턴 키워드는 하나의 규칙을 하나의 조치로만 구성할 수 있는 것처럼, 디폴트로 설정할 수 있습니다. 그러나 조치 전용 규칙은 MQDLH를 가지고 있지만, 테이블의 다른 규칙에 따라 아직 처리되지 않은 큐의 모든 메시지에 적용됩니다.
3. 규칙 테이블은 DLQ 핸들러가 시작될 때 유효화되고, 오류도 그 때 플래그됩니다. 규칙 테이블은 언제든지 변경할 수 있지만, 그러한 변경사항은 DLQ 핸들러가 재시작되지 않으면 적용되지 않습니다.
4. DLQ 핸들러는 메시지, MQDLH 또는 메시지 설명자의 내용을 변경하지 않습니다. DLQ 핸들러는 항상 메시지 옵션 MQPMO_PASS_ALL_CONTEXT를 사용하여 다른 큐에 메시지를 넣습니다(put).
5. DLQ 핸들러는 MQOO_INPUT_AS_Q_DEF 옵션을 사용하여 DLQ를 엽니다.
6. DLQ 핸들러의 여러 인스턴스가 동시에 같은 규칙 테이블을 사용하여 같은 큐에 대해 실행될 수 있습니다. 그러나 DLQ와 DLQ 핸들러 사이는 일대일 관계가 되는 것이 더 일반적입니다.

모든 DLQ 메시지 처리

DLQ 핸들러는 표시되었으나 제거되지 않은 DLQ의 모든 메시지에 대한 레코드를 보존합니다. DLQ에서 작은 메시지 서브셋을 추출하기 위한 필터로 DLQ 핸들러를 사용할 경우, DLQ 핸들러는 처리하지 않은 DLQ의 메시지들에 대한 레코드를 계속 보존해야 합니다. 또한 DLQ 핸들러는 DLQ가 FIFO(first-in-first-out)로 정의되었다고 해도 DLQ에 도착하는 새로운 메시지가 표시될 것인지 보장할 수는 없습니다. 그러므로 큐가 비어 있지 않으면, 모든 메시지를 점검하기 위해 정기적으로 DLQ를 다시 스캔해야 합니다. 이러한 이유로, DLQ에는 될 수 있으면 몇 개의 메시지만 있도록 해야 합니다. 여러 가지 이유로 제거하거나 다른 큐로 전달할 수 없는 메시지가 큐에 누적되는 경우, DLQ 핸들러의 워크로드가 증가되어 DLQ 자체가 완전히 채워질 위험이 있게 됩니다.

DLQ 핸들러

고유 측정 방법을 사용하면 DLQ 핸들러가 DLQ를 비울 수 있습니다. 예를 들어, 단순히 DLQ에 메시지들을 두는 ACTION(IGNORE)을 사용하지 않도록 합니다 (ACTION(IGNORE)은 테이블의 다른 규칙에서 명시적으로 언급되지 않은 메시지들에 대해 가정됩니다). 대신, 그렇게 하지 않으면 무시되는 메시지들에 대해 다른 큐로 이동하는 조치를 사용하십시오. 예를 들면 다음과 같습니다.

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

마찬가지로, 테이블의 최종 규칙은 테이블의 이전 규칙들에서 언급되지 않은 메시지의 처리를 위해서는 포괄적인 규칙이어야 합니다. 예를 들어, 테이블의 최종 규칙은 다음과 같을 수 있습니다.

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

이 조치는 테이블의 최종 규칙에 해당되는 메시지들을 큐 REALLY.DEAD.QUEUE에 전달하도록 합니다. 이 큐에서 해당 메시지들은 수동으로 처리될 수 있습니다. 그러한 규칙을 사용하지 않으면, 메시지들은 DLQ에 영구적으로 남아 있게 됩니다.

DLQ 핸들러 규칙 테이블 예

다음은 단일 제어 데이터 항목과 몇 개의 규칙을 포함하는 규칙 테이블의 예입니다.

```
*****
*           An example rules table for the runmqdlq command           *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.

* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
```

DLQ 핸들러

- * condition, attempt to forward the message to its
- * destination queue. Make 5 attempts at approximately
- * 60-second intervals (the default value for RETRYINT).

```
REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)
```

- * The AAAA corporation are always sending messages with incorrect
- * addresses. When we find a request from the AAAA corporation,
- * we return it to the DLQ (DEADQ) of the reply-to queue manager
- * (&REPLYQM).
- * The AAAA DLQ handler attempts to redirect the message.

```
MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +  
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)
```

- * The BBBB corporation never do things by half measures. If
- * the queue manager BBBB.1 is unavailable, try to
- * send the message to BBBB.2

```
DESTQM(bbbb.1) +  
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)
```

- * The CCCC corporation considers itself very security
- * conscious, and believes that none of its messages
- * will ever end up on one of our DLQs.
- * Whenever we see a message from a CCCC queue manager on our
- * DLQ, we send it to a special destination in the CCCC organization
- * where the problem is investigated.

```
REPLYQM(CCCC.*) +  
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)
```

- * Messages that are not persistent run the risk of being
- * lost when a queue manager terminates. If an application
- * is sending nonpersistent messages, it should be able
- * to cope with the message being lost, so we can afford to
- * discard the message.

```
PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
```

- * For performance and efficiency reasons, we like to keep
- * the number of messages on the DLQ small.
- * If we receive a message that has not been processed by
- * an earlier rule in the table, we assume that it
- * requires manual intervention to resolve the problem.
- * Some problems are best solved at the node where the
- * problem was detected, and others are best solved where
- * the message originated. We don't have the message origin,
- * but we can use the REPLYQM to identify a node that has
- * some interest in this message.
- * Attempt to put the message onto a manual intervention
- * queue at the appropriate node. If this fails,
- * put the message on the manual intervention queue at
- * this node.

```
REPLYQM('?*') +
```

```
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)  
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

DLQ 핸들러

제9장 도구화 이벤트

MQSeries 도구 이벤트를 사용하여 큐 관리자 작업을 모니터링할 수 있습니다. 이 장에서는 도구 이벤트에 대한 간단한 소개를 제공합니다. 완전한 설명은 *MQSeries Programmable System Management*에 있는 도구 이벤트 관련 부분을 참조하십시오.

도구 이벤트의 개념

도구 이벤트는 큐 관리자가 사전 정의된 일련의 조건을 검색할 때마다 이벤트 메시지는 특수 메시지가 생성되도록 합니다. 예를 들어, 다음 상황에서 *Queue Full* 이벤트가 발생합니다.

- Queue Full 이벤트가 지정된 큐에서 사용 가능할 경우
- 응용프로그램이 큐에 메시지를 넣기(put) 위해 MQPUT 호출을 발행했으나, 큐가 가득 차서 호출에 실패한 경우

도구 이벤트를 발생시킬 수 있는 기타 상황은 다음과 같습니다.

- 큐의 메시지 수가 임계값 한계에 도달한 경우
- 지정된 시간 내에 큐 서비스가 제공되지 않은 경우
- 채널 인스턴스가 시작되거나 정지된 경우
- Compaq OpenVMS용 MQSeries 시스템에서, 응용프로그램이 권한이 없는 사용자 ID를 지정하여 큐를 열려는 경우

채널 이벤트를 제외하고, 모든 도구 이벤트를 생성하려면 먼저 사용 가능하게 되어야 합니다.

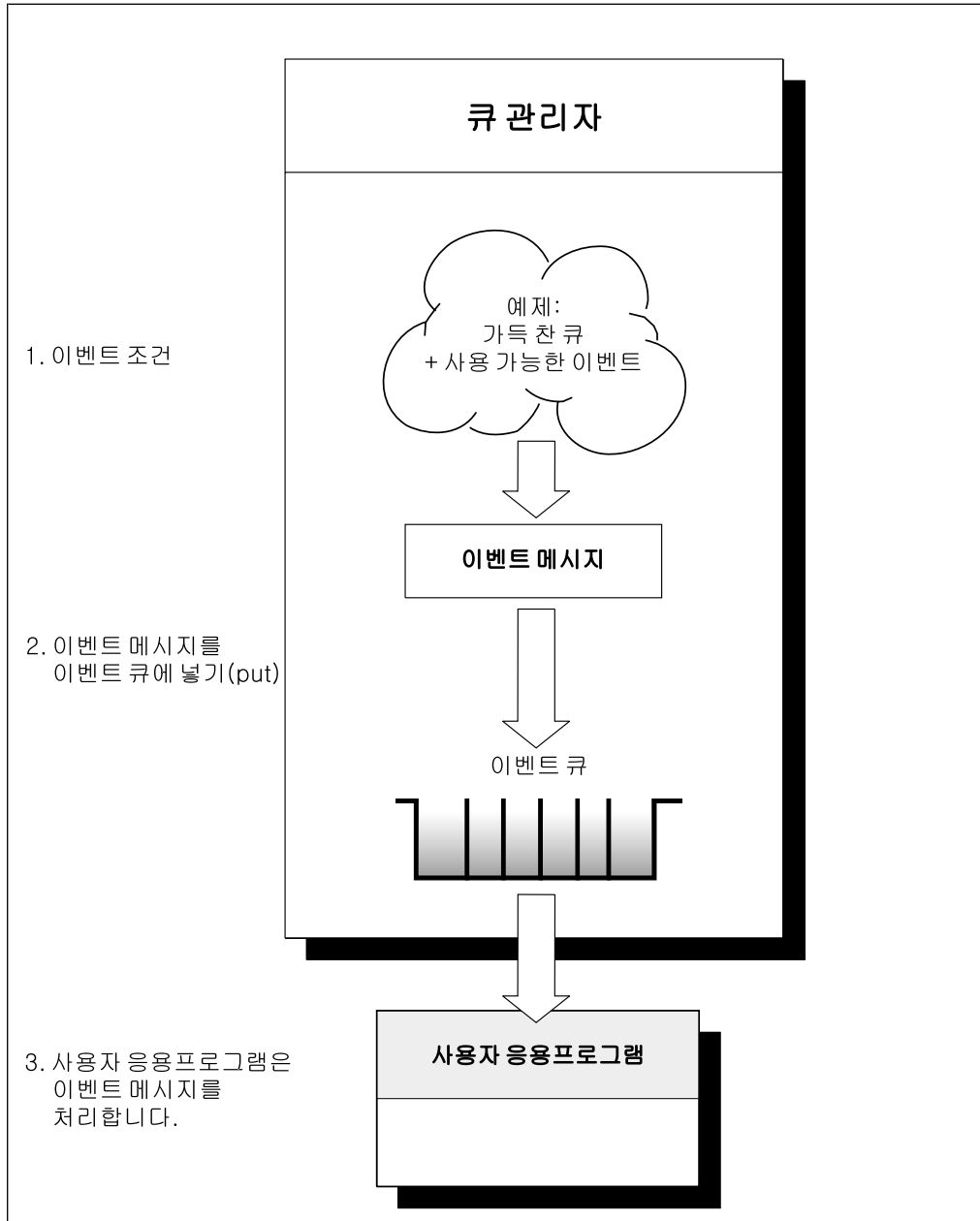


그림 9. 도구 이벤트에 대한 이해. 큐관리자가 이벤트에 대한 조건이 충족되었는지를 감지할 때, 이벤트 메시지를 적절한 이벤트 큐에 넣습니다(put).

이벤트를 발생시키는 조건에 관한 정보가 있는 이벤트 메시지는 이벤트 큐에 넣어집니다(put). 응용프로그램은 분석을 위해 이 큐에서 이벤트 메시지를 검색할 수 있습니다.

이벤트를 사용하는 이유

사용자가 이벤트 큐를 리모트 큐로 지정하면, 도구 이벤트를 지원하는 노드에 대해 모든 이벤트 큐를 단일 큐 관리자에 넣을(put) 수 있습니다. 그런 다음 단일 노드에서 큐 관리자의 네트워크를 모니터링하기 위해 생성된 이벤트를 사용할 수 있습니다. 그림10에 이에 대한 설명이 있습니다.

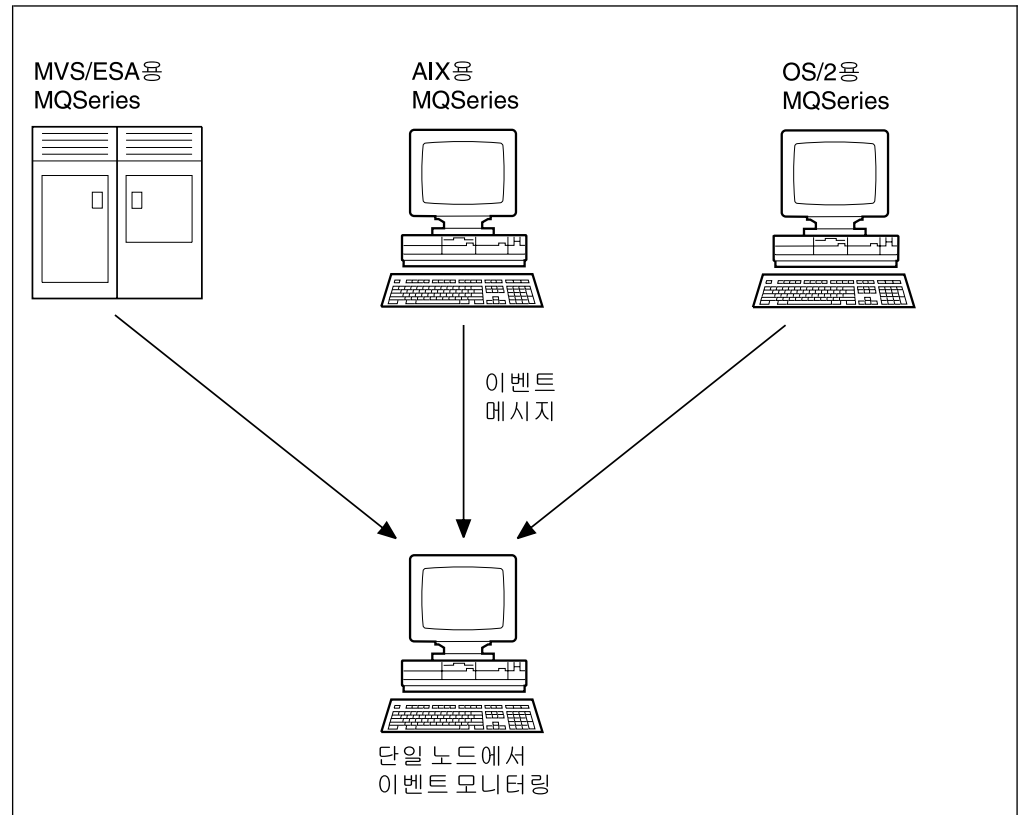


그림 10. 단일 노드상의 서로 다른 플랫폼 사이의 큐 관리자 모니터링

이벤트 유형

MQSeries 이벤트는 다음과 같은 범주로 나누어집니다.

큐 관리자 이벤트

이 이벤트들은 큐 관리자 내의 자원 정의에 관련됩니다. 예를 들어, 응용프로그램이 존재하지 않는 큐에 메시지를 넣으려고(put) 하는 경우입니다.

성능 이벤트

이러한 이벤트는 자원이 임계값에 도달했음을 알립니다. 예를 들어, 큐 용량 한계에 도달했거나, 가져오기(get) 다음에 큐에 사전 정의된 시간 내에 서비스되지 않은 경우입니다.

트리거된 이벤트 큐 사용

이벤트가 생성될 때 이벤트 큐에 놓여지는 이벤트 메시지가 (사용자 작성) 모니터링 응용프로그램을 시작할 수 있도록 트리거를 사용하여 이벤트 큐를 설정할 수 있습니다. 이 응용프로그램은 이벤트 메시지를 처리하고 적절한 조치를 취할 수 있습니다. 예를 들어, 특정 이벤트는 오퍼레이터에게 알려져야 하고, 다른 이벤트는 일부 관리 작업을 자동으로 시작하는 응용프로그램을 시작할 수 있습니다.

이벤트 사용 및 사용 안함

이벤트의 유형에 따라 큐 관리자나 큐 속성, 또는 둘 다에 대해 적절한 값을 지정하여 이벤트를 사용 가능 또는 불가능하게 할 수 있습니다. 다음 중 하나를 사용하여 이를 수행하십시오.

- MQSC 명령. 자세한 정보는 *MQSeries* 명령 참조서를 참조하십시오.
- UNIX 시스템, OpenVMS 시스템 및 OS/2에서의 큐 관리자에 대한 PCF 명령. 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.
- MQAI 명령. 자세한 정보는 *MQSeries Administration Interface Programming Guide and Reference*를 참조하십시오.

이벤트 사용은 이벤트 범주에 따라 결정됩니다.

- 큐 관리자 이벤트는 큐 관리자에서 속성을 설정하여 사용합니다.
- 성능 이벤트는 전체적으로 큐 관리자에서 사용 가능해야 합니다. 그렇지 않으면 전혀 성능 이벤트가 발생할 수 없습니다. 그런 다음 적절한 큐 속성을 설정하면 특정 성능 이벤트를 사용할 수 있습니다. 또한 큐 용량 상한과 같은, 이벤트가 발생할 수 있는 조건을 지정해야 합니다.
- 채널 이벤트는 자동으로 발생하므로 특별한 조치를 취할 필요가 없습니다. 채널 이벤트를 모니터하지 않으려면, 채널 이벤트 큐의 넣기(put)를 금지할 수 있습니다.

이벤트 메시지

이벤트 메시지에는 이벤트 유형, 이벤트를 발생시킨 응용프로그램의 이름과 같은 이벤트 출처에 관련된 정보가 있습니다. 성능 이벤트의 경우에는 큐에 대한 간단한 통계적 요약이 있습니다.

이벤트 메시지 형식은 PCF 응답 메시지의 형식과 비슷합니다. 메시지 데이터는 *MQSeries Programmable System Management*에 설명된 데이터 구조를 사용하여 사용자가 작성한 관리 프로그램을 통해 검색할 수 있습니다.

제10장 트랜잭션식 지원

*MQSeries Application Programming Guide*에 이 장의 주제에 대한 완전한 소개가 수록되어 있습니다. 여기에는 간략한 소개만 제공됩니다.

응용프로그램은 갱신 세트를 작업 단위로 분류할 수 있습니다. 이들 갱신은 대개는 논리적으로 관련되며, 모두 데이터 무결성을 보존할 수 있어야 합니다. 하나의 갱신이 성공해도 다른 갱신에 실패하면 데이터 무결성이 손실됩니다.

작업 단위는 성공적으로 완료될 때 **확약**됩니다. 이 때 작업 단위 안에서 갱신된 모든 사항은 영구적이 되거나 되돌릴 수 없습니다. 작업 단위가 실패하면 모든 갱신사항은 **백아웃**됩니다. 동기점 통합은 작업 단위가 무결성에 대해 **확약**되거나 **백아웃**되는 프로세스입니다.

로컬 작업 단위에서는 MQSeries 큐 관리자의 자원만 갱신됩니다. 여기에서 1단계 **확약 프로세스**를 사용하여 큐 관리자 자체에 의해 동기점 통합이 제공됩니다.

글로벌 작업 단위에서는 XA 준수 데이터베이스 등 다른 자원 관리자에 속하는 자원도 갱신됩니다. 여기에서는 2단계 **확약 프로시저**가 사용되며, 큐 관리자 자체에 의해 작업 단위가 통합될 수 있습니다.

요약하면 큐 관리자 자원은 로컬 또는 전역 작업 단위의 일부로 갱신할 수 있습니다.

로컬 작업 단위

갱신될 자원이 MQSeries 큐 관리자의 자원일 경우에 로컬 작업 단위를 사용하십시오. 갱신사항은 MQCMIT 명령어(verb)를 사용하여 **확약**되거나 MQBACK을 사용하여 **백아웃**됩니다.

전역 작업 단위

XA를 준수하는 데이터베이스 관리자에 대한 갱신사항도 포함해야 할 경우에는 전역 작업 단위를 사용하십시오. 여기에서 통합은 큐 관리자에 대해 내부 또는 외부 통합이 될 수 있습니다.

큐 관리자 통합

전역 작업 단위는 MQBEGIN 명령어(verb)를 사용하여 시작된 후 MQCMIT를 사용하여 **확약**되거나 MQBACK을 사용하여 **백아웃**됩니다. 2단계 **확약 프로세스**가 사용되므로, Oracle과 같은 XA 준수 자원 관리자에 대해 먼저 **확약** 준비를 요청합니다. 모든 자원 관리자가 준비되어야만 **확약** 요청을 하게 됩니다. 만약 임의의 자원 관리자가 **확약** 준비를 할 수 없다는 신호를 보낼 경우, 각 자원 관리자에게 **확약** 대신 **백아웃**을 요청합니다.

외부 통합

여기에서 통합은 IBM CICS, Transarc Encina 또는 BEA Tuxedo와 같은 XA 준수 트랜잭션 관리자가 수행합니다. 트랜잭션 관리자는 작업 단위의 시작과 확약을 제어합니다. MQBEGIN, MQCMIT 및 MQBACK 명령어(verb)는 사용할 수 없습니다.

이 장에서는 전역 작업 단위에 대한 지원을 가능하게 하는 방법에 대해 설명합니다(로컬 작업 단위에 대한 지원은 특별히 사용하지 않아도 됩니다).

이 장에는 다음과 같은 절이 있습니다.

- 『데이터베이스 통합』
- 134 페이지의 『Oracle 구성』
- 139 페이지의 『관리 작업』

데이터베이스 통합

큐 관리자가 글로벌 작업 단위 자체를 통합할 때에는 MQ 작업 단위내에서 데이터베이스 갱신을 통합할 수 있습니다. 즉, 혼합 MQI와 SQL 응용프로그램을 작성할 수 있고 MQCMIT 및 MQBACK 명령어(verb)를 사용하여 큐와 데이터베이스에 대한 변경 사항을 함께 확약하거나 롤백할 수 있습니다.

큐 관리자는 2단계 확약 프로토콜을 사용하여 이를 실행합니다. 작업 단위를 확약할 때, 큐 관리자는 먼저 참여하는 각 데이터베이스 관리자에게 갱신을 확약할 준비가 되었는지 여부를 묻습니다. 큐 관리자를 포함하여 모든 참여자들이 확약할 준비가 된 경우에만 모든 큐 및 데이터베이스 갱신이 확약됩니다. 하나의 참여자라도 갱신을 준비할 수 없는 경우, 작업 단위는 대신 백아웃됩니다.

확약 프로토콜 중에 큐 관리자와 임의의 데이터베이스 관리자와의 연결이 끊어지는 경우, 완벽한 복구 지원이 제공됩니다. 데이터베이스 관리자가 인다우트(in-doubt) 상태(즉, 준비하도록 호출되었지만 아직 확약 또는 백아웃 결정을 수신하지 못한 상태)에 있는 동안 사용 불가능하게 되면, 큐 관리자는 작업 단위의 결과가 성공적으로 전달될 때까지 그 작업 단위의 결과를 기억합니다. 마찬가지로, 큐 관리자가 불완전한 확약 조작을 미해결 상태로 종료하면, 큐 관리자가 다시 시작할 때 이 조작을 기억합니다.

MQI 명령어(verb)인 MQBEGIN은 데이터베이스 갱신사항도 포함할 작업 단위를 표시하는데 사용해야 합니다. *MQSeries Application Programming Guide*에서는 같은 작업 단위내에서 MQSeries 및 데이터베이스 갱신을 수행하는 샘플 프로그램을 설명합니다.

큐 관리자는 *X/Open Distributed Transaction Processing: The XA Specification*(ISBN 1 872630 24 3)에 설명된 것처럼 XA 인터페이스를 사용하여 데이터베이스 관리자와

통신합니다. 이는 큐 관리자가 이 표준을 따르는 데이터베이스 관리자외도 통신할 수 있음을 의미합니다. 그러한 데이터베이스 관리자를 XA를 준수하는 데이터베이스 관리자라고도 합니다.

표5에서는 MQSeries Version 5.1 제품군에서 지원하는 XA 준수 데이터베이스 관리자를 설명합니다.

표 5. XA 준수 관계형 데이터베이스

MQSeries 제품	DB2	Oracle	Sybase
AIX용 MQSeries	예	예	예
HP-UX용 MQSeries	예	예	아니오
Sun Solaris용 MQSeries	예	예	예
Compaq OpenVMS Alpha용 MQSeries	아니오	예	아니오
Windows NT 및 Windows 2000용 MQSeries	예	아니오	예

제한사항

다음 제한사항은 데이터베이스 통합 지원에 적용됩니다.

- MQSeries 작업 단위내에서 데이터베이스 갱신사항을 통합할 수 있는 기능은 MQI 클라이언트 응용프로그램에서 지원되지 않습니다.
- MQI 갱신 및 데이터베이스 갱신은 같은 큐 관리자 서버 시스템에서 수행되어야 합니다.
- 데이터베이스 서버는 큐 관리자 서버와 다른 시스템에 상주할 수 있습니다. 이러한 경우, 데이터베이스는 데이터베이스 관리자에서 제공하는 XA 준수 클라이언트 기능을 통해 액세스해야 합니다.
- 큐 관리자가 XA를 준수해도, 전역 작업 단위에서 참여자로 다른 큐 관리자를 구성할 수 없습니다. 그 이유는 한번에 하나의 연결만 지원되기 때문입니다.

데이터베이스 연결

큐 관리자에 대한 표준 연결을 설정하는 응용프로그램은 별도의 로컬 큐 관리자 에이전트 프로세스에서 스레드와 연관됩니다. 응용프로그램이 MQBEGIN을 발행하면, 그 응용프로그램과 에이전트 프로세스는 작업 단위에 관련될 데이터베이스에 연결되어야 합니다. 데이터베이스 연결은 응용프로그램이 큐 관리자에 연결되어 있는 동안 유지됩니다. 이것은 데이터베이스가 제한된 수의 사용자 또는 연결을 지원할 경우 고려해야 할 중요한 사항입니다.

연결 수를 줄이는 한 가지 방법은 응용프로그램이 MQCONNX 호출을 사용하여 빠른 경로 바인딩을 요청하는 것입니다. 이 경우, 응용프로그램과 로컬 큐 관리자 에이전트는 같은 프로세스가 되어 결국 단일 데이터베이스 연결을 공유할 수 있습니다. 이를 수

행하기 전에, *MQSeries Application Programming Guide*에서 빠른 경로 응용프로그램에 적용할 제한사항 목록을 참조하십시오.

데이터베이스 관리자 구성

큐 관리자가 통합한 전역 작업 단위에 데이터베이스 관리자가 참여할 수 있으려면 먼저 몇 가지 작업을 수행해야 합니다.

1. 데이터베이스 관리자에 대한 XA 스위치 로드 파일을 작성하십시오.
2. 큐 관리자의 구성 파일인 `qm.ini`에서 데이터베이스 관리자를 정의하십시오.
스위치 로드 파일의 이름을 포함한 여러가지 항목을 `qm.ini`에서 정의해야 합니다.

스위치 로드 파일 작성

지원되는 데이터베이스 관리자의 스위치 로드 파일을 작성하기 위한 지시사항은 134 페이지의 『Oracle 스위치 로드 파일 작성』에서 제공합니다.

설치 프로시저에 대한 자세한 정보는 MQSeries 설치 문서를 참조하십시오.

스위치 로드 파일을 생성하기 위해 사용되는 샘플 소스 모듈에는 모두 MQStart라는 단일 함수가 있습니다. 스위치 로드 파일이 로드될 때, 큐 관리자에서 이 함수를 호출하면 XA 스위치라는 구조의 주소를 리턴합니다. 스위치 로드 파일은 데이터베이스 관리자가 제공하는 라이브러리에 링크되어 있는데, 이 라이브러리는 MQSeries가 해당 데이터베이스를 호출하도록 합니다.

Oracle용 스위치 로드 파일을 빌드하는데 사용하는 샘플 소스 모듈은 `oraswit.c`입니다.

데이터베이스 관리자 정의

데이터베이스 관리자에 대해 스위치 로드 파일을 작성하였으면, 큐 관리자에 대해 그 위치를 지정해야 합니다. 이것은 XAResourceManager 스탠자에서 큐 관리자의 `qm.ini` 파일에서 수행합니다.

큐 관리자가 통합하려고 하는 각 데이터베이스 관리자에 대해 XAResourceManager 스탠자를 추가해야 합니다.

XAResourceManager 스탠자의 속성은 다음과 같습니다.

Name=name

데이터베이스 관리자 인스턴스를 식별하는 사용자 선택 문자열.

이름은 필수로, 최대 31자까지 가능하며 고유해야 합니다. 이 이름은 더 복잡한 구성에서 그 고유성을 유지하기 위해 예를 들어 갱신되는 데이터베이스의 이름을 포함할 수도 있지만, 단순히 데이터베이스 관리자의 이름일 수 있습니다.

1. XA 스위치 로드 파일이란 동적으로 로드된 오브젝트로서 큐 관리자와 데이터베이스 관리자가 서로 통신할 수 있도록 합니다.

선택하는 이름은 큐 관리자가 **dspmqrn** 명령이 사용될 때 메시지나 출력에서 이 데이터베이스 관리자 인스턴스를 참조하기 위해 그 이름을 사용하기 때문에 의미가 있어야 합니다.

이름을 선택하였으면, 이 속성을 변경하지 마십시오. 구성 정보 변경에 대한 정보는 143 페이지의 『구성 정보 변경』을 참조하십시오.

SwitchFile=name

데이터베이스 관리자의 XA 스위치 로드 파일의 완전한 이름입니다. 이것은 필수 속성입니다.

XAOpenString=string

이것은 데이터베이스 매니저의 `xa_open` 입력 항목점에서 전달되는 데이터 문자열입니다. 이 문자열의 형식은 특정 데이터베이스 관리자에 따라 다르지만, 보통 갱신될 데이터베이스의 이름을 식별해야 합니다.

이것은 선택적 속성으로 이를 생략하면 공백 문자열로 간주됩니다.

XACloseString=string

이것은 데이터베이스 매니저의 `xa_close` 항목 점에서 전달되는 데이터 문자열입니다. 이 문자열의 형식은 특정 데이터베이스 관리자에 따라 다릅니다.

이것은 선택적 속성으로, 이를 생략하면 공백 문자열로 간주됩니다.

ThreadOfControl=THREAD|PROCESS

ThreadOfControl 값은 THREAD 또는 PROCESS가 될 수 있습니다. 큐 관리자는 이를 사용하여 일련화를 수행합니다.

데이터베이스 관리자가 “스레드 안전” 상태일 경우, *ThreadOfControl*의 값은 THREAD가 될 수 있고, 큐 관리자는 동시에 여러 스레드에서 데이터베이스 관리자를 호출할 수 있습니다.

데이터베이스 관리자가 스레드 안전 상태가 아니면, *ThreadOfControl*의 값은 PROCESS여야 합니다. 큐 관리자는 특정 프로세스 내에서 한번에 하나의 호출만 수행되도록 데이터베이스 관리자에 대한 모든 호출을 일련화합니다.

이 속성들에 대한 추가 설명은 195 페이지의 『XAResourceManager 스탠자』를 참조하십시오.

134 페이지의 『Oracle 구성』에서는 지원되는 각각의 데이터베이스 관리자로 MQSeries를 구성하기 위해 수행해야 하는 특정 작업들에 대한 자세한 정보를 제공합니다.

Oracle 구성

다음 작업을 수행해야 합니다.

- 아직 수행하지 않았으면 Oracle 레벨을 점검하고 패치를 적용하십시오.
- 환경 변수 설정을 점검하십시오.
- Oracle XA 지원을 사용 가능하도록 하십시오.
- Oracle 스위치 로드 파일을 작성하십시오.
- XAResourceManager 구성 정보를 `qm.ini` 파일에 추가하십시오.
- 필요하면 Oracle 구성 매개변수를 변경하십시오.

Oracle에 대한 최소 지원 레벨

OpenVMS에서의 Oracle에 대한 최소 지원 레벨은 8.1.6입니다.

환경 변수 설정 점검

Oracle 환경 변수는 응용프로그램 프로세스에서 뿐만 아니라 큐 관리자 프로세스에 대해서도 설정되어야 합니다. 특히, 다음 환경 변수는 항상 큐 관리자 시작 이전에 설정해야 합니다.

ORACLE_HOME

Oracle 홈 디렉토리입니다.

ORACLE_SID

사용할 Oracle SID입니다.

Oracle XA 지원 사용

Oracle XA 지원을 사용할 수 있는지 확인해야 합니다. 특히, Oracle 공유 라이브러리가 작성되어 있어야 합니다. Oracle 공유 라이브러리는 Oracle XA 라이브러리 설치시 작성됩니다.

Oracle8을 설치하는 동안 라이브러리가 자동으로 빌드됩니다. 라이브러리를 재빌드하려면, *Oracle8 Administrator's Reference*에서 사용자 플랫폼에 적절한 사항을 참조하십시오.

Oracle 스위치 로드 파일 작성

Oracle 스위치 로드 파일을 작성하는 가장 간단한 방법은 샘플 파일을 사용하는 것입니다. Oracle 스위치 로드 파일을 작성하기 위해 사용하는 소스 코드를 135 페이지의 그림11에서 볼 수 있습니다.


```

#include <cmqc.h>
#include "xa.h"

extern struct xa_switch_t xaosw;

struct xa_switch_t * MQENTRY MQStart(void)
{
    return(&xaosw);
}

```

그림 11. Oracle 스위치 로드 파일 oraswit.c에 대한 소스 코드

포함된 xa.h 헤더 파일은 oraswit.c와 같은 디렉토리에 위치하여 MQSeries와 함께 제공됩니다.

OpenVMS 시스템에서 Oracle 스위치 로드 파일 작성

OpenVMS 시스템에서 Oracle 스위치 로드 파일을 작성하려면, Oracle 클라이언트 라이브러리 "oraclient_v816.exe"에 oraswit.c를 링크하고 컴파일해야 합니다.

1. Oracle 스위치 로드 파일 oraswit를 빌드할 디렉토리를 작성하십시오.
2. mqs_examples:[xatm]에서 다음 파일들을 복사하여 위에서 작성한 디렉토리에 놓으십시오.
 - xa.h
 - oraswit.c
3. 복사한 소스 파일(oraswit.c)을 컴파일하십시오. 예를 들면 다음과 같습니다.

```
$ cc oraswit0.c
```

4. 스위치 로드 파일을 생성하십시오.

```

$ link/share oraswit0.obj, sys$input/options
ora_root:[util]oraclient_v816.exe/share
SYMBOL_VECTOR=(MQStart=PROCEDURE)

```

5. MQStart 프로시저는 생성된 이미지에서 런타임시 동적으로 로드됩니다. 그러므로, ".exe" 확장자 없이 생성된 파일을 지시하기 위해 논리 이름을 시스템 논리 이름 테이블에 정의하거나 그 파일을 sys\$share에 복사해야 합니다. 예를 들어 빌드된 파일의 위치가 disk\$a_device:[a_directory]oraswit0.exe일 경우, 다음 중 하나를 사용하십시오.

Oracle 구성

```
$ define/sys/exec oraswit0 disk$a_device:[a_directory]oraswit0
```

또는

```
$ copy disk$a_device:[a_directory]oraswit0.exe sys$share:oraswit0.  
exe
```

Oracle용 XAResourceManager 구성 정보 추가

다음 단계는 큐 관리자의 `qm.ini` 구성 파일을 수정하여 전역 작업 단위의 참여자로 Oracle을 정의하는 것입니다. 다음 속성을 갖는 XAResourceManager 스탠지를 추가해야 합니다.

Name=name

이 속성은 필수입니다. 이 참여자에 대해 적당한 이름을 선택하십시오. 갱신될 데이터베이스의 이름을 포함할 수 있습니다.

SwitchFile=name

이 속성은 필수입니다. Oracle 스위치 로드 파일의 완전한 이름입니다.

XAOpenString=string

Oracle용 XA 열기 문자열은 다음과 같은 형식을 갖습니다.

```
Oracle_XA+Acc=P//|P/userName/passWord
      +SesTm=sessionTimeLimit
      [+DB=dataBaseName]
      [+GPwd=P/groupPassWord]
      [+LogDir=logDir]
      [+MaxCur=maximumOpenCursors]
      [+SqlNet=connectString]
```

여기서

Acc= 필수이며 사용자 액세스 정보를 지정하는 데 사용됩니다. **P//**는 어떤 명확한 사용자 정보나 암호 정보가 제공되지 않았다는 것과 **ops\$login** 형식이 사용됨을 나타냅니다. **P/userName/passWord**는 올바른 ORACLE 사용자 ID와 해당 암호를 나타냅니다.

SesTm=

필수이며 시스템이 트랜잭션을 자동으로 삭제하기 전에 그 트랜잭션이 비활성 상태로 있을 수 있는 최대 시간을 지정하는 데 사용됩니다. 시간 단위는 초입니다.

DB= 데이터베이스 이름을 지정하는 데 사용됩니다. `DataBaseName`은 Oracle 사전 처리 컴파일러에서 데이터베이스를 식별하기 위해 사용하는 이름입니다. 이 필드는 응용프로그램이 명확하게 데이터베이스 이름을 지정할 경우에만 필요합니다(즉, SQL문에 AT 절을 사용하십시오).

GPwd=

GPwd는 서버 보안 암호를 지정하는 데 사용됩니다. **P/groupPassWord**는 서버 보안 그룹 암호 이름입니다. 서버 보안 그룹은 같은 ORACLE 인스턴스에 대해 실행 중인 다른 응용프로그램들에 대해 임시의 보안 레벨을 제공합니다. 디폴트는 ORACLE에서 정의한 서버 보안 그룹입니다.

LogDir=

LogDir은 Oracle XA 라이브러리 오류 및 추적 정보를 기록할 수 있는 로

Oracle 구성

컬 시스템에서 디렉토리를 지정하는 데 사용합니다. 값을 지정하지 않으면, 현재 디렉토리로 지정됩니다. 사용자 mqm이 이 디렉토리에 대해 쓰기 액세스 권한을 가져야 합니다.

MaxCur=

MaxCur은 데이터베이스가 열릴 때 할당될 커서 수를 지정하는 데 사용됩니다. 이것은 사전 처리 컴파일러 옵션인 maxopencursors와 같은 목적으로 사용됩니다.

SqlNet=

SqlNet은 시스템에 로그인하기 위해 사용되는 SQL*Net 연결 문자열을 지정하는 데 사용합니다. 연결 문자열은 SQL*Net V1 문자열, SQL*Net V2 문자열 또는 SQL*Net V2 별명이 될 수 있습니다. 이 필드는 큐 관리자와 분리된 시스템에서 Oracle을 설정할 때 필요합니다.

자세한 정보는 *Oracle8 Server Application Developer's Guide*(부품 번호 A54642-01)를 참조하십시오.

XACloseString=string

Oracle에는 XA 닫기 문자열이 필요하지 않습니다.

ThreadOfControl=THREAD|PROCESS

Compaq OpenVMS용 MQSeries 플랫폼에서는 이 매개변수를 지정하지 않아도 됩니다.

이 속성들 각각에 대한 자세한 설명은 195 페이지의 『XAResourceManager 스텐자』를 참조하십시오.

그림12에서, 갱신될 데이터베이스는 MQBankDB입니다. 모든 오류 정보 및 추적 정보가 같은 장소에 기록되도록 LogDir을 XA 열기 문자열에 추가하는 것이 좋습니다. 이것은 Oracle 스위치 로드 파일이 생성된 후 sys\$share 디렉토리에 복사되었음을 가정한 것입니다.

```
XAResourceManager:  
Name=Oracle MQBankDB  
SwitchFile=sys$share:oraswit0  
XAOpenString=Oracle_XA+Acc=P/jim/tiger+SesTm=35+LogDir=/tmp/ora.log+DB=MQBankDB
```

그림 12. Oracle용 샘플 XAResourceManager 항목

Oracle 구성 매개변수 변경

큐 관리자와 사용자 응용프로그램은 Oracle에 연결시 XA 열기 문자열에 지정된 사용자 ID를 사용합니다.

- 데이터베이스 특권 열기 문자열에 지정된 Oracle 사용자 ID는 DBA_PENDING_TRANSACTIONS 보기에 액세스할 수 있는 특권을 가지고 있어야 합니다.
필요한 특권은 다음 명령을 사용하여 얻을 수 있습니다. 여기서 userID는 액세스가 제공되는 사용자 ID입니다.

```
grant select on DBA_PENDING_TRANSACTIONS to userID;
```

보안에 대한 자세한 정보는 85 페이지의 『제7장 MQSeries 오브젝트 보호』를 참조하십시오.

관리 작업

일반적인 조작에서, 구성 단계를 완료한 후에는 최소한의 관리만 필요합니다. 큐 관리자는 데이터베이스 관리자의 사용 가능 여부에 관계없이 작업을 수행하므로, 관리 작업을 더욱 손쉽게 할 수 있습니다. 특히 이는 다음을 의미합니다.

- 각 데이터베이스 관리자를 먼저 시작하지 않고도 언제든지 큐 관리자를 시작할 수 있습니다.
- 데이터베이스 관리자 중 하나가 사용 불가능하게 될 경우에도 큐 관리자를 정지하여 재시작하지 않아도 됩니다.

이와 같이, 큐 관리자는 데이터베이스 관리자와는 독립적으로 시작 및 정지할 수 있으며 데이터베이스 관리자가 이를 지원할 경우 그 반대 경우도 가능합니다.

큐 관리자와 데이터베이스 관리자 사이에 접속이 끊겼을 경우에는, 두 관리자가 다시 사용 가능하게 될 시점에 재동기화를 수행해야 합니다.

재동기화는 데이터베이스에 관련된 인다우트(in-doubt) 작업 단위가 완료되는 프로세스입니다. 일반적으로, 재동기화는 사용자가 개입하지 않아도 자동으로 발생합니다. 큐 관리자는 데이터베이스 관리자에게 인다우트(in-doubt) 상태에 있는 작업 단위 목록을 요청합니다. 그러면 큐 관리자는 데이터베이스 관리자가 이러한 각각의 인다우트(in-doubt) 작업 단위를 확약 또는 롤백하도록 지시합니다.

큐 관리자가 정지하면, 재시작할 때 각 데이터베이스 관리자 인스턴스와 재동기화를 수행해야 합니다. 하나의 개별 데이터베이스 관리자가 사용 불가능하게 될 경우, 큐 관리자에서 해당 데이터베이스 관리자의 가용성을 인식하면 그 데이터베이스 관리자를 다시 동기화해야 합니다.

큐 관리자는 새로운 전역 작업 단위가 시작하면, 자동으로 사용 불가능한 데이터베이스 관리자에 접속을 시도합니다. 또한, 모든 인다우트(in-doubt) 작업 단위를 명확하게 해석하기 위해 **rsvmqtrn** 명령을 사용할 수 있습니다.

인다우트(in-doubt) 작업 단위

데이터베이스 관리자는 PREPARE 명령을 받은 후 큐 관리자와의 접속이 끊길 경우 인다우트(in-doubt) 작업 단위들과 함께 남아 있을 수 있습니다. 데이터베이스 관리자는 큐 관리자로부터 COMMIT 또는 ROLLBACK 결과를 수신할 때까지, 갱신사항과 연관되는 데이터베이스 잠금을 보유해야 합니다.

이러한 잠금으로 다른 응용프로그램이 데이터베이스 레코드를 갱신하지 못하거나 읽지 못할 수 있으므로, 가능하면 빨리 재동기화를 수행해야 합니다.

어떠한 이유로 큐 관리자가 자동으로 데이터베이스를 재동기화할 때까지 기다릴 수 없는 경우에는, 데이터베이스 관리자가 제공하는 기능을 사용하여 수동으로 데이터베이스 갱신사항을 확약하거나 롤백할 수 있습니다. 이것을 수동 결정이리는데, 데이터 무결성을 손상시킬 가능성이 있으므로 마지막 방법으로 사용해야 합니다. 다른 모든 참여자가 롤백하면 데이터베이스 갱신사항이 확약될 수 있으며, 그 반대도 가능합니다.

그러나, 데이터베이스를 재시작할 때 큐 관리자를 재시작하거나 **rsvmqtrn** 명령을 사용하여 자동으로 재동기화를 시작하는 것이 더 좋습니다.

dspmqtrn 명령 사용

데이터베이스 관리자가 사용 불가능할 때, **dspmqtrn** 명령을 사용하여 그 데이터베이스와 관련되는 미해결 작업 단위(UOW) 상태를 점검할 수 있습니다.

데이터베이스 관리자가 사용 불가능하게 되면, 2단계 확약 프로세스에 들어가기 전에 참여하고 있던 모든 처리 UOW가 롤백됩니다. 데이터베이스는 다음에 재시작할 때 해당 처리 UOW를 롤백합니다.

dspmqtrn 명령은 하나 이상의 참여자가 인다우트(in-doubt) 상태인 작업을 표시하고 큐 관리자의 COMMIT 또는 ROLLBACK을 대기합니다.

각 작업 단위의 참여자 상태가 각각 표시됩니다. 작업 단위가 특정 자원 관리자의 자원을 갱신하지 않았을 경우에는 표시되지 않습니다.

인다우트(in-doubt) 작업 단위에 관련하여, 자원 관리자는 다음 중 하나를 수행했다고 볼 수 있습니다.

- 준비 자원 관리자가 갱신사항을 확약할 준비가 되어 있습니다.
- 확약 자원 관리자가 갱신사항을 확약했습니다.
- 롤백 자원 관리자가 갱신사항을 롤백했습니다.
- 참여 자원 관리자가 참여했지만, 갱신사항을 준비, 확약 또는 롤백하지 않았습니다.

큐 관리자는 재시작시 참여자의 개별 상태를 기억하지 않습니다. 큐 관리자를 재시작했지만 데이터베이스 관리자에 접속할 수 없는 경우, 해당 데이터베이스 관리자가 참여하

고 있던 인다우트(in-doubt) 작업 단위는 재시작시 해석되지 않습니다. 이러한 경우에, 데이터베이스 관리자는 재동기화와 같은 경우가 있을 때까지 준비 상태로 보고됩니다.

dspmqtrn 명령은 인다우트(in-doubt) UOW를 표시할 때 먼저 참여할 수 있는 모든 가능한 자원 관리자를 나열합니다. 이 관리자들에게는 고유한 ID인 *RMID*가 할당되는데, 이 ID는 인다우트(in-doubt) UOW에 관한 자원 관리자의 상태를 보고할 때 자원 관리자의 이름 대신 사용됩니다.

그림13은 다음 명령을 실행했을 때의 결과를 보여줍니다.

```
dspmqtrn -m MY_QMGR
```

```
AMQ7107: Resource manager 0 is MQSeries.
AMQ7107: Resource manager 1 is Oracle MQBankDB
AMQ7107: Resource manager 2 is Oracle MQFeeDB

AMQ7056: Transaction number 0,1.
      XID: formatID 5067085, gtrid_length 12, bqual_length 4
          gtrid [3291A5060000201374657374]
          bqual [00000001]
AMQ7105: Resource manager 0 has committed.
AMQ7104: Resource manager 1 has prepared.
AMQ7104: Resource manager 2 has prepared.
```

그림 13. Sample dspmqtrn output

그림13의 출력은 큐 관리자와 연관된 세 개의 자원 관리자가 있음을 보여줍니다. 첫 번째는 자원 관리자 0으로, 이는 큐 관리자 자체입니다. 다른 두 개의 자원 관리자 인스턴스는 MQBankDB와 MQFeeDB Oracle 데이터베이스입니다.

예에서는 단일 인다우트(in-doubt) 작업 단위만 보여줍니다. 메시지는 세 개의 모든 자원 관리자에 대해 발행됩니다. 이는 큐 관리자와 작업 단위 내의 두 Oracle 데이터베이스에 대해 갱신이 수행되었음을 의미합니다.

큐 관리자, 자원 관리자 0에 대한 갱신이 확약되었습니다. Oracle 데이터베이스에 대한 갱신은 준비 상태에 있는데, 이는 Oracle이 MQBankDB와 MQFeeDB 데이터베이스에 대한 갱신을 확약하기 위해 호출되기 전에 사용 불가능하게 되었음을 의미합니다.

인다우트(in-doubt) 작업 단위에는 XID라고 하는 외부 ID가 있습니다. 이것은 Oracle에서 갱신을 연관시키는 ID입니다.

rsvmqtrn 명령 사용

그림13에 표시된 출력은 두 개의 Oracle 데이터베이스에 아직 확약 결정이 전달되지 않은 단일 인다우트(in-doubt) UOW를 보여줍니다.

관리 작업

이 작업 단위를 완료하려면, Oracle이 다음에 사용 가능하게 될 때 큐 관리자와 Oracle을 재동기화해야 합니다. 큐 관리자는 새로운 작업 단위가 시작될 때 Oracle과의 재접속을 시도합니다. 또는, `rsvmqtrn` 명령을 사용하여 큐 관리자가 명확하게 재동기화를 수행하도록 할 수 있습니다. 인다우트(in-doubt) 작업 단위와 연관된 데이터베이스가 가능한 빠르게 해제될 수 있도록 Oracle이 재시작 된 후 바로 이를 수행해야 합니다.

이것은 큐 관리자가 모든 인다우트(in-doubt) 작업 단위를 해석하도록 지시하는 `-a` 옵션을 사용하여 이루어집니다. 다음 예에서, Oracle이 재시작되어 큐 관리자가 인다우트(in-doubt) 작업 단위를 해석할 수 있었습니다.

```
rsvmqtrn -m MY_QMGR -a
```

모든 인다우트(in-doubt) 트랜잭션이 해석되었습니다.

혼합 결과 및 오류

큐 관리자가 2단계 확약 프로토콜을 사용하더라도 일부 작업 단위가 혼합 결과로 완료될 수 있습니다. 이는 일부 참여자는 갱신사항을 확약하고 일부는 백아웃한 경우입니다.

혼합 결과로 완료되는 작업 단위는 공유 자원이 일관적인 상태에 있지 않다는 면에서 중요한 의미를 갖습니다.

혼합 결과는 보통 큐 관리자에서 인다우트(in-doubt) 작업 단위를 직접 해석하도록 허용하는 대신 작업 단위에 대해 수동 결정을 할 경우에 발생합니다.

큐 관리자는 수동 결정에 의한 손상을 감지할 때마다 FFST 정보를 생성하고 다음의 두 메시지 중 하나를 사용하여 오류 로그에 실패를 기록합니다.

- 데이터베이스 관리자가 확약 대신 롤백한 경우:

```
AMQ7606 A transaction has been committed but one or more resource managers have rolled back.
```

- 데이터베이스 관리자가 롤백 대신 확약한 경우:

```
AMQ7607 A transaction has been rolled back but one or more resource managers have committed.
```

수동 결정으로 손상된 데이터베이스를 식별하는 추가 메시지가 발행됩니다. 이제 일관성이 복구되도록 사용자는 손상된 데이터베이스에 대해 로컬로 복구 단계를 수행해야

합니다. 손상된 데이터베이스의 복구는, 먼저 잘못 파악되거나 롤백된 갱신사항을 판별한 후에 수동으로 데이터베이스 변경을 취소하거나 다시 실행해야 하는 복잡한 프로시저입니다.

소프트웨어 오류로 손상이 발생할 가능성은 적습니다. AMQ7112 메시지는 이런 문제로 인해 손상된 작업 단위의 트랜잭션 번호를 보고합니다. 참여자가 일관성이 없는 상태에 있을 수 있습니다.

```
rsvmqtrn -m MY_QMGR

AMQ7107: Resource manager 0 is MQSeries.
AMQ7107: Resource manager 1 is Oracle MQBankDB
AMQ7107: Resource manager 2 is Oracle MQFeedB

AMQ7112: Transaction number 0,1 has encountered an error.
        XID: formatID 5067085, gtrid_length 12, bqual_length 4
        gtrid [3291A5060000201374657374]
        bqual [00000001]
AMQ7105: Resource manager 0 has committed.
AMQ7104: Resource manager 1 has prepared.
AMQ7104: Resource manager 2 has rolled back.
```

그림 14. 오류가 있는 트랜잭션에 대한 *dspmqrn* 출력 샘플

큐 관리자는 다음에 재시작할 때까지 그러한 실패에 대한 복구를 시도하지 않습니다. 그림14의 예는, 작업 단위를 해석하기 위해 **rsvmqtrn**이 발행되었지만 큐 관리자 1인 MQBankDB 데이터베이스에 대한 갱신사항이 준비 상태에 있음을 의미합니다.

구성 정보 변경

큐 관리자가 전역 작업 단위의 통합을 성공적으로 시작한 후에는, *qm.ini* 파일에 있는 모든 *XAResourceManager* 스탠자에 변경사항을 작성할 때는 주의해야 합니다.

필요한 경우 언제나 *qm.ini* 파일을 변경할 수는 있지만, 큐 관리자가 재시작될 때까지 변경사항은 적용되지 않습니다. 예를 들어, 데이터베이스 관리자로 전달된 XA 열기 문자열을 변경해야 할 경우, 변경사항이 적용되도록 하려면 큐 관리자를 재시작해야 합니다.

XAResourceManager 스탠자를 제거할 경우, 큐 관리자가 해당 데이터베이스 관리자에 접속할 수 있는 기능도 제거된다는 점에 주의해야 합니다.

XAResourceManager 스탠자에서 이름 속성은 절대 변경하지 마십시오. 이 속성은 큐 관리자에 대해 데이터베이스 관리자 인스턴스를 고유하게 식별합니다. 이 고유 ID가 변경될 경우, 큐 관리자는 데이터베이스 관리자 인스턴스가 제거되었고 완전히 새로운 인스턴스가 추가되었다고 가정합니다. 큐 관리자는 데이터베이스를 인다우트(in-doubt) 상태로 둔 채 미해결 작업 단위를 계속 이전 이름과 연관시킵니다.

데이터베이스 관리자 인스턴스 제거

구성에서 영구적으로 데이터베이스나 데이터베이스 관리자를 제거하려면, 먼저 데이터베이스가 인다우트(in-doubt) 상태가 아닌지 확인해야 합니다. 큐 관리자를 재시작하기 전에 이 점검을 수행해야 합니다. 대부분의 데이터베이스 관리자는 인다우트(in-doubt) 트랜잭션을 나열하기 위한 명령을 제공합니다. 인다우트(in-doubt) 트랜잭션이 없는 경우 해당 XAResourceManager 스탠자를 제거하기 전에 큐 관리자가 데이터베이스 관리자와 재동기화할 수 있도록 해야 합니다.

이 프로시저를 수행하지 못하면, 큐 관리자는 해당 데이터베이스가 관련된 모든 인다우트(in-doubt) 작업 단위를 계속 기억합니다. 큐 관리자가 재시작할 때마다 경고 메시지(AMQ7623)가 발행됩니다. 이 큐 관리자를 사용하여 해당 데이터베이스를 다시 구성할 예정이 아니라면, **rsvmqtrn** 명령의 **-r** 옵션을 사용하여 큐 관리자가 인다우트(in-doubt) 트랜잭션에서의 해당 데이터베이스 참여를 잊도록 지시할 수 있습니다.

주: 큐 관리자는 동기점 처리가 모든 참여자에 대해 완료될 때만 최종적으로 트랜잭션에 대해 잊게 됩니다.

임시로 XAResourceManager 스탠자를 제거해야 할 경우가 있습니다. 이런 경우에는 나중에 쉽게 복구할 수 있도록 그 스탠자를 주석으로 처리하는 것이 좋습니다. 큐 관리자가 특정 데이터베이스나 데이터베이스 관리자와 접속할 때마다 오류가 발생할 경우에 이 조치를 사용할 수 있습니다. 연관된 XAResourceManager 항목을 제거하면 큐 관리자가 다른 모든 참여자를 포함하는 전역 작업 단위를 시작할 수 있습니다. 다음은 XAResourceManager 스탠자를 주석으로 처리한 예입니다.

```
# This database has been temporarily removed
#XAResourceManager:
# Name=Oracle MQBankDB
# SwitchFile=sys$share:oraswit0
# XAOpenString=MQBankDB
```

그림 15. 주석 처리된 XAResourceManager 스탠자

제11장 복구 및 재시작

메시징 시스템은 시스템에 입력된 메시지들이 해당 목적지로 배달되도록 합니다. 따라서 시스템에서 메시지를 추적하는 방법과 시스템이 어떤 이유로 실패할 경우 메시지를 복구하는 방법을 제공해야 합니다.

MQSeries는 메시지의 수신, 전송 및 전달을 핸들링하는 큐 관리자의 활동에 대한 레코드(로그)를 유지보수하여 메시지가 손실되지 않도록 합니다. 이러한 로그를 사용하여 다음과 같은 세 가지 유형의 복구를 수행합니다.

1. 재시작 복구: 계획된 방식으로 MQSeries를 정지할 때 사용합니다.
2. 손상 복구: 예상치 못한 실패로 MQSeries가 정지된 경우에 사용합니다.
3. 매체 복원: 손상된 오브젝트를 복원하기 위해 사용합니다.

모든 경우, 복구 작업은 큐 관리자가 정지될 때의 상태로 복원합니다. 처리 중인 트랜잭션은 롤백되어, 큐 관리자가 정지할 때 확약되지 않은 메시지들은 큐에서 제거됩니다. 복구 작업은 모든 지속 메시지를 복원하며, 비지속 메시지는 프로세스 중에 손실됩니다.

이 장의 나머지 부분에서는 복구 및 재시작의 개념에 대해 자세히 소개하고 문제점이 발생할 경우 복구하는 방법을 알려줍니다. 다음과 같은 주제를 설명합니다.

- 『메시지 손실 여부 확인(로그 기록)』
- 148 페이지의 『체크포인팅 - 완전한 복구』
- 151 페이지의 『로그 크기 계산』
- 152 페이지의 『로그 관리』
- 155 페이지의 『복구에 로그 사용』
- 157 페이지의 『MQSeries 로그 파일 보호』
- 158 페이지의 『백업 및 복원』
- 159 페이지의 『복구 시나리오』
- 161 페이지의 『dmpmqlog 명령을 사용하여 로그의 내용 덤프』.

메시지 손실 여부 확인(로그 기록)

MQSeries는 중요한 모든 변경사항을 큐 관리자가 제어하는 데이터로 로그에 기록합니다. 여기에는 오브젝트의 작성 및 삭제, 모든 지속 메시지 갱신, 트랜잭션 상태, 오브젝트 속성에 대한 변경사항 및 채널 활동이 포함됩니다. 따라서 다음을 수행하여 메시지 큐에 대한 모든 갱신사항을 복구하는데 필요한 정보를 포함합니다.

- 큐 관리자 변경사항 레코드를 보존합니다.
- 재시작 프로세스에서 사용할 큐 갱신 레코드를 보존합니다.
- 하드웨어 및 소프트웨어 실패 후에 데이터를 복원할 수 있게 합니다.

로그 형식

MQSeries는 다음과 같은 두 가지 구성요소로 구성됩니다.

1. 하나 이상의 로그 데이터 파일
2. 로그 제어 파일

데이터가 기록되는 로그 파일은 여러 개가 있습니다. 151 페이지의 『로그 크기 계산』에 설명된 것처럼 파일 개수와 크기를 정의하거나, 시스템 디폴트로 지정된 세 개의 파일(각각 4 MB)을 사용할 수 있습니다.

큐 관리자를 작성할 때, 정의하는 로그 파일 수는 할당된 1차 로그 파일 수입니다. 수를 지정하지 않으면, 디폴트 값이 사용됩니다. 로그 경로를 변경하지 않은 경우에는 다음 디렉토리에 작성됩니다.

MQS_ROOT: [MQM.LOG.QmName.ACTIVE]

MQSeries는 이 1차 로그 파일에서 시작하지만, 로그가 가득차기 시작하면 2차 로그 파일을 할당합니다. 이것은 동적으로 수행되며 로그 공간의 필요성이 줄어들면 2차 로그 파일을 제거합니다. 디폴트로 두 개의 2차 로그 파일을 할당하여 8 MB의 추가 디스크 공간을 제공합니다. 디폴트 값을 변경할 수 있습니다. 183 페이지의 『제13장 MQSeries 구성』을 참조하십시오.

로그 제어 파일

로그 제어 파일에는 로그 파일의 사용을 모니터링하는 데 필요한 정보(크기 및 위치, 다음 사용 가능한 파일의 이름 등)가 있습니다.

주: 큐 관리자를 시작할 때 작성된 로그의 크기는 응용프로그램이 핸들링할 메시지들의 크기와 볼륨을 수용하기에 충분해야 합니다. 디폴트 로그 수와 크기를 요구사항에 맞게 수정하십시오. 디폴트 값을 변경하는 방법은 151 페이지에 설명되어 있습니다.

로그 기록 유형

MQSeries에서 로그 기록에 사용되는 파일 수는 파일 크기, 수신한 메시지 수 및 메시지 길이에 따라 다릅니다. 큐 관리자 활동 레코드를 유지보수하는 두 가지 방법으로는 순환 로깅과 선형 로깅이 있습니다.

순환 로깅

시스템이 정지했을 때 처리 중이었던 트랜잭션을 롤백하기 위해 로그를 사용하여 복구를 재시작하려면 순환 로깅을 사용하십시오.

순환 로깅은 모든 재시작 데이터를 일련의 로그 파일에 보존합니다. 로그 기록은 첫 번째 파일부터 시작해서 모든 파일에 순서대로 채워집니다. 그리고 나서, 첫 번째 파일로 돌아가서 다시 시작됩니다. 제품을 사용하는 동안 이 작업이 계속되므로 로그 파일은 부족하지 않습니다.

위는 순환 로깅에 대한 간단한 설명입니다. 그러나 주의할 점이 한 가지 있습니다. 데이터 손실 없이 큐 관리자를 재시작하는 데 필요한 로그 항목은 큐 관리자 데이터 복구가 완료되어 더이상 필요하지 않을 때까지 보존됩니다. 로그 파일을 다시 사용하기 위해 해제하는 메커니즘은 148 페이지의 『체크포인팅 - 완전한 복구』에 설명되어 있습니다. 따라서 MQSeries가 필요할 경우 로그 용량을 확장하기 위해 2차 로그 파일을 사용한다는 점을 알아야 합니다.

선형 로깅

재시작 복구와 매체 또는 정방향 복구를 모두 수행하려면 선형 로깅을 사용하십시오(로그 내용을 재생하여 손실되거나 손상된 데이터를 다시 작성합니다).

선형 로깅은 연속되는 일련의 파일에 로그 데이터를 보존합니다. 공간이 다시 사용되지 않으므로, 큐 관리자가 작성된 이후에 기록된 레코드를 항상 검색할 수 있습니다.

디스크 공간은 제한되어 있으므로, 일부 보존 양식을 고려해야 합니다. 이것은 로그에 대한 디스크 공간을 관리하여 필요에 따라 기존 공간을 재사용하거나 확장하는 관리 작업입니다.

선형 로깅에 사용되는 로그 파일은 메시지 플로우나 큐 관리자 경과 시간에 따라 많아질 수 있습니다. 그러나 이 중에서 활성화되는 파일이 몇 가지 있습니다. 활성 파일에는 큐 관리자를 재시작하는 데 필요한 로그 항목이 있습니다. 일반적으로 활성 로그 파일 수는 구성 파일에 정의된 1차 로그 파일 수와 동일합니다(로그 파일 수를 정의하는 방법에 대한 자세한 내용은 151 페이지의 『로그 크기 계산』을 참조하십시오).

로그 파일의 활성 여부를 제어하는 키 이벤트는 체크포인트입니다. MQSeries 체크포인트는 큐 관리자를 성공적으로 재시작할 수 있는 정보가 포함된 일련의 로그 레코드입니다. 이전에 기록된 정보는 큐 관리자를 재시작하는 데 필요하지 않으므로 비활성 상태라고 할 수 있습니다(체크포인팅에 대한 추가 정보는 148 페이지의 『체크포인팅 - 완전한 복구』를 참조하십시오).

비활성 로그 파일이 언제까지 필요한지 결정해야 합니다. 더이상 조작에 필요하지 않으면 비활성 로그 파일을 보존하거나 삭제할 수 있습니다. 로그 파일 처리에 대한 추가 정보는 152 페이지의 『로그 관리』를 참조하십시오.

2차 로그 파일에 새 체크포인트가 기록되거나, 나중에 1차 로그 파일에 새 체크포인트가 기록되면, 첫 번째 파일은 비활성화되고 새로운 1차 파일이 형식화되어 1차 풀의 끝에 추가됩니다. 이때, 로그 기록에 사용할 수 있는 1차 파일들이 복원됩니다. 여기서 1차 로그 파일 풀은 확장된 로그 파일 목록에서 현재 파일 세트로 표시될 수 있습니다. 앞에 설명한 것처럼, 이것은 조작 요구사항에 따라 비활성 파일을 관리하는 작업입니다.

2차 로그 파일이 선형 로그에 대해 정의되어 있어도 정상적인 조작에서는 사용되지 않습니다. 트랜잭션이 장기화되면 재시작을 위해 로그 파일이 계속 필요하므로 활성 풀에서 파일을 해제할 수 없습니다. 이런 경우에는 2차 파일이 형식화되어 활성 로그 파일 풀에 추가됩니다.

사용 가능한 2차 파일 수가 한계에 이르면, 로그 활동이 필요한 대부분의 추가 조작 요청은 응용프로그램에 리턴되는 MQRC_RESOURCE_PROBLEM으로 거부됩니다.

하드웨어 실패가 없을 경우, 두 유형의 로그 기록을 통해 예상치 못한 전원 손실에 대비할 수 있습니다.

체크포인팅 - 완전한 복구

메시지 큐에 대한 지속적 갱신은 두 단계에서 발생합니다. 먼저 갱신을 나타내는 레코드가 로그에 기록된 후 큐 파일이 갱신됩니다. 따라서 로그 파일은 큐 파일보다 더 최근의 파일이 될 수 있습니다. MQSeries는 재시작 처리가 일관된 지점에서 시작될 수 있도록 체크포인트를 사용합니다. 체크포인트는 로그에 설명된 레코드가 큐에 있는 레코드와 같을 때의 시점입니다. 체크포인트 자체는 큐 관리자를 재시작하기 위해 필요한 일련의 로그 레코드로 구성됩니다. 예를 들어, 모든 트랜잭션(즉, 작업 단위) 상태는 체크포인트에서 활성 상태입니다.

체크포인트는 MQSeries에서 자동으로 생성됩니다. 큐 관리자를 시작하거나 종료할 때, 로그 기록 공간이 부족할 때, 1000개의 조작이 기록될 때마다 체크포인트가 사용됩니다. 큐는 추가 메시지들을 핸들링하므로, 체크포인트 레코드는 현재 큐 상태와 일관되지 않습니다.

MQSeries는 재시작할 때 로그에서 최근 체크포인트 레코드를 찾습니다. 이 정보는 모든 체크포인트 끝에서 갱신되는 체크포인트 파일에 있습니다. 체크포인트 레코드는 로그와 데이터 가장 최근의 일관된 지점을 나타냅니다. 이 체크포인트의 데이터는 큐를 체크포인트시의 상태로 다시 빌드하는 데 사용됩니다. 큐가 재작성될 때 시스템에 장애가 있거나 다운되기 전의 상태로 돌아가도록 로그가 재생됩니다.

MQSeries는 로그의 헤드와 테일에 대한 내부 포인터를 유지보수합니다. 헤드 포인터는 복구하는 메시지 데이터와 일치하는 최근 체크포인트로 이동합니다.

체크포인트는 복구 작업을 효율적으로 수행하고 1차 및 2차 로그 파일의 재사용을 제어하기 위해 사용됩니다.

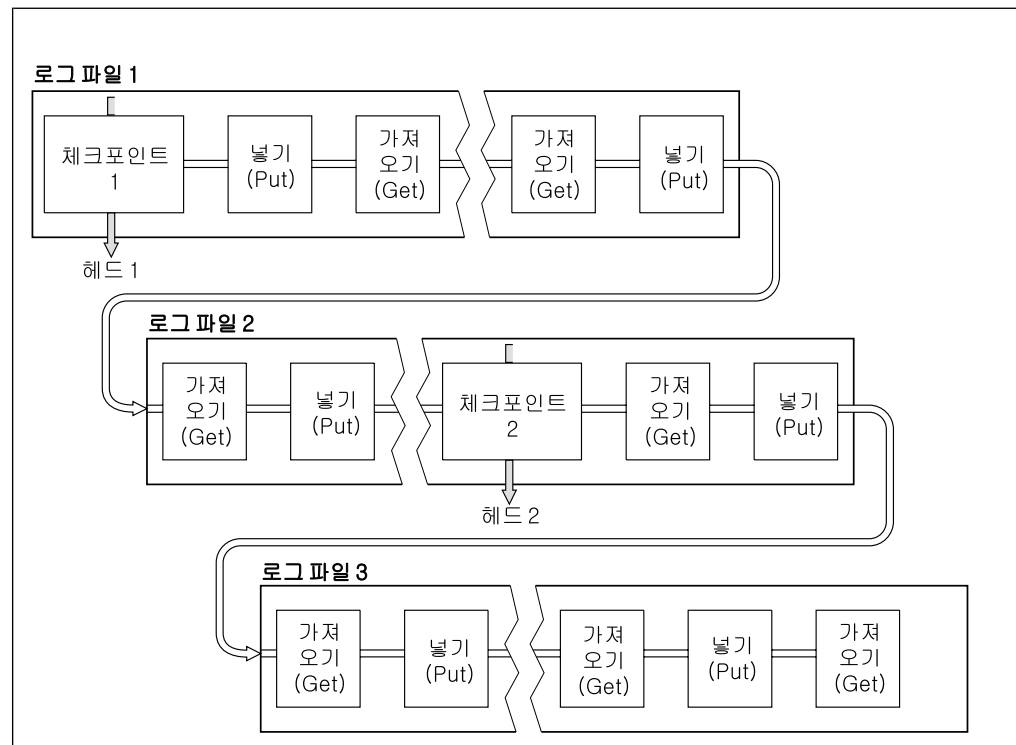


그림 16. 체크포인트링. 간단하게 로그 파일 끝만 표시합니다.

그림16에서, 최근 체크포인트인 체크포인트 2 이전의 모든 레코드는 MQSeries에서 더 이상 필요하지 않습니다. 큐는 체크포인트 정보와 이후의 로그 항목에서 복구될 수 있습니다. 순환 로깅의 경우, 체크포인트 이전에 비워진 파일들은 다시 사용할 수 있습니다. 선형 로깅의 경우, 비워진 로그 파일은 정상 조작시 액세스할 필요가 없으므로 비활성 상태가 됩니다. 예에서 큐 헤드 포인터는 최근 체크포인트인 체크포인트 2 지점으로 이동하고, 이 지점은 새로운 큐 헤드인 헤드 2가 됩니다. 로그 파일 1은 이제 다시 사용할 수 있습니다.

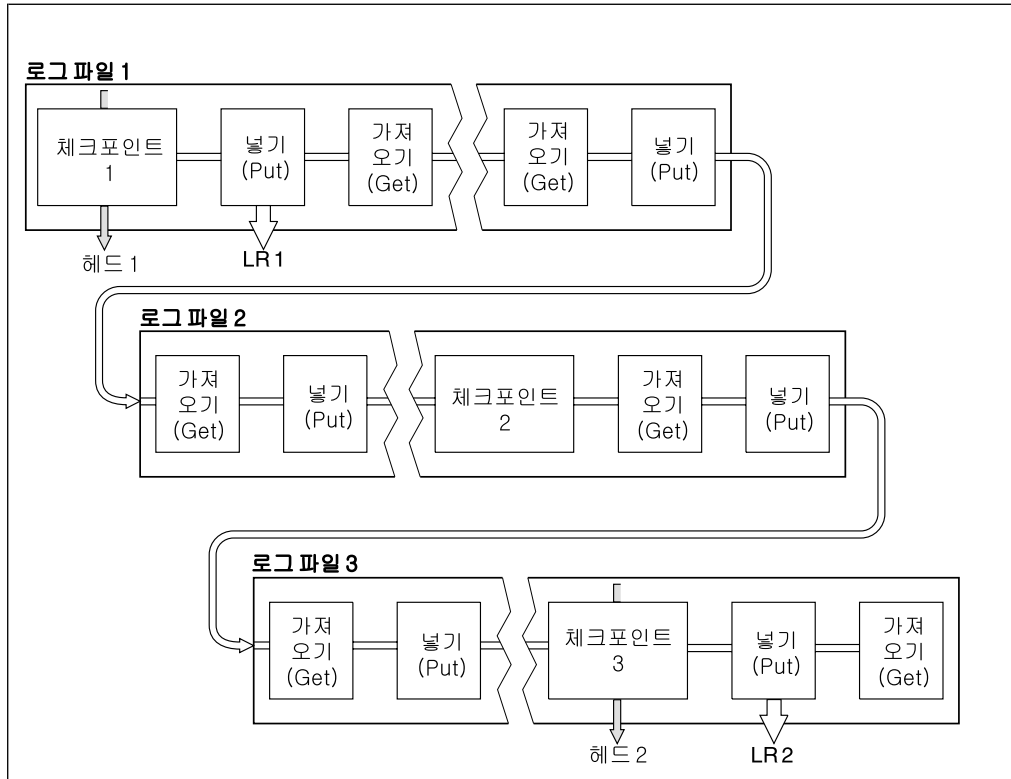


그림 17. 장기 실행 트랜잭션에서의 체크포인팅. 간단하게 로그 파일 끝만 표시됩니다.

그림17은 장기 실행 트랜잭션이 로그 파일의 재사용에 주는 영향을 표시합니다. 예에서 장기 실행 트랜잭션의 결과, 첫 번째 체크포인트가 표시된 후에 LR 1으로 표시된 로그에 입력 항목이 발생했습니다. LR 2로 표시된 트랜잭션은 세 번째 체크포인트 이후까지 완료되지 않습니다. LR 1 이후의 모든 로그 정보는 해당 트랜잭션의 복구를 위해 보유됩니다(필요할 경우, 복구를 완료할 때까지).

장기 실행 트랜잭션이 완료되면, LR 2에서 로그의 헤드는 최근에 기록된 체크포인트인 체크포인트 3으로 이동합니다. 체크포인트 3, 헤드 2 이전의 로그 레코드를 포함하는 파일들은 더이상 필요하지 않습니다. 순환 로깅을 사용 중이면, 공간을 다시 사용할 수 있습니다.

장기 실행 트랜잭션이 완료되기 전에 1차 로그 파일이 완전히 채워질 경우, 가능하면 로그가 가득 차지 않도록 2차 로그 파일이 사용됩니다.

로그 헤드가 이동할 때 순환 로깅을 사용 중이면, 1차 로그 파일을 재사용할 수 있게 되고 로거는 사용 가능한 첫 번째 1차 파일을 다시 사용합니다. 선형 로깅을 사용 중이면, 로그 헤드는 계속 활성 풀 아래로 이동하고 첫 번째 파일이 비활성 상태가 됩니다. 새로운 1차 파일이 형식화된 후 풀의 맨 아래에 추가되어 차후 로그 기록 활동에 사용됩니다.

로그 크기 계산

큐 관리자가 순환 로깅과 선형 로깅 중 사용할 방법을 결정하고 난 후에는 큐 관리자에 필요한 로그의 크기를 계산해야 합니다. 로그의 크기는 다음 로그 구성에 따라 결정됩니다.

LogFilePages

각 1차 및 2차 로그 파일의 크기(4 KB 페이지 단위)

LogPrimaryFiles

사전에 할당된 1차 로그 파일 수

LogSecondaryFiles

1차 로그 파일이 가득찰 때 작성하여 사용할 수 있는 2차 로그 파일 수.

표6에서는 다양한 조작에 대해 큐 관리자가 기록하는 데이터 양을 보여줍니다. 큐 관리자가 수행하는 대부분의 조작은 최소량의 로그 공간을 요구하지만, 지속 메시지를 큐에 넣을(put) 때 메시지를 복구할 수 있도록 모든 메시지 데이터가 로그에 기록되어야 합니다. 따라서 로그의 크기는 일반적으로 큐 관리자가 핸들링해야 하는 지속 메시지 수와 크기에 따라 다릅니다.

표 6. 로그 오버헤드 크기(모든 값은 근사값임)

조작	크기
지속 메시지 넣기(put)	750 바이트 + 메시지 길이. 메시지가 크면, 각각 300 바이트의 오버헤드를 가진 15700 바이트 세그먼트로 나누어집니다.
메시지 가져오기(get)	260 바이트
동기점, 확약	750 바이트
동기점, 롤백	1000 바이트 + 롤백할 가져오기(get) 또는 넣기(put)에 대해 각각 12 바이트
오브젝트 작성	1500 바이트
오브젝트 삭제	300 바이트
속성 대체	1024 바이트
매체 이미지 레코드	800 바이트 + 이미지. 이미지는 각각 300 바이트의 오버헤드를 가진 15700 바이트 세그먼트로 나누어 집니다.
체크포인트	750 바이트 + 각 활성 작업 단위마다 200 바이트. 성능상의 이유로 버퍼링된 미확약 넣기(put) 또는 가져오기(get)에 대해 추가 데이터를 기록할 수도 있습니다.

주:

- 1차 및 2차 로그 파일 수는 큐 관리자를 시작할 때마다 변경될 수 있습니다.
- 로그 파일 크기는 변경될 수 없으므로 큐 관리자를 작성하기 전에 판별해야 합니다.

3. 1차 로그 파일 수와 로그 파일 크기는 큐 관리자를 작성할 때 사전에 할당된 로그 공간의 양을 판별합니다. 이 공간을 구성할 때, 작은 로그 파일을 여러 개 구성하는 것보다는 큰 로그 파일을 몇 개 구성하는 것이 좋습니다.
4. 1차 및 2차 로그 파일 수의 합계는 63개 이하여야 합니다. 이는 장기 실행 트랜잭션에서처럼 재시작 복구에 대해 큐 관리자가 사용할 수 있는 로그 공간의 최대 양을 제한합니다. 큐 관리자가 매체 복원을 위해 사용할 수 있는 로그 공간에는 이 한계가 적용되지 않습니다.
5. 순환 로깅이 사용될 경우, 큐 관리자는 1차 로그 공간을 다시 사용합니다. 이는 큐 관리자가 기록해야 하는 데이터 양보다 큐 관리자의 로그가 작을 수 있음을 의미합니다. 로그 파일이 가득 차면 큐 관리자는 한계에 이를 때까지 2차 로그 파일을 할당하므로 다음 1차 로그 파일은 사용할 수 없게 됩니다.
6. 1차 로그 파일은 체크포인트 중에 다시 사용할 수 있습니다. 큐 관리자는 로그 공간이 적어지므로 체크포인트를 사용하기 전에 1차 및 2차 로그 공간을 모두 고려합니다.

1차 로그 파일을 2차 로그 파일보다 더 많이 정의하지 않으면, 큐 관리자는 체크포인트를 사용하기 전에 2차 로그 파일을 할당할 수 있습니다. 그러면 1차 로그 파일을 다시 사용할 수 있게 됩니다.

로그 관리

시간이 경과할수록 기록된 로그 레코드 중 일부는 큐 관리자를 재시작하는 데 필요하지 않게 되므로, 큐 관리자는 로그 파일에서 비워진 공간을 이용할 수 있습니다. 이 작업이 수행될 경우 할당된 공간이 바로 재사용되므로 사용되는 디스크 공간의 양이 감소하는 것을 볼 수는 없습니다.

로그 레코드 중에서 마지막 완료 체크포인트 시작 이후로 기록된 레코드와 활성 트랜잭션에 의해 기록된 레코드만이 큐 관리자를 재시작하는 데 필요합니다. 따라서 오랫동안 체크포인트를 사용하지 못한 경우나, 장기 실행 트랜잭션이 로그 레코드를 오래 전에 기록한 경우에 로그가 채워질 수 있습니다. 큐 관리자는 첫 번째 문제점을 피하기 위해 자주 체크포인트를 사용하려고 합니다.

장기 실행 트랜잭션이 로그를 채울 때, 로그 레코드를 기록하려는 시도가 실패하면 일부 MQI 호출에서 MQRC_RESOURCE_PROBLEM을 리턴합니다(처리 중인 모든 트랜잭션을 확약하거나 롤백하기 위한 공간이 예약되어 있으므로, MQCMIT 또는 MQBAC은 실패하지 않습니다).

큐 관리자는 로그 공간을 너무 많이 소비하는 트랜잭션을 롤백합니다. 이와 같이 트랜잭션이 롤백되는 응용프로그램은 동일한 트랜잭션에서 동기점을 지정하는 후속 MQPUT 또는 MQGET 조작을 수행할 수 없습니다. 이 상태에서 동기점 아래에 메시지를 넣거나(put) 가져오려고(get) 하면 MQRC_BACKED_OUT이 리턴됩니다. 그러면 응용프로

그램이 MQRC_BACKED_OUT을 리턴하는 MQCMIT나 MQBACK을 발행하여 새로운 트랜잭션을 시작할 수도 있습니다. 너무 많은 로그 공간을 소비하는 트랜잭션이 롤백되면 로그 공간이 해제되므로, 큐 관리자가 계속해서 정상적으로 조작을 수행하게 됩니다.

로그가 채워지면 메시지가 발행됩니다(AMQ7463). 또한 장기 실행 트랜잭션으로 인해 공간이 해제되지 못하여 로그가 채워지면 메시지 AMQ7465가 발행됩니다.

마지막으로 비동기 내부 관리 프로세스가 레코드를 핸들링하는 것보다 더 빨리 로그에 기록될 경우, 메시지 AMQ7466이 발행됩니다. 이 메시지가 표시되면, 로그 파일 수를 늘리거나 큐 관리자에서 처리하는 데이터 양을 줄여야 합니다.

디스크가 가득 찰 경우에 발생하는 사항

큐 관리자 로그 기록 구성요소를 이용하여 디스크나 로그 파일이 가득 찰 경우에 대비할 수 있습니다. 로그를 포함하는 디스크가 채워지면, 큐 관리자는 메시지 AMQ6708을 발행하고 오류 레코드가 취해집니다.

로그 파일은 로그 레코드가 파일에 기록되는 대로 확장되는 것이 아니라, 최대 크기로 작성됩니다. 이는 MQSeries가 새 파일을 작성할 때만 디스크 공간을 모두 사용할 수 있다는 의미입니다. 따라서 레코드를 로그에 기록할 때는 공간이 사용되지 않습니다. MQSeries는 항상 기존 로그 파일에서 사용 가능한 공간의 크기를 인식하고 그에 따라 파일 내에서 공간을 관리합니다.

로그 파일을 포함하는 드라이브를 채울 경우, 일부 디스크 공간을 비울 수도 있습니다. 선형 로그를 사용 중이면, 다른 드라이브나 디바이스로 복사할 수 있는 로그 디렉토리에 비활성 로그 파일이 있을 수 있습니다. 계속해서 공간이 부족하면, 큐 관리자 구성 파일에서 로그 구성이 올바른지 점검하십시오. 로그가 사용 가능한 공간을 초과하지 않도록 1차 또는 2차 로그 파일 수를 줄일 수도 있습니다. 기존 큐 관리자에 대해 로그 파일 크기는 변경할 수 없다는 점에 유의하십시오. 큐 관리자는 모든 로그 파일 크기가 동일한 것으로 간주합니다.

로그 파일 관리

순환 로깅을 사용할 경우에는 로그 파일을 보유할 공간이 충분한지 확인하십시오. 시스템을 구성할 때 확인해야 합니다(188 페이지의 『LogDefaults 스탠자』 및 193 페이지의 『Log 스탠자』 참조). 필요할 때 작성할 2차 파일에 요구되는 공간을 포함하여, 로그에서 사용되는 디스크 공간의 양은 구성된 디스크 크기에 따라 제한됩니다.

선형 로그를 사용할 경우에는 데이터가 기록되면서 로그 파일이 계속 추가되므로 시간이 지날수록 사용되는 디스크 공간량이 커집니다. 데이터 기록율이 높으면, 디스크 공간도 새로운 로그 파일에 의해 빠르게 소비됩니다.

시간이 지나면 선형 로그의 이전 로그 파일은 큐 관리자를 재시작하거나 손상된 오브젝트의 매체 복원을 수행하는 데 더이상 필요하지 않게 됩니다. 정기적으로 큐 관리자는 필요한 로그 파일을 나타내기 위해 메시지 쌍을 발행합니다.

- 메시지 AMQ7467은 큐 관리자를 재시작하는 데 필요한 가장 오래된 로그 파일의 이름을 제공합니다. 이 로그 파일과 새로운 모든 로그 파일은 큐 관리자를 재시작하는 동안 사용 가능해야 합니다.
- 메시지 AMQ7468은 매체 복원을 수행하는 데 필요한 가장 오래된 로그 파일의 이름을 제공합니다.

이 파일보다 더 오래된 로그 파일은 온라인 상태가 아니어도 됩니다. 이러한 파일은 손상 복구를 위해 테이프와 같은 보존 매체에 복사하고, 활성 로그 디렉토리에서 제거할 수 있습니다. 재시작에는 필요하지 않지만 매체 복원에 필요한 로그 파일은 보존을 위해 오프로드될 수도 있습니다.

필요한 로그 파일을 찾을 수 없으면, 연산자 메시지 AMQ6767이 발행됩니다. 로그 파일과 모든 후속 로그 파일이 큐 관리자에 대해 사용 가능하게 만들고 조작을 재시도하십시오.

주: 매체 복원을 수행할 때, 필요한 모든 로그 파일이 동시에 로그 파일 디렉토리에서 사용가능해야 합니다. 필요한 모든 로그 파일을 보유하는 데 디스크 공간이 부족하지 않도록 복구하려면 오브젝트의 표준 매체 이미지가 있는지 확인하십시오.

로그 파일 위치

로그 파일의 위치를 선택할 때, MQSeries가 디스크 공간 부족으로 인해 새 로그를 형식화할 수 없을 경우에는 조작에 막대한 영향을 준다는 점에 유의하십시오.

순환 로그를 사용할 경우에는 최소한 구성된 1차 로그 파일에 대해 드라이브에 충분한 공간이 있는지 확인하십시오. 또한 로그가 커졌을 경우에 필요한 최소 하나의 2차 로그 파일용 공간도 남겨 두어야 합니다.

선형 로그를 사용할 경우에는 더 많은 공간을 남겨 두어야 합니다. 데이터가 기록되면서 로그에서 소비되는 공간이 계속 증가하기 때문입니다.

로그 파일은 큐 관리자 데이터와 분리된 디스크 드라이브에 두어야 합니다. 이는 성능에도 좋은 영향을 줍니다. 로그 파일을 이중복사 배열로 여러 개의 디스크 드라이브에 둘 수도 있습니다. 그러면 로그가 있는 드라이브에 오류가 있을 경우 보호할 수 있습니다. 이중복사가 없으면, MQSeries 시스템의 마지막 백업으로 되돌아가야 합니다.

복구에 로그 사용

데이터가 손상될 수 있는 경우에는 여러 가지가 있습니다. Compaq OpenVMS용 MQSeries는 다음 상황에서 복구할 수 있도록 도와줍니다.

- 손상된 데이터 오브젝트
- 시스템 전원 손실
- 통신 장애
- 손상된 로그 볼륨

이 절에서는 위의 문제점에서 복구하기 위해 로그를 사용하는 방법에 대해 설명합니다.

문제점에서 복구

MQSeries는 통신 장애나 전원 손실에서 복구할 수 있습니다. 간혹 다른 유형의 문제점(부주의로 인한 파일 삭제 등)에서 복구할 수도 있습니다.

통신 장애가 발생할 때, 메시지는 수신하는 응용프로그램에서 제거될 때까지 큐에 남아 있습니다. 메시지가 전송될 경우, 그 메시지는 성공적으로 전송될 때까지 트랜스미션 큐에 남아 있습니다. 통신 장애로부터 복구하려면 일반적으로 장애가 발생한 링크를 사용하여 채널을 재시작하는 것으로도 충분합니다.

전원 손실이 발생하면 큐 관리자가 재시작될 때 MQSeries가 큐를 장애시의 상태로 복원합니다. 그러면 어떤 지속 메시지도 손실되지 않습니다. 비지속 메시지는 제거되어 MQSeries가 정지될 때 유지되지 않습니다.

갑작스러운 손상 등으로 인해 MQSeries 오브젝트를 사용할 수 없는 경우가 있습니다. 그러면 시스템 전체나 일부를 복구해야 합니다. 필요한 조치는 손상이 감지된 시기, 선택된 로그 방식이 매체 복원을 지원하는지의 여부 및 손상된 오브젝트에 따라 결정됩니다.

매체 복원

매체 복원은 선형 로그에 대해서만 기록된 정보에서 오브젝트를 재작성하는 것을 의미합니다. 매체 복원은 순환 로깅에 대해 지원되지 않습니다. 예를 들어, 오브젝트 파일이 부주의로 삭제되거나 일부 다른 이유로 사용할 수 없게 되면, 매체 복원을 사용하여 이를 다시 작성할 수 있습니다. 오브젝트의 매체 복원에 필요한 로그의 정보를 매체 이미지라고 합니다. 매체 이미지는 **rcdmqimg** 명령을 사용하여 수동으로 기록하거나, 특정 상황에서 자동으로 기록할 수 있습니다.

매체 이미지는 오브젝트 자체가 재작성될 수 있는 오브젝트의 이미지를 포함하는 일련의 로그 레코드입니다.

오브젝트를 재작성하기 위해 필요한 첫 번째 로그 레코드를 매체 복원 레코드라고 하며, 이는 오브젝트의 최근 매체 이미지의 시작입니다. 각 오브젝트의 매체 복원 레코드에는 체크포인트 중에 기록된 정보의 조각들 중 하나입니다.

매체 이미지로부터 오브젝트를 재작성할 때, 마지막 이미지를 사용한 이후로 오브젝트에 대해 수행된 갱신을 설명하는 로그 레코드를 재생해야 할 경우도 있습니다.

예를 들어, 지속 메시지를 큐에 넣기(put) 전에 큐 오브젝트 이미지를 가져온 로컬 큐를 고려해 보십시오. 그 오브젝트의 최근 이미지를 재작성하려면, 이미지 자체를 재생하는 것 뿐만 아니라, 메시지를 큐에 넣는(put) 작업을 기록하는 로그 항목을 재생해야 합니다.

오브젝트가 작성되면, 기록된 로그 레코드에는 오브젝트를 완전히 다시 작성하기 위한 충분한 정보가 있어야 합니다. 이 레코드는 오브젝트의 첫 번째 매체 이미지를 구성합니다. 후속 매체 이미지는 다음과 같은 경우에 큐 관리자가 자동으로 기록합니다.

- 종료할 때마다 모든 프로세스 오브젝트와 로컬이 아닌 큐의 이미지를 가져옵니다.
- 큐를 비울 때 로컬 큐 이미지를 가져옵니다.

매체 이미지는 300 페이지의 『rcdmqimg(매체 이미지 기록)』에 설명된 것처럼 **rcdmqimg** 명령을 사용하여 수동으로 기록할 수도 있습니다.

매체 이미지 복구

MQSeries는 일부 오브젝트가 손상된 것을 발견할 경우에 매체 이미지로부터 그 오브젝트를 복구합니다. 특히, 이것은 정상적인 큐 관리자 시동시 손상된 것으로 발견되는 오브젝트에 적용됩니다. 큐 관리자가 마지막으로 종료되었을 때 트랜잭션이 완료되지 않으면, 영향을 받는 큐도 시동 조작을 완료하기 위해 자동으로 복구됩니다.

다른 오브젝트는 **rcrmqobj** 명령을 사용하여 수동으로 복구해야 합니다. 이 명령은 로그의 레코드를 재생하여 MQSeries 오브젝트를 재작성합니다. 오브젝트는 이미지가 저장된 시간과 재작성 명령이 발행된 시간 사이의 모든 적용가능한 로그 이벤트와 함께 로그에서 발견된 최근 이미지로부터 재작성됩니다. MQSeries 오브젝트가 손상된 경우, 수행할 수 있는 올바른 조치는 오브젝트를 삭제하거나 이러한 방식으로 재작성하는 것뿐입니다. 그러나, 비지속 메시지는 이 방식으로 복구할 수 없습니다.

rcrmqobj 명령에 대한 추가 정보는 303 페이지의 『rcrmqobj(오브젝트 재작성)』를 참조하십시오.

오브젝트의 매체 복원을 시도할 때 매체 복원 레코드를 포함하는 로그 파일과 모든 후속 로그 파일이 로그 파일 디렉토리에서 사용 가능해야 합니다. 필요한 파일을 찾을 수 없으면, 연산자 메시지 AMQ6767이 발행되고 매체 복원 조작은 실패합니다. 재작성할 오브젝트의 표준 매체 이미지를 가져오지 않으면, 오브젝트를 재작성하는 데 필요한 모든 로그 파일을 보유할 수 있는 디스크 공간이 부족하게 됩니다.

시동시 손상된 오브젝트 복구

큐 관리자가 시동시 손상된 오브젝트를 발견할 때 수행할 조치는 오브젝트의 유형과 큐 관리자가 매체 복원을 지원하도록 구성되어 있는지의 여부에 따라 결정됩니다.

큐 관리자 오브젝트가 손상된 경우, 큐 관리자는 오브젝트를 복구할 수 없으면 큐 관리자를 시작할 수 없습니다. 큐 관리자가 선형 로그로 구성되어 있는 상태에서 매체 복원을 지원할 경우, MQSeries는 자동으로 매체 이미지로부터 MQSeries 오브젝트를 재작성하려고 합니다. 선택한 로그 방법이 매체 복원을 지원하지 않을 경우, 큐 관리자 백업을 복원하거나 큐 관리자를 삭제할 수 있습니다.

큐 관리자가 정지했을 때 트랜잭션이 활성 중이었으면, 이 트랜잭션 내에서 넣거나(put) 가져간(get) 지속 미확약 메시지를 포함하는 로컬 큐도 큐 관리자를 성공적으로 시작하는 데 필요합니다. 이러한 로컬 큐 중 어느 하나가 손상된 것으로 발견되고, 큐 관리자가 매체 복원을 지원할 경우, 큐 관리자는 자동으로 매체 이미지로부터 그 큐를 재작성하려고 합니다. 큐 중에서 하나라도 복구할 수 없으면 MQSeries가 시작되지 않습니다.

미확약 메시지를 포함하는 손상된 로컬 큐가 매체 복원을 지원하지 않는 큐 관리자에서 시동 처리 중에 발견되면, 그 큐는 손상된 오브젝트로 표시되고 큐의 미확약 메시지는 무시됩니다. 이러한 큐 관리자에 대해 손상된 오브젝트의 매체 복원을 수행할 수 없으므로 이를 삭제하는 수밖에 없기 때문입니다. 손상을 보고하기 위해 메시지 AMQ7472가 발행됩니다.

다른 시기에 손상된 오브젝트 복구

오브젝트 매체 복원은 시동시에만 자동으로 수행됩니다. 다른 때 오브젝트 손상이 감지되면 연산자 메시지 AMQ7472가 발행되고 그 오브젝트를 사용하는 대부분의 조치가 실패합니다. 큐 관리자가 시작된 후에 어떤 시점에서든 큐 관리자 오브젝트가 손상될 경우, 큐 관리자는 선제 종료를 수행합니다. 오브젝트가 손상되면 이를 삭제하고, 큐 관리자가 선형 로그를 사용 중일 경우 **rcrmqobj** 명령을 사용하여 매체 이미지로부터 복구할 수 있습니다(추가 정보는 303 페이지의 『rcrmqobj(오브젝트 재작성)』를 참조하십시오).

MQSeries 로그 파일 보호

MQSeries 큐 관리자가 실행될 때 수동으로 로그 파일을 제거해서는 안됩니다. 사용자가 부주의로(또는 고의로) 큐 관리자가 재시작하는 데 필요한 로그 파일을 삭제할 경우, MQSeries는 오류를 발행하지 않고 지속 메시지를 포함하는 데이터를 계속 처리합니다. 큐 관리자는 정상적으로 종료하지만 재시작에 실패합니다. 메시지의 매체 복원은 불가능합니다.

활성 큐 관리자에서 사용되는 로그를 제거하는 권한을 가진 사용자는 다른 중요한 큐 관리자 자원(권한 파일, 큐 파일, 오브젝트 카탈로그 및 MQSeries 실행 파일 등)을 삭

제하는 권한도 가지고 있습니다. 따라서 MQSeries가 자체를 보호할 수 없는 방식으로 실행 중이거나 휴지 상태인 큐 관리자를 손상시킬 수 있습니다.

사용자에게 향상된 특권이나 MQM 권한 ID를 부여할 때 주의하도록 하십시오.

백업 및 복원

정기적으로 하드웨어 오류로 인한 손상에 대해 보호하기 위해 큐 관리자 데이터를 백업할 수 있습니다. 그러나 종종 메시지 데이터의 수명이 짧기 때문에 백업하지 않을 수도 있습니다.

MQSeries 백업

큐 관리자의 데이터를 백업하려면 다음을 수행해야 합니다.

1. 큐 관리자가 실행되지 않도록 하십시오.

큐 관리자가 실행 중이면 **endmqm** 명령으로 정지하십시오.

주: 실행 중인 큐 관리자를 백업할 경우, 백업은 파일이 복사될 때 처리 중인 갱신으로 인해 일관성을 잃을 수 있습니다.

2. 큐 관리자가 해당 데이터와 로그 파일을 위치시키는 디렉토리를 찾으십시오.

구성 파일의 정보를 사용하여 이 디렉토리를 판별할 수 있습니다. 자세한 정보는 183 페이지의 『제13장 MQSeries 구성』을 참조하십시오.

주: 디렉토리에 나타나는 이름을 이해하는 것이 약간 어려울 수도 있습니다. 이것은 MQSeries를 사용 중인 플랫폼과 호환되도록 이름을 변환하기 때문입니다. 이름 변환에 대한 자세한 정보는 25 페이지의 『MQSeries 파일 이름 이해』를 참조하십시오.

3. 모든 큐 관리자의 데이터와 모든 서브디렉토리를 포함하여 로그 파일 디렉토리의 사본을 가져오십시오.

파일 중 누락한 것이 없는지 확인하십시오(특히 로그 제어 파일과 구성 파일). 디렉토리 중 일부는 비어 있을 수도 있지만, 이후 날짜의 백업을 복원할 경우 모두 필요할 수 있으므로, 이 디렉토리들도 저장하는 것이 좋습니다.

4. 파일 소유권을 보존하십시오. BACKUP 명령과 /BY_OWNER 매개변수로 이를 수행할 수 있습니다.

MQSeries 복원

큐 관리자의 데이터를 백업하려면 다음을 수행해야 합니다.

1. 큐 관리자가 실행되지 않도록 하십시오.

2. 큐 관리자가 해당 데이터와 로그 파일을 위치시키는 디렉토리를 찾으십시오. 이 정보는 구성 파일에 있습니다.

3. 백업한 데이터가 위치할 디렉토리의 내용을 지우십시오.
4. 백업한 큐 관리자 데이터와 로그 파일을 해당 위치에 복사하십시오.

결과 디렉토리 구조를 점검하여 필요한 모든 디렉토리가 있는지 확인하십시오.

MQSeries 디렉토리 및 서브디렉토리에 대한 자세한 정보는 343 페이지의 『부록C. 디렉토리 구조』를 참조하십시오.

로그 파일 뿐만 아니라 로그 제어 파일도 있는지 확인하십시오. 또한 MQSeries가 복원된 데이터에 대해 올바른 위치에서 볼 수 있도록 MQSeries 및 큐 관리자 구성 파일에 일관성이 있는지 점검하십시오.

데이터를 올바르게 백업하고 복원하면 큐 관리자가 시작됩니다.

주: 큐 관리자 데이터와 로그 파일이 서로 다른 디렉토리에 있어도, 디렉토리를 동시에 백업하고 복원해야 합니다. 큐 관리자 데이터 및 로그 파일의 경과 시간이 다를 경우, 큐 관리자가 올바른 상태에 있지 않으며 시작되지 않을 수 있습니다. 큐 관리자가 시작되어도 데이터는 거의 손상될 것입니다.

복구 시나리오

이 절에서는 발생할 수 있는 문제점과 이 문제점에서 복구하는 방법을 보여줍니다.

디스크 드라이브 오류

큐 관리자 데이터나 로그, 또는 모두를 포함하는 디스크 드라이브에 문제가 발생할 수도 있습니다. 데이터 손실이나 손상 등의 문제점이 있습니다. 세 가지 경우에서 유지되는 데이터의 부분이 다릅니다.

모든 경우, 먼저 손상이 있는 디렉토리 구조를 점검한 뒤 필요하다면 손상을 복구해야 합니다. 큐 관리자 데이터를 손실한 경우, 큐 관리자 디렉토리 구조가 손상되었을 위험이 있습니다. 그러한 경우, 큐 관리자를 재시작하기 전에 수동으로 디렉토리 트리를 재작성해야 합니다. 구조적 손상을 점검할 경우, 사용하는 로그 기록 유형에 따라 수행할 수 있는 여러 가지 대안이 있습니다.

- 디렉토리 구조에 주요 손상이 있거나 로그가 손상된 경우, 구성 파일, 로그 및 큐 관리자 디렉토리를 포함하여, 오래된 모든 파일을 제거하여 QMgrName 레벨로 다시 되돌리고, 마지막 백업을 복원한 후 큐 관리자를 재시작하십시오.
- 매체 복원에서 선행 로깅을 사용할 경우, 디렉토리 구조가 손상되지 않았는지 확인하고 큐 관리자를 재시작하십시오. 큐 관리자가 재시작되지 않을 경우, 백업을 복원하십시오. 큐 관리자가 재시작될 경우, MQSC를 사용하여 다른 오브젝트가 손상되었는지 점검하십시오. `rcrmqobj` 명령을 사용하여 찾은 항목을 복구하십시오. 예를 들면, 다음과 같습니다.

```
rcrmqobj -m QMgrName -t * *
```

여기서 QMgrName은 복구될 큐 관리자입니다. -t * *는 임의 유형의 오브젝트가 복구됨을 나타냅니다. 하나 또는 두 개의 오브젝트만 손상된 것으로 보고된 경우에는 여기에 이름과 유형으로 그 오브젝트를 지정할 수 있습니다.

주: 이 명령은 채널에 적용되지 않습니다.

- 매체 복원 및 손상되지 않은 로그에서 선형 로깅을 사용할 경우, 기존의 로그 파일과 로그 제어 파일을 변경되지 않은 상태로 두고 큐 관리자 데이터의 백업을 복원할 수도 있습니다. 큐 관리자를 시작하면 로그의 변경사항이 적용되어 큐 관리자가 다시 실패 상태가 됩니다.

이 방법은 두 가지 사실을 바탕으로 합니다. 먼저 체크포인트 파일이 큐 관리자 데이터의 일부로 복원된다는 중요한 사실입니다. 이 파일에는 일관된 큐 관리자를 제공하기 위해 로그의 데이터 중 어느 정도가 적용되어야 하는지 판별하는 정보가 들어 있습니다.

두 번째로, 백업시 큐 관리자 시작에 필요했던 가장 오래된 로그 파일과, 모든 후속 로그 파일들이 로그 파일 디렉토리에서 사용 가능해야 합니다.

이것이 가능하지 않으면, 큐 관리자 데이터 및 로그(동시에 사용된 것 모두)의 백업을 복원해야 합니다.

- 순환 로깅, 또는 매체 복원이 없는 선형 로깅의 경우, 최근 백업에서 큐 관리자를 복원해야 합니다. 백업을 복원했으면, 큐 관리자를 재시작하고 위와 같이 손상된 오브젝트를 점검하십시오. 그러나, 매체 복원이 없으므로 손상된 오브젝트를 재작성할 다른 방법을 찾아야 합니다.

손상된 큐 관리자 오브젝트

큐 관리자 오브젝트가 정상 조작 중에 손상된 것으로 보고되었으면, 큐 관리자는 선제 종료로 수행합니다. 사용하는 로그 기록 유형에 따라 이러한 상황에서 복구할 수 있는 두 가지 방법이 있습니다.

- 선형 로깅의 경우, 손상된 오브젝트를 포함하는 파일을 수동으로 삭제한 후 큐 관리자를 재시작하십시오. 손상된 오브젝트의 매체 복원이 자동으로 이루어집니다.
- 순환 로깅 또는 선형 로깅의 경우, 큐 관리자 데이터 및 로그의 마지막 백업을 복원한 후 큐 관리자를 재시작하십시오.

손상된 단일 오브젝트

단일 오브젝트가 정상 조작 중에 손상된 것으로 보고되면, 사용하는 로그 기록 유형에 따라 복구할 수 있는 두 가지 방법이 있습니다.

- 선형 로깅의 경우, 매체 이미지로부터 오브젝트를 재작성하십시오.

- 순환 로깅의 경우, 큐 관리자 데이터 및 로그의 마지막 백업을 복원하고 큐 관리자를 재시작하십시오.

자동 매체 복원 실패

선형 로그를 사용하여 큐 관리자를 시동하는 데 필요한 로컬 큐가 손상되고, 자동 매체 복원이 실패할 경우, 큐 관리자 데이터 및 로그의 마지막 백업을 복원하고 큐 관리자를 재시작하십시오.

dmpmqlog 명령을 사용하여 로그의 내용 덤프

dmpmqlog 명령은 큐 관리자 로그의 내용을 덤프하는 데 사용할 수 있습니다. 디폴트로, 모든 활성 로그 레코드가 덤프됩니다. 즉, 명령은 로그의 헤드부터 덤프를 시작합니다. 일반적으로 이것은 마지막으로 완료된 체크포인트의 시작부터 이루어집니다.

로그는 큐 관리자가 실행되고 있지 않은 경우에만 덤프될 수 있습니다. 큐 관리자는 종료시 체크포인트를 사용하므로, 로그의 활성 부분에는 항상 약간의 로그 레코드만 있습니다. 그러나 **dmpmqlog** 명령은 다음 옵션 중 하나를 사용하여 더 많은 로그 레코드를 덤프하여 덤프의 시작 위치를 변경하도록 지시할 수 있습니다.

- 가장 간단한 옵션은 로그의 베이스에서 덤프를 시작하는 것입니다. 로그의 베이스는 로그의 헤드를 포함하는 로그 파일에서 첫 번째 로그 레코드입니다. 이 경우에 덤프되는 추가 데이터 양은 로그 파일에서 로그 헤드의 위치에 따라 결정됩니다. 로그 파일의 시작 부분에 있으면, 적은 양의 추가 데이터만 덤프됩니다. 헤드가 로그 파일의 끝 부분에 있으면, 상당히 많은 데이터가 덤프됩니다.
- 또다른 옵션은 덤프의 시작 위치가 개별 로그 레코드로 지정되도록 합니다. 각 로그 레코드는 고유한 *LSN(Log Sequence Number)*로 식별됩니다. 순환 로깅의 경우, 이 시작 로그 레코드는 로그 베이스 이전에 있을 수 없습니다. 이 제한은 선형 로그에 적용되지 않습니다. 비활성 로그 파일은 명령 실행 이전에 복구해야 할 수도 있습니다. 이 옵션의 경우, 시작 위치로 올바른 LSN을 지정해야 합니다. 이것은 이전 **dmpmqlog** 출력에서 가져와야 합니다.

예를 들어, 선형 로깅을 사용할 경우 마지막 **dmpmqlog** 출력으로부터 `nextlsn`을 지정할 수 있습니다. 다음 LSN은 Log File Header에 표시되며 이는 기록될 다음 로그 레코드의 LSN을 나타냅니다. 그러므로, 이를 시작 위치로 사용하여 로그가 마지막으로 덤프된 이후로 기록된 모든 로그 레코드를 형식화할 수 있습니다.

- 세 번째 옵션은 선형 로그 전용입니다. 주어진 로그 파일 범위에서 로그 레코드 형식화를 시작하도록 덤프에 지시합니다. 이 경우, 로그 덤프는 활성 로그 파일과 동일한 디렉토리에서 이 로그 파일과 각각의 연속 파일이 발견될 것으로 예상합니다. 이 옵션은 순환 로그에는 적용될 수 없습니다. 로그 덤프가 로그 베이스 이전에 로그 레코드에 액세스할 수 없기 때문입니다.

dmpmqlog 사용

dmpmqlog 명령의 출력은 Log File Header로 일련의 형식화된 로그 레코드입니다. 큐 관리자는 몇 개의 로그 레코드를 사용하여 데이터에 대한 변경사항을 기록합니다.

형식화된 일부 정보는 내부적으로만 사용됩니다. 다음 목록에는 가장 유용한 로그 레코드가 포함되어 있습니다.

Log File Header

각 로그에는 항상 **dmpmqlog** 명령에 의해 형식화되는 첫 번째 사항인 단일 로그 파일 헤더가 있습니다. 여기에는 다음과 같은 필드가 있습니다.

<i>logactive</i>	1차 로그 범위 수.
<i>loginactive</i>	2차 로그 범위 수.
<i>logsize</i>	범위당 4 KB 페이지 수.
<i>baselsn</i>	로그의 헤드를 포함하는 로그 범위의 첫 번째 LSN.
<i>nextlsn</i>	기록될 다음 로그 레코드의 LSN.
<i>headlsn</i>	로그의 헤드에 있는 로그 레코드의 LSN.
<i>tailsn</i>	로그의 테일 위치를 식별하는 LSN.
<i>hflag1</i>	로그가 CIRCULAR 또는 LOG RETAIN(선형) 중 어느 상태인지를 식별합니다.
<i>HeadExtentID</i>	로그의 헤드를 포함하는 로그 범위.

Log Record Header

로그 내의 각 로그 레코드에는 다음과 같은 정보를 포함하는 고정된 헤더가 있습니다.

<i>LSN</i>	로그 순차 번호.
<i>LogRecdType</i>	로그 레코드의 유형.
<i>XTranid</i>	이 로그 레코드와 연관되는 트랜잭션 ID(있는 경우). MQI의 <i>TranType</i> 은 MQ 전용 트랜잭션을 나타냅니다. XA의 <i>TranType</i> 은 다른 자원 관리자와 함께 포함됩니다. 동일한 작업 단위 내에 포함된 갱신사항은 <i>XTranid</i> 가 같습니다.
<i>QueueName</i>	이 로그 레코드와 연관되는 큐(있는 경우).
<i>Qid</i>	큐의 고유한 내부 ID.
<i>PrevLSN</i>	같은 트랜잭션내에서 이전 로그 레코드의 LSN(있는 경우).

Start Queue Manager

큐 관리자의 시작을 기록합니다.

<i>StartDate</i>	큐 관리자가 시작된 날짜.
<i>StartTime</i>	큐 관리자가 시작된 시간.

Stop Queue Manager

큐 관리자의 정지를 기록합니다.

StopDate 큐 관리자가 정지된 날짜.

StopTime 큐 관리자가 정지된 시간.

ForceFlag 사용된 종료 유형.

Start Checkpoint

큐 관리자 체크포인트의 시작을 표시합니다.

End Checkpoint

큐 관리자 체크포인트의 끝을 표시합니다.

ChkPtLSN 이 체크포인트를 시작한 로그 레코드의 LSN.

Put Message

큐에 넣는(put) 지속 메시지를 기록합니다. 메시지를 동기점 하에 넣었으면(put), 로그 레코드 헤더에는 널이 아닌 *XTanid*가 있습니다. 레코드의 나머지는 다음이 포함됩니다.

SpcIndex 큐에 있는 메시지의 ID. 이것은 큐로부터 해당 메시지를 가져오기 (get) 위해 사용된 해당 MQGET와 대조하기 위해 사용할 수 있습니다. 이 경우, 같은 *QueueName* 및 *SpcIndex*를 포함하는 후속 *Get Message* 로그 레코드를 발견할 수 있습니다. 이 시점에서 후속 메시지를 그 큐에 넣기(put) 위해 *SpcIndex* ID를 다시 사용할 수 있습니다.

Data 이 로그 레코드에 대해 16진 덤프에 포함되는 것으로, 뒤에 메시지 설명자(eyecatcher MD)와 메시지 데이터 자체가 표시되는 다양한 내부 데이터입니다.

Put Part

단일 로그 레코드에 대해 너무 큰 지속 메시지는 단일 *Put Message* 레코드로 기록되고, 그 뒤로 여러 개의 *Put Part* 로그 레코드가 표시됩니다.

Data 이전 로그 레코드가 중지된 메시지 데이터에서 계속합니다.

Get Message

지속 메시지 가져오기(get)만 기록됩니다. 메시지가 동기점 하에 가져왔으면, 로그 레코드 헤더에는 널이 아닌 *XTanid*가 있습니다. 레코드의 나머지는 다음이 포함됩니다.

SpcIndex 큐에서 가져온(get) 메시지를 식별합니다. 같은 *QueueName* 및 *SpcIndex*를 포함하는 가장 최근의 *Put Message* 로그 레코드는 가져온(get) 메시지를 식별합니다.

QPriority 큐에서 가져온(get) 메시지의 우선순위.

Start Transaction

새로운 트랜잭션의 시작을 나타냅니다. MQI의 TranType은 MQ 전용 트랜잭션을 나타냅니다. XA의 TranType은 다른 자원 관리자와 관련되는 것을 나타냅니다. 이 트랜잭션에서 수행된 모든 갱신은 동일한 *XTranid*를 갖습니다.

Prepare Transaction

큐 관리자가 지정된 *XTranid*와 연관되는 갱신을 확약할 준비가 되어 있음을 나타냅니다. 이 로그 레코드는 다른 자원 관리자와 관련되는 2단계 확약의 일부로 기록됩니다.

Commit Transaction

큐 관리자가 트랜잭션에 의해 수행된 모든 갱신을 확약하였음을 나타냅니다.

Rollback Transaction

이 로그 레코드는 트랜잭션을 롤백하기 위한 큐 관리자 의도를 나타냅니다.

End Transaction

이 로그 레코드는 트랜잭션 롤백의 끝을 표시합니다.

Transaction Table

이 레코드는 동기점에서 기록됩니다. 지속적으로 갱신된 각 트랜잭션의 상태를 기록합니다. 각 트랜잭션에 대해 다음 정보가 기록됩니다.

- XTranid* 트랜잭션 ID.
- FirstLSN* 트랜잭션과 연관되는 첫 번째 로그 레코드의 LSN.
- LastLSN* 트랜잭션과 연관되는 마지막 로그 레코드의 LSN.

Transaction Participants

이 로그 레코드는 큐 관리자의 XA 트랜잭션 관리자 구성요소에 의해 기록됩니다. 트랜잭션에 참여하는 외부 자원 관리자들을 기록합니다. 각 참여자에 대해 다음 정보가 기록됩니다.

- RMName* 자원 관리자의 이름.
- RMId* 자원 관리자 ID. 이것은 자원 관리자가 참여하는 전역 트랜잭션을 기록하는 후속 *Transaction Prepared* 로그 레코드에도 기록됩니다.
- SwitchFile* 이 자원 관리자에 대한 스위치 로드 파일.
- XAOpenString* 이 자원 관리자에 대한 XA 열기 문자열.
- XACloseString* 이 자원 관리자에 대한 XA 열기 문자열.

Transaction Prepared

이 로그 레코드는 큐 관리자의 XA 트랜잭션 관리자 구성요소에 의해 기록됩니다. 이것은 지정된 전역 트랜잭션이 성공적으로 준비되었음을 나타냅니다. 각각의 참여

dmpmqlog 사용

하는 자원 관리자가 예약되도록 지시됩니다. 준비된 각 자원 관리자의 *RMID*는 로그 레코드에 기록됩니다. 큐 관리자 자체가 트랜잭션에 참여할 경우, *RMID*가 0인 *Participant Entry*가 제시됩니다.

Transaction Forget

이 로그 레코드는 큐 관리자의 XA 트랜잭션 관리자 구성요소에 의해 기록됩니다. 이것은 각 참여자에게 예약 결정이 전달된 경우에 *Transaction Prepared* 로그 레코드 다음에 나옵니다.

Purge Queue

RUNMQSC CLEAR 명령 등을 사용하여 큐의 모든 메시지가 제거되었음을 기록합니다.

Queue Attributes

큐의 속성에 대한 초기화와 변경을 기록합니다.

Create Object

MQSeries 오브젝트 작성을 기록합니다.

ObjName 작성된 오브젝트의 이름.

UserId 작성을 수행하는 사용자 ID.

Delete Object

MQSeries 오브젝트 삭제를 기록합니다.

ObjName 삭제된 오브젝트의 이름.

166 페이지의 그림18에 **dmpmqlog** 명령의 출력 예가 나와 있습니다. 특정 로그 레코드의 LSN에서 시작하는 덤프는 다음 명령을 사용하여 생성되었습니다.

```
dmpmqlog -m "testqm" -s 0:0:0:44162
```

dmpmqlog 사용

AMQ7701: DMPMQLOG command is starting.

LOG FILE HEADER

```
counter1 . . . : 23                counter2 . . . : 23
FormatVersion . : 2                logtype . . . : 10
logactive . . . : 3                loginactive . . : 2
logsize . . . . : 1024             pages
baselsn . . . . : <0:0:0:0>
nextlsn . . . . : <0:0:0:60864>
lowtranlsn . . . : <0:0:0:0>
minbufflsn . . . : <0:0:0:58120>
headlsn . . . . : <0:0:0:58120>
taillsn . . . . : <0:0:0:60863>
logfilepath . . : ""
hflag1 . . . . : 1
                -> CONSISTENT
                -> CIRCULAR
HeadExtentID . . : 1                LastEID . . . . : 846249092
LogId . . . . . : 846249061         LastCommit . . . : 0
FirstArchNum . . : 4294967295      LastArchNum . . . : 4294967295
nextArcFile . . . : 4294967295     firstRecFile . . . : 4294967295
firstDlFile . . . : 4294967295    lastDeleteFile . . : 4294967295
RecHeadFile . . . : 4294967295    FileCount . . . . : 3
frec_trunclsn . . : <0:0:0:0>
frec_readlsn . . : <0:0:0:0>
frec_extnum . . . : 0                LastCid . . . . : 0
onlineBkupEnd . . : 0                softmax . . . . : 4194304
```

LOG RECORD - LSN <0:0:0:44162>

HLG Header: lreclsize 212, version 1, rmid 0, eyecatcher HLRH

```
LogRecdType . . : ALM Start Checkpoint (1025)
Eyecatcher . . . : ALRH                Version . . . . . : 1
LogRecdLen . . . : 192                LogRecdOwnr . . . : 1024 (ALM)
XTranid . . . . : TranType: NULL
QueueName . . . . : NULL
Qid . . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>
```

No data for Start Checkpoint Record

그림 18. dmpmqlog 출력 예 (1/13)

LOG RECORD - LSN <0:0:0:44374>

HLG Header: lreclsize 220, version 1, rmid 0, eyecatcher HLRH

```

LogRecdType . . . : ATM Transaction Table (773)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 200                      LogRecdOwnr . . . : 768   (ATM)
XTranid . . . . . : TranType: NULL
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

```

Version : 1

TranCount : 0

LOG RECORD - LSN <0:0:0:44594>

HLG Header: lreclsize 1836, version 1, rmid 0, eyecatcher HLRH

```

LogRecdType . . . : Transaction Participants (1537)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 1816                    LogRecdOwnr . . . : 1536  (T)
XTranid . . . . . : TranType: NULL
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

```

Id. : TLPH

Version : 1

Flags : 3

Count : 2

Participant Entry 0

RMName : DB2 MQBankDB

RMId : 1

SwitchFile . . . : /Development/sbolam/build/devlib/tstxasw

XAOpenString . . :

XACloseString . . :

Participant Entry 1

RMName : DB2 MQBankDB

RMId : 2

SwitchFile . . . : /Development/sbolam/build/devlib/tstxasw

XAOpenString . . :

XACloseString . . :

그림 18. dmpmqlog 출력 예 (2/13)

dmpmqlog 사용

```
LOG RECORD - LSN <0:0:0:46448>
*****

HLG Header: lreclsize 236, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ALM End Checkpoint (1026)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 216                      LogRecdOwnr . . : 1024 (ALM)
XTranid . . . . : TranType: NULL
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

ChkPtLSN . . . . : <0:0:0:44162>
OldestLSN . . . . : <0:0:0:0>
MediaLSN . . . . : <0:0:0:0>

LOG RECORD - LSN <0:0:0:52262>
*****

HLG Header: lreclsize 220, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Start Transaction (769)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 200                      LogRecdOwnr . . : 768 (ATM)
XTranid . . . . : TranType: MQI      TranNum{High 0, Low 1}
QueueName . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . : 1
SoftLogLimit . . : 10000
```

그림 18. dmpmqlog 출력 예 (3/13)

LOG RECORD - LSN <0:0:0:52482>

HLG Header: lreclsize 730, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Put Message (257)
 Eyecatcher . . . : ALRH Version : 1
 LogRecdLen . . . : 710 LogRecdOwnr . . . : 256 (AQM)
 XTranid : TranType: MQI TranNum{High 0, Low 1}
 QueueName : Queue1
 Qid : {Hash 196836031, Counter: 0}
 ThisLSN : <0:0:0:0>
 PrevLSN : <0:0:0:52262>

Version : 3
 SpcIndex : 1
 PrevLink.Locn . . . : 36 PrevLink.Length : 8
 PrevDataLink . . . : {High 0, Low 2048}
 Data.Locn : 2048 Data.Length . . . : 486

```

Data . . . . . :
00000: 41 51 52 48 00 00 00 04 FF FF FF FF FF FF FF FF AQRH.....
00016: 00 00 00 00 00 00 00 00 00 00 00 01 00 01 01 C0 .....&#192;
00032: 00 00 00 00 00 00 00 01 00 00 00 22 00 00 00 00 .....
00048: 00 00 00 00 41 4D 51 20 74 65 73 74 71 6D 20 20 ....AMQ testqm
00064: 20 20 20 20 33 80 2D D2 00 00 10 13 00 00 00 00 3€-&#30;.....
00080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00112: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 .....
00128: 00 00 00 00 00 00 00 22 00 00 00 00 00 00 00 00 .....
00144: 00 00 00 00 00 00 00 C9 2C B5 C0 25 FF FF FF FF .....&#26;,&#181;&#192;%...
00160: 4D 44 20 20 00 00 00 01 00 00 00 00 00 00 00 08 MD .....
00176: 00 00 00 00 00 00 01 11 00 00 03 33 20 20 20 20 .....3
00192: 20 20 20 20 00 00 00 00 00 00 00 01 20 20 20 20 .....
00208: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 .....
00224: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 .....
00240: 20 20 20 20 20 20 20 20 20 20 20 20 74 65 73 74 test
00256: 71 6D 20 20 20 20 20 20 20 20 20 20 20 20 20 20 qm
00272: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 .....
00288: 20 20 20 20 20 20 20 20 20 20 20 20 73 62 6F 6C sbo1
00304: 61 6D 20 20 20 20 20 20 04 37 34 38 30 00 00 00 am .7480...
00320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00336: 00 00 00 00 00 00 00 00 20 20 20 20 20 20 20 20 .....
00352: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 .....
00368: 20 20 20 20 20 20 20 20 00 00 00 06 75 74 7A 61 .....utza
00384: 70 69 20 20 20 20 20 20 20 20 20 20 20 20 20 20 pi
00400: 20 20 20 20 20 20 20 20 31 39 39 37 30 35 31 39 19970519
00416: 31 30 34 32 31 35 32 30 20 20 20 20 00 00 00 00 10421520 ....
00432: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00448: 50 65 72 73 69 73 74 65 6E 74 20 6D 65 73 73 61 Persistent messa
00464: 67 65 20 70 75 74 20 75 6E 64 65 72 20 73 79 6E ge put under syn
00480: 63 70 6F 69 6E 74 cpoint
  
```

그림 18. dmpmqlog 출력 예 (4/13)

dmpmqlog 사용

```

LOG RECORD - LSN <0:0:0:53458>
*****

HLG Header: lrecsize 734, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Put Message (257)
Eyecatcher . . . : ALRH                      Version . . . . : 1
LogRecdLen . . . : 714                      LogRecdOwnc . . . : 256      (AQM)
XTranid . . . . : TranType: NULL
QueueName . . . . : Queue2
Qid . . . . . : {Hash 184842943, Counter: 2}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:0>

Version . . . . : 3
SpIndex . . . . : 1
PrevLink.Locn . . : 36                      PrevLink.Length : 8
PrevDataLink . . : {High 0, Low 2048}
Data.Locn . . . . : 2048                    Data.Length . . . : 490
Data . . . . . :
00000: 41 51 52 48 00 00 00 04 FF FF FF FF FF FF FF FF  AQRH.....
00016: 00 00 00 00 00 00 00 00 00 00 00 01 00 01 01 C0  .....&#192;
00032: 00 00 00 00 00 00 00 01 00 00 00 26 00 00 00 00  .....&;...
00048: 00 00 00 00 41 4D 51 20 74 65 73 74 71 6D 20 20  ....AMQ testqm
00064: 20 20 20 20 33 80 2D D2 00 00 10 13 00 00 00 00  3E- &#30;.....
00080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00112: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01  .....
00128: 00 00 00 00 00 00 00 26 00 00 00 00 00 00 00 00  .....&;.....
00144: 00 00 00 00 00 00 00 C9 2C B6 D8 DD FF FF FF FF  .....&#26;,&#216;.....
00160: 4D 44 20 20 00 00 00 01 00 00 00 00 00 00 00 08  MD .....
00176: 00 00 00 00 00 00 01 11 00 00 03 33 20 20 20 20  .....3
00192: 20 20 20 20 00 00 00 00 00 00 00 01 20 20 20 20  .....
00208: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00224: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00240: 20 20 20 20 20 20 20 20 20 20 20 20 20 74 65 73 74  test
00256: 71 6D 20 20 20 20 20 20 20 20 20 20 20 20 20 20  qm
00272: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00288: 20 20 20 20 20 20 20 20 20 20 20 20 73 62 6F 6C  sbol
00304: 61 6D 20 20 20 20 20 20 04 37 34 38 30 00 00 00  am .7480...
00320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00336: 00 00 00 00 00 00 00 20 20 20 20 20 20 20 20 20  .....
00352: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00368: 20 20 20 20 20 20 20 20 00 00 00 06 75 74 7A 61  ....utza
00384: 70 69 20 20 20 20 20 20 20 20 20 20 20 20 20 20  pi
00400: 20 20 20 20 20 20 20 20 31 39 39 37 30 35 31 39  19970519
00416: 31 30 34 33 32 37 30 36 20 20 20 20 00 00 00 00  10432706 ....
00432: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00448: 50 65 72 73 69 73 74 65 6E 74 20 6D 65 73 73 61  Persistent messa
00464: 67 65 20 6E 6F 74 20 70 75 74 20 75 6E 64 65 72  ge not put under
00480: 20 73 79 6E 63 70 6F 69 6E 74  syncpoint

```

그림 18. dmpmqlog 출력 예 (5/13)

LOG RECORD - LSN <0:0:0:54192>

HLG Header: lreclen 216, version 1, rmid 0, eyecatcher HLRH

```

LogRecdType . . . : ATM Commit Transaction (774)
Eyecatcher   . . . : ALRH                      Version . . . . . : 1
LogRecdLen   . . . : 196                      LogRecdOwnr . . . : 768   (ATM)
XTranid      . . . : TranType: MQI             TranNum{High 0, Low 1}
QueueName    . . . : NULL
Qid          . . . : {NULL_QID}
ThisLSN      . . . : <0:0:0:0>
PrevLSN      . . . : <0:0:0:52482>

```

Version : 1

LOG RECORD - LSN <0:0:0:54408>

HLG Header: lreclen 220, version 1, rmid 0, eyecatcher HLRH

```

LogRecdType . . . : ATM Start Transaction (769)
Eyecatcher   . . . : ALRH                      Version . . . . . : 1
LogRecdLen   . . . : 200                      LogRecdOwnr . . . : 768   (ATM)
XTranid      . . . : TranType: MQI             TranNum{High 0, Low 3}
QueueName    . . . : NULL
Qid          . . . : {NULL_QID}
ThisLSN      . . . : <0:0:0:0>
PrevLSN      . . . : <0:0:0:0>

```

Version : 1

SoftLogLimit . . : 10000

LOG RECORD - LSN <0:0:0:54628>

HLG Header: lreclen 240, version 1, rmid 0, eyecatcher HLRH

```

LogRecdType . . . : AQM Get Message (259)
Eyecatcher   . . . : ALRH                      Version . . . . . : 1
LogRecdLen   . . . : 220                      LogRecdOwnr . . . : 256   (AQM)
XTranid      . . . : TranType: MQI             TranNum{High 0, Low 3}
QueueName    . . . : Queue1
Qid          . . . : {Hash 196836031, Counter: 0}
ThisLSN      . . . : <0:0:0:0>
PrevLSN      . . . : <0:0:0:54408>

```

Version : 2

SpIndex : 1 QPriority : 0

PrevLink.Locn . . : 36 PrevLink.Length : 8

PrevDataLink . . . : {High 4294967295, Low 4294967295}

그림 18. dmpmqlog 출력 예 (6/13)

dmpmqlog 사용

```
LOG RECORD - LSN <0:0:0:54868>
*****

HLG Header: lreclsize 240, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Get Message (259)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 220                      LogRecdOwnc . . . : 256      (AQM)
XTranid . . . . . : TranType: NULL
QueueName . . . . : Queue2
Qid . . . . . : {Hash 184842943, Counter: 2}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

Version . . . . . : 2
SpIndex . . . . . : 1                      QPriority . . . . . : 0
PrevLink.Locn . . : 36                    PrevLink.Length : 8
PrevDataLink . . : {High 4294967295, Low 4294967295}
LOG RECORD - LSN <0:0:0:55108>
*****

HLG Header: lreclsize 216, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Commit Transaction (774)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 196                      LogRecdOwnc . . . : 768      (ATM)
XTranid . . . . . : TranType: MQI      TranNum{High 0, Low 3}
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:54628>

Version . . . . . : 1

LOG RECORD - LSN <0:0:0:55324>
*****

HLG Header: lreclsize 220, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Start Transaction (769)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 200                      LogRecdOwnc . . . : 768      (ATM)
XTranid . . . . . : TranType: XA
      XID: formatID 5067085, gtrid_length 14, bqual_length 4
      gtrid [3270BDB40000102374657374716D]
      bqual [00000001]
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

Version . . . . . : 1
SoftLogLimit . . : 10000
```

그림 18. dmpmqlog 출력 예 (7/13)

```

LOG RECORD - LSN <0:0:0:55544>
*****

HLG Header: lreclen 738, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : AQM Put Message (257)
Eyecatcher . . . : ALRH                               Version . . . . : 1
LogRecdLen . . . : 718                               LogRecdOwnc . . . : 256   (AQM)
XTranid . . . . : TranType: XA
  XID: formatID 5067085, gtrid length 14, bqual_length 4
      gtrid [3270BDB40000102374657374716D]
      bqual [00000001]
QueueName . . . . : Queue2
Qid . . . . . : {Hash 184842943, Counter: 2}
ThisLSN . . . . : <0:0:0:0>
PrevLSN . . . . : <0:0:0:55324>

Version . . . . . : 3
Spclndex . . . . : 1
PrevLink.Locn . . : 36                               PrevLink.Length : 8
PrevDataLink . . : {High 0, Low 2048}
Data.Locn . . . . : 2048                             Data.Length . . . : 494
Data . . . . . :
00000: 41 51 52 48 00 00 00 04 FF FF FF FF FF FF FF FF  AQRH.....
00016: 00 00 00 00 00 00 00 00 00 00 00 01 00 01 01 C0  .....&#192;
00032: 00 00 00 00 00 00 00 00 01 00 00 00 2A 00 00 00 00  .....*....
00048: 00 00 00 01 41 4D 51 20 74 65 73 74 71 6D 20 20  ....AMQ testqm
00064: 20 20 20 20 33 80 2D D2 00 00 10 13 00 00 00 00  3E-#30;.....
00080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00096: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00112: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01  .....
00128: 00 00 00 00 00 00 00 00 2A 00 00 00 00 00 00 00  .....*.....
00144: 00 00 00 00 00 00 00 00 C9 2C B8 3E E8 FF FF FF FF  .....&#26;,&#184;>.....
00160: 4D 44 20 20 00 00 00 01 00 00 00 00 00 00 00 08  MD .....
00176: 00 00 00 00 00 00 01 11 00 00 03 33 20 20 20 20  .....3
00192: 20 20 20 20 00 00 00 00 00 00 00 01 20 20 20 20  .....
00208: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00224: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00240: 20 20 20 20 20 20 20 20 20 20 20 20 74 65 73 74  .....
00256: 71 6D 20 20 20 20 20 20 20 20 20 20 20 20 20 20  qm          test
00272: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00288: 20 20 20 20 20 20 20 20 20 20 20 20 73 62 6F 6C  .....          sbol
00304: 61 6D 20 20 20 20 20 20 04 37 34 38 30 00 00 00  am          .7480...
00320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00336: 00 00 00 00 00 00 00 00 20 20 20 20 20 20 20 20  .....
00352: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  .....
00368: 20 20 20 20 20 20 20 20 20 00 00 00 06 75 74 7A 61  .....          ....utza
00384: 70 69 20 20 20 20 20 20 20 20 20 20 20 20 20 20  pi
00400: 20 20 20 20 20 20 20 20 31 39 39 37 30 35 31 39  .....          19970519
00416: 31 30 34 34 35 38 37 32 20 20 20 20 00 00 00 00  10445872  ....
00432: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
00448: 41 6E 6F 74 68 65 72 20 70 65 72 73 69 73 74 65  Another persiste
00464: 6E 74 20 6D 65 73 73 61 67 65 20 70 75 74 20 75  nt message put u
00480: 6E 64 65 72 20 73 79 6E 63 70 6F 69 6E 74  nder syncpoint

```

그림 18. dmpmqlog 출력 예 (8/13)

dmpmqlog 사용

```
LOG RECORD - LSN <0:0:0:56282>
*****

HLG Header: lreclsize 216, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Prepare Transaction (770)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 196                      LogRecdOwnc . . . : 768      (ATM)
XTranid . . . . . : TranType: XA
  XID: formatID 5067085, gtrid_length 14, bqual_length 4
  gtrid [3270BDB40000102374657374716D]
  bqual [00000001]
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:55544>

Version . . . . . : 1

LOG RECORD - LSN <0:0:0:56498>
*****

HLG Header: lreclsize 708, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : Transaction Prepared (1538)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 688                      LogRecdOwnc . . . : 1536    (T)
XTranid . . . . . : TranType: XA
  XID: formatID 5067085, gtrid_length 14, bqual_length 4
  gtrid [3270BDB40000102374657374716D]
  bqual [00000001]
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

Id . . . . . : TLPR
Version . . . . : 1                      Flags . . . . . : 1
Count . . . . . : 3

Participant Entry 0
RMId . . . . . : 0                      State . . . . . : 2

Participant Entry 1
RMId . . . . . : 1                      State . . . . . : 2

Participant Entry 2
RMId . . . . . : 2                      State . . . . . : 2
```

그림 18. dmpmqlog 출력 예 (9/13)


```

LOG RECORD - LSN <0:0:0:57206>
*****

HLG Header: lreclsize 216, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Commit Transaction (774)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 196                      LogRecdOwnr . . . : 768   (ATM)
XTranid . . . . . : TranType: XA
    XID: formatID 5067085, gtrid_length 14, bqual_length 4
        gtrid [3270BDB40000102374657374716D]
        bqual [00000001]
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:56282>

Version . . . . . : 1
LOG RECORD - LSN <0:0:0:57440>
*****

HLG Header: lreclsize 224, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : Transaction Forget (1539)
Eyecatcher . . . : ALRH                      Version . . . . . : 1
LogRecdLen . . . : 204                      LogRecdOwnr . . . : 1536  (T)
XTranid . . . . . : TranType: XA
    XID: formatID 5067085, gtrid_length 14, bqual_length 4
        gtrid [3270BDB40000102374657374716D]
        bqual [00000001]
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

Id . . . . . : TLFG
Version . . . . . : 1                      Flags . . . . . : 0

```

그림 18. dmpmqlog 출력 예 (10/13)

dmpmqlog 사용

```
LOG RECORD - LSN <0:0:0:58120>
*****

HLG Header: lreclsize 212, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ALM Start Checkpoint (1025)
Eyecatcher . . . : ALRH                               Version . . . . . : 1
LogRecdLen . . . : 192                               LogRecdOwnr . . . : 1024 (ALM)
XTranid . . . . . : TranType: NULL
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

No data for Start Checkpoint Record

LOG RECORD - LSN <0:0:0:58332>
*****

HLG Header: lreclsize 220, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ATM Transaction Table (773)
Eyecatcher . . . : ALRH                               Version . . . . . : 1
LogRecdLen . . . : 200                               LogRecdOwnr . . . : 768 (ATM)
XTranid . . . . . : TranType: NULL
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

Version . . . . . : 1
TranCount . . . . : 0
```

그림 18. dmpmqlog 출력 예 (11/13)

LOG RECORD - LSN <0:0:0:58552>

HLG Header: lreclsize 1836, version 1, rmid 0, eyecatcher HLRH

```

LogRecdType . . : Transaction Participants (1537)
Eyecatcher   . . : ALRH                      Version . . . . : 1
LogRecdLen   . . : 1816                      LogRecdOwnc . . : 1536 (T)
XTranid      . . : TranType: NULL
QueueName    . . : NULL
Qid          . . : {NULL_QID}
ThisLSN      . . : <0:0:0:0>
PrevLSN      . . : <0:0:0:0>

```

```

Id . . . . . : TLPH
Version . . . : 1                      Flags . . . . . : 3
Count . . . . : 2

```

Participant Entry 0

```

RMName . . . . : DB2 MQBankDB
RMId . . . . . : 1
SwitchFile . . : /Development/sbolam/build/devlib/tstxasw
XAOpenString . :
XACloseString . :

```

Participant Entry 1

```

RMName . . . . : DB2 MQFeeDB
RMId . . . . . : 2
SwitchFile . . : /Development/sbolam/build/devlib/tstxasw
XAOpenString . :
XACloseString . :

```

LOG RECORD - LSN <0:0:0:60388>

HLG Header: lreclsize 236, version 1, rmid 0, eyecatcher HLRH

```

LogRecdType . . : ALM End Checkpoint (1026)
Eyecatcher   . . : ALRH                      Version . . . . : 1
LogRecdLen   . . : 216                      LogRecdOwnc . . : 1024 (ALM)
XTranid      . . : TranType: NULL
QueueName    . . : NULL
Qid          . . : {NULL_QID}
ThisLSN      . . : <0:0:0:0>
PrevLSN      . . : <0:0:0:0>

```

```

ChkPtLSN . . . : <0:0:0:58120>
OldestLSN . . . : <0:0:0:0>
MediaLSN . . . : <0:0:0:0>

```

그림 18. dmpmqlog 출력 예 (12/13)

dmpmqlog 사용

```
LOG RECORD - LSN <0:0:0:60624>
*****

HLG Header: lrecsize 240, version 1, rmid 0, eyecatcher HLRH

LogRecdType . . . : ALM Stop Queue Manager (1028)
Eyecatcher . . . : ALRH                               Version . . . . . : 1
LogRecdLen . . . : 220                               LogRecdOwnc . . . : 1024 (ALM)
XTranid . . . . . : TranType: NULL
QueueName . . . . : NULL
Qid . . . . . : {NULL_QID}
ThisLSN . . . . . : <0:0:0:0>
PrevLSN . . . . . : <0:0:0:0>

Version . . . . . : 1
StopDate . . . . : 19970519                          StopTime . . . . : 10490868
SessionNumber . . : 0                                ForceFlag . . . . : Quiesce

AMQ7702: DMPMQLOG command has finished successfully.
```

그림 18. dmpmqlog 출력 예 (13/13)

166 페이지의 그림 18에 대한 참고 사항

1. *Log File Header*의 *headlsn* 값은 <0:0:0:58120>입니다. 이것은 다른 시작 LSN을 요청하지 않은 덤프가 시작된 곳입니다.
2. *nextlsn*은 <0:0:0:60864>입니다. 이것은 큐 관리자가 다음에 재시작될 때 기록할 첫 번째 로그 레코드의 LSN이 됩니다.
3. *HeadExtentID*는 1로, 이는 현재 로그 파일 S0000001.LOG에 상주하는 로그의 헤드를 나타냅니다.
4. 형식화된 첫 번째 로그 레코드는 *Start Checkpoint* 로그 레코드입니다. 체크포인트는 *End CheckPoint* 레코드가 <0:0:0:46448>에 있을 때까지 여러 로그 레코드로 퍼집니다.
5. 체크포인트에서 기록된 레코드 중 하나는 <0:0:0:44594>에 있는 *Transaction Participants* 로그 레코드입니다. 이것은 큐 관리자에 의해 통합된 전역 트랜잭션에 참여하는 자원 관리자들에게 대해 자세히 보여줍니다.
6. <0:0:0:52262>에 있는 *Start Transaction* 로그 레코드는 트랜잭션의 시작을 표시합니다. *XTranid*는 이것이 MQ 갱신만 포함하는 로컬 트랜잭션을 나타내는 MQI의 *TranType*을 보여줍니다.
7. 다음 로그 레코드는 트랜잭션을 시작한 동기점 하에 지속 MQPUT을 기록하는 *Put Message* 로그 레코드입니다. MQPUT이 큐 *Queue1*에 대해 수행되었고 메시지 데이터는 *Persistent message put under syncpoint*로 기록됩니다. 이 메시지는 *SpcIndex 1*이 할당되는데, 이는 이 메시지의 이후 MQGET와 일치됩니다.

8. LSN <0:0:0:53458>에 있는 다음 로그 레코드도 *Put Message* 레코드입니다. 이 지속 메시지는 다른 큐인 *Queue2*에 넣었지만(*put*), *XTranid*가 *NULL*이어서 동기점 하에 수행되지 않았습니다. 이것도 *SpcIndex*가 1이며 이는 이 특수 큐의 고유한 ID입니다.
9. LSN <0:0:0:54192>에 있는 다음 로그 레코드는 동기점 하에 넣은(*put*) 메시지를 요약합니다.
10. 로그 레코드 <0:0:0:54408> 및 <0:0:0:54628>에서, *Queue1* 큐에 대한 동기점 하에 MQGET에 의해 새 트랜잭션이 시작됩니다. *Get Message* 로그 레코드의 *SpcIndex*는 1로, 이는 <0:0:0:52262>의 *Queue1*에 넣은(*put*) 동일 메시지였음을 나타냅니다.
11. 다음 로그 레코드는 다른 *Put Message* 로그 레코드에 의해 *Queue2*에 넣은(*put*) 메시지를 가져옵니다(*get*).
12. 동기점 하의 MQGET는 <0:0:0:55108>에 있는 *Commit Transaction* 로그 레코드에 의해 표시된 대로 요약되었습니다.
13. 마지막으로 MQBEGIN은 <0:0:0:55324>에 있는 *Start Transaction* 로그 레코드에서 전역 트랜잭션을 시작하는 데 사용됩니다. 이 로그 레코드의 *XTranid*는 *TranType*이 XA입니다.
14. 다음 *Put Message*는 *Queue2*에 넣은(*put*) 지속 메시지를 기록합니다. 이것은 이전 로그 레코드와 동일한 *XTranid*를 공유합니다.
15. *Transaction Prepared* 로그 레코드가 이 *Xtranid*에 대해 기록될 경우, 트랜잭션은 전체적으로 요약되어야 합니다. 그러한 로그 레코드가 없으면 트랜잭션이 롤백된 표시로 볼 수 있습니다. 이 경우, *Transaction Prepared* 로그 레코드가 <0:0:0:56498>에서 발견됩니다. 이것은 큐 관리자 자체를 *RMIId*가 0인 참여자로 기록합니다. 두 개의 추가 참여자가 있고, *RMIIds* 1과 2를 이전 *Transaction Participants* 로그 레코드와 대조할 수 있습니다.
16. 요약 단계 중에 큐 관리자의 XA 트랜잭션 관리자 구성요소는 참여자로부터 개별 응답을 기록하지 않습니다. 로그는 큐 관리자 갱신이 요약되었는지의 여부만 나타냅니다. <0:0:0:57206>에 있는 *Commit Transaction* 로그 레코드는 메시지가 *Queue2*로 요약되었음을 나타냅니다.
17. <0:0:0:57440>에 있는 *Transaction Forget*은 요약 결정이 다른 두 개의 자원 관리자라도 전달되었음을 나타냅니다. 이러한 자원 관리자의 갱신 요약 실패는 큐 관리자의 오류 로그에서 진단됩니다.

dmpmqlog 사용

제12장 이름 서비스 사용

이름 서비스는 하나의 큐 관리자에 연결된 응용프로그램이 로컬 큐를 열 수 있도록 하는 설치 가능 서비스입니다. 실제로 이러한 큐는 SCOPE 속성이 CELL로 설정된 다른 시스템의 다른 큐 관리자에 정의된 큐입니다.

응용프로그램은 이 방식으로 열린 큐에 있는 리모트 큐에 대해 허용된 모든 조작을 수행할 수 있습니다. 여기서는 DCE를 사용하지 않는 고유 구성요소를 작성할 수 있는 상태이더라도 분산 컴퓨팅 환경(DCE)을 사용합니다.

제공된 이름 서비스 구성요소를 사용하려면, 이름 서비스와 해당되는 설치 구성요소를 큐 관리자에 대해 정의해야 합니다. 큐 관리자 구성 파일(qm.ini)에서 적절한 스탠지를 삽입하면 됩니다. 자세한 내용은 *MQSeries Programmable System Management*를 참조하십시오. 일부 DCE를 구성해야 할 수도 있습니다.

다른 큐 관리자에 있는 큐를 공유하기 위한 DEC 사용

큐 관리자가 분산 컴퓨팅 환경(DCE) 셀 내에 있는 노드에 있으면, 그 큐 관리자들이 큐를 공유하도록 설정할 수 있습니다. 그러면 응용프로그램은 하나의 큐 관리자에 연결하고도 다른 노드의 다른 큐 관리자에 있는 큐를 열 수 있습니다. 응용프로그램은 큐가 실제로 다른 큐 관리자에 상주하는 것으로 인식하지 않습니다(보통, 큐 관리자는 큐가 해당되는 큐 관리자에 존재하지 않을 경우에 로컬 응용프로그램에서의 열기 요청을 거부합니다).

공유 큐에 대한 구성 작업

이 절에서는 DCE 셀 내에 있는 노드에 상주하는 큐 관리자에 대해 공유 큐를 설정하는 방법에 대해 설명합니다.

각 큐 관리자에 대해 다음을 수행하십시오.

1. 필요한 이름 서비스 스탠지를 큐 관리자 구성 파일에 추가하여 이름 서비스를 구성하십시오. 이 스탠지의 내용은 *MQSeries Programmable System Management*에 설명되어 있습니다. 이름 서비스를 실행하려면, 큐 관리자를 재시작해야 합니다.
2. **endmqm** 명령을 사용하여 실행 중인 큐 관리자를 정지하십시오.
3. **strmqm** 명령을 사용하여 큐 관리자를 재시작하십시오.
4. 큐 관리자 사이의 메시징을 위한 채널을 설정하십시오. 72 페이지의 『리모트 관리를 위한 채널 및 트랜스미션 큐 준비』를 참조하십시오.

큐 공유

공유할 모든 큐에 대해 SCOPE 속성을 CELL로 지정하십시오. 예를 들어, 다음 MQSC 명령을 사용하십시오.

```
DEFINE QLOCAL(GREY.PUBLIC.QUEUE) SCOPE(CELL)
```

또는

```
ALTER QLOCAL(PINK.LOCAL.QUEUE) SCOPE(CELL)
```

작성되거나 변경된 큐는 DCE 셀의 노드에 있는 큐 관리자에 속해야 합니다.

DCE 구성

공된 이름 서비스 구성요소를 사용하려면, OSF 분산 컴퓨팅 환경(DCE)이 설치해야 합니다. 이 서비스는 하나의 큐 관리자에 연결되어 있는 응용프로그램이 같은 DCE 셀에서 다른 큐 관리자에 속해 있는 큐를 열 수 있도록 합니다.

제공된 이름 서비스를 실행할 수 있도록 하는 DCL 셸 스크립트 예가 `mq_examples:dcesetu.com`에 제공되어 있습니다.

제13장 MQSeries 구성

이 장에서는 개인 큐 관리자나 노드의 동작을 변경하여 설치 필요사항에 맞게 변경하는 방법에 대해 설명합니다.

MQSeries를 제어하는 일련의 구성 속성(또는 매개변수)에 지정된 값을 수정하여 MQSeries 구성 정보를 변경할 수 있습니다.

이 구성 정보를 변경하는 방법과 MQSeries에서 변경사항이 저장되는 장소는 플랫폼마다 다릅니다. Compaq OpenVMS용 MQSeries의 사용자는 **MQSeries** 구성 파일을 편집하여 구성 정보를 변경할 수 있습니다.

이 장의 내용은 다음과 같습니다.

- 185 페이지의 『MQSeries 구성 정보 변경에 대한 속성』에서 MQSeries 구성 정보를 수정하기 위해 사용할 수 있는 속성에 대해 설명합니다.
- 191 페이지의 『큐 관리자 구성 정보 변경』에서 큐 관리자 구성 정보를 수정하기 위해 사용할 수 있는 속성에 대해 설명합니다.
- 200 페이지의 『mqs.ini 및 qm.ini 파일 예』에서 Compaq OpenVMS용 MQSeries에 대한 mqs.ini 및 qm.ini 파일의 예를 제공합니다.

MQSeries 구성 파일

Compaq OpenVMS용 MQSeries의 사용자는 다음에서 MQSeries 구성 속성을 수정할 수 있습니다.

- 노드에서 전체적으로 MQSeries를 변경하기 위한 MQSeries 구성 파일(mqs.ini). 노드에는 각각 하나의 mqs.ini 파일이 있습니다.
- 특정 큐 관리자를 변경하기 위한 큐 관리자 구성 파일(qm.ini). 노드의 각 큐 관리자에 대해 하나의 qm.ini 파일이 있습니다.

구성 파일(스탠자 파일 또는 *.ini* 파일이라고도 함)에는 하나 이상의 스탠자가 있습니다. 스탠자는 공통 함수를 가지고 있거나 시스템의 일부(예: 로그 함수, 채널 함수 및 설치 가능 서비스)를 정의하는 파일의 일련의 행입니다.

구성 파일에 대한 변경사항은 다음에 큐 관리자를 시작할 때까지 적용되지 않습니다.

구성 파일 편집

구성 파일을 편집하기 전에, 해당 파일을 백업하여 필요할 경우에 다시 사용할 수 있도록 하십시오.

구성 파일

다음 중 한가지 방법으로 구성 파일을 편집할 수 있습니다.

- 노드에서 큐 관리자의 구성을 변경하는 명령을 사용하여 자동으로 편집
- 표준 텍스트 편집기를 사용하여 수동으로 편집

설치 후에 MQSeries 구성 파일에서 디폴트 값을 편집할 수 있습니다.

구성 파일 속성에서 올바르지 않은 값을 설정할 경우, 값은 무시되며 연산자 메시지가 발행되어 문제점을 표시합니다(결과는 속성을 전체적으로 누락시키는 것과 같습니다).

새로운 큐 관리자를 작성할 때, 다음을 수행해야 합니다.

- MQSeries 구성 파일 백업
- 새로운 큐 관리자 구성 파일 백업

구성 파일을 편집해야 할 경우

예를 들어, 다음과 같은 경우에 구성 파일을 편집해야 합니다.

- 구성 파일이 손실되었을 경우. 가능하다면 백업에서 복구하십시오.
- 하나 이상의 큐 관리자를 새로운 디렉토리로 이동하려는 경우.
- 디폴트 큐 관리자를 변경하려는 경우. 이는 실수로 기존 큐 관리자를 삭제했을 경우에 수행하게 됩니다.
- IBM 지원 센터에서 구성 파일을 편집하도록 권고했을 경우.

구성 파일 우선순위

구성 파일의 속성 값은 다음 우선순위에 따라 설정됩니다.

- 명령행에 입력된 매개변수는 구성 파일에 정의된 값보다 우선합니다.
- qm.ini 파일에 정의된 값은 mqs.ini 파일에 정의된 값보다 우선합니다.

구성 파일에 대한 변경사항 구현

구성 파일을 편집할 경우, 변경사항은 큐 관리자에 의해 즉시 구현되지 않습니다. MQSeries 구성 파일에 대한 변경사항은 MQSeries가 시작된 경우에만 구현됩니다. 큐 관리자 구성 파일에 대한 변경사항은 큐 관리자가 시작될 때 구현됩니다. 변경할 때 큐 관리자가 실행 중이면, 시스템에서 변경사항이 인식되도록 큐 관리자를 정지한 후 재시작해야 합니다.

MQSeries 구성 파일, mqs.ini

MQSeries 구성 파일인 mqs.ini에는 노드상의 모든 큐 관리자에 관련되는 정보가 들어 있습니다. 이것은 설치시 자동으로 작성됩니다. 특히, mqs.ini 파일은 각 큐 관리자와 연관되는 데이터를 찾는 데 사용됩니다.

mqs.ini 파일은 디폴트로 데이터 디렉토리인 **MQS_ROOT:[MQM]**에 저장됩니다.

mqs.ini 파일에는 다음 사항이 있습니다.

- 큐 관리자의 이름
- 디폴트 큐 관리자의 이름
- 각 파일과 연관되는 파일 위치

mqs.ini 내용에 대한 자세한 정보는 『MQSeries 구성 정보 변경에 대한 속성』을 참조하십시오.

큐 관리자 구성 파일 qm.ini

큐 관리자 구성 파일인 qm.ini에는 특정 큐 관리자에 관련되는 정보가 들어 있습니다. 각 구성 파일마다 하나의 큐 관리자 구성 파일이 있습니다. qm.ini 파일은 연관되는 큐 관리자를 작성할 때 자동으로 작성됩니다.

qm.ini 파일은 큐 관리자가 사용하는 디렉토리 트리의 루트에 위치합니다.

예를 들어, Compaq OpenVMS용 MQSeries에서 QMNAME이라는 큐 관리자에 대한 구성 파일의 경로와 이름은 다음과 같습니다.

```
MQS_ROOT:[MQM.QMGRS.QMNAME]QM.INI
```

주: 큐 관리자 이름은 최대 48자까지 가능합니다. 그렇다고 해서 그 이름이 올바르거나 고유하다고 보장할 수는 없습니다. 그러므로, 디렉토리 이름은 큐 관리자 이름을 기초로 생성됩니다. 이러한 프로세스를 이름 변환이라고 합니다. 설명을 보려면 25 페이지의 『MQSeries 파일 이름 이해』를 참조하십시오.

qm.ini에 대한 자세한 정보는 191 페이지의 『큐 관리자 구성 정보 변경』을 참조하십시오.

MQSeries 구성 정보 변경에 대한 속성

다음 속성 그룹이 mqs.ini에 표시됩니다.

- AllQueueManagers 스탠자
- 187 페이지의 『ClientExitPath 스탠자』
- 187 페이지의 『DefaultQueueManager 스탠자』
- 187 페이지의 『ExitProperties 스탠자』
- 188 페이지의 『LogDefaults 스탠자』
- 190 페이지의 『QueueManager 스탠자』

mqs.ini 샘플은 200 페이지의 『mqs.ini 및 qm.ini 파일 예』에 나와 있습니다.

AllQueueManagers 스탠자

AllQueueManagers 스탠자는 다음을 지정할 수 있습니다.

- 큐 관리자와 연관되는 파일이 저장되는 qmgrs 디렉토리의 경로
- EBCDIC 형식 데이터를 ASCII 형식으로 변환하는 방법

MQSeries 구성 파일 변경

DefaultPrefix=directory_name

이 속성은 qmgrs 디렉토리의 경로를 지정합니다. 그 경로에서 큐 관리자 데이터가 보존됩니다.

큐 관리자에 대한 디폴트 접두부를 변경할 경우, 설치시 작성된 디렉토리 구조를 복제해야 합니다(343 페이지의 『부록C. 디렉토리 구조』 참조).

특히, qmgrs 구조를 작성해야 합니다. 디폴트 접두부를 변경하기 전에 MQSeries를 정지하고, 구조가 새로운 위치로 이동되고 접두부가 변경된 후에만 MQSeries를 재시작해야 합니다.

디폴트 접두부를 변경하는 다른 방법으로는, 논리 MQSPREFIX를 사용하여 **crtmqm** 명령에 대한 DefaultPrefix를 대체할 수 있습니다.

ConvEBCDICNewline=NL_TO_LF|TABLE|ISO

EBCDIC 코드 페이지에는 ASCII 코드 페이지에서 지원되지 않는 줄바꾸기(NL) 문자가 있습니다. ASCII의 일부 ISO 변형에도 그와 같은 것이 있습니다.

EBCDIC NL 문자를 ASCII 형식으로 변환할 때 MQSeries가 사용할 방법을 지정하려면 ConvEBCDICNewline 속성을 사용하십시오.

NL_TO_LF

모든 EBCDIC의 ASCII 변환에 대해 EBCDIC NL 문자(X'15')가 ASCII 줄바꾸기 문자, LF(X'0A')로 변환되도록 하려면 NL_TO_LF를 지정하십시오.

NL_TO_LF가 디폴트입니다.

TABLE

모든 EBCDIC의 ASCII 변환에 대해 플랫폼에서 사용되는 변환 테이블에 따라 EBCDIC NL 문자가 변환되도록 하려면, TABLE을 지정하십시오.

이 변환 유형의 효과는 플랫폼과 언어에 따라 다양할 수 있습니다. 같은 플랫폼에서 다른 CCSID를 사용할 경우에도 작동이 다를 수 있습니다.

ISO

다음은 원할 경우에 ISO를 지정하십시오.

- TABLE 메소드를 사용하여 ISO CCSID를 변환하려는 경우
- NL_TO_LF 메소드를 사용하여 다른 모든 CCSID를 변환하려는 경우

가능한 ISO CCSID를 표7에서 보여줍니다.

표7. 가능한 ISO CCSID 목록

CCSID	코드 세트
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7

표 7. 가능한 ISO CCSID 목록 (계속)

CCSID	코드 세트
916	ISO8859-8
920	ISO8859-9
1051	roman8

ASCII CCSID가 ISO 서브세트가 아닐 경우, ConvEBCDICNewline 디폴트는 NL_TO_LF입니다.

데이터 변환에 대한 자세한 정보는 *MQSeries Application Programming Guide* 또는 82 페이지의 『데이터 변환』을 참조하십시오.

ClientExitPath 스탠자

ClientExitPath 스탠자는 클라이언트의 채널 종료 위치에 대한 디폴트 경로를 지정합니다.

ExitsDefaultPath=defaultprefix

ExitsDefaultPath 속성은 플랫폼에 대한 디폴트 접두부를 지정합니다.

DefaultQueueManager 스탠자

DefaultQueueManager 스탠자는 노드에 대한 디폴트 큐 관리자를 지정합니다.

Name=default_queue_manager

디폴트 큐 관리자가 큐 관리자 이름이 명확하게 지정되지 않은 명령을 처리합니다. DefaultQueueManager 속성은 새로운 디폴트 큐 관리자를 작성할 경우에 자동으로 갱신됩니다. 부주의로 새로운 디폴트 큐 관리자를 작성한 후 다시 원래 큐 관리자로 되돌리려면, DefaultQueueManager 속성을 수동으로 변경해야 합니다.

ExitProperties 스탠자

ExitProperties 스탠자는 큐 관리자 종료 프로그램에서 사용하는 구성 옵션을 지정합니다.

CLWLMode=SAFE|FAST

클러스터 워크로드 종료인 CLWL은 MQAPI 호출(MQOPEN 또는 MQPUT 등과 같은)에 대한 응답에서 열리는 클러스터의 클러스터 큐를 지정할 수 있게 합니다. CLWL 종료는 CLWLMode 속성에서 지정하는 값에 따라 FAST 모드나 SAFE 모드에서 실행됩니다. CLWLMode 속성을 지정하지 않으면, 클러스터 워크로드 종료는 SAFE 모드에서 실행됩니다.

SAFE

SAFE 옵션은 CLWL 종료는 큐 관리자에 대해 분리된 프로세스에서 실행됨을 지정합니다. 이것이 디폴트입니다.

MQSeries 구성 파일 변경

SAFE 모드에서 실행할 때 사용자 작성 CLWL에서 문제점이 발생하면, 다음 사항이 발생합니다.

- CLWL 서버 프로세스(amqzlw0)가 실패합니다.
- 큐 관리자가 CLWL 서버 프로세스를 재시작합니다.
- 오류 로그가서 사용자에게 오류를 보고합니다. MQAPI 호출이 진행 중일 경우, 잘못된 리턴 코드 형식으로 통지를 수신합니다.

큐 관리자 무결성은 보존됩니다.

주: 분리된 프로세스에서의 CLWL 종료 실행과 연관되는 오버헤드가 발생하며, 이는 성능에 영향을 줄 수 있습니다.

FAST

큐 관리자 프로세스에서 클러스터 종료를 인라인으로 실행하려면 FAST를 지정하십시오.

이 옵션을 지정하면 SAFE 모드에서의 실행과 연관되는 오버헤드를 피할 수 있기 때문에 성능이 개선되지만, 큐 관리자 무결성은 감소됩니다. 그러므로, CLWL 종료에 문제점이 없고 특별히 성능의 오버헤드가 우려될 경우에만 FAST 모드에서 CLWL 종료를 실행해야 합니다.

CLWL 종료가 FAST 모드에서 실행 중일 때 문제점이 발생하면, 큐 관리자는 실패하게 되며 큐 관리자 무결성이 손상될 수 있는 위험이 발생합니다.

LogDefaults 스탠자

LogDefaults 스탠자는 노드에 대한 디폴트 로그 속성을 지정합니다. 로그 속성은 큐 관리자를 작성할 때 디폴트 값으로 사용되지만, **crtmqm** 명령에서 로그 속성을 지정할 경우에 대체할 수 있습니다. 이 명령에 대한 자세한 내용은 266 페이지의 『crtmqm(큐 관리자 작성)』을 참조하십시오.

큐 관리자가 작성되면, 그 큐 관리자에 대한 로그 속성을 **qm.ini** 파일의 해당 로그 스탠자로 읽을 수 있습니다.

DefaultPrefix 속성(AllQueueManagers 스탠자에 있음) 및 *LogPath* 속성(LogDefaults 스탠자에 있음)을 사용하면 큐 관리자와 해당되는 로그를 서로 다른 물리적 드라이브에 있도록 할 수 있습니다. 동일한 드라이브에 있도록 하는 것이 디폴트이지만, 이 방법을 사용하는 것이 좋습니다.

로그 크기 계산에 대한 정보는 151 페이지의 『로그 크기 계산』을 참조하십시오.

주: 다음 매개변수 목록에 제공된 한계는 MQSeries에서 설정된 한계입니다. 운영 체제 한계는 가능한 최대 로그 크기를 줄일 수도 있습니다.

LogPrimaryFiles=3|2-62

1차 로그 파일은 나중을 위해 작성시 할당된 로그 파일입니다.

가능한 1차 로그 파일 수의 최소값은 2이고 최대값은 62입니다. 디폴트는 3입니다.

1차 및 2차 로그 파일의 총 개수는 63을 초과할 수 없으며 3이상이어야 합니다.

이 값은 큐 관리자가 작성될 때 **crtmqm** 명령의 **-lp** 매개변수에 의해 대체됩니다.

LogSecondaryFiles=2|1-61

2차 로그 파일은 1차 파일이 모두 소모될 때 할당되는 로그 파일입니다.

2차 로그 파일 수의 최소값은 1이고 최대값은 61입니다. 디폴트는 2입니다.

1차 및 2차 로그 파일의 총 개수는 63을 초과할 수 없으며 3이상이어야 합니다.

이 값은 큐 관리자가 작성될 때 **crtmqm** 명령의 **-ls** 매개변수에 의해 대체됩니다.

LogFilePages=number

로그 데이터가 로그 파일에 일련의 파일에 보유됩니다. 로그 파일 크기는 4 KB 페이지 단위로 지정됩니다.

Compaq OpenVMS용 MQSeries의 경우, 디폴트 로그 파일 페이지 수는 1024이고, 로그 파일 크기는 4 MB 단위로 제공됩니다. 로그 파일 페이지 수의 최소값은 64이고 최대값은 16 384입니다.

이 값은 큐 관리자가 작성될 때 **crtmqm** 명령의 **-lf** 매개변수에 의해 대체됩니다.

LogType=CIRCULAR|LINEAR

LogType 속성은 사용될 유형을 정의하기 위해 사용됩니다. 디폴트는 CIRCULAR입니다.

CIRCULAR

시스템이 정지되었을 때 진행 중이었던 트랜잭션을 롤백하기 위해 로그를 사용하여 복구를 재시작하려는 경우 이 값을 사용하십시오.

순환 로깅에 대한 추가 설명은 146 페이지의 『순환 로깅』을 참조하십시오.

LINEAR

재시작 복구와 매체 또는 전달 복구(로그의 내용을 재생하여 유실되거나 손상된 데이터를 작성할 경우) 모두를 수행하려면 이 값을 사용하십시오.

선형 로깅에 대한 추가 설명은 147 페이지의 『선형 로깅』을 참조하십시오.

디폴트 로그 유형을 변경하려면, **mq5.ini** 파일에서 LogType 속성을 편집하면 됩니다. 또는, **crtmqm** 명령의 **-ll** 매개변수로 선형 로깅을 지정하여 디폴트를 대체할 수 있습니다. 큐 관리자가 작성되고 나면 로그 기록 방법을 변경할 수 없습니다.

MQSeries 구성 파일 변경

LogBufferPages=17|4-32

쓰기를 위해 버퍼 레코드에 할당된 메모리 양을 구성할 수 있습니다. 버퍼 크기는 4 KB 페이지 단위로 지정됩니다.

버퍼 페이지 수의 최소값은 4이고 최대값은 32입니다. 버퍼 수가 더 커지면 처리량이 더 높아집니다. 특히, 메시지의 크기가 큰 경우가 이에 해당합니다.

디폴트 버퍼 페이지 수는 17로, 이는 68 KB와 같습니다.

이 값은 큐 관리자를 작성하거나 시작할 때 검사하여 증가하거나 감소할 수 있습니다. 그러나 큐 관리자가 재시작할 때까지 변경한 값은 적용되지 않습니다.

LogDefaultPath=directory_name

큐 관리자에 대한 로그 파일이 상주하는 디렉토리를 지정할 수 있습니다. 디렉토리는 큐 관리자가 기록할 수 있는 로컬 디바이스에 존재해야 하며, 메시지 큐와 다른 드라이브에 있어야 합니다. 다른 드라이브를 지정하면 시스템 실패시 추가 보호가 제공됩니다.

Compaq OpenVMS용 MQSeries의 디폴트는 MQS_ROOT:[MQM.LOG]입니다.

또는, -ld 플래그를 사용하여 **crtmqm** 명령에서 디렉토리의 이름을 지정할 수 있습니다. 큐 관리자가 작성될 때, 하나의 디렉토리가 큐 관리자 디렉토리 아래에 작성되는데 이 디렉토리에 로그 파일이 위치하게 됩니다. 이 디렉토리의 이름은 큐 관리자 이름을 기초로 합니다. 그러면 로그 파일 경로가 고유해지며 디렉토리 이름 길이에 대한 제한사항도 따르게 됩니다.

crtmqm 명령에서 -ld를 지정하지 않을 경우, mqs.ini 파일에 있는 LogDefaultPath 속성 값이 디폴트로 사용되는데, 이것은 MQS_ROOT:[MQM.LOG]입니다.

큐 관리자 이름은 여러 큐 관리자가 서로 다른 로그 디렉토리를 사용하도록 로그 파일 디렉토리 이름에 추가됩니다.

큐 관리자가 작성되었을 때 LogPath 값이 qm.ini 파일의 Log 스탠자에서 작성되어, 큐 관리자의 로그 파일에 대한 완전한 디렉토리 이름을 제공합니다. 이 값은 큐 관리자가 시작되거나 삭제될 때 로그 파일을 찾는 데 사용됩니다.

QueueManager 스탠자

모든 큐 관리자에 대해 하나의 QueueManager 스탠자가 있습니다. 이 속성들은 큐 관리자 이름과, 그 큐 관리자와 연관되는 파일을 포함하는 디렉토리의 이름을 지정합니다. 이 디렉토리의 이름은 큐 관리자 이름을 기초로 하지만, 큐 관리자 이름이 올바른 파일 이름이 아니면 변환됩니다.

이름 변환에 대한 자세한 정보는 25 페이지의 『MQSeries 파일 이름 이해』를 참조하십시오.

Name=queue_manager_name

이 속성은 큐 관리자의 이름을 지정합니다.

Prefix=prefix

이 속성은 큐 관리자 파일이 저장되는 장소를 지정합니다. 디폴트로, 이것은 mqs.ini 파일에서 AllQueueManager 스탠자의 DefaultPrefix 속성에 지정된 값과 같습니다.

Directory=name

이 속성은 큐 관리자 파일이 저장되는 서브디렉토리의 이름을 지정합니다. 대체 접두부 값을 지정하지 않았을 경우 일반적으로 MQS_ROOT:[MQM.QMGRS] 아래에 지정됩니다. 이 이름은 큐 관리자 이름을 기초로 하지만, 중복되는 이름이 있거나 큐 관리자 이름이 올바른 파일 이름이 아니면 변환할 수 있습니다.

큐 관리자 구성 정보 변경

다음 속성 그룹은 주어진 큐 관리자에 특정한 qm.ini 파일에 표시되며, mqs.ini에 설정된 값을 대체하기 위해 사용됩니다.

- 『Service 스탠자』
- 192 페이지의 『ServiceComponent 스탠자』
- 193 페이지의 『Log 스탠자』
- 195 페이지의 『XAResourceManager 스탠자』
- 196 페이지의 『Channels 스탠자』
- 198 페이지의 『LU62 및 TCP 스탠자』
- 200 페이지의 『ExitPath 스탠자』

Service 스탠자

Service 스탠자는 설치 가능한 서비스의 이름과 해당 서비스에 대한 시작점 수를 지정합니다. 사용되는 서비스마다 하나의 Service 스탠자가 있어야 합니다.

서비스 내의 각 구성요소에 대해, ServiceComponent 스탠자가 있어야 하는데, 이 스탠자는 그 구성요소의 코드를 포함하는 모듈의 이름과 경로를 식별합니다. 자세한 정보는 ServiceComponent 스탠자를 참조하십시오.

Name=AuthorizationService|NameService

필요한 서비스의 이름을 지정합니다.

AuthorizationService

MQSeries의 경우, 권한 서비스 구성요소는 오브젝트 권한 관리자 또는 OAM으로 알려져 있습니다.

Compaq OpenVMS용 MQSeries에서, AuthorizationService 스탠자와 해당되는 연관 ServiceComponent 스탠자는 큐 관리자가 작성될 때 자동으로 추가되지만, 큐 관리자를 작성하기 전에 mqsnoaut 논리를 설정하여 mqsnoaut를

MQSeries 구성 파일 변경

통해 대체될 수 있습니다(자세한 정보는 89 페이지의 『오브젝트 권한 관리자 사용 안함』을 참조하십시오). 다른 ServiceComponent 스탠지는 수동으로 추가해야 합니다.

NameService

제공된 이름 서비스를 사용하려면 NameService 스탠자가 `qm.ini` 파일에 수동으로 추가되어야 합니다.

EntryPoints=number-of-entries

서비스에 대해 정의된 시작점 수를 지정합니다. 여기에는 초기화 및 종료 시작점이 포함됩니다.

설치 가능 서비스 및 구성요소에 대한 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.

일반적인 보안 서비스에 대한 자세한 정보는 85 페이지의 『제7장 MQSeries 오브젝트 보호』를 참조하십시오.

ServiceComponent 스탠자

ServiceComponent 스탠자는 해당되는 구성요소의 코드를 포함하는 모듈의 이름과 경로를 식별합니다.

각 서비스마다 여러 개의 ServiceComponent 스탠자가 있을 수 있지만, 각 ServiceComponent 스탠자는 해당되는 서비스 스탠자와 일치해야 합니다.

Compaq OpenVMS용 MQSeries에서, 권한 서비스 스탠자는 디폴트로 제시되며, 연관된 구성요소인 OAM이 활성화됩니다.

Service=service_name

필요한 서비스의 이름을 지정합니다. 이 이름은 Service 스탠자의 Name 속성에 지정된 값과 일치해야 합니다.

Name=component_name

서비스 구성요소의 설명적 이름을 지정합니다. 이 이름은 고유해야 하며, MQSeries 오브젝트의 이름에 대해 올바른 문자만 포함해야 합니다(예: 큐 이름). 이 이름은 서비스가 생성하는 연산자 메시지에서 발생합니다. 그러므로, 이 이름은 회사의 등록상표나 이와 유사한 구분 문자열로 시작하는 것이 좋습니다.

Module=module_name

이 구성요소의 코드를 포함할 모듈 이름을 지정합니다.

주: 전체 경로 이름을 지정하십시오.

ComponentDataSize=size

각 호출에서 구성요소에 전달되는 구성요소 데이터 영역의 크기를 바이트 단위로 지정합니다. 구성요소 데이터가 필요하지 않으면 0을 지정하십시오.

설치 가능 서비스 및 구성요소에 대한 자세한 정보는 *MQSeries Programmable System Management*를 참조하십시오.

Log 스탠자

Log 스탠자는 특정 큐 관리자에 대한 로그 속성을 지정합니다. 디폴트로, 이 속성들은 **crtmqm** 명령에서 특정 매개변수에 의해 대체되지 않는한 큐 관리자가 작성될 때 **mqm.ini** 파일의 LogDefaults 스탠자에 지정된 설정에서 상속됩니다. 자세한 정보는 188 페이지의 『LogDefaults 스탠자』 및 266 페이지의 『crtmqm(큐 관리자 작성)』을 참조하십시오.

이러한 특정 큐 관리자가 다른 큐 관리자와 다르게 구성되어야 할 경우에만 이 스탠자의 속성을 변경하십시오.

qm.ini 파일의 속성에 지정된 값은 큐 관리자가 시작될 때 읽혀집니다. 파일은 큐 관리자가 작성될 때 작성됩니다.

로그 크기 계산에 대한 정보는 151 페이지의 『로그 크기 계산』을 참조하십시오.

주: 다음 매개변수 목록에 제공된 한계는 MQSeries에서 설정된 한계입니다. 운영 체제 한계는 가능한 최대 로그 크기를 줄일 수도 있습니다.

LogPrimaryFiles=3|2-62

1차 로그 파일은 나중을 위해 작성시 할당된 로그 파일입니다.

가능한 1차 로그 파일 수의 최소값은 2이고 최대값은 62입니다. 디폴트는 3입니다.

1차 및 2차 로그 파일의 총 개수는 63을 초과할 수 없으며 3이상이어야 합니다.

값은 큐 관리자를 작성하거나 시작할 때 검사합니다. 큐 관리자가 작성된 후에 이를 변경할 수 있습니다. 그러나, 값 변경은 큐 관리자가 재시작할 때까지 적용되지 않으므로 즉시 효력이 발생하지는 않습니다.

LogSecondaryFiles=2|1-61

2차 로그 파일은 1차 파일이 모두 소모될 때 할당되는 로그 파일입니다.

2차 로그 파일 수의 최소값은 1이고 최대값은 61입니다. 디폴트는 2입니다.

1차 및 2차 로그 파일의 총 개수는 63을 초과할 수 없으며 3이상이어야 합니다.

값은 큐 관리자를 시작할 때 검사합니다. 이 값을 변경할 수는 있지만, 변경사항은 큐 관리자가 재시작할 때까지 적용되지 않으며, 재시작해도 즉시 효력이 발생하지는 않습니다.

LogFilePages=number

로그 데이터가 로그 파일에는 일련의 파일에 보유됩니다. 로그 파일 크기는 4 KB 페이지 단위로 지정됩니다.

MQSeries 구성 파일 변경

Compaq OpenVMS용 MQSeries에서, 디폴트 로그 파일 페이지 수는 1024이고, 로그 파일 크기는 4 MB 단위로 제공됩니다. 로그 파일 페이지 수의 최소값은 64 이고 최대값은 16 384입니다.

주: 큐 관리자 작성시 지정된 로그 파일 크기는 기존 큐 관리자에 맞게 변경할 수 있습니다.

LogType=CIRCULAR|LINEAR

LogType 속성은 큐 관리자에서 사용될 로그 기록 유형을 정의합니다. 그러나, 큐 관리자가 작성되고 나면 사용될 로그 기록 유형을 변경할 수 없습니다. 요구하는 로그 기록 유형의 큐 관리자를 작성하는 방법에 대해서는 188 페이지의 『LogDefaults 스탠자』에 있는 LogType 속성 설명을 참조하십시오.

CIRCULAR

시스템이 정지되었을 때 진행 중이었던 트랜잭션을 롤백하기 위해 로그를 사용하여 복구를 재시작하려는 경우 이 값을 사용하십시오.

순환 로깅에 대한 추가 설명은 146 페이지의 『순환 로깅』을 참조하십시오.

LINEAR

재시작 복구와 매체 또는 전달 복구(로그의 내용을 재생하여 유실되거나 손상된 데이터를 작성할 경우) 모두를 수행하려면 이 값을 사용하십시오.

선형 로깅에 대한 추가 설명은 147 페이지의 『선형 로깅』을 참조하십시오.

LogBufferPages=17|4-32

쓰기를 위해 버퍼 레코드에 할당된 메모리 양을 구성할 수 있습니다. 버퍼 크기는 4 KB 페이지 단위로 지정됩니다.

버퍼 페이지 수의 최소값은 4이고 최대값은 32입니다. 버퍼 수가 더 커지면 처리량이 더 높아집니다. 특히, 메시지의 크기가 큰 경우가 이에 해당합니다.

디폴트 버퍼 페이지 수는 17로, 이는 68 KB와 같습니다.

이 값은 큐 관리자를 작성하거나 시작할 때 검사하여 증가하거나 감소할 수 있습니다. 그러나 큐 관리자가 재시작할 때까지 변경한 값은 적용되지 않습니다.

LogPath=directory_name

큐 관리자에 대한 로그 파일이 상주하는 디렉토리를 지정할 수 있습니다. 디렉토리는 큐 관리자가 기록할 수 있는 로컬 디바이스에 존재해야 하며, 메시지 큐와 다른 드라이브에 있어야 합니다. 다른 드라이브를 지정하면 시스템 실패시 추가 보호가 제공됩니다.

디폴트는 MQS_ROOT:[MQM.LOG]입니다.

-ld 플래그를 사용하여 **crtmqm** 명령에서 디렉토리의 이름을 지정할 수 있습니다. 큐 관리자가 작성될 때, 하나의 디렉토리가 큐 관리자 디렉토리 아래에 작성되는데 이 디렉토리에 로그 파일이 위치하게 됩니다. 이 디렉토리의 이름은 큐 관리자 이

름을 기초로 합니다. 그러면 로그 파일 경로를 고유하게 할 수 있으며 디렉토리 이름 길이에 대한 제한사항을 따르게 됩니다.

crtmqm 명령에서 **-ld**를 지정하지 않을 경우, **mq5.ini** 파일에 있는 **LogDefaultPath** 속성 값이 사용됩니다.

주: Compaq OpenVMS용 MQSeries에서, 사용자 ID **mqm** 및 그룹 **mqm**은 로그 파일에 대해 완전한 권한을 가지고 있어야 합니다. 이 파일들의 위치를 변경할 경우, 이 권한을 사용자 스스로 부여해야 합니다. 이것은 로그 파일이 제품과 함께 제공된 디폴트 위치에 있을 경우에는 필요하지 않습니다.

XAResourceManager 스탠자

XAResourceManager 스탠자는 큐 관리자에 의해 통합된 전역 작업 단위에 관련될 자원 관리자를 지정합니다.

전역 작업 단위에 참여하는 자원 관리자의 각 인스턴스에 대해 **qm.ini**에서 하나의 XAResourceManager 스탠자가 요구됩니다. **mq5.ini**를 통해 디폴트 값은 제공되지 않습니다.

qm.ini에 XAResourceManager 속성을 추가하는 방법에 대한 자세한 정보는 130 페이지의 『데이터베이스 통합』을 참조하십시오.

Name=name(필수)

이 속성은 자원 관리자 인스턴스를 식별합니다.

Name 값의 길이는 최대 31자까지 가능하며 **qm.ini** 내에서 고유해야 합니다. XA-스위치 구조에서 정의한 대로 자원 관리자의 이름을 사용할 수 있습니다. 그러나, 같은 자원 관리자의 여러 인스턴스를 사용 중이면, 각 인스턴스에 대한 고유한 이름을 구성해야 합니다. 예를 들어, Name 문자열에 데이터베이스의 이름을 포함시켜서 고유성을 보장할 수 있습니다.

MQSeries는 메시지와 **dspmqrn** 명령의 출력에서 Name 값을 사용합니다.

연관된 큐 관리자가 시작되었고 자원 관리자 이름이 사용되고 있으면, 자원 관리자 인스턴스의 이름을 변경하거나 해당 항목을 **qm.ini**에서 삭제하지 않는 것이 좋습니다.

SwitchFile=name(필수)

이 속성은 자원 관리자의 XA 스위치 구조를 포함하는 로드 파일의 완전한 이름을 지정합니다.

XAOpenString=string(선택적)

이 속성은 자원 관리자의 **xa_open** 시작점으로 전달될 데이터 문자열을 지정합니다. 문자열 내용은 자원 관리자 자체에 따라 달라집니다. 예를 들어, 자원 관리자의 인스턴스가 액세스하려는 데이터베이스를 식별할 수 있습니다. 이 속성을 정의하는

MQSeries 구성 파일 변경

방법에 대한 자세한 정보는 137 페이지의 『Oracle용 XAResourceManager 구성 정보 추가』를 참조하고 적절한 문자열에 대해서는 자원 관리자 문서를 참조하십시오.

XACloseString=string(선택적)

이 속성은 자원 관리자의 xa_close 시작점으로 전달될 데이터 문자열을 지정합니다. 문자열 내용은 자원 관리자 자체에 따라 달라집니다. 이 속성을 정의하는 방법에 대한 자세한 정보는 137 페이지의 『Oracle용 XAResourceManager 구성 정보 추가』를 참조하고 적절한 문자열에 대해서는 데이터베이스 문서를 참조하십시오.

ThreadOfControl=THREAD|PROCESS

ThreadOfControl 속성에 설정된 값은 고유한 다중스레드 프로세스 중 하나에서 자원 관리자를 호출하기 위해 필요로 할 때 일련화 목적을 위해 큐 관리자에서 사용됩니다.

THREAD

자원 관리자가 전체적으로 “스레드 주의” 상태임을 의미합니다. 다중스레드 MQSeries 프로세스에서, XA 함수 호출은 동시에 여러 스레드로부터 외부 자원 관리자에 대해 수행될 수 있습니다.

PROCESS

자원 관리자가 “스레드 안전” 상태가 아님을 의미합니다. 다중스레드 MQSeries 프로세스에서 한번에 하나의 XA 함수만 자원 관리자에 대해 수행될 수 있습니다.

ThreadOfControl 항목은 다중스레드 응용프로그램 프로세스에서 큐 관리자에 의해 발행된 XA 함수 호출에 적용되지 않습니다. 일반적으로, 다른 스레드에 동시 작업 단위가 있는 응용프로그램은 각 자원 관리자에 의해 이러한 조작 모드가 지원 되도록 요구합니다.

Channels 스탠자

Channels 스탠자에는 채널에 대한 정보가 들어 있습니다.

MaxChannels=100|number

이 속성은 허용되는 최대 채널 수를 지정합니다. 디폴트는 100입니다.

MaxActiveChannels=MaxChannels_value

이 속성은 언제든지 활성화될 수 있는 최대 채널 수를 지정합니다. 디폴트는 MaxChannels 속성에 지정된 값입니다.

MaxInitiators=3|number

이 속성은 시작기의 최대 수를 지정합니다.

MQIBINDTYPE=FASTPATH|STANDARD

이 속성은 응용프로그램에 대한 바인딩을 지정합니다.

FASTPATH

채널이 MQCONNX FASTPATH를 사용하여 연결합니다. 즉, 에이전트 프로세스가 없습니다.

STANDARD

채널이 STANDARD를 사용하여 연결합니다.

AdoptNewMCA=NO|SVR|SDR|RCVR|CLUSRCVR|ALL|FASTPATH

MQSeries가 채널을 시작하려는 요청을 수신하지만 이미 같은 채널에 대해 amqcrsta 프로세스가 존재한다는 사실이 발견되면, 기존 프로세스는 새로운 것이 시작되기 전에 정지해야 합니다. AdoptNewMCA 속성은 기존 프로세스의 종료와 지정된 채널 유형에 대한 새로운 프로세스의 시동을 제어할 수 있게 합니다.

주어진 채널 유형에 대해 AdoptNewMCA 속성을 지정하지만 채널이 이미 실행 중이거나 새로운 채널이 실패할 경우,

1. 새로운 채널은 종료할 것을 요구하여 이전 채널을 정지시키려고 합니다.
2. 이전 채널 서버가 AdoptNewMCATimeout 대기 간격이 만기될 때까지 이 요청에 응답하지 않으면, 이전 채널 서버에 대한 프로세스(또는 스레드)는 강제 종료됩니다.
3. 이전 채널 서버가 2 단계 후에 종료되지 않았는데, 그 후 두 번째로 AdoptNewMCATimeout 대기 간격이 만기되면, MQSeries는 “CHANNEL IN USE” 오류와 함께 채널을 종료합니다.

다음 목록에서 하나 이상의 값을 쉼표나 공백으로 구분하여 지정합니다.

NO AdoptNewMCA 기능이 필요하지 않습니다. 이것이 디폴트입니다.

SVR 서버 채널 채택

SDR 송신자 채널 채택

RCVR

수신자 채널 채택

CLUSRCVR

클러스터 수신자 채널 채택

ALL FASTPATH 채널을 제외한 모든 채널 유형 채택

FASTPATH

FASTPATH 채널일 경우 채널 채택. 이것은 적절한 채널 유형(예: AdoptNewMCA=RCVR,SVR,FASTPATH)도 지정된 경우에만 발생합니다.

주!

AdoptNewMCA 속성은 큐 관리자의 내부 설계로 인해 FASTPATH 채널에 대해 예상치 못한 형태로 작동될 수 있습니다. 그러므로, FASTPATH 채널에 대해 AdoptNewMCA 속성을 사용할 때는 주의해야 합니다.

AdoptNewMCATimeout=60|1--3600

이 속성은 새로운 프로세스가 이전 프로세스를 종료하기 위해 기다려야 하는 시간(초)을 지정합니다. 1--3600 범위 내의 값을 초 단위로 지정하십시오. 디폴트 값은 60입니다.

AdoptNewMCACheck=QM|ADDRESS|NAME|ALL

AdoptNewMCACheck 속성은 AdoptNewMCA 속성을 사용할 때 필요한 유형 점검을 지정할 수 있게 합니다. 채널이 부주의나 고의로 종료되지 않게 하려면 가능할 경우 다음과 같은 세 가지 유형의 점검을 모두 수행해야 합니다. 최소한 채널 이름 일치 여부를 점검하십시오.

다음에서 하나 이상의 값을 쉼표나 공백으로 구분하여 지정하십시오.

QM 이것은 큐 관리자 이름이 일치하는지 리스너 프로세스가 점검해야 함을 의미합니다.

ADDRESS

이것은 리스너 프로세스가 통신 주소를 점검해야 함을 의미합니다(예: TCP/IP 주소).

NAME

이것은 채널 이름이 일치하는지 리스너 프로세스가 점검해야 함을 의미합니다.

ALL 리스너 프로세스가 큐 관리자 이름, 통신 주소 및 채널 이름 일치에 대해 점검하도록 합니다.

AdoptNewMCACheck=NAME,ADDRESS는 FAP1, FAP2, FAP3의 디폴트입니다.

AdoptNewMCACheck=NAME,ADDRESS,QM은 FAP4 이상에 대해 디폴트입니다.

LU62 및 TCP 스탠자

이 스탠지들은 네트워크 프로토콜 구성 매개변수를 지정하며, 채널에 대한 디폴트 속성을 대체합니다.

주: 디폴트 값에 대한 변경사항을 나타내는 속성만 지정해야 합니다.

LU62

다음 속성을 지정할 수 있습니다.

TPName

이 속성은 리모트 사이트에서 시작할 TP 이름을 지정합니다.

LocalLU

이것은 로컬 시스템에서 사용할 논리 단위의 이름입니다.

TCP

다음 속성을 지정할 수 있습니다.

Port=1414|port_number

이 속성은 TCP/IP 세션에 대한 디폴트 포트 번호를 십진 표기법으로 지정합니다. MQSeries에 대해 “잘 알려진” 포트 번호는 1414입니다.

KeepAlive=YES|NO

KeepAlive 함수를 ON 또는 OFF로 전환하려면 이 속성을 사용하십시오. KeepAlive=YES는 연결의 다른 끝이 계속 사용 가능한지 TCP/IP가 정기적으로 점검하도록 합니다. 그렇지 않으면, 채널이 닫힙니다.

ListenerBacklog=number

TCP/IP에서 수신할 때, 최대 미해결 연결 요청 수가 설정됩니다. 이것은 리스너가 요청을 승인하도록 TCP/IP 포트에서 기다리는 요청의 백로그로 간주될 수 있습니다. 디폴트 리스너 백로그 값이 표8에 나와 있습니다.

표 8. 디폴트 미해결 연결 요청(TCP)

플랫폼	디폴트 ListenerBacklog 값
OS/390	255
OS/2 Warp	10
Windows NT Server	100
Windows NT Workstation	5
AS/400	255
Sun Solaris	100
HP-UX	20
AIX V4.2 이상	100
AIX V4.1 이하	10
다른 모든 플랫폼	5

백로그가 표8에 표시된 값에 도달하면, TCP/IP 연결이 거부되고 채널을 시작할 수 없게 됩니다.

MCA 채널의 경우, 채널이 RETRY 상태가 되면서 나중에 연결을 재시도하게 됩니다.

클라이언트 연결의 경우, 클라이언트는 MQCONN에서 MQRC_Q_MGR_NOT_AVAILABLE 이유 코드를 수신하고 나중에 연결을 재시도해야 합니다.

MQSeries 구성 파일 변경

ListenerBacklog 속성은 TCP/IP 리스너에 대한 디폴트 미해결 요청 수를 대체할 수 있게 합니다.

주: 일부 운영 체제는 표시된 디폴트보다 더 큰 값을 지원합니다. 필요하다면, 이 표의 내용은 연결 한계에 도달하는 것을 방지하기 위해 사용할 수 있습니다.

ExitPath 스탠자

ExitDefaultPath=*string*

ExitDefaultPath 속성은 다음의 위치를 지정합니다.

- 클라이언트에 대한 채널 종료
- 서버에 대한 채널 종료와 데이터 변환 종료

종료 경로는 클라이언트의 경우 mqs.ini 파일의 ClientExitPath 스탠자가 읽으며 서버의 경우는 이 (ExitPath) 스탠자가 읽습니다.

mqs.ini 및 qm.ini 파일 예

201 페이지의 그림19는 Compaq OpenVMS용 MQSeries에서의 mqs.ini 파일 예를 보여줍니다.

```

#####
** Module Name: mqs.ini                                **
** Type      : MQSeries Configuration File            **
** Function   : Define MQSeries resources for the node **
**           :                                         **
#####
** Notes      :                                         **
** 1) This is an example MQSeries configuration file   **
**           :                                         **
#####
AllQueueManagers:
  #####
  ** The path to the qmgrs directory, below which queue manager data **
  ** is stored                                                         **
  #####
  DefaultPrefix=mqs_root:[mqm]
ClientExitPath:
  ExitsDefaultPath=mqs_root:[mqm.exits]
LogDefaults:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=1024
  LogType=CIRCULAR
  LogBufferPages=17
  LogDefaultPath=mqs_root:[mqm.log]
QueueManager:
  Name=saturn.queue.manager
  Prefix=mqs_root:[mqm]
  Directory=saturn$queue$manager
DefaultQueueManager:
  Name=saturn.queue.manager
QueueManager:
  Name=pluto.queue.manager
  Prefix=mqs_root:[mqm]
  Directory=pluto$queue$manager

```

그림 19. Compaq OpenVMS용 MQSeries 시스템용 MQSeries 구성 파일 예

202 페이지의 그림20은 속성 그룹이 Compaq OpenVMS용 MQSeries의 큐 관리자 구성 파일에 어떻게 배열되는 지를 보여줍니다.

MQSeries 구성 파일 변경

```
#####  
** Module Name: qm.ini *#  
** Type : MQSeries queue manager configuration file *#  
# Function : Define the configuration of a single queue manager *#  
#####  
** Notes : *#  
** 1) This file defines the configuration of the queue manager *#  
#####  
ExitPath:  
  ExitsDefaultPath=mqm_root:[mqm.exits]  
  
Service:  
  Name=AuthorizationService  
  EntryPoints=9  
  
ServiceComponent:  
  Service=AuthorizationService  
  Name=MQSeries.UNIX.auth.service  
  Module=amqzfu  
  ComponentDataSize=0  
  
Service:  
  Name=NameService  
  EntryPoints=5  
  
ServiceComponent:  
  Service=NameService  
  Name=MQSeries.DCE.name.service  
  Module=amqzfa  
  ComponentDataSize=0  
  
Log:  
  LogPrimaryFiles=3  
  LogSecondaryFiles=2  
  LogFilePages=1024  
  LogType=CIRCULAR  
  LogBufferPages=17  
  LogPath=mqm_root:[mqm.log.saturn$queue$manager]  
  
XAResourceManager:  
  Name=Oracle Resource Manager Bank  
  SwitchFile=sys$share:oraswit0.exe  
  XAOpenString=MQBankDB  
  XACloseString=  
  ThreadOfControl=PROCESS  
  
CHANNELS:  
  MaxChannels = 20 ; Maximum number of Channels allowed.  
 ; Default is 100.  
  MaxActiveChannels = 10 ; Maximum number of Channels allowed to be  
 ; active at any time. The default is the  
 ; value of MaxChannels.  
  
TCP:  
  KeepAlive = Yes ; TCP/IP entries.  
 ; Switch KeepAlive on
```

그림 20. Compaq OpenVMS용 MQSeries의 큐 관리자 구성 파일 예

참고:

노드의 MQSeries는 큐 관리자 및 로그에 대해 디폴트 위치를 사용합니다.

큐 관리자 saturn.queue.manager는 노드에 대한 디폴트 큐 관리자입니다. 이 큐 관리자
자와 연관되는 파일에 대한 디렉토리는 자동으로 OpenVMS 파일 시스템에 맞는 올바른
파일 이름으로 변환되었습니다.

MQSeries 구성 파일 변경

큐 관리자와 연관되는 데이터를 찾기 위해 MQSeries 구성 파일이 사용되므로, 존재하지 않거나 올바르지 않은 구성 파일은 일부 또는 모든 MQSeries 명령이 실패할 수 있습니다. 또한, 응용프로그램은 MQSeries 구성 파일에 정의되지 않은 큐 관리자에 연결할 수 없습니다.

제14장 문제점 판별

이 장에서는 Compaq OpenVMS용 MQSeries 사용시 발생하는 문제점을 처리하는 방법을 제시합니다.

모든 문제점이 즉시 해결되지는 않습니다. 예를 들어, 성능 문제점은 하드웨어의 한계로 발생할 수도 있습니다. 또한 문제점의 원인이 MQSeries 코드에 있다고 생각되면, IBM 지원 센터에 문의하십시오. 이 장에는 다음 절이 포함되어 있습니다.

- 『기본 점검사항』
- 209 페이지의 『공통 프로그래밍 오류』
- 209 페이지의 『다음 수행 사항』
- 212 페이지의 『응용프로그램 설계 고려사항』
- 214 페이지의 『올바르지 않은 출력』
- 217 페이지의 『오류 로그』
- 221 페이지의 『데드-레터 큐』
- 221 페이지의 『구성 파일 및 문제점 판별』
- 222 페이지의 『MQSeries 추적 사용』
- 223 페이지의 『FFST(First Failure Support Technology)』
- 228 페이지의 『클라이언트의 문제점 판별』

기본 점검사항

MQSeries의 문제점은 보통 다음 구성요소 중 하나에서 발생합니다.

- MQSeries
- 네트워크
- 응용프로그램
- 근본적인 운영 체제

다음 절에서는 문제점을 조사하는 동안 고려해야 하는 기본적인 질문을 제공합니다.

이전에 MQSeries가 성공적으로 실행되었습니까?

MQSeries가 이전에 제대로 실행되지 않았으면, 올바르게 설정되지 않았을 것입니다. MQSeries가 올바르게 설치되거나 설정되었는지 점검하려면 *Compaq OpenVMS Alpha용 MQSeries 빠른 참조 버전 5.1*을 참조하십시오.

오류 메시지가 있습니까?

MQSeries는 오류 로그를 사용하여 MQSeries 자체의 조작에 대한 메시지, 사용자가 시작하는 큐 관리자, 사용 중인 채널에서 수신되는 오류 데이터를 캡처합니다. 문제점과 연관되는 메시지가 기록되었는지 보려면 오류 로그를 점검하십시오.

오류 로그의 내용과 위치에 대한 정보는 217 페이지의 『오류 로그』를 참조하십시오.

문제점을 설명하는 리턴 코드가 있습니까?

응용프로그램에서 MQI(Message Queue Interface) 호출 실패를 나타내는 리턴 코드를 받은 경우, 리턴 코드의 설명을 보려면 *MQSeries Application Programming Reference*를 참조하십시오.

문제점을 다시 만들 수 있습니까?

문제점을 다시 만들 수 있을 경우에는 문제점이 있는 상황을 고려해 보십시오.

- 문제점이 명령이나 동등한 관리 요청에 의해 발생했습니까?
다른 방법으로 입력한 경우에 조작이 작동됩니까? 명령행에 입력한 명령이 작동할 경우 명령 서버가 정지되지 않았는지, SYSTEM.ADMIN.COMMAND.QUEUE의 큐 정의가 변경되었는지 점검하십시오.
- 프로그램에서 발생했습니까? 모든 MQSeries 시스템과 모든 큐 관리자에서 실패했습니까? 아니면 일부에서만 실패했습니까?
- 문제점이 발생할 때 항상 시스템에서 실행되고 있는 것으로 보이는 응용프로그램을 식별할 수 있습니까? 이런 경우에는 그 응용프로그램을 조사하여 오류가 있는지 살펴보십시오.

마지막 성공적인 실행 이후로 변경을 수행했습니까?

변경이 최근에 수행된 것으로 간주할 때, MQSeries 시스템, 인터페이스하는 다른 프로그램, 하드웨어 및 새로운 응용프로그램에 대해서도 고려해 보십시오. 인식하지 못하는 새로운 응용프로그램이 시스템에서 실행되었을 가능성도 있습니다.

- 큐 정의를 변경, 추가 또는 삭제했습니까?
- 채널 정의를 변경하거나 추가했습니까? 변경사항은 응용프로그램에서 요구하는 근본적인 통신 정의나 MQSeries 채널 정의에 대해 수행되었을 것입니다.
- 응용프로그램이 사용자가 수행한 변경사항에 대한 결과로 가져올 수 있는 리턴 코드를 처리합니까?

이전에 응용프로그램이 성공적으로 실행되었습니까?

문제점에 하나의 특정 응용프로그램이 관련되어 있는 것으로 나타날 경우, 이전에 그 응용프로그램이 성공적으로 실행되었는지 살펴보십시오.

이 질문에 대해 예로 응답하기 전에, 다음을 고려해 보십시오.

- 성공적인 최종 실행 이후로 응용프로그램을 변경했습니까?
이런 경우, 응용프로그램의 새로운 부분이나 수정된 부분에 오류가 있을 가능성이 있습니다. 변경사항을 살펴보고 문제점에 대한 명백한 이유를 찾을 수 있는지 확인하십시오. 이전 레벨의 응용프로그램을 사용하여 재시도할 수 있습니까?
- 이전에 모든 응용프로그램 기능을 충분히 연습했습니까?
이전에 실행된 적이 없는 응용프로그램의 일부가 처음 사용된 경우에 이런 문제점이 발생할 수 있습니까? 이런 경우, 응용프로그램의 해당 부분에 오류가 있을 가능성이 있습니다. 실패할 때 어떤 응용프로그램이 수행되고 있었는지 찾아보고, 오류에 대해 그 프로그램의 해당 부분에서 소스 코드를 점검하십시오.
프로그램이 이전의 많은 상황에서 성공적으로 실행되었으면, 현재 큐 상태 및 오류가 발생하였을 때 처리 중이었던 파일을 점검하십시오. 프로그램에서 간혹 사용되는 경로가 실행되도록 하는 일부 특수한 데이터 값을 포함하고 있을 수도 있습니다.
- 응용프로그램이 모든 리턴 코드를 점검합니까?
MQSeries 시스템이 사소한 방식으로 변경되어, 응용프로그램이 변경 결과로 수신하는 리턴 코드를 점검하지 않습니까? 예를 들어 응용프로그램이 액세스하는 큐를 공유할 수 있다고 가정합니까? 큐가 독점적으로 재정의된 경우, 더이상 그 큐에 액세스할 수 없음을 나타내는 리턴 코드를 응용프로그램이 처리할 수 있습니까?
- 응용프로그램이 다른 MQSeries 시스템에서 실행됩니까?
문제점을 발생시키는 이 MQSeries 시스템의 설정 방식에 대해 다른 방식이 있을 수 있습니까? 예를 들어, 동일한 메시지 길이나 우선순위를 사용하여 큐를 정의했습니까?

이전에 응용프로그램이 성공적으로 실행되지 않은 경우

응용프로그램이 아직 성공적으로 실행된 적이 없다면 이 응용프로그램을 신중히 조사하여 오류를 찾아내야 합니다.

코드를 보기 전에, 코드가 작성된 프로그래밍 언어에 따라 변환기 또는 컴파일러 및 연계 편집기에서 출력을 조사하여(적용 가능한 경우) 오류가 보고되었는지 살펴보십시오.

응용프로그램이 로드 라이브러리로의 변환, 컴파일 또는 링크 편집에 실패한 경우, 이를 호출하려고 하면 실행에 실패하게 됩니다. 응용프로그램 빌드에 대한 정보는 *MQSeries Application Programming Reference*를 참조하십시오.

문서에서 이러한 각각의 단계가 오류 없이 수행되었다면, 응용프로그램의 코딩 논리를 고려해야 합니다. 문제점의 증상이 실패한 기능과 오류가 있는 코드 부분을 나타내니까? MQSeries 응용프로그램에 대한 문제점을 발생하는 일반 오류의 일부 예에 대해서는 209 페이지의 『공통 프로그래밍 오류』를 참조하십시오.

문제점이 네트워크의 특정 부분에만 영향을 줍니까?

문제점의 영향을 받는 네트워크의 특정 부분(예: 리모트 큐)을 식별할 수 있습니다. 리모트 메시지 큐 관리자에 대한 링크가 작동하지 않을 경우, 메시지는 리모트 큐로 흐를 수 없습니다.

두 시스템 사이의 연결과 MQSeries의 상호통신 구성요소가 시작되었는지 점검하십시오.

메시지가 트랜스미션 큐에 이르렀는지 점검하고, 트랜스미션 큐와 리모트 큐의 로컬 큐 정의를 점검하십시오.

네트워크 관련 변경사항을 수행했거나 문제점에 대해 고려할 수 있는 MQSeries 정의를 변경했습니까?

특정 시기에 문제점이 발생했습니까?

특정 시기에 문제점이 발생할 경우에는 시스템 로드의 영향을 받았을 수 있습니다. 일반적으로 최대 시스템 로드는 정오와 자정이므로, 이 시기는 로드와 따른 문제점이 발생할 가능성이 큰 시기입니다(MQSeries 네트워크가 여러 시간대에 분산되어 있을 경우, 최대 시스템 로드는 일부 다른 시기에 발생하는 것으로 보일 수도 있습니다).

문제점이 간헐적입니까?

간헐적으로 발생하는 문제점은 프로세스가 서로 독립적으로 실행될 수 없다고 여겨질 경우에 발생할 수 있습니다. 예를 들어, 프로그램은 이전 프로세스가 완료되기 전에 대기 옵션을 지정하지 않고 MQGET 호출을 발행할 수 있습니다. 간헐적인 문제점은 또한 메시지를 넣는(put) 호출이 불확실할 때(즉, 확약되거나 백아웃되기 전에) 응용프로그램이 큐에서 메시지를 가져오려고(get) 할 경우에도 나타납니다.

서비스 갱신사항을 적용했습니까?

서비스 갱신사항이 MQSeries에 적용된 경우, 갱신 조치가 성공적으로 완료되었고 어떤 오류 메시지도 생성되지 않았는지 점검하십시오.

- 갱신사항에 특수 지시사항이 있습니까?
- 테스트를 실행하여 갱신이 올바르게 완전하게 적용되었는지 확인했습니까?
- MQSeries가 이전 서비스 레벨로 복원될 경우 문제점이 계속 존재합니까?
- 설치에 성공한 경우, 패치 오류에 대해 IBM 지원 센터를 점검하십시오.
- 다른 프로그램에 패치가 적용되었으면, MQSeries가 인터페이스하는 방식에 주는 영향에 대해 고려해 보십시오.

갱신사항을 적용해야 합니까?

MQSeries는 근본적인 운영 체제(OpenVMS)와 다양한 네트워킹 제품(TCP/IP와 같은)에 의존합니다. 적절한 벤더와 함께 이 제품에 대해 필요한 모든 서비스 갱신사항을 적용했는지 확인하십시오.

공통 프로그래밍 오류

다음 목록의 오류는 MQSeries 프로그램을 실행하는 중에 발견되는 문제점의 가장 일반적인 원인을 보여줍니다. MQSeries 시스템에 대한 문제점은 다음 오류 중 하나 이상에 의해 발생할 수 있다는 점을 고려하십시오.

- 큐가 사실상 서로 독점적일 때 공유할 수 있다는 가정을 합니다.
- MQI 호출에서 올바르지 않은 매개변수를 전달합니다.
- MQI 호출에서 충분하지 않은 매개변수를 전달합니다. 이것은 MQI가 처리할 응용 프로그램에 대해 완료 및 이유 코드를 설정할 수 없음을 의미합니다.
- MQI 요청의 리턴 코드를 점검하는 데 실패합니다.
- 올바르지 않은 길이가 지정된 변수를 전달합니다.
- 올바르지 않은 순서로 매개변수를 전달합니다.
- *MsgId* 및 *CorrelId*를 올바르게 초기화하는 데 실패합니다.

다음 수행 사항

예비 점검으로 문제점의 원인을 찾을 수도 있습니다. 이런 경우, 바로 문제점을 해결할 수 있어야 합니다. MQSeries 라이브러리(관련 서적 참조)와 다른 사용 허가된 프로그램의 라이브러리에 있는 기타 서적들을 참조할 수 있습니다.

원인을 아직 찾지 못한 경우에는 문제점을 자세히 파악해야 합니다.

이 절의 목적은 예비 점검으로 문제점을 찾을 수 없는 경우에 그 문제점의 원인을 식별할 수 있도록 안내하는 것입니다.

시스템에 대해 어떤 변경사항도 설정되지 않았고 응용프로그램에 대해 문제점이 없으면, 문제점 증상을 가장 잘 설명하는 옵션을 선택하십시오.

- 210 페이지의 『올바르지 않은 출력이 확보되었습니까?』
- 210 페이지의 『PCF 명령으로부터 응답을 수신하는 데 실패했습니까?』
- 212 페이지의 『문제점이 리모트 큐에만 영향을 줍니까?』

이 증상 중에 문제점이 없다면, 시스템의 다른 구성요소에 의해 발생한 경우를 고려해 보십시오.

다음

올바르지 않은 출력이 확보되었습니까?

이 서적에서 응용프로그램에 대해 『올바르지 않은 출력』의 예는 다음과 같습니다.

- 예상하고 있었던 메시지를 수신하지 못합니다.
- 예상치 못한 정보나 손상된 정보를 포함하는 메시지를 수신합니다.
- 예상하지 못한 메시지(예를 들어, 목적이 다른 응용프로그램이었던 메시지)를 수신합니다.

모든 경우에, 응용프로그램이 사용 중인 큐 또는 큐 관리자 별명이 올바르게 지정되었는지 점검하고 네트워크에 대한 변경사항을 조정하십시오.

MQSeries 오류 메시지가 생성될 경우(접두부로 『AMQ』과 붙어 있음), 오류 로그를 보아야 합니다. 추가 정보는 217 페이지의 『오류 로그』를 참조하십시오.

PCF 명령으로부터 응답을 수신하는 데 실패했습니까?

명령을 발행하였으나 응답을 수신하지 못하였으면, 다음 질문을 고려해 보십시오.

- 명령 서버가 실행 중입니까?

dspmqcsv 명령으로 작업하여 명령 서버의 상태를 점검하십시오.

- 이 명령에 대한 응답이 명령 서버가 실행되고 있지 않은 것으로 나타날 경우, **strmqcsv** 명령을 사용하여 이를 시작하십시오.
- 명령에 대한 응답이 SYSTEM.ADMIN.COMMAND.QUEUE가 MQGET 요청에 대해 사용되지 않음을 나타낼 경우, MQGET 요청에 대해 큐를 사용하십시오.

- 데드-레터 큐에 응답이 송신되었습니까?

데드-레터 큐 헤더 구조에는 문제점을 설명하는 이유와 피드백 코드가 들어 있습니다. 데드-레터 큐 헤더 구조(MQDLH)에 대한 정보는 *MQSeries Application Programming Reference*를 참조하십시오.

데드-레터 큐에 메시지가 있으면, 제공된 찾아보기 샘플 응용프로그램(amqsbcg)을 사용하여 MQGET 호출로 큐에서 메시지를 찾아볼 수 있습니다. 샘플 응용프로그램은 이름 지정된 큐 관리자에 대해 이름 지정된 큐에서 모든 메시지를 단계적으로 거쳐서 이름 지정된 큐에 있는 모든 메시지에 대해 메시지 설명자와 메시지 컨텍스트 필드를 표시합니다.

- 오류 로그로 메시지가 송신되었습니까?

추가 정보는 217 페이지의 『오류 로그』를 참조하십시오.

- 큐가 넣기(put) 및 가져오기(get) 조작에 대해 사용 가능합니까?

- *WaitInterval*의 길이가 충분합니까?

MQGET 호출이 시간 종료되었으면, MQCC_FAILED의 완료 코드와 MQRC_NO_MSG_AVAILABLE의 이유 코드가 리턴됩니다(*WaitInterval* 필드와

MQGET로부터의 완료 및 이유 코드에 대한 정보는 *MQSeries Application Programming Reference*를 참조하십시오.

- 사용자 고유의 응용프로그램을 사용하여 명령을 SYSTEM.ADMIN.COMMAND.QUEUE에 넣을(put) 경우, 동기점을 사용해야 합니까?
동기점에서 사용자의 요청 메시지를 특별히 제외시키지 않았으면, 응답 메시지를 수신하기 전에 동기점을 사용해야 합니다.
- 큐의 MAXDEPTH 및 MAXMSGL 속성의 높이가 충분합니까?
- *CorrelId* 및 *MsgId* 필드를 올바르게 사용하고 있습니까?
응용프로그램에서 *MsgId* 및 *CorrelId*의 값은 큐로부터 모든 메시지를 수신하도록 설정하십시오.

명령 서버를 정지한 후 재시작하여, 생성된 오류 메시지에 응답하십시오.

시스템이 계속 응답하지 않으면, 큐 관리자나 MQSeries 시스템 전체적으로 문제점이 있을 수 있습니다. 먼저 개인 큐 관리자를 정지하고 실패한 큐 관리자를 판별하십시오. 이것으로도 문제점이 밝혀지지 않으면, MQSeries를 정지하였다가 재시작하여 오류 로그에서 생성된 메시지에 응답하십시오.

재시작 후에도 문제점이 계속 발생하면 IBM 지원 센터에 문의하십시오.

일부 큐가 실패했습니까?

큐 서브세트에만 문제점이 발생한 것으로 여겨지면, 문제점이 있는 것으로 생각되는 로컬 큐를 점검하십시오.

1. 각 큐에 대한 정보를 표시하십시오. MQSC 명령 DISPLAY QUEUE를 사용하여 정보를 표시하십시오.
2. 표시된 데이터를 사용하여 다음 점검사항을 수행하십시오.
 - CURDEPTH가 MAXDEPTH이면, 큐가 처리되고 있지 않다는 의미입니다. 모든 응용프로그램이 정상적으로 실행되고 있는지 점검하십시오.
 - CURDEPTH가 MAXDEPTH가 아니면, 다음 큐 속성을 점검하여 올바른지 확인하십시오.
 - 트리거가 사용될 경우:
 - 트리거 모니터가 실행 중입니까?
 - 트리거 용량이 너무 많지 않습니까? 즉, 트리거 이벤트를 자주 충분히 생성합니까?
 - 프로세스 이름이 올바릅니까?
 - 프로세스가 사용 가능하고 작동되고 있습니까?
 - 큐를 시작할 수 있습니까? 그렇지 않으면, 다른 응용프로그램이 입력하기 위해 이미 큐를 열었을 것입니다.
 - 큐가 GET 및 PUT에 대해 적절하게 사용할 수 있습니까?

다음

- 큐에서 메시지를 가져오는(get) 응용프로그램 프로세스가 없으면, 이런 상황이 발생하는 원인을 판별하십시오. 응용프로그램이 시작되어야 했거나, 연결이 중단되었거나, 어떤 이유로 인해 MQOPEN 호출이 실패했습니다.

큐 속성 IPPROCS 및 OPPROCS를 점검하십시오. 이러한 속성은 입력 및 출력을 위해 큐가 열려 있는지의 여부를 나타냅니다. 값이 0이면, 입력과 출력 중 어떤 조작도 발생할 수 없음을 나타냅니다. 값이 변경되었거나 큐가 열렸었지만 지금은 닫혀 있는 것입니다.

메시지를 넣거나(put) 가져올(get) 때의 상태를 점검해야 합니다.

문제점을 해결할 수 없으면 IBM 지원 센터에 문의하십시오.

문제점이 리모트 큐에만 영향을 줍니까?

문제점이 리모트 큐에만 영향을 미칠 경우, 다음을 점검하십시오.

- 필요한 채널이 시작되었고 트리거 작업이 가능한지, 필요한 시작기가 실행 중인지 점검하십시오.
- 메시지를 리모트 큐에 넣어야(put) 하는 프로그램에 문제점이 보고되지 않았는지 점검하십시오.
- 분산 큐잉 프로세스를 시작하기 위해 트리거를 사용할 경우, 트랜스미션 큐가 트리거를 설정했는지 점검하십시오. 또한 채널 시작기가 실행 중인지 점검하십시오.
- 채널 오류나 문제점을 나타내는 메시지에 대해 오류 로그를 점검하십시오.
- 필요하다면 채널을 수동으로 시작하십시오. 이를 수행하는 방법에 대해서는 *MQSeries 상호통신*을 참조하십시오.

채널을 정의하는 방법에 대해서는 *MQSeries 상호통신*을 참조하십시오.

응용프로그램 설계 고려사항

여러가지 방식에 있어서 부적절한 프로그램 설계가 성능에 영향을 줄 수 있습니다. 이것은 다른 작업의 성능에 영향을 미치면서도 프로그램이 제대로 수행되는 것처럼 표시될 수 있으므로 감지하기가 어렵습니다. MQSeries를 호출하는 프로그램에 고유한 몇 가지 문제점이 이 절에서 설명됩니다.

응용프로그램 설계에 대한 자세한 정보는 *MQSeries Application Programming Guide*를 참조하십시오.

메시지 길이의 효과

MQSeries에서 100 MB까지의 데이터를 보유하는 것이 허용되지만, 메시지의 데이터 양은 메시지를 처리하는 응용프로그램의 성능에 영향을 미칩니다. 응용프로그램으로부터 최상의 성능을 얻으려면, 중요한 데이터만 메시지로 송신해야 합니다. 예를 들어, 은행

계정 차변에 기입하는 요청에서 클라이언트로부터 서버 응용프로그램으로 전달되어야 하는 정보는 계정 번호와 차변 기입 금액입니다.

메시지 지속성의 효과

지속 메시지는 로그에 기록됩니다. 메시지를 로그에 기록하면 응용프로그램 성능이 떨어지므로, 중요한 데이터에 대해서만 지속 메시지를 사용해야 합니다. 큐 관리자가 정지하거나 실패할 경우 메시지의 데이터가 버려질 수 있으면, 비지속 메시지를 사용하십시오.

특정 메시지 검색

MQGET 호출은 보통 큐에서 첫 번째 메시지를 검색합니다. 메시지 설명자에서 메시지와 상호관계 ID(*MsgId* 및 *CorrelId*)를 사용하여 특정 메시지를 지정할 경우, 큐 관리자는 그 메시지를 찾을 때까지 큐를 검색해야 합니다. 이 방식으로 MQGET 호출을 사용하면 응용프로그램 성능에 영향을 줍니다.

다른 길이의 메시지를 포함하는 큐

큐의 메시지 길이가 다를 경우, 응용프로그램은 메시지 크기를 판별하기 위해 0으로 설정된 *BufferLength* 필드와 함께 MQGET 호출을 사용할 수 있으므로, 호출이 실패해도 메시지 데이터의 크기를 리턴합니다. 그러면 응용프로그램은 호출을 반복하여 첫 번째 호출에서 측정된 메시지의 ID와 올바른 크기의 버퍼를 지정할 수 있습니다. 그러나 같은 큐에 서비스를 제공하는 다른 응용프로그램이 있으면, 두 번째 MQGET 호출은 두 호출 사이에 다른 응용프로그램이 검색한 메시지를 검색하는 데 시간을 소비하므로 응용프로그램의 성능이 감소될 수 있습니다.

응용프로그램이 고정 길이의 메시지를 사용할 수 없을 경우, 이 문제점에 대한 다른 해결책은 큐가 승인할 수 있는 최대 메시지 크기를 찾은 후 MQGET 호출에서 이 값을 사용하는 것입니다. 큐에 대한 메시지의 최대 크기는 큐의 *MaxMsgLength* 속성에 저장됩니다. 그러나 이 큐 속성의 값이 100 MB(Compaq OpenVMS용 MQSeries에서 허용되는 최대값) 이상이 될 수 있으므로, 이 방법은 더 많은 양의 저장영역을 사용할 수 있습니다.

동기점 빈도

동기점 내에서 확약 없이 다양한 MQPUT 호출을 발행하는 프로그램은 성능 문제점을 가져올 수 있습니다. 영향을 받는 큐는 현재 액세스할 수 없는 메시지로 채워지고, 다른 작업은 이러한 메시지를 가져오기(get) 위해 기다릴 수 있습니다. 저장영역 측면과, 메시지를 가져오려는(get) 작업과 연관된 스레드의 측면에서 이를 고려해야 합니다.

MQPUT1 호출의 사용

큐에 넣을(put) 메시지가 하나일 경우에만 MQPUT1 호출을 사용하십시오. 여러 개의 메시지를 넣으려면 MQOPEN 호출과 그 뒤에 오는 일련의 MQPUT 호출, 단일 MQCLOSE 호출을 사용하십시오.

올바르지 않은 출력

“올바르지 않은 출력”이란 용어는 여러 가지 방식으로 해석될 수 있습니다. 이 서적에서는 문제점 판별을 목적으로 하고 있으며 용어의 의미에 대해서는 210 페이지의 『올바르지 않은 출력이 확보되었습니까?』에 설명되어 있습니다.

이 절에서는 두 가지 유형의 올바르지 않은 출력에 대해 설명됩니다.

- 예상할 때 표시되지 않는 메시지들
- 올바르지 않은 정보나 손상된 정보를 포함하는 메시지들

응용프로그램에 분산 큐잉 사용이 포함될 경우 발견할 수 있는 추가 문제점에 대해서도 논의됩니다.

큐에 표시되지 않는 메시지

예상할 때 메시지가 나타나지 않을 경우 다음을 점검하십시오.

- 메시지를 큐에 성공적으로 넣었습니까(put)?
 - 큐가 올바르게 정의되었습니까? 예를 들어, MAXMSGL 크기가 충분합니까?
 - 큐에 메시지를 넣을(put) 수 있습니까?
 - 큐가 이미 가득 찼습니까? 그러면 응용프로그램이 필수 메시지를 큐에 넣을 수 없습니다.
- 큐로부터 메시지를 가져올(get) 수 있습니까?
 - 동기점을 사용해야 합니까?

동기점 내에서 메시지를 넣거나(put) 검색할 경우, 그 메시지들은 복구 단위가 확약될 때까지 다른 작업에 사용할 수 없습니다.
 - 대기 간격이 충분합니까?

MQGET 호출에 대한 옵션으로 대기 간격을 설정할 수 있습니다. 응답을 위해 충분히 기다리도록 설정해야 합니다.

- 메시지나 상호관계 ID에 의해 식별되는 특정 메시지를 기다리고 있습니까(MsgId 또는 CorrelId)?

올바른 MsgId 또는 CorrelId를 가진 메시지를 기다리고 있는지 점검하십시오. 성공적인 MQGET 호출은 두 값을 모두 검색된 메시지의 값으로 설정하므로, 다른 메시지를 성공적으로 가져오려면 이 값들을 재설정해야 합니다.

또한 큐에서 다른 메시지를 가져올(get) 수 있는지 점검하십시오.

- 다른 응용프로그램이 큐로부터 메시지를 가져올(get) 수 있습니까?
- 예상하는 메시지가 지속 메시지로 정의되었습니까?
지속 메시지로 정의되지 않고 MQSeries가 재시작되면, 메시지가 손실됩니다.
- 다른 응용프로그램이 큐에 대한 독점적 액세스를 가지고 있습니까?

큐에서 잘못된 점을 찾을 수 없는 상태에서 MQSeries가 실행 중이면, 메시지를 큐에 넣기(put) 위해 예상한 프로세스에 대해 다음을 점검하십시오.

- 응용프로그램이 시작되었습니까?
트리거되어야 했을 경우에는 올바른 트리거 옵션을 지정했는지 점검하십시오.
- 응용프로그램이 정지했습니까?
- 트리거 모니터가 실행 중입니까?
- 트리거 프로세스가 올바르게 정의되었습니까?
- 응용프로그램이 올바르게 완료했습니까?
작업 로그에서 비정상 종료에 관한 내용을 찾으십시오.
- 응용프로그램이 변경사항을 파악했습니까 아니면 백아웃했습니까?

여러 트랜잭션이 큐에 서비스를 제공할 경우, 서로 충돌할 수 있습니다. 예를 들어, 하나의 트랜잭션이 버퍼 길이가 0인 MQGET 호출을 발행하여 메시지 길이를 찾은 후, 그 메시지의 *MsgId*를 지정하는 특정 MQGET 호출을 발행한다고 가정합니다. 그러나 그동안 다른 트랜잭션이 그 메시지에 대해 성공적인 MQGET 호출을 발행하므로, 첫 번째 응용프로그램은 이유 코드 MQRC_NO_MSG_AVAILABLE을 수신합니다. 다중 서버 환경에서 실행되는 응용프로그램은 이런 상황에 대처할 수 있도록 설계되었습니다.

메시지는 수신될 수 있었지만 응용프로그램이 어떤 방식으로든지 처리에 실패하였다고 간주해 보십시오. 예를 들어, 예상된 메시지 형식에서의 오류로 프로그램이 이를 거부했습니까? 이런 경우에는 『예상치 못하거나 손상된 정보를 포함하는 메시지』를 참조하십시오.

예상치 못하거나 손상된 정보를 포함하는 메시지

메시지에 포함된 정보가 응용프로그램에서 예상한 것이 아니거나 어떤 방식으로든지 손상된 경우에는 다음 사항을 고려하십시오.

- 사용자의 응용프로그램이나 큐에 메시지를 넣은(put) 응용프로그램이 변경되었습니까?
변경을 인식해야 하는 모든 시스템에 모든 변경사항이 동시에 반영되도록 하십시오.
예를 들어 메시지 데이터 형식이 변경되었을 수 있습니다. 이런 경우 두 응용프로그램이 모두 변경사항을 인식하도록 다시 컴파일해야 합니다. 하나의 응용프로그램이 다시 컴파일되지 않았으면, 데이터는 다른 응용프로그램에 대해 손상된 것으로 표시됩니다.
- 응용프로그램이 올바르게 않은 큐로 메시지를 송신하고 있습니까?

올바르지 않은 출력

응용프로그램이 수신하는 메시지가 실제로 다른 큐를 지원하는 응용프로그램에 대한 것이 아닌지 점검하십시오. 필요하다면, 권한이 없는 응용프로그램이 올바르게 않은 큐에 메시지를 넣지(put) 않도록 보안 정의를 변경하십시오.

응용프로그램이 알리아스 큐를 사용했으면, 별명이 올바른 큐를 지시하는지 점검하십시오.

- 이 큐에 대해 트리거 정보가 올바르게 지정되었습니까?

사용자의 응용프로그램이 시작되어야 했는지, 아니면 다른 응용프로그램이 시작되어야 했는지 점검하십시오.

이러한 점검으로 문제점을 해결할 수 없으면, 메시지를 송신하는 프로그램과 이를 수신하는 프로그램 모두에 대해 응용프로그램 논리를 점검해야 합니다.

분산 큐를 사용할 경우 올바르게 않은 출력에 대한 문제점

응용프로그램이 분산 큐잉을 사용할 경우, 다음 사항도 고려해야 합니다.

- 송신 및 수신 시스템에 MQSeries가 올바르게 설치되어 있고 분산 큐잉에 대해 올바르게 구성되어 있습니까?

- 두 시스템 사이에 사용 가능한 링크가 있습니까?

두 시스템 모두 사용 가능하고 MQSeries에 연결되어 있는지 점검하십시오. 두 시스템 사이의 연결과, 두 큐 관리자 사이의 채널이 활성화 상태인지 점검하십시오.

- 송신 시스템에 트리거가 설정되어 있습니까?

- 리모트 시스템으로부터 응답 메시지를 대기 중인 메시지가 있습니까?

리모트 시스템에서 트리거가 활성화되었는지 점검하십시오.

- 큐가 이미 가득 찼습니까?

그러면 응용프로그램이 필수 메시지를 큐에 넣을 수 없습니다. 이런 경우, 메시지를 데드-레터 큐에 넣었는지(put) 점검하십시오.

데드-레터 큐 헤더에는 메시지를 타겟 큐에 넣을(put) 수 없는 원인을 설명하는 이유와 피드백 코드가 들어 있습니다. 데드-레터 큐 헤더 구조에 대한 정보는 *MQSeries Application Programming Reference*를 참조하십시오.

- 송신 큐 관리자와 수신 큐 관리자 사이에 불일치가 있습니까?

예를 들어, 메시지 길이는 수신되는 큐 관리자가 처리할 수 있는 것보다 더 길 수 있습니다.

- 송신 채널과 수신 채널의 채널 정의가 호환 가능합니까?

예를 들어, 순차 번호 랩에서 불일치가 발생하면 분산 큐잉 구성요소가 정지됩니다. 분산 큐잉에 대한 자세한 정보는 *MQSeries 상호통신*을 참조하십시오.

- 데이터 변환이 관련되어 있습니까? 송신 응용프로그램과 수신 응용프로그램 사이의 데이터 형식이 다를 경우, 데이터 변환이 필요합니다. 자동 변환은 형식이 내장 형식 중 하나로 인식될 경우 MQGET가 발행될 때 발생합니다.

데이터 세트가 변환에 대해 인식되지 않으면, 사용자 고유 루틴으로 변환을 수행할 수 있도록 데이터 변환 종료가 사용됩니다.

위의 예외는 데이터를 MVS/ESA용 MQSeries에 송신할 경우에 발생합니다.

데이터 변환에 대한 추가 정보는 *MQSeries* 상호통신을 참조하십시오.

오류 로그

MQSeries는 오류 로그를 사용하여 MQSeries 자체의 조작에 대한 메시지, 사용자가 시작하는 큐 관리자, 사용 중인 채널에서 수신되는 오류 데이터를 캡처합니다.

오류 로그의 위치는 큐 관리자 이름이 알려져 있는지, 오류가 클라이언트와 연관되는지의 여부에 따라 결정됩니다.

- 큐 관리자 이름이 알려지고 큐 관리자가 사용 가능할 경우:

MQS_ROOT:[MQM.QMGRS.QMgrName.ERRORS]AMQERR01.LOG

- 큐 관리자가 사용 불가능할 경우:

MQS_ROOT:[MQM.QMGRS.\$SYSTEM.ERRORS]AMQERR01.LOG

- 클라이언트 응용프로그램에 대해 오류가 발생한 경우:

MQS_ROOT:[MQM.ERRORS]AMQERR01.LOG

- FFST(First Failure Support Technology) - 223 페이지의 『FFST를 조사하는 방법』 참조.

주: 클라이언트의 경우, 오류는 클라이언트의 루트 드라이브에 저장됩니다.

로그 파일

설치시 [MQM.QMGRS.\$SYSTEM.ERRORS] 디렉토리는 QMGRS 파일 경로에 작성됩니다. 오류 서브디렉토리는 이름이 다음과 같이 최고 세 개의 오류 로그 파일을 포함할 수 있습니다.

- AMQERR01.LOG
- AMQERR02.LOG
- AMQERR03.LOG

큐 관리자를 작성하고 나면, 큐 관리자에서 필요할 때 세 개의 오류 로그 파일이 작성됩니다. 이 파일들의 이름은 \$SYSTEM 이름과 같으며(즉, AMQERR01, AMQERR02 및 AMQERR03), 각각의 용량은 256 KB입니다. 파일은 작성하는 각 큐 관리자의 오류 서브디렉토리에 위치합니다.

오류 메시지가 생성됨에 따라 메시지들은 AMQERR01에 위치됩니다. AMQERR01이 256 KB 이상이면 AMQERR02에 복사됩니다. 복사 이전에 AMQERR02는 AMQERR03.LOG에 복사됩니다. AMQERR03의 이전 내용(있는 경우)은 버려집니다.

오류 로그

그러므로 최근 오류 메시지는 항상 AMQERR01에 위치하고, 다른 파일들은 오류 메시지의 실행 기록을 관리하기 위해 사용됩니다.

채널에 관련되는 모든 메시지는 큐 관리자 이름이 알려져 있지 않거나 큐 관리자를 사용할 수 없으면 해당되는 큐 관리자의 오류 파일에도 위치합니다. 큐 관리자 이름을 사용할 수 없거나 그 이름을 판별할 수 없으면, 채널 관련 메시지가 [MQM.QMGRS.\$SYSTEM.ERRORS] 서브디렉토리에 위치합니다.

오류 로그 파일의 내용을 조사하려면 일반 OpenVMS 편집기를 사용하십시오.

이전 오류

위의 오류 로그가 아직 설정되지 않았는데 오류가 발생하는 일부 특수한 경우가 있습니다. MQSeries는 이런 오류를 오류 로그에 기록하려고 합니다. 로그의 위치는 큐 관리자가 설정된 상태에 따라 결정됩니다.

예를 들어, 구성 파일이 손상되어 어떤 위치 정보도 판별될 수 없으면, 오류는 설치시 루트 디렉토리 mqm에서 작성한 오류 디렉토리에 기록됩니다.

MQSeries 구성 파일을 읽을 수 있고 AllQueueManagers 스탠자의 DefaultPrefix 속성을 읽을 수 있으면, 오류는 DefaultPrefix[.errors] 디렉토리에 기록됩니다.

구성 파일에 대한 추가 정보는 183 페이지의 『제13장 MQSeries 구성』을 참조하십시오.

연산자 메시지

Compaq OpenVMS용 MQSeries에서, 연산자 메시지는 대개 명령에서 올바르지 않은 매개변수를 사용하는 등의 작업을 사용자가 직접 수행함으로써 발생하는 일반 오류를 식별합니다. 연산자 메시지는 메시지 범주가 표준 위치에 설치된 자국어(NLS) 가능 메시지입니다.

이 메시지들은 연관된 창에 기록되고(있을 경우), 큐 관리자 디렉토리의 오류 로그 AMQERR01.LOG에도 기록됩니다. 예를 들면 다음과 같습니다.

```
MQS_ROOT:[MQM.QMGRS.QUEUE$MANAGER.ERRORS]
```

일부 오류는 큐 관리자 디렉토리의 AMQERR01.LOG 파일에 기록되고 다른 오류는 오류 로그의 \$SYSTEM 디렉토리 사본에 기록됩니다.

오류 로그 예

이 예는 Compaq OpenVMS용 MQSeries 오류 로그의 일부를 나타낸 것입니다.

```
...
06/29/00 09:41:39 AMQ7467: The oldest log file required to start queue
manager BKMI is S0000000.LOG.
```

```
EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this may
```

```

be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory. If you move any of the log
files required to recreate objects from their media images, you will
have to restore them to recreate the objects.
-----
06/29/00 09:41:39 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any
log files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory.
-----
06/29/00 09:42:05 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this
may be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory. If you move any of the log
files required to recreate objects from their media images,
you will have to restore them to recreate the objects.
-----
06/29/00 09:42:05 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any
log files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory.
-----
06/29/00 09:42:06 AMQ8003: MQSeries queue manager started.

EXPLANATION: MQSeries queue manager BKM1 started.
ACTION: None.
-----
06/29/00 09:42:06 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this
may be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory. If you move any of the
log files required to recreate objects from their media images,
you will have to restore them to recreate the objects.
-----
06/29/00 09:42:06 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images.
Any log files prior to this will not be accessed by media recovery
operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
-----
06/29/00 09:46:27 AMQ7030: Request to quiesce the queue manager accepted.
The queue manager will stop when there is no further work for it to
perform.

EXPLANATION: You have requested that the queue manager end when there is no
more work for it. In the meantime, it will refuse new applications
that attempt to start, although it allows those already running to
complete their work.
ACTION: None.
-----
06/29/00 09:46:43 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this may be
required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive
medium to release space in the log directory. If you move any of the
log files required to recreate objects from their media images, you

```

```

will have to restore them to recreate the objects.
-----
06/29/00 09:46:43 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any
log files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
-----
06/29/00 09:46:44 AMQ8004: MQSeries queue manager ended.

EXPLANATION: MQSeries queue manager BKM1 ended.
ACTION: None.
-----
06/29/00 09:46:59 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this
may be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory. If you move any of the log files
required to recreate objects from their media images, you will have to
restore them to recreate the objects.
-----
06/29/00 09:47:00 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any log
files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
-----
06/29/00 09:47:08 AMQ7472: Object TEST1, type queue damaged.

EXPLANATION: Object TEST1, type queue has been marked as damaged. This
indicates that the queue manager was either unable to access the object in
the file system, or that some kind of inconsistency with the data in
the object was detected.
ACTION: If a damaged object is detected, the action performed depends on
whether the queue manager supports media recovery and when the damage
was detected. If the queue manager does not support media recovery,
you must delete the object as no recovery is possible. If the queue manager
does support media recovery and the damage is detected during the processing
performed when the queue manager is being started, the queue manager will
automatically initiate media recovery of the object. If the queue
manager supports media recovery and the damage is detected once the queue
manager has started, it may be recovered from a media image using the
rcrmqobj command or it may be deleted.
-----
06/29/00 09:47:09 AMQ8003: MQSeries queue manager started.

EXPLANATION: MQSeries queue manager BKM1 started.
ACTION: None.
-----
06/29/00 09:47:09 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this may be
required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory. If you move any of the log files
required to recreate objects from their media images, you will have to
restore them to recreate the objects.
-----
06/29/00 09:47:10 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any log
files prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
-----
06/29/00 09:47:47 AMQ7081: Object TEST1, type queue recreated.

```

```

EXPLANATION: The object TEST1, type queue was recreated from its media
image.
ACTION: None.
-----
06/29/00 11:22:10 AMQ7467: The oldest log file required to start queue
manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to restart the queue manager. Log records older than this may
be required for media recovery.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory. If you move any of the log files
required to recreate objects from their media images, you will have
to restore them to recreate the objects.
-----
06/29/00 11:22:10 AMQ7468: The oldest log file required to perform media
recovery of queue manager BKM1 is S0000000.LOG.

EXPLANATION: The log file S0000000.LOG contains the oldest log record
required to recreate any of the objects from their media images. Any log files
prior to this will not be accessed by media recovery operations.
ACTION: You can move log files older than S0000000.LOG to an archive medium
to release space in the log directory.
-----
06/29/00 11:22:11 AMQ8004: MQSeries queue manager ended.

EXPLANATION: MQSeries queue manager BKM1 ended.
ACTION: None.
-----
...

```

데드-레터 큐

특정 이유로 전달할 수 없는 메시지는 데드-레터 큐에 위치됩니다. MQSC DISPLAY QUEUE 명령을 발행하여 큐에 메시지가 있는지 점검할 수 있습니다. 큐에 메시지가 있으면, 제공된 찾아보기 샘플 응용프로그램(amqsbcg)을 사용하여 MQGET 호출을 통해 큐에서 메시지를 찾아볼 수 있습니다. 샘플 응용프로그램은 이름 지정된 큐 관리자에 대해 이름 지정된 큐에서 모든 메시지를 단계적으로 거쳐서 이름 지정된 큐에 있는 모든 메시지에 대해 메시지 설명자와 메시지 컨텍스트 필드를 표시합니다.

큐에 메시지를 넣는(put) 이유에 따라 데드-레터 큐에 있는 메시지의 처리 방법을 결정해야 합니다.

사용 중인 각 큐 관리자에 데드-레터 큐가 없으면 문제점이 발생할 수 있습니다. **crtmqm** 명령을 사용하여 큐 관리자가 작성될 때, 자동으로 SYSTEM.DEAD.LETTER.QUEUE라는 데드-레터 큐가 디폴트 오브젝트로서 작성됩니다. 그러나 이 큐는 큐 관리자에 대한 데드-레터 큐로 정의되어 있지 않습니다. 51 페이지의 『데드-레터 큐의 정의』를 참조하십시오.

구성 파일 및 문제점 판별

대개 구성 파일에 오류가 있으면 큐 관리자를 찾을 수 없으므로 『큐 관리자를 사용할 수 없는』 오류를 가져옵니다.

다음은 구성 파일을 점검할 때 주의할 사항입니다.

- 구성 파일이 있는지 확인하십시오.

구성 파일

- 다음과 같은 적절한 사용권한이 있는지 확인하십시오.

```
MQS.INI;1 MQM (RWED, RWED, RW, R)
(identifier=MQM, ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
```

- MQSeries 구성 파일이 올바른 큐 관리자와 로그 디렉토리를 참조하는지 확인하십시오.

MQSeries 추적 사용

Compaq OpenVMS용 MQSeries는 추적 기능을 위해 다음과 같은 명령을 사용합니다.

- **strmqtrc** - 332 페이지의 『strmqtrc(MQSeries 추적 시작)』 참조
- **dspmqtrc** - 283 페이지의 『dspmqtrc(MQSeries 형식화 추적 출력 표시)』 참조
- **endmqtrc** - 295 페이지의 『endmqtrc(MQSeries 추적 종료)』 참조

추적 기능은 추적되는 각 엔티티에 대해 하나의 파일을 사용하여 각 추적 정보를 적절한 파일에 기록합니다.

추적과 연관되는 파일은 `MQS_ROOT:[MQM.TRACE]` 디렉토리에 작성됩니다.

이 디렉토리의 파일에는 큐 관리자의 세부사항과 모든 이전 추적 및 모든 `$$SYSTEM` 추적에 대한 세부사항이 포함되어 있습니다.

추적 파일 이름

추적 파일 이름은 다음 방식으로 구성됩니다.

```
AMQppppppppp.TRC
```

여기서 `ppppppppp`는 추적을 생성하는 프로세스의 ID(PID)입니다.

주:

1. Compaq OpenVMS용 MQSeries에서, 프로세스 ID의 값은 항상 8자입니다.
2. 추적되는 엔티티의 일부로 실행 중인 각 프로세스에 대해 하나의 추적 파일이 있습니다.

샘플 추적 데이터

다음 샘플은 OpenVMS 추적에서 추출된 것입니다.


```

...
ID      ELAPSED_MSEC   DELTA_MSEC     APPL      SYSCALL KERNEL  INTERRUPT
30d     0 0 MQS CEI Exit!. 12484.1 xcsWaitEventSem rc=10806020
30d     0 0 MQS CEI Exit! 12484.1 zcpReceiveOnLink rc=20805311
30d     0 0 MQS FNC Entry 12484.1 zxcProcessChildren
30d     0 0 MQS CEI Entry. 12484.1 xcsRequestMutexSem
30d     1 0 MQS CEI Entry.. 12484.1 xcsHSHMEMBtoPTR
30d     1 0 MQS CEI Exit... 12484.1 xcsHSHMEMBtoPTR rc=00000000
30d     1 0 MQS FNC Entry.. 12484.1 xllSemGetVal
30d     1 0 MQS FNC Exit... 12484.1 xllSemGetVal rc=00000000
30d     1 0 MQS FNC Entry.. 12484.1 xllSemReq
30d     1 0 MQS FNC Exit... 12484.1 xllSemReq rc=00000000
30d     1 0 MQS CEI Exit.. 12484.1 xcsRequestMutexSem rc=00000000
30d     2 0 MQS CEI Entry. 12484.1 xcsReleaseMutexSem
30d     2 0 MQS CEI Entry.. 12484.1 xcsHSHMEMBtoPTR
30d     2 0 MQS CEI Exit... 12484.1 xcsHSHMEMBtoPTR rc=00000000
30d     2 0 MQS FNC Entry.. 12484.1 xllSemRel
30d     2 0 MQS FNC Exit... 12484.1 xllSemRel rc=00000000
30d     2 0 MQS CEI Exit.. 12484.1 xcsReleaseMutexSem rc=00000000
30d     2 0 MQS CEI Entry. 12484.1 xcsHSHMEMBtoPTR
...

```

그림 21. 샘플 Compaq OpenVMS용 MQSeries 추적

주:

1. 이 예에서는 데이터가 잘려 있습니다. 그러나 실제 추적에서는 완전한 함수 이름과 리턴 코드가 표시됩니다.
2. 리턴 코드는 리터럴이 아니라 값으로 제공됩니다.

FFST(First Failure Support Technology)

Compaq OpenVMS용 MQSeries에서 일반적으로 FFST 로그에 기록되는 정보는 MQS_ROOT:[MQM.ERRORS] 디렉토리의 파일에 기록됩니다.

이 오류들은 대개 복구할 수 없는 심각한 오류로써 시스템의 구성 문제점이나 MQSeries 내부 오류를 나타냅니다.

FFST를 조사하는 방법

파일 이름은 AMQnnnnnnnn_mm.FDC 형식으로 지정됩니다. 여기서,

nnnnnnnn

오류를 보고하는 프로세스 ID입니다.

mm 순번입니다(보통 0).

프로세스가 FFST를 작성할 때, 시스템 오류 로그에도 항목을 기록합니다. 레코드에는 자동 문제점 추적시 도움이 되도록 FFST 파일의 이름이 표시되어 있습니다.

MQSeries First Failure Symptom Report

```

=====
Date/Time      :- Monday January 29 21:32:03 GMT 2001
Host Name     :- CELERY (Unknown)
PIDS          :- 5697175
LVLS         :- 510
Product Long Name :- MQSeries for OpenVMS Alpha
Vendor        :- IBM
Probe Id      :- ZX005025
Application Name :- MQM
Component     :- zxcProcessChildren
Build Date    :- Jan  8 2001
Userid       :- [400,400] (SJACKSON)
Program Name  :- AMQZXMA0.EXE
Process      :- 202001DA
Thread       :- 00000001
QueueManager  :- JJJH
Major Errorcode :- zrcX_PROCESS_MISSING
Minor Errorcode :- OK
Probe Type    :- MSGAMQ5008
Probe Severity :- 2
Probe Description :- AMQ5008: An essential MQSeries process 538968541
                  cannot be found and is assumed to be terminated.
Arith1       :- 538968541 202001dd
VMS Errorcode :- -SYSTEM-W-NONEXPR, nonexistent process (000008E8)

```

JPI Quota information:

```

=====
ASTCNT=247/250(98%) *          BIOCNT=500/500(100%) *
BYTCNT=183616/183616(100%) *   DIOCNT=250/250(100%) *
ENQCNT=4885/5000(97%) *       FILCNT=241/250(96%) *
PAGFILCNT=975280/1000000(97%) * TQCNT=246/250(98%) *
FREPTCNT=2147483647          APTCNT=0
GPGCNT=5808                  PPGCNT=5872
VIRTPEAK=203264              DFWSCNT=1392
WSAUTH=2784                  WSAUTHEXT=65536
WSEXTENT=65536               WSPeAK=11680
WSQUOTA=2784                  WSSIZE=15792
CPULIM=0                      MAXDETACH=0
MAXJOBS=0                     JOBPRCCNT=2
PAGEFLTS=2895                 PRCCNT=2/100(2%) +
(*) - % resource remaining, (+) - % resource used

```

Privilege and rights information:

```

=====
CURPRIV=bugchk detach netmbx prmgbl sysgbl sysprv tmpmbx world
IMAGPRIV=bugchk prmgbl sysgbl world
AUTHPRIV=bugchk detach netmbx prmgbl sysgbl sysprv tmpmbx world
SJACKSON                      INTERACT
REMOTE                          MQM
SYS
IMAGE RIGHTS=
SYS$NODE_CELERY

```

SYI information:

```

=====
ACTIVE CPU=1/1(100%) +          CLUSTER NODES=1
FREE_GBLPAGES=16000528/16174643(98%) * GBLPAGFIL=1000000
FREE_GBLSECTS=936/1550(60%) *      MEMSIZE=16384
PAGEFILE_FREE=16888/16888(100%) *   PAGE_SIZE=8192
SWAPFILE_FREE=936/936(100%) *       MAXPROCESSCNT=102
PROCSECTCNT=64                     BALSETCNT=100
WSMAX=65536                          NPAGEDYN=2269184
NPAGEVIR=9437184                    PAGEDYN=1597440
VIRTUALPAGECNT=2147483647           LOCKIDTBL_MAX=109437
PQL_DASTLM=24                        PQL_MASTLM=100
PQL_DBIOLM=32                       PQL_MBIOLM=100
PQL_DBYTLM=65536                    PQL_MBYTLM=100000
PQL_DCPULM=0                         PQL_MCPULM=0
PQL_DDIOLM=32                       PQL_MDIOLM=100
PQL_DFILLM=128                      PQL_MFILLM=100
PQL_DPGFLQUOTA=65536                PQL_MPGFLQUOTA=32768
PQL_DPRCLM=32                       PQL_MPRCLM=10
PQL_DTQELM=16                       PQL_MTQELM=0
PQL_DWSDEFAULT=1392                 PQL_MWSDEFAULT=1392
PQL_DWSQUOTA=2784                   PQL_MWSQUOTA=2784
PQL_DWSEXTENT=65536                 PQL_MWSEXTENT=65536

```

PQL_DENQLM=128	PQL_MENQLM=300
PQL_DJTQUOTA=4096	PQL_MJTQUOTA=0
CLISYMTBL=750	DEFMBXMSG=256
DEFMBXBUFQUO=1056	CHANNELCNT=5000
DLCKEXTRASTK=2560	PIOPAGES=575
CTLPAGES=256	CTLINGLIM=35
(*) - % resource remaining, (+) - % resource used	

```

MQM Function Stack
zxcProcessChildren
xcsFFST
MQM Trace History

```

```

--> xllFreeSem
<-- xllFreeSem rc=OK
--> xcsFreeQuickCell
--> xllSpinLockRequest
<-- xllSpinLockRequest rc=OK
--> xstFreeCell
<-- xstFreeCell rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
<-- xcsFreeQuickCell rc=OK
<-- xcsCloseEventSem rc=OK
--> xcsFreeMemBlock
--> xstFreeMemBlock
--> xcsRequestThreadMutexSem
<-- xcsRequestThreadMutexSem rc=OK
--> xcsReleaseThreadMutexSem
<-- xcsReleaseThreadMutexSem rc=OK
--> xstFreeBlockFromSharedMemSet
--> xllSpinLockSlowRequest
<-- xllSpinLockSlowRequest rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
--> xstFreeBlockInExtent
--> xcsQueryMutexSem
<-- xcsQueryMutexSem rc=OK
--> xcsRequestMutexSem
--> xllSemReq
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemReq rc=OK
<-- xcsRequestMutexSem rc=OK
--> xclDeleteMutexMem
--> xllCSCloseMutex
--> xihHANDLEtoSUBPOOLFn
--> xihGetConnSPDetailsFromList
--> xihGetConnSPDetails
<-- xihGetConnSPDetails rc=OK
<-- xihGetConnSPDetailsFromList rc=OK
<-- xihHANDLEtoSUBPOOLFn rc=OK
--> xllSpinLockSlowRequest
<-- xllSpinLockSlowRequest rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
--> xllFreeSem
<-- xllFreeSem rc=OK
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
--> xcsFreeQuickCell
--> xllSpinLockRequest
<-- xllSpinLockRequest rc=OK
--> xstFreeCell
<-- xstFreeCell rc=OK
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
<-- xcsFreeQuickCell rc=OK
<-- xllCSCloseMutex rc=OK
<-- xclDeleteMutexMem rc=OK
--> xstSerialiseExtent
--> xllSpinLockRequest
<-- xllSpinLockRequest rc=OK
<-- xstSerialiseExtent rc=OK
--> xstFreeChunk
--> xstDeleteChunk

```

```

    <-- xstDeleteChunk rc=OK
    --> xstInsertChunk
    <-- xstInsertChunk rc=OK
    <-- xstFreeChunk rc=OK
    --> xstReleaseSerialisationOnExtent
    --> xllSpinLockRelease
    <-- xllSpinLockRelease rc=OK
    <-- xstReleaseSerialisationOnExtent rc=OK
    <-- xstFreeBlockInExtent rc=OK
    <-- xstFreeBlockFromSharedMemSet rc=OK
    <-- xstFreeMemBlock rc=OK
    <-- xcsFreeMemBlock rc=OK
    <-- zcpDeleteIPC rc=OK
    --> xcsReleaseMutexSem
    --> xllSemRel
    --> vms_mtx
    --> vms_get_lock
    <-- vms_get_lock rc=OK
    <-- vms_mtx rc=OK
    <-- xllSemRel rc=OK
    <-- xcsReleaseMutexSem rc=OK
    --> xcsFreeMemBlock
    --> xstFreeMemBlock
    --> xcsRequestThreadMutexSem
    <-- xcsRequestThreadMutexSem rc=OK
    --> xcsReleaseThreadMutexSem
    <-- xcsReleaseThreadMutexSem rc=OK
    --> xstFreeBlockFromSharedMemSet
    --> xllSpinLockSlowRequest
    <-- xllSpinLockSlowRequest rc=OK
    --> xllSpinLockRelease
    <-- xllSpinLockRelease rc=OK
    --> xstFreeBlockInExtent
    --> xcsQueryMutexSem
    <-- xcsQueryMutexSem rc=OK
    --> xcsRequestMutexSem
    --> xllSemReq
    --> vms_mtx
    --> vms_get_lock
    <-- vms_get_lock rc=OK
    <-- vms_mtx rc=OK
    <-- xllSemReq rc=OK
    <-- xcsRequestMutexSem rc=OK
    --> xclDeleteMutexMem
    --> xllCSCloseMutex
    --> xihHANDLEtoSUBPOOLFn
    --> xihGetConnSPDetailsFromList
    --> xihGetConnSPDetails
    <-- xihGetConnSPDetails rc=OK
    <-- xihGetConnSPDetailsFromList rc=OK
    <-- xihHANDLEtoSUBPOOLFn rc=OK
    --> xllSpinLockSlowRequest
    <-- xllSpinLockSlowRequest rc=OK
    --> xllSpinLockRelease
    <-- xllSpinLockRelease rc=OK
    --> xllFreeSem
    <-- xllFreeSem rc=OK
    --> vms_mtx
    --> vms_get_lock
    <-- vms_get_lock rc=OK
    <-- vms_mtx rc=OK
    --> xcsFreeQuickCell
    --> xllSpinLockRequest
    <-- xllSpinLockRequest rc=OK
    --> xstFreeCell
    <-- xstFreeCell rc=OK
    --> xllSpinLockRelease
    <-- xllSpinLockRelease rc=OK
    <-- xcsFreeQuickCell rc=OK
    <-- xllCSCloseMutex rc=OK
    <-- xclDeleteMutexMem rc=OK
    --> xstSerialiseExtent
    --> xllSpinLockRequest
    <-- xllSpinLockRequest rc=OK
    <-- xstSerialiseExtent rc=OK
    --> xstFreeChunk
    --> xstDeleteChunk
    <-- xstDeleteChunk rc=OK
    --> xstInsertChunk
    <-- xstInsertChunk rc=OK

```

```

<-- xstFreeChunk rc=OK
--> xstReleaseSerialisationOnExtent
--> xllSpinLockRelease
<-- xllSpinLockRelease rc=OK
<-- xstReleaseSerialisationOnExtent rc=OK
<-- xstFreeBlockInExtent rc=OK
<-- xstFreeBlockFromSharedMemSet rc=OK
<-- xstFreeMemBlock rc=OK
<-- xcsFreeMemBlock rc=OK
<-- zxcCleanupAgent rc=OK
--> xcsReleaseMutexSem
--> xllSemRel
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemRel rc=OK
<-- xcsReleaseMutexSem rc=OK
--> xcsCheckProcess
--> kill
<-- kill rc=OK
<-- xcsCheckProcess rc=OK
--> xcsCheckProcess
--> kill
<-- kill rc=OK
<-- xcsCheckProcess rc=OK
--> xcsRequestMutexSem
--> xllSemReq
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemReq rc=OK
<-- xcsRequestMutexSem rc=OK
--> xcsReleaseMutexSem
--> xllSemRel
--> vms_mtx
--> vms_get_lock
<-- vms_get_lock rc=OK
<-- vms_mtx rc=OK
<-- xllSemRel rc=OK
<-- xcsReleaseMutexSem rc=OK
--> xcsCheckProcess
--> kill
<-- kill rc=Unknown(FFFF)
<-- xcsCheckProcess rc=xecP_E_INVALID_PID
--> xcsBuildDumpPtr
--> xcsGetMem
<-- xcsGetMem rc=OK
<-- xcsBuildDumpPtr rc=OK
<-- xcsBuildDumpPtr rc=OK
<-- xcsBuildDumpPtr rc=Unknown(4B)
--> xcsFFST

```

ECAnchor					
6A91B0				5A584541	ZXEA
6A91C0	03000000	E8030000	03220000	0100DA01è...."....Ú..
6A91D0	2C05A500	D4040000	0A00DA01	01000000	,.¥.Ô....Ú.....
6A91E0	B0000000	0A00DA01	03000000	E8030000	°....Ú.....è...
6A91F0	03220000	0100DA01	07000000	00000000	"....Ú.....
6A9200	283F9700	03000000	F0030000	08410000	(?--.....A..
6A9210	0300DA01	03000000	F0030000	08410000	..Ú.....A..
6A9220	0300DA01	01000000	B4000000	0200DA01	..Ú.....Ú..
6A9230	03000000	E8030000	03220000	0100DA01è...."....Ú..
6A9240	00000000	00000000	00000000	00000000
6A9250	00000000	00000000	00000000	00000000
6A9260	00000000	01000000	B4000000	0200DA01Ú..
6A9270	03000000	E8030000	03220000	0100DA01	,...è...."....Ú..
6A9280	B41F0000	0200DA01	01000000	B4000000Ú.....
6A9290	0200DA01	03000000	E8030000	03220000	..Ú.....è...."
6A92A0	0100DA01	DA012020	00000000	EFCD0300	..Ú.Ú..Í..
6A92B0	00000000	00000000	00000000	00000000
6A92C0	00000000	00000000	01000000	00000000
6A92D0	44454641	554C5400	00000000	00000000	DEFAULT.....
6A92E0	00000000	00000000	00000000	00000000
6A92F0	00000000	00000000	00000000	00000000
6A9300	2F6D7173	5F726F6F	742F6D71	6D000000	/mq_s_root/mqm...
6A9310	00000000	00000000	00000000	00000000
6A9320	to 6A93F0 suppressed, lines same as above				
6A9400	5A435048	01000000	B8000000	0400DA01	ZCPH....¸....Ú..
6A9410	03000000	F0030000	08410000	0300DA01A....Ú..

FFST

```

6A9420 DC040000 0400DA01 01000000 B8000000  &#220;.....&#218;.....&#184;...
6A9430 0400DA01 03000000 F0030000 08410000  ..&#218;.....A..
6A9440 0300DA01 00000000 00000000 00000000  ..&#218;.....
6A9450 00000000 00000000 00000000 00000000  .....
6A9460 to 6A9470 suppressed, lines same as above
6A9480 00000000 00000000 5A435048 01000000  .....ZCPH....
6A9490 B8000000 0500DA01 03000000 F0030000  &#184;.....&#218;.....
6A94A0 08410000 0300DA01 DC040000 0500DA01  .A....&#218;.&#220;.....&#218;..
6A94B0 01000000 B8000000 0500DA01 03000000  ...&#184;.....&#218;.....
6A94C0 F0030000 08410000 0300DA01 4C4B0000  .....A....&#218;..LK..
6A94D0 0500DA01 01000000 B8000000 0500DA01  ..&#218;.....&#184;.....&#218;..
6A94E0 03000000 F0030000 08410000 0300DA01  .....A....&#218;..
6A94F0 07000000 00090000 50140000 0100DA01  .....P....&#218;..
6A9500 01000000 B0000000 0100DA01 03000000  .....&#218;.....
6A9510 E8030000 03220000 0100DA01 98170000  &#232;....."....&#218;..~...
6A9520 0200DA01 01000000 B4000000 0200DA01  ..&#218;.....&#218;..
6A9530 03000000 E8030000 03220000 0100DA01  ...&#232;....."....&#218;..
6A9540 00000000 00000000 08000000 3C0A0000  .....<...
6A9550 50140000 0100DA01 01000000 B0000000  P....&#218;.....°...
6A9560 0100DA01 03000000 E8030000 03220000  ..&#218;.....&#232;....."..
6A9570 0100DA01 08180000 0200DA01 01000000  ,&#218;.....&#218;.....
6A9580 B4000000 0200DA01 03000000 E8030000  .....&#218;.....&#232;...
6A9590 03220000 0100DA01 00000000 00000000  ."....&#218;.....
6A95A0 00000000 00000000 00000000 00000000  .....
6A95B0 to 6A9560 suppressed, lines same as above
6A9660 01000000 00000000 78180000 0200DA01  ....x....&#218;..
6A9670 01000000 B4000000 0200DA01 03000000  .....&#218;.....
6A9680 E8030000 03220000 0100DA01 09000000  &#232;....."....&#218;.....
6A9690 780B0000 50140000 0100DA01 01000000  x...P....&#218;.....
6A96A0 B0000000 0100DA01 03000000 E8030000  °....&#218;.....&#232;...
6A96B0 03220000 0100DA01 00000000 00000000  ."....&#218;.....
6A96C0 00000000 00000000 00000000 00000000  .....
6A96D0 to 6A9750 suppressed, lines same as above
6A9760 00000000 00000000 DD012020 00000000  .....&#221;.. ....
6A9770 00000000 00000000 01000000 09000000  .....
6A9780 00000000

```

기능 스택 및 추적 실행 기록은 문제점 판별시 도움이 되도록 IBM에서 사용합니다. 대부분의 경우 지원 센터를 통해 제기되는 문제점과는 별도로, FFST가 생성될 때 시스템 관리자가 수행할 수 있는 사항은 거의 없습니다.

그러나 해결할 수 있는 문제점도 있습니다. 내부 함수 중 하나를 호출할 때 FFST에서 『할당량 초과』 또는 『다바이스의 공간 부족』이란 설명을 표시할 경우, 이것은 관련 SYSGEN 매개변수 한계를 초과한 것입니다.

문제점을 해결하려면 시스템 매개변수를 조정하여 내부 한계를 늘려야 합니다. 자세한 정보는 183 페이지의 『제13장 MQSeries 구성』을 참조하십시오.

클라이언트의 문제점 판별

MQI 클라이언트 응용프로그램은 클라이언트가 아닌 MQI 응용프로그램과 동일한 방식으로 MQRC_* 이유 코드를 수신합니다. 그러나 이제 클라이언트와 연관되는 오류 상태에 대한 추가 이유 코드를 알 수 있습니다. 예를 들면 다음과 같습니다.

- 리모트 시스템이 응답하지 않음
- 통신 회선 오류
- 올바른지 않은 시스템 주소

일반적으로 오류가 가장 많이 발생하는 시간은 응용프로그램이 MQCONN을 발행하고 MQRC_Q_MQR_NOT_AVAILABLE 응답을 수신할 때입니다. 클라이언트 로그 파일에 기록된 오류 메시지에 오류 원인이 설명되어 있습니다. 메시지는 오류 원인에 따라 서버에 기록될 수도 있습니다.

클라이언트 종료

클라이언트가 종료되어도 서버에서 프로세스를 수행하기 위해 큐를 열어 놓을 수 있습니다. 대개 이것은 통신 계층에서 상대방과의 연결이 중단되었음을 감지할 때까지의 짧은 시간입니다.

클라이언트의 오류 메시지

클라이언트 시스템에서 오류가 발생할 경우, 서버와 연관되는 오류 파일에 오류 메시지를 넣습니다(put). 오류가 없으면, 클라이언트 코드는 클라이언트 시스템의 루트 디렉토리에 있는 오류 로그에 오류 메시지를 넣으려고 합니다.

OS/2, UNIX 및 OpenVMS 시스템 클라이언트

OS/2, UNIX 및 OpenVMS 시스템 클라이언트의 오류 메시지는 각각의 MQSeries 서버 시스템에 있는 오류 로그에 위치합니다. 일반적으로 이 파일은 OpenVMS 시스템의 MQS_ROOT:[MQM.ERRORS] 디렉토리와 UNIX 시스템의 /var/mqm/errors에 나타납니다.

DOS 및 Windows 클라이언트

로그 파일 AMQERR01.LOG의 위치는 MQDATA 환경 변수에 의해 설정됩니다. MQDATA에 의해 대체되지 않을 경우, 디폴트 위치는 다음과 같습니다.

C:\

DOS 환경에서 작업할 때는 환경 변수 MQDATA가 포함됩니다.

추적 및 오류 정보를 저장하기 위해 클라이언트 코드에서 사용되는 디폴트 라이브러리이며, qm.ini 파일이 저장되는 디렉토리 이름도 포함됩니다(NetBIOS 설정에 필요한 경우). 지정하지 않으면, 디폴트로 C 드라이브가 사용됩니다.

이 라이브러리에 있는 디폴트 파일의 이름은 다음과 같습니다.

AMQERR01.LOG

오류 메시지의 경우

AMQERR01.FDC

최초 실패 데이터 캡처 메시지의 경우

클라이언트 문제점 판별

제15장 성능 조정

이 장에서는 OpenVMS 시스템의 성능을 조정하여 MQSeries로부터 최상의 성능을 얻는 방법에 대해 설명합니다.

MQSeries와 같은 제품에서, 모든 환경에 알맞은 다양한 성능 조정 OpenVMS 매개변수 값을 정의한다는 것은 불가능합니다. 가장 효율적인 값은 대체로 MQSeries 자체의 워크로드와 OpenVMS 시스템에 의해 판별됩니다. *Compaq OpenVMS용 MQSeries Alpha, 버전 5.1* 빠른 시작의 매개변수 설정에서 가능한 최소 또는 초기 값을 제공하기는 하지만, 이 값들은 큐 관리자에 대한 워크로드가 커질수록 증가해야 합니다. 이를 수행하는 프로세스를 "성능 조정"이라고 합니다.

OpenVMS 시스템의 성능 조정에 대해서는 *OpenVMS Performance Management*에 설명되어 있습니다. 그 서적에 있는 정보와 특히 MQSeries에 관련되는 다음 사항들을 참조하십시오.

- 일부 성능 조정 매개변수는 시스템에 전체적으로 적용됩니다(예: GBLPAGES). 이 매개변수들은 SYSGEN 유틸리티에 의해 제어되기 때문에 SYSGEN 매개변수라고도 합니다. 정기적으로 사용되면, AUTOGEN FEEDBACK 메커니즘은 시스템에서 사용하는 자원을 모니터링하고 SYSGEN 매개변수를 자동으로 조정하여 변경 워크로드를 추적합니다. 그러면 시스템을 적절히 조정된 상태로 유지하는 데 필요한 수작업이 크게 감소할 뿐만 아니라 리소스 부담으로 인한 오류 발생을 피할 수 있습니다. MQSeries에 관련되는 SYSGEN 매개변수는 다음과 같습니다.

GBLPAGES, GBLSECTIONS 및 GBLPAGFIL

큐 관리자는 공유(전역) 메모리를 통해 통신하는 일련의 통합 프로세스로 구현됩니다. 그러므로, 전역 메모리를 제어하는 SYSGEN 매개변수 (GBLPAGES, GBLSECTIONS 및 GBLPAGFIL)의 크기가 충분하도록 해야 합니다. *Compaq OpenVMS용 MQSeries Alpha, 버전 5.1* 빠른 시작에서 이 매개변수들의 가능한 초기 값이 제공되어 있습니다. MQSeries 사용자 수가 늘어나면, 전역 메모리 수요가 증가되기 때문에 해당 SYSGEN 매개변수도 증가되어야 합니다.

CHANNELCNT

큐 관리자는 프로세스간 통신과 동기화 메커니즘으로 OpenVMS 편지함을 사용합니다. 이 편지함은 채널을 통해 액세스할 수 있으므로, SYSGEN 매개변수 CHANNELCNT가 충분히 커야 합니다. 대부분의 경우에는 설치시 설정된 값으로도 충분합니다. 그러나 여러 개의 활성 MQSeries 프로세스가 있는, 과중하게 로드된 시스템에서는 이 값을 증가시켜야 합니다.

SYSGEN 매개변수를 명확하게 설정하려면, *OpenVMS Performance Management* 에 설명된 대로 MODPARAMS.DAT 파일을 수정하십시오.

- 일부 OpenVMS 조정 매개변수는 개인 사용자 이름이나 프로세스(예: PGFLQUOTA) 에 적용됩니다. 이 매개변수들은 일반적으로(모든 경우에 적용되는 것은 아님) AUTHORIZE 유틸리티에 의해 제어됩니다. 이러한 매개변수를 자동으로 조정하는 방법은 없습니다. 그러나 이 매개변수들이 특정 프로세스에서 사용할 수 있는 자원의 양에는 한계가 있으므로, 일반적으로 필요한 것보다 많이 자원을 설정하여 로드가 크게 발생할 경우에 대비한 여분의 용량을 제공해야 합니다. MQSeries 관련 프로세스 특정 매개변수는 다음과 같습니다.

PGFLQUOTA

이 매개변수는 프로세스에서 사용할 수 있는 페이지파일 공간의 양을 제어합니다. MQSeries 프로세스는 일반적으로 크기가 아주 크거나 많은 수의 메시지를 이동하므로, 많은 페이지파일 공간을 소비합니다.

PRCLM

이 매개변수는 주어진 프로세스가 작성할 수 있는 서브프로세스 수를 제어합니다. 대부분의 MQSeries 프로세스가 실행 제어기의 서브프로세스로 작성되므로, 시스템은 PRCLM에 대해 높은 값을 요구합니다.

ENQLM, ASTLM, TQELM

앞서 언급한 것과 같이, 큐 관리자는 일련의 통합 프로세스로 구현됩니다. 이 프로세스들은 OpenVMS 잠금 관리자, ATS(Asynchronous System Traps) 및 타이머를 사용하여 활동을 동기화합니다. 이 자원들의 사용을 제한하는 세 개의 매개변수는 큐 관리자 필요성에 부합되도록 충분히 큰 값으로 설정해야 합니다.

프로세스 고유 매개변수의 값 설정

이러한 매개변수를 설정하는 가장 일반적인 방법은 적절한 사용자 이름(MQSeries의 경우 보통 MQM)에 대한 값을 조정하는 것입니다.

그러나 OpenVMS에서, 일부 프로세스 할당량은 같은 작업의 모든 프로세스 즉, 상위 프로세스와 상위 프로세스의 서브프로세스인 다른 모든 프로세스에서 공유합니다. 이 범주에서의 프로세스 할당량에는 BYTLM, FILLM, PGFLQUOTA, PRCLM, TQELM, ENQLM이 포함되는데 이들을 *풀에 지정된 할당량*이라고 합니다.

대부분의 큐 관리자 프로세스는 실행 제어기의 서브프로세스로 작성되므로, 모든 큐 관리자 프로세스가 풀에 지정된 할당량을 공유하게 됩니다. 그러므로 일반적으로 단일 프로세스에 필요한 값보다 큰 값으로 이 할당량을 설정해야 합니다. 이는 특히 PGFLQUOTA에 적용되는데, 그 이유는 이 매개변수가 큐 관리자 프로세스에서 일괄적으로 작성할 수 있는 가상 메모리 양을 제한하기 때문입니다. 따라서 실행 제어기가 시작되면, 제어기는 AUTHORIZE가 유지하는 권한 파일의 초기 할당 값을 확보하지

않고 적정한 값으로 할당 값을 설정합니다. 이렇게 되면, AUTHORIZE는 더이상 이 값들을 수정하는데 사용할 수 없습니다. 그 대신에 다음 논리 이름을 사용하여 명확한 할당 값으로 이를 대체할 수 있습니다.

```
MQS_ASTLM
MQS_BIOLM
MQS_BYTLM
MQS_DIOLM
MQS_ENQLM
MQS_FILLM
MQS_PGFLQUOTA
MQS_PRCLM
MQS_TQELM
```

SYS\$MANAGER:MQS_SYSTARTUP.COM이라는 파일을 사용하여 이 논리를 설정할 수 있습니다. MQSeries에서는 편집과 이름 변경이 가능한 SYS\$MANAGER:MQS_SYSTARTUP.TEMPLATE 파일을 제공합니다. 예를 들어, PGFLQUOTA 매개변수에 대해 다른 값을 제공하려면 다음을 수행하십시오.

1. .TEMPLATE 파일 MQS_SYSTARTUP.TEMPLATE를 .COM 파일로 복사하십시오.
2. 방금 작성한 MQS_SYSTARTUP.COM 파일을 편집하여 mqs_pgflquota와 같이 프로세스 할당량에 해당하는 논리를 정의하는 행에 대한 주석 처리를 해제(활성화)하십시오.
3. 새로운 값을 정의하십시오.
예를 들면 다음과 같습니다.

```
$! DEFINE/SYSTEM MQS_PGFLQUOTA 1000000
```

다음과 같이 될 수 있습니다.

```
$ DEFINE/SYSTEM MQS_PGFLQUOTA 5000000
```

4. 논리를 정의하려면 mqs_systartup 파일을 실행하십시오. 예를 들면 다음과 같습니다.

```
$ @sys$manager:mqs_systartup
```

일반적으로, 이것은 시스템 시동 프로시저의 일부로 수행됩니다.

풀에 지정된 할당량의 부족 현상은 일반적으로 새 클라이언트 응용프로그램이 큐 관리자에 연결하는데 실패하거나 새 연결에 너무 많은 자원을 소비하여 일부 다른 응용프로그램이 연결에 실패할 경우 현저하게 드러납니다.

프로세스 고유 매개변수 설정

또한 실행 제어기는 여러 할당량에 대해 명확한 설정을 가지고 시작하기 때문에, SYSGEN PQL_D* 매개변수는 EC에 적용되지 않는다는 점에 유의하십시오.

제16장 OpenVMS용 MQSeries 및 클러스터링

OpenVMS 클러스터 및 *MQSeries 큐 관리자 클러스터*는 서로 독립적인 별개의 두 클러스터입니다.

주: 클러스터라는 용어가 사용되면 이는 MQSeries 큐 관리자 클러스터를 가리킵니다. OpenVMS 클러스터는 항상 *OpenVMS 클러스터*라고 합니다.

MQSeries 큐 관리자 클러스터는 OpenVMS 클러스터 상호통신 프로토콜, OpenVMS 클러스터 분배 잠금 관리자나 OpenVMS 클러스터 파일 시스템을 반드시 사용하지 않습니다. MQSeries 클러스터의 큐 관리자들 사이의 모든 통신은 지원되는 프로토콜 중 하나를 사용하여 MQSeries 채널을 통합합니다. 따라서 동일한 OpenVMS 클러스터의 부분이 아닌 OpenVMS 시스템에서 실행되는 큐 관리자에 대해 MQSeries 큐 관리자 클러스터를 구성할 수 있습니다.

MQSeries 큐 관리자가 OpenVMS 클러스터 내에 구성되어 있는 경우, MQSeries 큐 관리자는 OpenVMS 클러스터 내에서 한번에 하나의 OpenVMS 노드(이 장의 이후에서는 노드라고 함)에서만 실행될 수 있습니다. 단일 MQSeries 큐 관리자의 기능은 OpenVMS 클러스터 내에서 여러 OpenVMS 노드에 거쳐 분배될 수 없습니다. 여러 OpenVMS 노드에서 MQSeries 큐 관리자를 시작하려고 하면 오류가 리턴됩니다. 그러나 MQSeries 클러스터에 여러 MQSeries 큐 관리자가 구성되어 있는 경우, 큐 관리자들은 OpenVMS 클러스터 내의 다른 OpenVMS 노드에서 실행될 수 있습니다.

OpenVMS 클러스터에서 MQSeries 큐 관리자에 대해 더 높은 레벨의 가용성을 제공하기 위해, MQSeries V5.1에 Failover 세트라는 새로운 기능을 도입했습니다. 이로서, 큐 관리자는 실패할 경우 다른 OpenVMS 클러스터 노드에서 자동으로 재시작할 수 있습니다. 이 기능은 MQSeries 큐 관리자 클러스터와 함께 사용하거나 클러스터 없이 사용할 수 있습니다(236 페이지의 『OpenVMS 클러스터 Failover 세트』 참조).

OpenVMS 클러스터에 MQSeries 설치

OpenVMS 클러스터에 Compaq OpenVMS Alpha용 MQSeries V5.1을 설치하는 것은 독립형 OpenVMS 시스템에 MQSeries를 설치하는 것과 아주 유사합니다. 그러나 설치하기 전에 다음을 고려해야 합니다.

- OpenVMS 클러스터에 여러 시스템 디스크가 있는 경우, MQSeries는 부팅된 노드가 있고 MQSeries를 실행해야 하는 각 시스템 디스크에 MQSeries가 설치되어야 합니다. MQSeries는 노드당 한번이 아니라 시스템 디스크당 한번만 설치해야 합니다.
- MQS_ROOT 디렉토리 구조를 보유하는 디스크는 디렉토리 구조 내에 포함된 큐 관리자를 실행할 OpenVMS 노드에서 시스템 전반에 걸쳐 마운트되어 있어야 합니다.

OpenVMS 클러스터에 설치

각 노드마다 다른 MQS_ROOT 디렉토리 구조를 가질 수 있습니다. 그러나 Failover 세트가 구성될 경우, Failover 세트의 각 OpenVMS 노드는 같은 MQS_ROOT 디렉토리 구조를 참조해야 합니다. MQSeries를 설치할 때 각 설치마다 MQS_ROOT 디렉토리를 지정해야 합니다(질문 'MQSeries 데이터 파일에 대한 루트 디바이스 입력'에 대한 응답에서).

- 큐 관리자에 대한 로그 파일을 포함하는 디스크가 MQS_ROOT를 포함하는 디스크와 다를 경우, 로그 파일을 포함하는 디스크는 Failover 세트의 모든 노드에서 시스템 전반에 걸쳐 마운트되어야 합니다.
- MQSeries는 디폴트 디렉토리 SYS\$SPECIFIC:[MQS_SERVER]을 가지고 있는 계정 MQM을 사용합니다. 이 디렉토리는 MQSeries가 설치된 노드에 대해서만 작성됩니다. 디렉토리는 같은 시스템 디스크에서 부팅을 하고, MQSeries를 실행해야 하는 각 추가 노드에 대해 작성되어야 합니다. 이것은 각 추가 노드에서 다음 DCL 명령을 실행하여 수행될 수 있습니다.

```
$create/directory sys$specific:[mq_server]/owner=[mq_server] -  
/protection=(s:rwed,o:rwed,g,w)  
$set sec/acl=(identifier=mqm,options=default,access=r+w+e+d+c) -  
sys$specific:[000000]mq_server.dir  
$set sec/acl=(identifier=mqm,access=r+w+e+d+c) -  
sys$specific:[000000]mq_server.dir
```

OpenVMS 클러스터 Failover 세트

OpenVMS 클러스터 Failover 세트의 개요

OpenVMS 클러스터 Failover 세트는 Compaq OpenVMS용 MQSeries, V5.1에서 사용 가능한 새로운 기능입니다. 이 세트를 사용하면 MQSeries 큐 관리자가 실패할 경우 OpenVMS 클러스터의 다른 OpenVMS 노드에서 자동으로 재시작할 수 있습니다. OpenVMS 클러스터 Failover 세트에서는 다음 유형의 실패가 지원됩니다.

- MQSeries 큐 관리자를 실행하는 OpenVMS 노드 정지
- MQSeries 큐 관리자를 실행하는 OpenVMS 노드의 시스템 손상
- MQSeries 큐 관리자의 명확한 종료 없이 MQSeries 큐 관리자를 실행하는 OpenVMS 노드 종료
- MQSeries 큐 관리자 실행 제어기 프로세스의 실패

다음 유형의 실패는 OpenVMS 클러스터 Failover 세트에서 지원되지 않습니다.

- 노드 또는 MQSeries 큐 관리자의 실패를 가져오지 않는 MQSeries 큐 관리자를 실행하는 OpenVMS 노드 결합
- 실행 제어기 프로세스를 제외하고 MQSeries 큐 관리자 프로세스의 실패. MQSeries 큐 관리자는 같은 노드에서 자동으로 재시작되지 않습니다.

OpenVMS 클러스터 Failover 세트 개요

- MQSeries 큐 관리자 큐 파일 및 로그 데이터가 있는 디스크의 소프트웨어 또는 하드웨어 실패
- MQSeries 큐 관리자 큐 파일 또는 로그 데이터 손상

OpenVMS 클러스터 Failover 세트는 MQSeries 채널용 TCP/IP 프로토콜을 사용하는 큐 관리자에 대해서만 지원됩니다. 다음의 TCP/IP 스택이 지원됩니다.

- OpenVMS용 Digital TCP/IP 서비스 V5.0A
- OpenVMS용 Porcess Software's TCPware V5.4
- OpenVMS용 Process Software's Multinet 4.3

OpenVMS 클러스터 Failover 세트 개념

OpenVMS 클러스터 Failover 세트는 MQSeries 큐 관리자를 실행할 가능성이 있는 OpenVMS 노드의 컬렉션입니다. OpenVMS 클러스터 Failover 세트에는 1 - 4개 사이의 OpenVMS 노드가 있을 수 있으며 모든 OpenVMS 노드는 동일한 OpenVMS 클러스터의 구성원이어야 합니다. OpenVMS 클러스터 Failover 세트는 하나의 MQSeries 큐 관리자에 대해 고유합니다. 하나의 OpenVMS 클러스터에 여러 개의 OpenVMS 클러스터 Failover 세트가 구성되어 있을 수 있습니다. OpenVMS 클러스터 Failover 세트에서 지원되는 큐 관리자 이름의 최대 길이는 25자입니다.

*Failover*는 MQSeries 큐 관리자가 지원되는 실패 발생시 다른 OpenVMS 노드에서 재시작되는 프로세스입니다. 이 프로세스가 완료되면 MQSeries 큐 관리자에 *Failover* 가 수행된 것입니다.

*Failback*은 실패가 해결된 후 원래의 OpenVMS 노드에서 MQSeries 큐 관리자가 재시작되는 프로세스입니다. OpenVMS 클러스터 Failover 세트는 자동 *Failback*을 지원하지 않지만, 수동으로 수행할 수는 있습니다. 이 프로세스가 완료되면 MQSeries 큐 관리자에 *Failback*이 수행된 것입니다.

Failover 모니터는 OpenVMS 클러스터 Failover 세트의 각 구성원에서 실행되는 프로세스입니다. *Failover* 모니터는 *Failover* 세트에 있는 모든 기능을 수행합니다. OpenVMS 클러스터 Failover 세트 내의 *Failover* 모니터는 이러한 기능을 제공하기 위해 상호 협력합니다. *Failover* 모니터는 **runmqfm** 명령을 사용하여 시작됩니다(이 명령에 대해서는 313 페이지의 『runmqfm(Failover 모니터 시작)』을 참조하십시오).

하나의 *Failover* 모니터는 감시자 *Failover* 모니터로 지정되며 이 *Failover* 모니터는 감시 상태에 있는 것입니다. *Failover* 세트에서 시작할 첫 번째 *Failover* 모니터는 초기 감시자 *Failover* 세트입니다. *Failover* 세트는 첫 번째 *Failover* 모니터가 시작될 때 활성 상태가 됩니다. 감시자 *Failover* 모니터가 실패하거나, 실행 중인 OpenVMS 노드가 실패할 경우, 다른 *Failover* 모니터가 감시자 *Failover* 모니터로 지정됩니다. 감시자 *Failover* 모니터는 MQSeries 큐 관리자가 실행 중인지 점검하고 지원되는 실패 발생

OpenVMS 클러스터 Failover 세트 개념

시 Failover 조작을 시작합니다. 다른 OpenVMS 노드에서 수행되어야 하는 조작은 감시자 Failover 모니터에 의해 실제로 조작을 수행하는 관련 OpenVMS 노드에서 실행되는 Failover 모니터로 전달됩니다.

OpenVMS 클러스터 Failover 세트는 DCL 명령 **failover**를 사용하여 관리됩니다. **failover** 명령은 OpenVMS 클러스터 Failover 세트의 어떤 노드에서도 사용할 수 있습니다. 모든 명령은 명령을 처리해야 하는 Failover 모니터를 결정하고 필요하면 이를 다른 Failover 모니터로 전달하는 감시자 Failover 모니터로 송신됩니다.

OpenVMS 클러스터 Failover 세트 구성 파일은 OpenVMS 노드 수 및 이름을 포함하여, OpenVMS 클러스터 Failover 세트의 세부사항을 보유합니다. 파일은 FAILOVER.INI이며 MQS_ROOT:[MQM.QMGRS.queueanagername] 디렉토리에 있습니다. 이것은 텍스트 편집기로 수정되는 텍스트 파일로, 첫 번째 Failover 모니터를 시작하기 전에 작성해야 합니다. FAILOVER.TEMPLATE라고 하는 템플릿 구성 파일은 MQS_EXAMPLES 디렉토리에 제공됩니다. 구성 파일의 매개변수는 동적으로 변경할 수 없습니다. 변경사항을 적용하려면 모든 Failover 모니터가 정지된 후 다시 시작되어야 합니다. Failover 모니터가 시작되지 않으면 MQSeries 큐 관리자의 자동 Failover가 발생하지 않을 수 있으므로 주의해야 합니다.

Failover 세트의 MQSeries 큐 관리자에 대해, 모든 MQSeries 명령은 **strmqm** 및 **endmqm** 명령을 제외하고 정상적으로 계속 작동합니다. MQSeries 큐 관리자가 활성 Failover 세트에 있을 때 이 두 가지 명령은 오류를 리턴합니다. **failover** 명령은 MQSeries 큐 관리자를 시작하고 종료하는 데 사용해야 합니다.

OpenVMS 노드 우선순위는 OpenVMS 클러스터 Failover 세트에 있는 각 OpenVMS 노드에 제공된 우선순위이며, 실패 발생 후 큐 관리자가 시작되어야 하는 OpenVMS 노드를 판별하는 데 사용됩니다. 가장 낮은 숫자의 우선순위 값을 갖는 OpenVMS 노드의 우선순위가 최상위입니다.

OpenVMS 클러스터 Failover 세트 TCP/IP 주소는 Failover 세트에 지정된 TCP/IP 주소입니다. Failover 세트 큐 관리자를 참조하는 모든 채널은 연결 이름에서 이 TCP/IP 주소를 지정하여 구성되어야 합니다. 각 OpenVMS 클러스터 Failover 세트는 고유한 TCP/IP 주소를 사용해야 합니다. OpenVMS 클러스터 Failover 세트의 모든 OpenVMS 노드는 동일한 서브넷에 있는 인터페이스 TCP/IP 주소를 가지고 있어야 하고 OpenVMS 클러스터 Failover 세트 TCP/IP 주소는 동일한 서브넷에 있어야 합니다.

OpenVMS 클러스터 Failover 세트 구성 준비

다음 단계들은 OpenVMS 클러스터 Failover 세트를 구성하기 전에 수행해야 합니다.

1. 큐 관리자가 없을 경우 **crtmqm**을 사용하여 큐 관리자를 작성하십시오.
2. OpenVMS 클러스터 Failover 세트에 대한 TCP/IP 주소를 확보하십시오.

OpenVMS 클러스터 Failover 세트 구성

3. OpenVMS 클러스터 TCP/IP Failover 세트 TCP/IP 주소를 사용할 수 있도록 MQSeries 채널을 작성하거나 수정하십시오.
4. OpenVMS 노드들이 OpenVMS 클러스터 Failover 세트에 있어야 하는지 판별하고 각각의 우선순위를 결정하십시오.
5. MQS_ROOT 논리가 OpenVMS 클러스터 Failover 세트에서 모든 OpenVMS 노드에 대해 같은 디렉토리를 참조하는지, 디스크가 모든 노드에서 시스템 전반에 마운트되어 있는지 확인하십시오. 디스크를 지원하는 노드가 사용 불가능하게 될 경우 해당 노드가 더이상 디스크에 액세스할 수 없으므로 MQS_ROOT 디렉토리 및 로그 파일을 포함하는 디스크는 Failover 세트의 한 노드에서 다른 노드로 지원되는 MSCP가 아니어야 합니다.

OpenVMS 클러스터 Failover 세트 구성

다음은 OpenVMS 클러스터 Failover 세트를 구성하는 데 필요한 단계입니다.

1. MQS_EXAMPLES:FAILOVER.TEMPLATE 파일을 MQS_ROOT:[MQM.QMGRS.queuemanagename] FAILOVER.INI에 복사하십시오.
2. MQS_ROOT:[MQM.QMGRS.queuemanagename] FAILOVER.INI 파일을 편집하여 이 OpenVMS 클러스터 Failover 세트 구성에 맞게 수정하십시오(240 페이지의 『FAILOVER.INI 구성 파일 편집』 참조).
3. **START_QM.COM**, **END_QM.COM** 및 **TIDY_QM.COM** 명령 프로시저를 편집하십시오(241 페이지의 『Failover 세트에서 사용되는 명령 프로시저』 참조).
4. Failover 모니터에서 사용되는 ICC 연관의 ICC 보안을 설정하십시오(248 페이지의 『ICC 연관에 대한 보안 설정』 참조).
5. **runmqfm -m queuemanagename** 명령을 사용하여 OpenVMS 클러스터 Failover 세트의 각 노드에서 Failover 모니터를 시작하십시오.
6. **failover -m queuemanagename -n nodename -s** 명령을 사용하여 큐 관리자를 시작하십시오.
7. 사이트 특정 종료룰을 수정하여 다음을 수행하십시오.
 - 큐 관리자가 종료될 때 노드에서 실행 중이면 그 큐 관리자를 종료하거나 이동하십시오.
 - Failover 모니터를 정지하십시오.

OpenVMS 클러스터 Failover 세트 구성 후 작업

다음은 클러스터 Failover 세트가 구성된 후 수행할 수 있는 작업입니다.

- **runmqfm** 명령을 사용하여 OpenVMS 클러스터 Failover 세트의 각 노드에서 Failover 모니터 프로세스를 시작할 수 있도록 시스템 시동 파일을 편집하십시오. **runmqfm** 명령은 MQSeries 시작 명령 다음에 와야 합니다.

OpenVMS 클러스터 Failover 세트 구성

- 시스템 시동시 자동으로 큐 관리자를 시작하기 위해 이 명령이 필요하면, 관련 노드의 시스템 시동에 명령을 위치시켜 Failover 모니터가 시작된 후에 큐 관리자를 시작하십시오. 노드에서 큐 관리자를 시작하는 명령은 **failover -m queuemanagename -n nodename -s**입니다.
- 시스템 종료시 Failover 모니터를 종료하려면 사이트 특정 종료를 수정하십시오. 또한 큐 관리자가 종료될 때 노드에서 실행 중이면 그 큐 관리자를 종료하거나 이동하십시오.

FAILOVER.INI 구성 파일 편집

FAILOVER.INI 파일은 각 OpenVMS 클러스터 Failover 세트에 알맞게 사용자 정의되어야 합니다. 각 필드의 의미는 표9에 나열되어 있습니다. MQS_EXAMPLES에 제공된 템플릿 구성 파일은 355 페이지의 『부록F. OpenVMS 클러스터 Failover 세트 템플릿』에 포함되어 있습니다. ‘#’ 문자로 시작하는 파일의 행들은 Failover 모니터 프로세스에 의해 읽혀질 때 무시됩니다. 파일 내에서 필드 이름의 대소문자는 템플릿 파일에 지정된 내용을 따라야 합니다. 각 필드 이름 다음에는 ‘=’ 문자와 연관된 값이 차례로 와야 합니다. 템플릿 파일의 모든 필드는 필수이므로 어떤 필드도 제거해서는 안 됩니다.

표9. FAILOVER.INI 파일 내의 필드 설명

필드 이름	설명
IpAddress	Failover 세트에서 사용할 TCP/IP 주소
PortNumber	큐 관리자의 리스너에서 사용할 TCP/IP 포트 번호
TimeOut	이 시간 종료 값은 EndCommand 프로시저에 전달됩니다. 241 페이지의 『Failover 세트에서 사용되는 명령 프로시저』 부분을 참조하십시오.
StartCommand	큐 관리자를 시작하는 데 사용되는 명령 프로시저
EndCommand	큐 관리자를 종료하는 데 사용되는 명령 프로시저
TidyCommand	OpenVMS 노드가 유지되는 큐 관리자가 실패한 후 노드에서 정리하기 사용되는 명령
LogDirectory	StartCommand, EndCommand 및 TidyCommand 프로시저에 의해 작성되는 로그 파일이 있는 디렉토리
NodeCount	Failover 세트의 노드 수. 이 필드 다음에 정의되는 노드 세 쌍의 수가 이 값과 일치해야 합니다. 지원되는 최대 노드 수는 4입니다.
NodeName	노드의 노드 이름. SCSNODE OpenVMS 시스템 매개변수에 지정된 값입니다.
Interface	OpenVMS용 Digital TCP/IP 서비스 TCP/IP 스택을 사용할 때 노드에 대한 TCP/IP 인터페이스 이름. 이것은 \$tcpip 인터페이스 표시 명령의 출력에서 확보할 수 있습니다. OpenVMS용 TCPware TCP/IP 또는 OpenVMS용 MultiNet TCP/IP 스택을 사용할 경우에는 이 필드가 사용되지 않지만, 디폴트 값 we0를 계속 지정해야 합니다(구성 파일에서 필드를 제거하지 마십시오).

표 9. FAILOVER.INI 파일 내의 필드 설명 (계속)

필드 이름	설명
Priority	Failover 세트 내에서 이 노드에 제공되는 우선순위입니다. 값은 1 - 10 사이여야 합니다. 값이 1이면 최상위 우선순위입니다. 여러 노드의 우선순위가 동일할 수 있습니다. 실패하거나 failover -s 또는 -f 명령에 지정된 특정 노드가 없으면, 큐 관리자는 사용 가능한 최상위 우선순위 노드에서 시작됩니다.

Failover 세트에서 사용되는 명령 프로시저

Failover 세트는 세 가지 명령 프로시저를 사용하여 해당 기능 중 일부를 구현합니다. 이 명령 프로시저의 위치는 FAILOVER.INI 구성 파일에서 필드 이름 StartCommand, EndCommand 및 TidyCommand로 지정되어 있습니다. 이 명령 프로시저에 대한 템플릿 파일(**START_QM.TEMPLATE**, **END_QM.TEMPLATE** 및 **TIDY_QM.TEMPLATE**)이 MQS_EXAMPLES에 제공됩니다. 이 파일들은 355 페이지의 『부록F. OpenVMS 클러스터 Failover 세트 템플릿』에 나열되어 있습니다.

명령 프로시저는 5개 또는 6개의 매개변수를 전달합니다. 이 매개변수는 표10에 나열되어 있습니다.

표 10. 명령 프로시저에 전달된 매개변수

매개변수	값
P1	큐 관리자 이름
P2	큐 관리자 디렉토리 이름
P3	클러스터 TCP/IP 주소
P4	노드 인터페이스 이름
P5	리스너 포트 번호
P6	큐 관리자 종료 시간 종료(EndCommand 프로시저의 경우)

StartCommand 프로시저는 다음 상황에서 큐 관리자를 시작하기 위해 사용됩니다.

- **failover** 명령의 **-s** 플래그와 함께 명시적으로 지정될 경우
- 큐 관리자가 **failover** 명령의 **-f** 플래그를 사용하여 다른 OpenVMS 노드로 이동할 경우
- 큐 관리자 실패 후 자동으로 재시작할 경우

디폴트로, StartCommand 프로시저는 노드에서 Failover 세트 TCP/IP 주소를 구성하여 큐 관리자를 실행한 후 **strmqm -m queuemanagename** 명령을 사용하여 큐 관리자를 시작합니다. 시스템 요구사항에 따라 다음 방법으로 명령 프로시저를 수정할 수 있습니다.

- **strmqm** 명령을 변경하십시오.
- 리스너 등의 추가 MQSeries 프로세스를 시작하는 명령을 추가하십시오.
- 응용프로그램 프로세스를 시작하는 명령을 추가하십시오.

StartCommand 프로시저는 큐 관리자가 시작된 후 모니터링 큐 관리자에 대해 1의 상태로 종료해야 합니다.

EndCommand 프로시저는 다음 상황에서 큐 관리를 종료하기 위해 사용됩니다.

- **failover** 명령의 -e 플래그로 명시적으로 지정될 경우
- 큐 관리자가 **failover** 명령의 -f 플래그를 사용하여 다른 OpenVMS 노드로 이동할 경우

디폴트로, EndCommand 프로시저는 **endmqm -i queuemanagename** 명령으로 큐 관리를 종료하려고 합니다. 구성 파일에 지정된 시간 종료 기간 내에 큐 관리자가 종료되지 않으면, 프로시저는 **endmqm -p queuemanagename** 명령으로 큐 관리를 종료하려고 합니다. 큐가 다른 시간 종료 기간 내에도 계속 종료되지 않으면, 큐 관리자는 실행 제어기 프로세스를 삭제하여 종료됩니다. 큐 관리자가 종료되면 Failover 세트 TCP/IP 주소 구성이 해제됩니다. 큐 관리자가 **endmqm** 명령을 사용하여 성공적으로 종료될 경우, SS\$_NORMAL 상태가 리턴됩니다. 큐 관리자가 실행 제어기 삭제로 종료될 경우, SS\$_ABORT 상태가 리턴됩니다. 큐 관리자가 세 번째 시간 종료 기간 후에 종료되지 않으면 SS\$_TIMEOUT 상태가 리턴됩니다. 이러한 상태는 EndCommand 프로시저의 결과를 판별하기 위해 감시자 Failover 모니터에서 사용되며 각각 Failover 세트의 상태를 설정합니다. 시스템 요구사항에 따라 다음 방법으로 명령 프로시저를 수정할 수 있습니다.

- 리스너 등의 추가 MQSeries 프로세스를 종료하는 명령을 추가하십시오.
- 응용프로그램 프로세스를 종료하는 명령을 추가하십시오.

TidyCommand 프로시저는 큐 관리자가 실패하였지만 OpenVMS 노드가 계속 실행될 경우 OpenVMS 노드에서 정리하는데 사용됩니다.

디폴트로, TidyCommand 프로시저는 Failover 세트 TCP/IP 주소의 구성을 해제합니다. 시스템 요구사항에 따라 다음 방법으로 명령 프로시저를 수정할 수 있습니다.

- 리스너와 같이 계속 실행 중인 MQSeries 프로세스를 종료하는 명령을 추가하십시오.
- 계속 실행 중인 응용프로그램 프로세스를 종료하는 명령을 추가하십시오.

디폴트로, 템플릿 파일은 OpenVMS 명령에 대해 Digital TCP/IP 서비스를 사용하여 TCP/IP 주소를 구성하거나 구성 해제할 수 있도록 설정됩니다. OpenVMS용 TCPware 또는 OpenVMS용 MultiNet를 사용 중이면, OpenVMS용 Digital TCP/IP 서비스 명령을 주석으로 처리하고(비활성화) OpenVMS용 TCPware 또는 OpenVMS용 MultiNet 명령의 주석 처리를 해제하십시오(활성화).

Failover 세트 관리

Failover 세트는 SYSTEM 계정이나 MQSeries 관리 계정으로 관리해야 합니다. **DCL runmqfm** 및 **failover**의 두 가지 명령을 사용하여 Failover 세트를 관리합니다.

runmqfm 명령은 **failover** 모니터를 시작하는 데 사용되고 **failover** 명령은 다른 모든 관리 작업을 수행합니다. 이 명령은 313 페이지의 『runmqfm(Failover 모니터 시작)』 및 296 페이지의 『failover(Failover 세트 관리)』에 설명되어 있습니다.

Failover 모니터 시동

Failover 모니터는 Failover 모니터를 시작하는 데 필요한 OpenVMS 노드에서 **runmqfm** 명령을 실행하여 시작됩니다. 예를 들어, 큐 관리자 TESTQM에 대해 Failover 모니터를 시작하려면 다음 명령을 사용하십시오.

```
$ runmqfm -m TESTQM
```

이것은 큐 관리자 이름을 기반으로 하며 이름이 _FM으로 끝나는 별도의 프로세스를 작성합니다. 이 예에서 프로세스 이름은 TESTQM_FM입니다. 이 프로세스는 monmq 활성 표시에 나열됩니다.

로그 파일이 필요할 경우, **runmqfm** 명령 출력의 경로를 재지정하여 로그 파일을 지정할 수 있으며 추가 디버그 정보는 -d 플래그를 지정하여 로그 파일에 표시할 수 있습니다. 예를 들면 다음과 같습니다.

```
$ runmqm -m TESTQM -d > sys$manager:fm.log
```

runmqfm 명령은 큐 관리자를 시작하지 않는 Failover 모니터 프로세스만 시작한다는 점에 유의하십시오.

Failover 세트 내에서 큐 관리자 시작

Failover 세트 내에서 큐 관리자를 시작하려면, 최소한 하나의 Failover 모니터가 실행 중이어야 하고, 큐 관리자를 시작할 노드에서 Failover 모니터가 실행되고 있어야 합니다. 큐 관리자는 **failover** 명령의 -s 플래그를 사용하여 시작됩니다. 이 명령은 Failover 세트 내의 OpenVMS 노드에서 실행될 수 있습니다. 예를 들어 큐 관리자 TESTQM을 BATMAN 노드에서 시작하려면, 다음 명령을 사용하십시오.

```
$ failover -m TESTQM -n BATMAN -s
```

사용 가능한 최상위 우선순위를 가진 OpenVMS 노드에서 큐 관리자를 시작해야 할 경우, 명령에서 -n 플래그를 생략하십시오. 예를 들면 다음과 같습니다.

Failover 모니터 시작

```
$ failover -m TESTQM -s
```

임의 노드에서 큐 관리자에 대해 Failover 모니터가 시작되었으면, **strmqm** 명령을 사용하여 큐 관리자를 시작할 수 없습니다. 그러나 모든 Failover 모니터가 큐 관리자에 대해 정지되었으면, 정상적으로 **strmqm** 명령을 사용할 수 있습니다.

Failover 세트 내에서 큐 관리자 종료

Failover 세트 내에서 큐 관리자를 종료하려면, 큐 관리자를 실행하는 노드에서 Failover 모니터가 실행되고 있어야 합니다. 큐 관리자는 **failover** 명령의 **-e** 플래그를 사용하여 종료됩니다. 이 명령은 Failover 세트 내의 OpenVMS 노드에서 실행될 수 있습니다. 예를 들어, 큐 관리자 TESTQM을 종료하려면 다음 명령을 사용하십시오.

```
$ failover -m TESTQM -e
```

임의 노드에서 큐 관리자에 대해 Failover 모니터가 시작되었으면, **endmqm** 명령을 사용하여 큐 관리자를 종료할 수 없습니다. 그러나 모든 Failover 모니터가 큐 관리자에 대해 정지되었으면, 정상적으로 **endmqm** 명령을 사용할 수 있습니다.

Failover 세트 내에서 큐 관리자 이동

Failover 세트 내에서 큐 관리자를 이동한다는 것은 현재 실행 중인 노드에서 큐 관리자를 정지하고 Failover 세트 내의 다른 노드에서 다시 큐 관리자를 시작한다는 의미입니다. Failover 세트 내에서 큐 관리자를 이동하려면, 현재 큐 관리자를 실행하는 노드와 큐 관리자를 이동하려는 노드에서 Failover 모니터가 실행 중이어야 합니다.

큐 관리자는 **failover** 명령의 **-f** 플래그를 사용하여 이동됩니다. 이 명령은 Failover 세트 내의 OpenVMS 노드에서 실행될 수 있습니다. 예를 들어 큐 관리자 TESTQM을 ROBIN 노드로 이동하려면 다음 명령을 사용하십시오.

```
$ failover -m TESTQM -n ROBIN -f
```

사용 가능한 최상위 우선순위를 가진 OpenVMS 노드로 큐 관리자를 이동해야 할 경우, 명령에서 **-n** 플래그를 생략하십시오. 예를 들면 다음과 같습니다.

```
$ failover -m TESTQM -f
```

Failover 세트 상태 표시

Failover 세트의 모든 상태를 설명하는 세 가지 유형의 상태가 있습니다.

- Failover 세트 큐 관리자 상태
- Failover 세트 노드 큐 관리자 상태(각 노드마다 하나씩)
- Failover 세트 노드 모니터 상태(각 노드마다 하나씩)

각 상태에 대해 가능한 값은 다음 세 개의 표에 설명되어 있습니다.

표 11. Failover 세트 큐 관리자 상태

상태	설명
STOPPED	큐 관리자가 Failover 세트에서 시작되었거나 완전하게 종료되었습니다.
STARTED	큐 관리자가 Failover 세트에서 시작되었습니다. Failover 세트는 큐 관리자가 실패할 경우 큐 관리자를 재시작하려고 합니다.

표 12. Failover 세트 노드 큐 관리자 상태

상태	설명
AVAILABLE	다른 노드에서 실패할 경우 큐 관리자가 이 노드에서 시작될 수 있도록 노드가 비워져 있습니다.
RUNNING	이 노드에서 큐 관리자가 실행 중입니다.
EXCLUDED	노드 자체가 실패하지 않고 불확실한 방식으로 노드에서 큐 관리자가 정지되었습니다. 다른 노드에서 큐 관리자가 실패한 경우, 이 노드에서 재시작되지 않습니다.

표 13. Failover 세트 노드 모니터 상태

상태	설명
STARTED	Failover 모니터가 이 노드에서 실행되고 있지만 감시자가 아닙니다.
WATCHING	Failover 모니터가 이 노드에서 실행되고 있으며 감시자입니다.
STOPPED	이 노드에서 실행 중인 Failover 모니터가 없습니다.

Failover 세트의 상태는 **failover** 명령의 -q 상태를 사용하여 표시됩니다. 최소한 하나의 Failover 모니터 프로세스가 실행되고 있어야 하고 명령은 Failover 세트 내의 임의의 노드에서 실행될 수 있습니다. 예를 들어, 큐 관리자 TESTQM에 대한 Failover 세트의 상태를 표시하려면 다음 명령을 사용하십시오.

```
$ failover -m TESTQM -q
```

명령의 샘플 출력은 다음과 같습니다.

DCL 기호 설정

```

83H8439, 5697-270 (C) Copyright IBM Corp. 1996. ALL RIGHTS RESERVED.
OpenVMS Cluster Failover Set - Configuration and State.
Queue Manager Name           : TESTQM
Sequence No                   : 11
TCP/IP Address                : 10.20.30.40
Listener Port Number         : 1414
Timeout to end the Queue Manager : 30
Queue Manager state in Failover Set : STARTED
OpenVMS Node - Configuration and State
Node name                     : BATMAN
Priority                       : 2
TCP/IP Interface              : we0
Queue Manager state           : RUNNING
Failover Monitor state        : WATCHING
Node name                     : ROBIN
Priority                       : 1
TCP/IP Interface              : we0
Queue Manager state           : EXCLUDED
Failover Monitor state        : STARTED
    
```

DCL 기호를 Failover 세트 상태로 설정

어떤 경우에는 DCL 명령 프로시저를 작성하여 Failover 세트를 제어해야 할 수도 있습니다. **failover** 명령의 -l 플래그는 Failover 세트의 상태를 나타내기 위해 세 개의 로컬 DCL 기호를 설정합니다. 이 기호들은 큐 관리자 상태를 기초로 조건부 조치를 취하는데 사용할 수 있습니다. 최소한 하나의 Failover 모니터 프로세스가 실행되고 있어야 하고 명령은 Failover 세트 내의 임의 노드에서 실행될 수 있습니다. 설정된 기호들은 표14에 나와 있습니다.

표14. DCL 기호 및 설명

DCL 기호 이름	설명
MQ\$QMGR_NODE	큐 관리자를 실행 중이면 OpenVMS 노드로 설정하고, 실행 중인 큐 관리자가 없으면 널(null) 문자열로 설정하십시오.
MQ\$AVAILABLE_NODES	큐 관리자를 실행하는 데 사용할 수 있는 OpenVMS 노드 목록으로 설정하십시오. 이는 큐 관리자가 AVAILABLE 상태이고 Failover 모니터가 실행 중인 노드들입니다.
MQ\$MONITOR_NODES	Failover 모니터가 실행 중인 OpenVMS 노드의 목록으로 설정하십시오.

예를 들어, 기호를 큐 관리자 TESTQM에 대한 Failover 세트의 상태로 설정하려면 다음 명령을 사용하십시오.

```
$ failover -m TESTQM -l
```


기호 설정 결과의 예는 다음과 같습니다.

```
MQS$AVAILABLE_NODES = ""
MQS$MONITOR_NODES = "BATMAN,ROBIN"
MQS$QMGR_NODE = "BATMAN"
```

Failover 모니터 프로세스 정지

OpenVMS 노드에서의 Failover 모니터 프로세스는 **failover** 명령의 **-h** 플래그를 사용하여 정지될 수 있습니다. 이 명령은 Failover 세트 내의 노드에서 실행될 수 있습니다. 예를 들어 노드 BATMAN에서 큐 관리자 TESTQM의 Failover 모니터를 정지하려면, 다음 명령을 사용하십시오.

```
$ failover -m TESTQM -n BATMAN -h
```

정지되는 Failover 모니터가 감시자 Failover 모니터일 경우, 다른 Failover 모니터가 감시자로 설정됩니다(있을 경우). 정지되는 Failover 모니터가 Failover 세트의 마지막 Failover 모니터일 경우, Failover 세트는 더이상 유지되지 않습니다. 이런 경우, 큐 관리자는 **strmqm** 및 **endmqm** 명령을 사용하여 시작하거나 종료할 수 있습니다. **failover** 명령의 **-h** 플래그는 큐 관리자를 종료하지 않습니다. Failover 모니터가 정지되는 OpenVMS 노드에서 큐 관리자가 실행 중이면, 큐 관리자는 계속 실행됩니다.

갱신이 진행되는 동안 명령 실행

플래그가 **-s**, **-e**, **-f** 및 **-c**인 **failover** 명령은 갱신으로 간주됩니다. 이 명령들이 진행되는 동안, 감시자 Failover 모니터에서 갱신 진행 플래그가 설정됩니다. 이 플래그가 설정되면 동시 갱신이 허용되지 않으므로 다른 갱신 및 Failover 모니터 정지 명령이 실패하게 됩니다. **-q** 및 **-l** 플래그와 같은 갱신 이외의 명령은 갱신이 진행되는 동안 계속 작동합니다.

간혹 갱신이 실패하여 갱신 진행 플래그가 설정된 상태로 남아 있을 수도 있습니다. **failover** 명령의 **-u** 플래그는 갱신 진행 플래그를 지웁니다. 이 명령은 주의하여 사용해야 합니다. 예를 들어, 큐 관리자 TESTQM에 대해 갱신 진행 플래그를 지우려면 다음 명령을 사용하십시오.

```
$ failover -m TESTQM -u
```

Failover 세트 상태 변경

어떤 상황에서는 Failover 세트의 상태를 변경해야 합니다. 이것은 **failover** 명령의 **-c** 플래그를 사용하여 보존됩니다. 대부분 실패 후 노드에서의 큐 관리자가 EXCLUDED

Failover 세트 상태 변경

상태이고 노드 클린업 후에 상태를 다시 AVAILABLE로 변경하려고 할 경우에 필요합니다. 예를 들어, 노드 BATMAN에서 큐 관리자 TESTQM에 대해 AVAILABLE 상태로 변경하려면 다음 명령을 사용하십시오.

```
$ failover -m TESTQM -n BATMAN -c -qmgr available
```

또한 임시로 노드 큐 관리자 상태를 AVAILABLE에서 EXCLUDED로 변경하여 큐 관리자를 실행하기 위한 후보로 간주되는 노드를 제외할 수 있습니다. 예를 들어, 노드 BATMAN에서 큐 관리자 TESTQM의 상태를 EXCLUDED로 변경하려면 다음 명령을 사용하십시오.

```
$ failover -m TESTQM - n BATMAN -c -qmgr excluded
```

다른 모든 상태를 변경할 수는 있지만 요청된 변경사항이 실행 중인 시스템에서 일관성이 있을 경우에만 변경사항이 적용됩니다. 예를 들어 Failover 모니터가 노드에서 실행 중인데 모니터 상태를 STOPPED로 변경하려고 하면, 이 변경은 적용되지 않습니다. EXCLUDED와 AVAILABLE 사이의 노드 큐 관리자 상태 변경과는 달리, 30초마다 감시자 Failover 모니터가 무결성 점검을 수행하고 실행 중인 시스템에 일관성이 없을 경우에는 상태를 변경하므로 상태 변경 명령을 사용할 필요가 없습니다.

ICC 연관에 대한 보안 설정

Failover 세트 모니터 및 클라이언트 프로그램은 OpenVMS ICC(Intra Cluster Communication) 호출을 사용하여 메시지를 전달합니다. 권한이 없는 사용자가 메시지를 Failover 모니터 프로세스로 송신하지 못하게 하려면, ICC 연관에 대한 보안이 **SYSS\$STARTUP:ICC\$SYSTARTUP.COM** 명령 프로시저에 구성되어 있어야 합니다.

각 Failover 세트는 두 개의 연관 이름을 사용합니다. 하나는 감시자 Failover 모니터와의 통신에 사용되는 큐 관리자의 이름에 대한 연관 이름이고, 다른 하나는 각 Failover 모니터와의 통신에 사용되는 것으로 **_MQ_FM**이 추가된 큐 관리자 이름과 연관된 이름입니다.

예는 Failover 세트의 각 노드에 대해 **ICC\$SYSTARTUP.COM**에서 요구되는 항목의 249 페이지의 그림22에 표시되어 있습니다. Failover 세트에 BATMAN 및 ROBIN이라고 하는 두 개의 노드가 있고 큐 관리자 이름은 TESTQM입니다.

```

$! ----- List Nodes with Special Actions -----
$!
$ nodeactions = "/BATMAN/ROBIN/"
$ if f$locate("/"+nodename+"/",nodeactions) .eq. f$length(nodeactions) -
then goto exit ! No action for this node
$ goto 'nodename' ! Go to action code for this node
$!
$! ----- Major Nodes -----
$BATMAN:
$ROBIN:
$!
$! Place in here calls to @SYSS$MANAGER:ICC$CREATE_SECURITY_OBJECT and
$! @SYSS$MANAGER:ICC$ADD_REGISTRY_TABLE that apply to FAilover nodes in the
$! cluster
$!
$!
$ @SYSS$MANAGER:ICC$CREATE_SECURITY_OBJECT ICC$::"TESTQM" -
"/owner=MQM/acl=((id=MQM,aaccess=open+access),(id=*,access=none))"
$!
$ @SYSS$MANAGER:ICC$CREATE_SECURITY_OBJECT 'nodename'::"TESTQM" -
"/owner=MQM/acl=((id=MQM,access=open+access),(id=*,access=none))"
$!
$ @SYSS$MANAGER:ICC$CREATE_SECURITY_OBJECT 'nodename'::"TESTQM_MQ_FM" -
"/owner=MQM/acl=((id=MQM,access=open+access),(id=*,access=none))"
$!
$ set security/class=logical_name_table icc$registry_table -
/acl=(id=MQM,access=read+write)
$!
$ GOTO EXIT
$!

```

그림 22. ICC\$SYSTARTUP.COM에 필요한 샘플 항목

ICC 연관 이름은 31자로 제한되므로, Failover 세트에서 사용될 경우 MQSeries 큐 관리자 이름으로 지원되는 최대 길이는 25자입니다. ICC 연관 보안 설정에 대한 추가 정보는 *OpenVMS System Manager's Manual*에서 볼 수 있습니다.

Failover 세트의 문제점 해결

start_qm.com, **end_qm.com** 및 **tidy_qm.com** 프로시저가 실행될 때, 로그 파일은 failover.ini 구성 파일에 지정된 LogDirectory에 기록됩니다. 로그 파일의 이름은 *qmgrname_procedurename.log*입니다. 예를 들어, 큐 관리자 이름이 TESTQM인 경우 **start_qm.com** 명령 프로시저는 testqm_start_qm.log라는 이름의 로그 파일을 생성합니다.

디폴트로, Failover 모니터는 로그 파일을 생성하지 않지만, 로그 파일은 **runmqfm** 명령의 경로 재지정 매개변수를 사용하여 지정할 수 있습니다. 추가 디버그 정보는 **runmqfm** 명령에 -d 매개변수를 지정하여 파일에 기록될 수 있습니다.

MQS_ROOT: [MQM.ERRORS]에서 FDC 파일이 생성되었는지 점검하십시오.

Failover 세트와 함께 OpenVMS용 MultiNet 사용

Failover 세트와 함께 OpenVMS용 MultiNet을 사용하려면, 클러스터 별명 서비스가 사용 가능해야 합니다. 클러스터 별명 서비스는 다음 명령으로 사용할 수 있습니다.

```
$ MULTINET CONFIGURE/SERVERS
SERVER-CONFIG> ENABLE CLUSTERALIAS
SERVER-CONFIG> EXIT
```

템플릿 명령 파일에는 단 하나의 클러스터 별명 주소가 있으며, Failover 세트에서 사용된다고 가정합니다. 그러나 사용되는 다른 클러스터 별명 주소가 있으면, 명령 프로시저는 다른 주소가 MULTINET_CLUSTER_IP_ALIASES 논리 이름에 남아 있도록 수정해야 합니다.

Failover 세트 사용 예

다음은 OpenVMS 클러스터에서 두 개의 노드 BATMAN과 ROBIN을 큐 관리자 TESTQM에 대한 Failover 세트로 구성하는 예입니다. Failover 세트 TCP/IP 주소는 10.20.30.40이고 TCP/IP 리스너 포트는 1414입니다. 노드 BATMAN가 1차 노드로 지정되고 ROBIN이 2차 노드로 지정됩니다. 초기에 큐 관리자는 BATMAN에서 시작되고 큐 관리자가 실패하면 ROBIN에서 재시작됩니다. 큐 관리자가 ROBIN에서 실행되면, 그 큐 관리자는 ROBIN이 재부팅될 때 BATMAN으로 돌아가는 데 실패하지 않습니다. 큐 관리자를 실행 중인 노드가 종료할 경우, 큐 관리자는 종료되고 Failover 모니터는 정지됩니다. 노드가 실행되고 있지 않으면, 큐 관리자는 종료하고 Failover 모니터만 정지됩니다.

failover.template 사용자 정의

failover.template 파일은 다음과 같이 수정되어 mqs_root:[mqm.qmgrs.testqm]failover.ini로 복사됩니다.

```

# FAILOVER.TEMPLATE
# Template for creating a FAILOVER.INI configuration file
# All lines beginning with a '#' are treated as comments
#
# OpenVMS Cluster Failover Set Configuration information
# -----
#
# The TCP/IP address used by the OpenVMS Cluster Failover Set
#
IpAddress=10.20.30.40
#
# The TCP/IP port number used by the MQSeries Queue Manager
#
PortNumber=1414
#
# The timeout used by the EndCommand command procedure
#
TimeOut=30
#
# The command procedure used to start the Queue Manager
#
StartCommand=@sys$manager:start_qm
#
# The command procedure used to end the Queue Manager
#
EndCommand=@sys$manager:end_qm
#
# The command procedure used to tidy up on a node after a
# Queue Manager failure but the OpenVMS node did not fail
#
TidyCommand=@sys$manager:tidy_qm
#
# The directory in which the log files for the start, end and
# tidy commands are written
#
LogDirectory=mqs_root:[mqm.errors]
#
# The number of nodes in the OpenVMS Cluster Failover Set. The
# number of nodes defined below must agree with this number
#
NodeCount=2
#
# The Name of the OpenVMS node
#
NodeName=BATMAN
#
# The TCP/IP interface name for the node
#Interface=we0
#
# The priority of the node
#
Priority=1
#
# The Name of the OpenVMS node
#
NodeName=ROBIN
#
# The TCP/IP interface name for the node
#
Interface=we0
#
# The priority of the node
#
Priority=2

```

그림 23. FAILOVER.INI 구성 파일을 작성하기 위한 failover.template

Failover 세트 명령 프로시저의 수정사항

명령 프로시저는 템플릿 파일에서 복사됩니다. 유일한 수정사항은 **start_qm.com**에서의 리스너 시작과 **end_qm.com**에서의 리스너 종료의 주석 처리가 해제(활성화)된다는 것입니다. 정지한 후 시작할 응용프로그램이 있으면, 적절한 명령이 명령 프로시저에 추가될 수 있습니다.

Failover 세트 시작 명령 프로시저 **start_failover_set.com**의 예

start_failover_set.com 명령 프로시저는 각 노드에서 Failover 모니터를 시작하고 조 건부로 큐 관리자를 시작하기 위해 사용됩니다. 프로시저는 **MQS_STARTUP.COM**

Failover 세트 예

명령 프로시저가 실행된 후에 시스템 시동시 호출됩니다. 프로시저는 두 개의 매개변수인 큐 관리자 이름과 1차 노드 이름을 전달합니다. 이 경우, 다음과 같이 호출됩니다.

```
$@start_failover_set testqm batman
```

start_failover_set.com 명령 프로시저는 Failover 모니터를 시작한 후 **failover** 명령에서 -l 매개변수를 사용하여 Failover 세트의 상태를 찾습니다. Failover 모니터가 **failover** 명령이 실행될 때 완전하게 시작되지 않기 때문에 명령은 매번 3회씩 재시도됩니다. 이 때 노드가 1차 노드이고 큐 관리자가 시작되지 않은 경우, **failover** 명령의 -s 매개변수를 사용하여 시작됩니다.

```

$on error then exit
$@sy$manager:mqs_symbols
$!
$! start_failover_set.com
$! -----
$! Command procedure to start a Failover Set Queue Manager during startup
$!
$! p1 = Queue Manager name
$! p2 = Primary Node name
$!
$! Check that the Queue Manager has been specified
$!
$if p1 .eqs ""
$then
$ Write sys$output "Queue Manager name omitted"
$ exit
$else
$ qmgr_name = p1
$endif
$!
$! Check that the primary node name has been specified
$!
$if p2 .eqs ""
$then
$ Write sys$output "Node name omitted"
$ exit
$else
$ primary_node = p2
$endif
$!
$! Get the node name of this node
$!
$this_node=f$getsyi("nodename")
$!
$! Start the Failover Monitor on this node
$!
$runmqfm -m 'qmgr_name'
$!
$! Check that the Failover Monitor has fully started
$! Wait up to 3 seconds
$!
$count = 0
$check_start:
$on error then continue
$!
$! Set the MQS$* symbols to the state of Failover Set
$! Wait up to 3 seconds
$!

```

그림 24. *start_failover_set* 명령 프로시저 (1/2)

Failover 세트 예

```
$failover -m 'qmgr_name' -l
$!
$! If an error is returned wait a second and try again
$!
$if ( ($status/8) .and %xffff ) .ne. 0 then goto wait
$!
$! If this node is not listed as running a monitor wait a second and try again
$!
$if f$locate( this_node, mqs$monitor_nodes ) .ne. f$length( mqs$monitor_nodes )
$then
$ goto start_qm
$endif
$wait:
$on error then exit
$count = count + 1
$!
$! If we have waited 3 seconds display an error and exit
$!
$if count .ge. 3
$then
$ write sys$output "Failover Monitor not started"
$ exit
$else
$ wait 00:00:01
$ goto check_start
$endif
$start_qm:
$!
$! Only start the Queue Manager on the primary node
$!
$if this_node .nes. primary_node
$then
$ write sys$output "Queue Manager not started on Secondary node"
$ exit
$endif
$!
$! Start the Queue Manager on the primary node if it is not already running.
$!
$if mqs$qmgr_node .eqs. ""
$then
$ failover -m 'qmgr_name' -n 'this_node' -s
$else
write sys$output "Queue Manager already started"
$endif
$exit
```

그림 24. *start_failover_set* 명령 프로시저 (2/2)

Failover 세트 종료 명령 프로시저 *end_failover_set.com*의 예

end_failover_set.com 명령 프로시저는 조건부로 큐 관리자를 종료한 후 각 노드에서 Failover 모니터를 종료하는 데 사용됩니다. **MQS_SHUTDOWN.COM** 명령 프로시저가 실행되기 전에 사이트 특정 종료에서 프로시저가 호출됩니다. 프로시저는 하나의 매개변수인 큐 관리자 이름을 전달합니다. 이 경우, 다음과 같이 호출됩니다.

```
$@start_failover_set testqm
```


end_failover_set.com 명령 프로시저는 **failover** 명령의 -l 매개변수를 사용하여 Failover 세트 상태를 확보합니다. 이때, 큐 관리자가 이 노드에서 실행되고 있으면 종료됩니다. 그리고 나서 Failover 모니터도 정지됩니다.

```

on error then exit
@sys$manager:mqs_symbols
$!
$! end_failover_set.com
$! -----
$! Command procedure to end a Failover Set Queue Manager during shutdown
$!
$! p1 = Queue Manager name
$!
$! Check that the Queue Manager has been specified
$!
$if p1 .eqs ""
$then
$ write sys$output "Queue Manager name omitted"
$ exit
$else
$ qmgr_name = p1
$endif
$!
$! Get the node name of this node
$!
$this_node=f$getsysi("nodename")
$!
$! Set the MQS$* symbols to the state of the Failover Set
$!
$failover -m 'qmgr_name' -l
$!
$! If an error then exit
$!
$if ( ($status/8) .and %xffff ) .ne. 0
$then
$ write sys$output "Error querying Failover Set"
$ exit
$endif
$!
$! If the Queue Manager is not running on this node then exit
$!
$ if mqs$qmgr_node .nes. this_node
$then
$ write sys$output "Queue Manager not running on this node"
$ goto halt_fm
$endif
$!
$! End the Queue Manager
$!
$failover -m gjtest -e
$halt_fm:
$!
$! Halt the Failover Monitor
$!
$failover -m gjtest -n 'this_node' -h

```

그림 25. *end_failover_set* 명령 프로시저

제2부 참조사항

제17장 MQSeries 제어 명령	259
MQSeries 오브젝트 이름 지정 규칙.	259
오브젝트 파일 보기	260
구문 다이어그램 읽는 방법.	260
구문 도움말.	261
예	262
MQSeries 리턴 코드.	263
crtmqcvx(데이터 변환)	264
crtmqm(큐 관리자 작성)	266
dltmqm(큐 관리자 삭제)	270
dmpmqlog(로그 덤프)	273
dspmqaout(권한 표시).	275
dspmqcsv(명령 서버 표시).	279
dspmqls(MQSeries 파일 표시)	281
dspmqrtrc(MQSeries 형식화 추적 출력 표시).	283
dspmqrtrn(MQSeries 트랜잭션 표시).	285
endmqcsv(명령 서버 종료).	287
endmqlsr(리스너 종료)	290
endmqm(큐 관리자 종료)	291
endmqtrc(MQSeries 추적 종료)	295
failover(Failover 세트 관리)	296
rcdmqimg(매체 이미지 기록)	300
rcrmqobj(오브젝트 재작성).	303
rsvmqtrn(MQSeries 트랜잭션 해석).	307
runmqchi(채널 시작기 실행)	309
runmqchl(채널 실행).	310
runmqdlq(데드-레터 큐 핸들러 실행)	311
runmqfm(Failover 모니터 시작)	313
runmqlsr(리스너 실행)	314
runmqsc(MQSeries 명령 실행)	316
runmqtmc(클라이언트 트리거 모니터 시작)	319
runmqtrm(트리거 모니터 시작)	320
setmqaut(권한 설정/재설정).	321
strmqcsv(명령 서버 시작)	329
strmqm(큐 관리자 시작)	330
strmqtrc(MQSeries 추적 시작)	332

제17장 MQSeries 제어 명령

이 장에는 Compaq OpenVMS용 MQSeries에서 사용되는 제어 명령에 대한 참조 자료가 있습니다. 이 장의 모든 명령은 OpenVMS DCL 프롬프트에서 발행할 수 있습니다.

명령 이름과 해당되는 플래그는 대소문자가 구별되지 않으므로, 대문자, 소문자 또는 대소문자를 혼합하여 입력할 수 있습니다. 그러나, 명령을 제어하기 위한 매개변수(큐 이름과 같은)에서는 대소문자가 구별됩니다. 자세한 정보는 27 페이지의 『제어 명령의 대소문자 구분』을 참조하십시오.

제어 명령을 사용하려면, 먼저 최종 재시동 이후로 `mqs_startup`이 한번 실행되어야 합니다.

MQSeries 오브젝트 이름 지정 규칙

일반적으로, MQSeries 오브젝트의 이름은 최대 48자까지 가능합니다. 이 규칙은 다음의 모든 오브젝트에 적용됩니다.

- 큐 관리자(그러나 OpenVMS 클러스터 Failover 세트에서 큐 관리자를 지원할 경우, 최대 길이는 25자입니다.)
- 큐
- 프로세스 정의
- 이름 목록
- 클러스터

채널 이름의 최대 길이는 20자입니다.

모든 MQSeries 이름에 대해 사용될 수 있는 문자는 다음과 같습니다.

- 대문자 A-Z
- 소문자 a-z
- 숫자 0-9
- 마침표(.)
- 밑줄(_)
- 정방향 슬래시(/) (참고 1 참조)
- 퍼센트 부호(%) (참고 1 참조)

주:

1. 정방향 슬래시 및 퍼센트는 특수 문자입니다. 그러나 이름의 첫번째 문자로 이 문자들을 사용할 수는 없습니다. 이 문자들 중 하나를 이름에 사용할 경우에는 그 이름은 항상 큰 따옴표로 묶어야 합니다.

이름

2. 선두 문자 또는 임베드된 공백은 허용되지 않습니다.
3. 자국어 문자는 허용되지 않습니다.
4. 이름을 큰 따옴표로 묶을 수 있는데, 이것은 특수 문자가 이름에 포함되어 있거나 대소문자가 보존되어야 할 경우에만 필수적입니다.

오브젝트 파일 보기

각 MQSeries 큐, 큐 관리자 또는 프로세스 오브젝트는 파일에 의해 표시됩니다. 이러한 오브젝트 이름이 올바르지 않은 경우도 있으므로, 필요하면 큐 관리자가 오브젝트 이름을 올바른 파일 이름으로 변환합니다. 25 페이지의 『MQSeries 파일 이름 이해』의 설명을 참조하십시오.

오브젝트의 실제 파일 이름을 표시하는 방법을 찾으려면 281 페이지의 『dspmqfls(MQSeries 파일 표시)』를 참조하십시오.

구문 다이어그램 읽는 방법

이 장에는 구문 다이어그램(『레일로드』 다이어그램이라고도 함)이 있습니다.

각 구문 다이어그램은 이중 오른쪽 화살표로 시작하여 오른쪽 및 왼쪽 화살표 창으로 끝납니다. 하나의 오른쪽 화살표로 시작하는 줄들은 연속 행을 표시하는 줄입니다. 구문 다이어그램은 왼쪽에서 오른쪽으로, 위에서 아래로 화살표의 방향을 따라 읽으면 됩니다.

구문 다이어그램에서 사용되는 다른 규칙들은 다음과 같습니다.

표 15. 구문 다이어그램을 읽는 방법

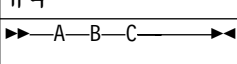
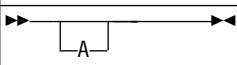
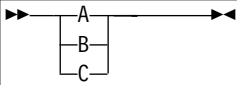
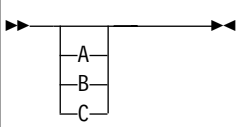
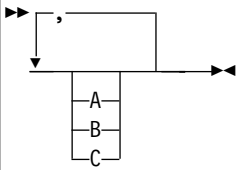
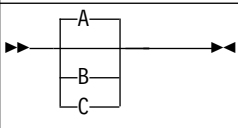
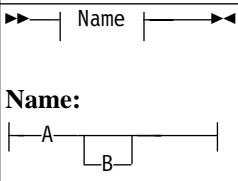
규칙	의미
	값 A, B, C를 지정해야 합니다. 필수 값은 구문 다이어그램의 기본 줄에 표시됩니다.
	값 A를 지정할 수 있습니다. 선택적 값은 구문 다이어그램의 기본 줄 아래에 표시됩니다.
	값 A, B, C는 대체값으로, 그 중 하나를 반드시 지정해야 합니다.
	값 A, B, C는 대체값으로, 그 중 하나를 지정할 수 있습니다.

표 15. 구문 다이어그램을 읽는 방법 (계속)

규칙	의미
	값 A, B 및 C에서 하나 이상을 지정할 수 있습니다. 여러 개 또는 반복되는 값에 대한 필수 분리 문자(이 예에서는 쉼표(,))가 화살표에 표시되어 있습니다.
	값 A, B, C는 대체값으로, 그 중 하나를 지정할 수 있습니다. 표시된 값을 전혀 지정하지 않으면, 디폴트 A(기본 선 위에 표시된 값)가 사용됩니다.
	구문 부분인 Name은 기본 구문 다이어그램과 분리되어 표시됩니다.

구문 도움말

명령어 뒤에 물음표를 입력하여 이 장에 있는 임의의 명령 구문에 대해 도움말을 볼 수 있습니다. MQSeries는 선택한 명령에 대해 요구되는 구문을 나열하여 이에 응답합니다. 구문은 명령과 연관되는 모든 매개변수와 변수를 보여줍니다. 다른 형식인 괄호는 매개변수의 필수 여부를 나타내기 위해 사용됩니다. 예를 들면 다음과 같습니다.

```
CmdName [-x OptParam ] ( -c | -b ) { -p principal } argument
```

여기서

CmdName

도움말이 요청된 명령어입니다.

[-x OptParam]

대괄호는 이것이 선택적 매개변수임을 나타냅니다.

(-c | -b)

필수 필드. 이 경우에 플래그 c 또는 b를 선택해야 합니다.

{ -p principal }

제공할 수 있는 선택적 변수 목록. 그러나, 이것이 표시되면 명령을 입력할 때 최소한 하나의 변수를 제공해야 합니다.

argument

이 명령과 함께 제공해야 하는 인수로, 조회에 대한 응답에 표시될 경우 필수입니다.

예

1. endmqm 입력 결과

```
endmqm [-z][-c | -i | -p] QMgrName
```

2. rcdmqimg 입력 결과

```
rcdmqimg [-z] [-m QMgrName] -t ObjType [GenericObjName]
```


MQSeries 리턴 코드

대부분의 MQSeries 명령(예: **crtmqm**)은 명령의 성공 또는 실패를 나타내기 위해 종료 상태 표시줄을 작성합니다.

명령 상태가 DCL 명령 상태에서 테스트될 경우, MQSeries 프로그램으로부터 리턴된 상태 값을 해석해야 합니다.

MQSeries 리턴 코드는 SYS\$MESSAGE:MQS_MSG.EXE라고 하는 메시지 파일에 정의됩니다.

파일의 리턴 코드와 연관되는 메시지 텍스트에 액세스하려면, 반드시 DCL SET MESSAGE 명령을 사용해야 합니다. 이 명령은 프로세스의 메시지 테이블에 메시지 코드를 로드합니다. 예를 들면 다음과 같습니다.

```
$ SET MESSAGE SYS$MESSAGE:MQS_MSG.EXE
```

이후에는 F\$MESSAGE 구문 함수를 사용하여 MQSeries 리턴 코드의 텍스트를 인쇄할 수 있습니다. 예를 들면 다음과 같습니다.

```
$ strmqm )(*bad-qm-name&$$#
The queue manager name is either not valid or not known
$ WRITE SYS$OUTPUT F$MESSAGE($STATUS)
%MQS-F-CSPRC_Q_MGR_NAM, Queue manager name error
```

OpenVMS 리턴 코드를 OS/2 또는 UNIX 시스템용 MQSeries에서 사용되는 리턴 코드 값으로 변환하려면, 다음의 DCL 등식을 사용하면 됩니다.

```
$ RC = $STATUS / 8 .AND. %XFFF
```

예를 들면 다음과 같습니다.

```
$ crtmqm &*)*(
The queue manager name is either not valid or not known
$ RC = $STATUS / 8 .AND. %XFFF
$ SHOW SYMBOL RC
RC = 72 Hex = 00000048 Octal = 00000000110
```

crtmqcvx(데이터 변환)

목적

데이터 유형 구조에서 데이터 변환을 수행하는 코드 부분을 작성하려면 **crtmqcvx** 명령을 사용하십시오. 이 명령은 C 구조를 변환하기 위해 종료에서 사용될 수 있는 C 함수를 생성합니다.

이 명령은 구조를 포함하는 또는 변환될 구조를 포함하는 입력 파일을 읽습니다. 그리고 나서, 코드 부분 또는 해당 구조로 변환될 부분을 포함하는 출력 파일을 작성합니다.

이 명령과 그 사용 방법에 대한 자세한 정보는 *MQSeries Application Programming Guide*를 참조하십시오.

구문

```
▶▶—crtmqcvx—SourceFile—TargetFile—▶▶
```

필수 매개변수

SourceFile

변환될 C 구조를 포함하는 입력 파일을 지정합니다.

TargetFile

구조를 변환하기 위해 생성된 코드 부분을 포함하는 출력 파일을 지정합니다.

리턴 코드

- 0** 명령이 정상적으로 완료됨
- 10** 명령이 예상치 못한 결과로 완료됨
- 20** 처리 중에 오류가 발생함

예제

다음 예는 소스 C 구조에 대해 데이터 변환 명령을 사용한 결과를 보여줍니다. 발행된 명령은 다음과 같습니다.

```
crtmqcvx source.tmp target.c
```

입력 파일인 source.tmp는 다음과 같이 표시됩니다.

```

/* This is a test C structure which can be converted by the */
/* crtmqcvx utility                                         */

struct my_structure
{
    int    code;
    MQLONG value;
};

```

명령에 의해 생성되는 출력 파일인 target.c를 아래에서 보여줍니다. 응용프로그램에서 이러한 코드 부분을 사용하여 데이터 구조를 변환할 수 있습니다. 그러나, 그렇게 할 경우 그 코드 부분에서 MQSeries 헤더 파일 amqsvmha.h에 제공된 매크로가 사용된다는 점을 알아야 합니다.

```

MQLONG Convertmy_structure(
    PMQBYTE *in_cursor,
    PMQBYTE *out_cursor,
    PMQBYTE in_lastbyte,
    PMQBYTE out_lastbyte,
    MQHCONN hConn,
    MQLONG  opts,
    MQLONG  MsgEncoding,
    MQLONG  ReqEncoding,
    MQLONG  MsgCCSID,
    MQLONG  ReqCCSID,
    MQLONG  CompCode,
    MQLONG  Reason)
{
    MQLONG ReturnCode = MQRC_NONE;

    ConvertLong(1); /* code */

    AlignLong();
    ConvertLong(1); /* value */

Fail:
    return(ReturnCode);
}

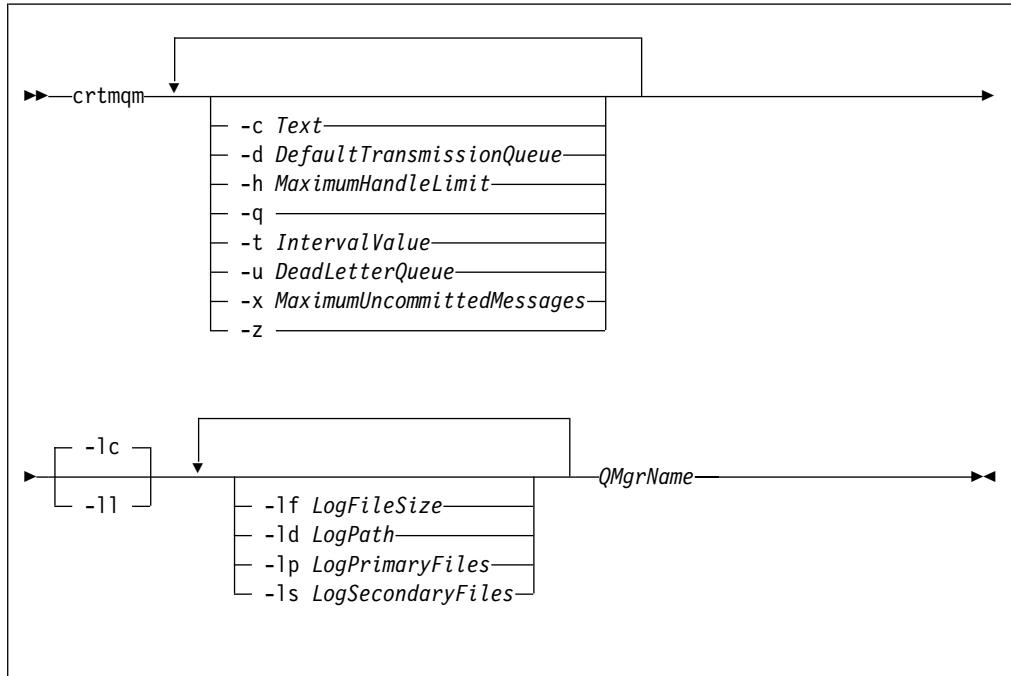
```

crtmqm(큐 관리자 작성)

목적

로컬 큐 관리자를 작성하려면 **crtmqm** 명령을 사용하십시오. 큐 관리자가 작성되었으면, **strmqm** 명령을 사용하여 시작하십시오.

구문



필수 매개변수

QMgrName

작성될 큐 관리자의 이름을 지정합니다. 이름은 48자 이하로 지정할 수 있습니다. 이것은 명령의 마지막 항목이어야 합니다.

선택적 매개변수

-c *Text*

이 큐 관리자에 대한 설명적 텍스트를 지정하십시오. 디폴트는 모두 공백입니다. 최대 64자까지 가능합니다. 대소문자가 혼합되어야 할 경우, 설명을 큰 따옴표로 묶어야 합니다.

-d *DefaultTransmissionQueue*

트랜스미션 큐가 해당되는 목적지에 대해 명확하게 정의되지 않은 경우에 리모트 메시지가 놓이는 로컬 트랜스미션 큐의 이름을 지정합니다. 디폴트는 없습니다.

-h *MaximumHandleLimit*

하나의 응용프로그램이 동시에 열 수 있는 최대 핸들 수를 지정합니다.

1 - 999 999 999 범위의 값을 지정하십시오. 디폴트 값은 256입니다.

-q 는 이 큐 관리자가 디폴트 큐 관리자임을 지정합니다. 새로운 큐 관리자는 디폴트로 기존의 큐 관리자를 바꿉니다.

실수로 이 플래그를 사용한 후, 다시 기존 큐 관리자를 디폴트 큐 관리자로 되돌리려고 하면, MQSeries 구성 파일에서 *DefaultQueueManager* 스텐자를 편집하십시오. 구성 파일에 대한 정보는 183 페이지의 『제13장 MQSeries 구성』을 참조하십시오.

-t *IntervalValue*

이 큐 관리자에 의해 제어되는 모든 큐에 대한 트리거 시간 간격을 밀리초 단위로 지정합니다. 이 값은 트리거가 일시중단될 때 트리거에서 생성하는 메시지를 수신한 후의 시간을 지정합니다. 즉, 큐에 메시지가 도착하여 트리거 메시지가 아니시에 이션 큐에 넣어질(put) 경우, 지정된 시간 내에 같은 큐에 도착하는 메시지들은 다른 트리거 메시지를 생성하지 않습니다.

트리거 시간 간격을 사용하여, 응용프로그램이 같은 큐에서 다른 처리 조건을 처리하도록 변경되기 전에 해당 트리거 조건으로 처리할 수 있는 충분한 시간이 허용되도록 할 수 있습니다. 발생하는 모든 트리거 이벤트를 볼 수도 있습니다. 그렇게 하려면, 이 필드에 낮은 값이나 0을 설정하십시오.

0 - 999 999 999 범위 내의 값을 지정하십시오. 디폴트는 999 999 999 밀리초 (11일을 초과하는 시간)입니다. 디폴트가 적용되도록 할 경우, 이는 첫번째 트리거 메시지 후 트리거가 사용되지 않음을 의미합니다. 그러나, 트리거는 큐 변경 명령을 사용하여 트리거 속성을 재설정함으로써 큐에 서비스를 제공하는 응용프로그램에서 다시 사용할 수 있습니다.

-u *DeadLetterQueue*

데드-레터(미배달 메시지) 큐로 사용될 로컬 큐의 이름을 지정합니다. 메시지가 해당되는 목적지로 라우트될 수 없을 경우 그 메시지들은 이 큐에 넣어(put)집니다.

속성이 생략될 경우의 디폴트는 데드-레터 큐가 아닙니다.

-x *MaximumUncommittedMessages*

하나의 동기점 아래에서 확약되지 않은 최대 메시지 수를 지정합니다. 즉, 다음의 합입니다.

- 큐에서 검색될 수 있는 메시지 수
- 큐에 넣어(put)질 수 있는 메시지 수
- 해당되는 작업 단위 내에 생성되는 트리거 메시지 수

이 한계는 동기점 외부에서 검색되거나 넣어지는(put) 메시지에는 적용되지 않습니다.

crtmqm

1 - 10 000 범위의 값을 지정하십시오. 디폴트 값은 1000개의 확장되지 않은 메시지는입니다.

-z 오류 메시지를 억제합니다.

이 플래그는 보통 MQSeries 내에서 원하지 않는 오류 메시지를 억제하기 위해 사용됩니다. 이 플래그를 사용하면 정보가 유실되므로, 명령행에 명령을 입력할 경우에는 이를 사용하지 않는 것이 좋습니다.

작성되는 큐 관리자에서 사용될 로그 기록을 정의하기 위해 다음과 같은 플래그 세트가 사용됩니다. 로그에 대한 자세한 정보는 155 페이지의 『복구에 로그 사용』을 참조하십시오.

-lc 순환 로깅이 사용됩니다. 이것은 디폴트 로그 기록 방법입니다.

-ll 선형 로깅이 사용됩니다.

-lf *LogFileSize*

로그 파일의 크기를 4 KB 단위로 지정합니다. 최소값은 64이고, 최대값은 16384입니다. 디폴트 값은 1024로, 디폴트 로그 크기 4 MB를 제공합니다.

-ld *LogPath*

로그 파일을 보유하기 위해 사용할 디렉토리를 지정합니다. 디폴트는 MQS_ROOT:[MQM.LOG]입니다. 디폴트는 MQSeries가 사용자에게 맞게 정의될 때도 변경할 수 있습니다.

사용자 ID mqm 및 그룹 mqm은 로그 파일에 대한 전체 권한을 가지고 있어야 합니다. 이 파일들의 위치를 변경할 경우, 이 권한을 사용자 스스로 부여해야 합니다. 이것은 로그 파일이 해당되는 디폴트 위치에 있을 경우에 자동으로 수행됩니다.

-lp *LogPrimaryFiles*

할당될 1차 로그 파일 수를 지정합니다. 디폴트 값은 3이고, 최소값은 2이며, 최대값은 62입니다.

-ls *LogSecondaryFiles*

할당될 2차 로그 파일 수를 지정합니다. 디폴트 값은 2이고, 최소값은 1이며, 최대값은 61입니다.

주: 총 로그 파일 수는 요청된 수에 관계없이 63으로 제한됩니다.

이전 매개변수 설명에 제공된 한계는 MQSeries에서 설정된 한계입니다. 운영 체제 한계는 가능한 최대 로그 크기를 줄일 수도 있습니다.

리턴 코드

- 0 큐 관리자가 작성됨
- 8 큐 관리자가 이미 존재함

- 49 큐 관리자 정지 중
- 69 저장영역 사용 불가능
- 70 큐 공간 사용불가능
- 71 예상치 못한 오류
- 72 큐 관리자 이름 오류
- 100 로그 위치가 올바르지 않음
- 111 큐 관리자가 작성되었으나, 제품 구성 파일에서 디폴트 큐 관리자 정의를 처리하는 중에 문제점이 발생했습니다. 디폴트 큐 관리자 스펙이 올바르지 않을 수도 있습니다.
- 115 올바르지 않은 로그 크기

예제

1. 이 명령은 Paint.queue.manager라고 하는 디폴트 큐 관리자를 작성합니다. 이 큐 관리자에는 Paint shop 설명이 제공됩니다. 또한, 이 명령은 선형 로깅이 사용됨을 지정합니다.

```
crtmqm -c "Paint shop" -ll -q "Paint.queue.manager"
```

2. 이 예는 로그 파일 수를 요청합니다. 두 개의 1차 로그 파일, 세 개의 2차 로그 파일이 지정되어 있습니다.

```
crtmqm -c "Paint shop" -ll -lp 2 -ls 3 -q "Paint.queue.manager"
```

3. 이 예에서, 다른 큐 관리자인 travel이 작성됩니다. 트리거 간격은 5000 밀리초(또는 5초)로 정의되고 해당되는 데드-레터 큐는 SYSTEM.DEAD.LETTER.QUEUE로 지정됩니다.

```
crtmqm -t 5000 -u SYSTEM.DEAD.LETTER.QUEUE "travel"
```

트리거 이벤트가 생성되었으면, 5초 동안 추가 트리거 이벤트를 사용할 수 없게 됩니다.

관련 명령

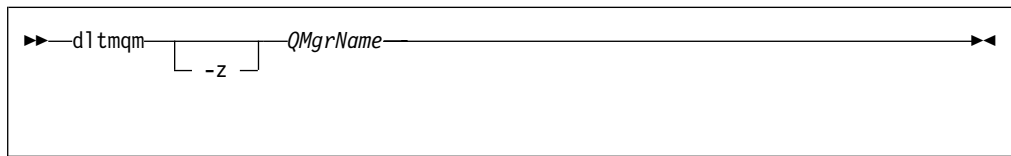
- strmqm** 큐 관리자 시작
- endmqm** 큐 관리자 종료
- dltmqm** 큐 관리자 삭제

dltmqm(큐 관리자 삭제)

목적

지정된 큐 관리자를 삭제하려면 **dltmqm** 명령을 사용하십시오. 이 큐 관리자와 연관되는 모든 오브젝트도 삭제됩니다. 큐 관리자를 삭제할 수 있으려면 **endmqm** 명령을 사용하여 이를 종료해야 합니다.

구문



필수 매개변수

QMgrName

삭제될 큐 관리자의 이름을 지정합니다.

선택적 매개변수

-z 오류 메시지를 억제합니다.

리턴 코드

- 0** 큐 관리자가 삭제됨
- 3** 큐 관리자가 작성됨
- 5** 큐 관리자 실행 중
- 16** 큐 관리자가 없음
- 49** 큐 관리자 정지 중
- 69** 저장영역 사용 불가능
- 71** 예상치 못한 오류
- 72** 큐 관리자 이름 오류
- 100** 로그 위치가 올바르지 않음
- 112** 큐 관리자가 삭제되었으나, 제품 구성 파일에서 디폴트 큐 관리자 정의를 처리하는 중에 문제점이 발생했습니다. 디폴트 큐 관리자 스펙이 올바르지 않을 수도 있습니다.

예제

1. 다음 명령은 큐 관리자 `saturn.queue.manager`를 삭제합니다.


```
dltmqm "saturn.queue.manager"
```

2. 다음 명령은 큐 관리자 travel을 삭제하고 이 명령으로 발생하는 모든 메시지를 억제합니다.

```
dltmqm -z "travel"
```

dltmqm

관련 명령

crtmqm	큐 관리자 작성
strmqm	큐 관리자 시작
endmqm	큐 관리자 종료

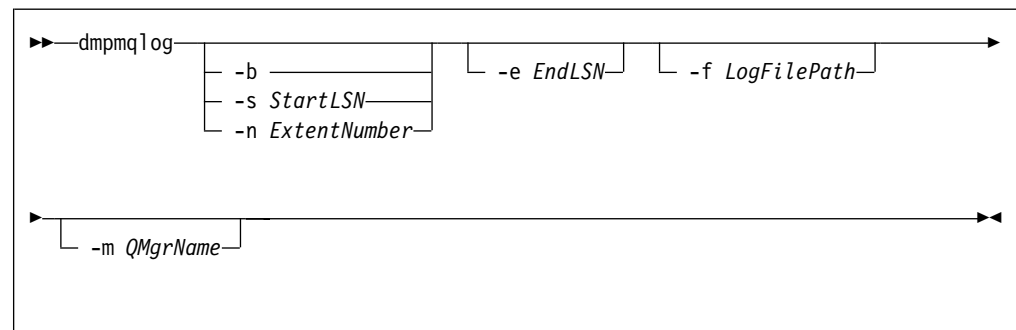
dmpmqlog(로그 덤프)

목적

MQSeries 시스템 로그의 형식화된 버전을 덤프하려면 **dmpmqlog** 명령을 사용하십시오.

덤프될 로그는 명령을 발행하기 위해 사용될 운영 체제와 동일한 환경에서 작성되어야 합니다.

구문



선택적 매개변수

덤프 시작점

덤프를 시작해야 할 LSN(Log Sequence Number)을 지정하려면 다음 명령 중 하나를 사용하십시오. 어떤 시작점도 지정하지 않으면, 덤프는 디폴트로 로그의 활성 부분에서 첫번째 레코드의 LSN에서 시작합니다.

-b 기본 LSN에서 덤프가 시작되어야 함을 지정합니다. 기본 LSN은 로그 활성 부분의 시작을 포함하는 로그 범위의 시작을 식별합니다.

-s *StartLSN*

지정된 LSN에서 덤프가 시작되어야 함을 지정합니다. LSN은 nnnn:nnnn:nnnn:nnnn 형식으로 지정됩니다.

순환 로깅을 사용할 경우, LSN 값은 로그의 기본 LSN 값보다 크거나 같아야 합니다.

-n *ExtentNumber*

지정된 범위 번호에서 덤프가 시작되어야 함을 지정합니다. 범위 번호는 0 - 9 999 999 사이입니다.

이 매개변수는 *LogType*이 (구성 파일 *qm.ini*에 기록된 대로) LINEAR인 큐 관리자에 대해서만 유효합니다.

dmpmqlog

-e *EndLSN*

지정된 LSN에서 덤프가 종료되어야 함을 지정합니다. LSN은 nnnn:nnnn:nnnn:nnnn 형식으로 지정됩니다.

-f *LogFilePath*

로그 파일에 대한 상대적이 아닌, 절대적 디렉토리 경로 이름입니다. 지정된 디렉토리에는 로그 헤더 파일(amqhlctl.lfh)과 active라고 하는 서브디렉토리가 있어야 합니다. 활성 서브디렉토리는 로그 파일을 포함해야 합니다. 디폴트로, 로그 파일은 mqs.ini 및 qm.ini 파일에 지정된 디렉토리에 있는 것으로 가정됩니다. 이 옵션이 사용되면, 큐 ID와 연관된 큐 이름은 큐 관리자 이름이 -m 옵션에 대해 명확하게 지정된 경우와 그 큐 관리자가 해당 디렉토리 경로에 오브젝트 카탈로그 파일을 가지고 있는 경우에만 덤프에 표시됩니다.

긴 파일 이름을 지원하는 시스템에서, 이 파일 이름은 qmqmobjcat로 지정되는데, 큐 ID를 큐 이름에 맵핑하려면 로그 파일이 작성될 때 사용된 그 파일이어야 합니다. 예로, 큐 관리자 이름이 qm1이면, 오브젝트 카탈로그 파일은 MQS_ROOT:[MQM.QMGRS.QM1.QMANAGER] 디렉토리에 위치됩니다. 이 맵핑을 수행하려면, 임시 큐 관리자를 작성하고(예: tmpq), 해당 오브젝트 카탈로그를 특정 로그 파일과 연관된 카탈로그로 대체한 후, 로그 파일에 대한 절대 디렉토리 경로와 함께 -m tmpq 및 -f 를 지정하여 dmpmqlog를 시작해야 합니다.

-m *QMgrName*

큐 관리자의 이름입니다. 이 매개변수를 생략하면, 디폴트 큐 관리자 이름이 사용됩니다.

지정하는 큐 관리자나 디폴트로 설정한 큐 관리자는 **dmpmqlog** 명령이 발행될 때 실행 상태가 아니어야 합니다. 마찬가지로, 큐 관리자는 **dmpmqlog**가 실행 중일 때 시작하지 말아야 합니다.

dspmqaout(권한 표시)

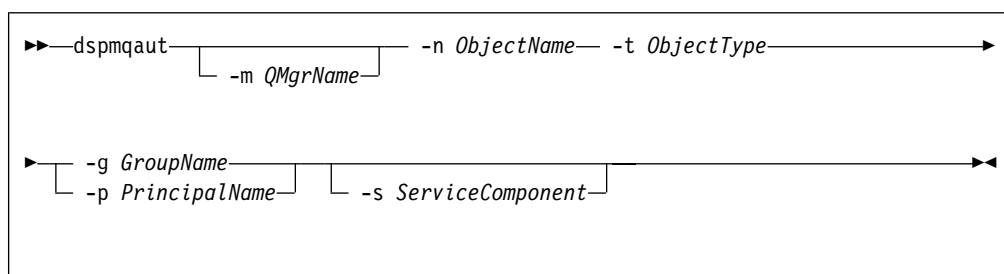
목적

지정된 오브젝트에 대한 현재 권한을 표시하려면 **dspmqaout** 명령을 사용하십시오.

하나의 그룹만 지정할 수 있습니다.

사용자 ID가 여러 그룹의 구성원일 경우, 이 명령은 모든 그룹의 결합된 권한을 표시합니다.

구문



필수 매개변수

-n *ObjectName*

조회할 오브젝트의 이름을 지정합니다.

이것은 조회가 큐 관리자 자체에 관련되지 않을 경우(포함되지 않아야 함) 필수 매개변수입니다.

큐 관리자, 큐 또는 프로세스 정의의 이름을 지정해야 합니다.

-t *ObjectType*

조회할 오브젝트의 유형을 지정합니다. 가능한 값은 다음과 같습니다.

queue 또는 **q** 오브젝트 유형 매개변수와 일치하는 큐(들).

qmgr 큐 관리자 오브젝트

process 또는 **prcs**

프로세스

namelist 또는 **nl**

이름 목록

선택적 매개변수

-m *QMgrName*

조회할 큐 관리자의 이름을 지정합니다.

dspmqaut

-g *GroupName*

조회할 사용자 그룹의 이름을 지정합니다. *하나의* 이름만 지정할 수 있습니다. 이 이름은 기존 권한 ID의 이름이어야 합니다.

-p *PrincipalName*

지정된 오브젝트에 대한 권한이 부여된 사용자의 이름을 지정합니다.

-s *ServiceComponent*

이 매개변수는 설치가능 권한 서비스를 사용할 경우에만 적용되며, 그렇지 않을 경우에는 무시됩니다.

설치가능 권한 서비스가 지원될 경우, 이 매개변수는 권한이 적용되는 권한 부여 서비스의 이름을 지정합니다. 이 매개변수는 선택적입니다. 이를 지정하지 않으면, 서비스에 대한 첫번째 설치가능 구성요소에 대해 권한 조회가 수행됩니다.

리턴된 매개변수

이 명령은 권한 목록을 리턴합니다. 이 목록에는 매개변수가 전혀 없거나 하나 또는 그 이상이 포함될 수 있습니다. 리턴된 각 권한 부여 매개변수는 지정된 그룹의 사용자 ID가 그 매개변수에 정의된 조작을 수행할 수 있는 권한을 가지고 있음을 의미합니다.

표16에서는 서로 다른 오브젝트 유형에 제공될 수 있는 권한을 보여줍니다.

표 16. *dspmqaut* 명령의 보안 권한

권한	큐	프로세스	Qmgr	이름 목록
all	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓
clr	✓			
connect			✓	
cpy	✓	✓	✓	✓
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
get	✓			
inq	✓	✓	✓	✓
passall	✓			
passid	✓			
put	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

다음 목록은 각 매개변수와 연관되는 권한을 정의합니다.

all	오브젝트에 관련되는 모든 조작을 사용합니다.
alladm	오브젝트에 관련되는 모든 관리 조작을 수행합니다.
allmqi	오브젝트에 관련되는 모든 MQI 호출을 사용합니다.
altusr	MQI 호출에 대한 대체 사용자 ID를 지정합니다.
browse	BROWSE 옵션으로 MQGET 호출을 발행하여 큐에서 메시지를 검색합니다.
chg	해당되는 명령 세트를 사용하여, 지정된 오브젝트의 속성을 변경합니다.
clr	큐를 지웁니다(PCF 명령 큐 지우기 전용).
connect	MQCONN 호출을 발행하여 지정된 큐 관리자에 응용프로그램을 연결합니다.
cpy	오브젝트(예: PCF 큐 복사 명령)의 정의를 복사합니다.
crt	해당되는 명령 세트를 사용하여, 지정된 유형의 오브젝트를 작성합니다.
dlt	해당되는 명령 세트를 사용하여, 지정된 오브젝트를 삭제합니다.
dsp	해당되는 명령 세트를 사용하여, 지정된 오브젝트의 속성을 표시합니다.
get	MQGET 호출을 발행하여 큐에서 메시지를 검색합니다.
inq	MQINQ 호출을 발행하여 특정 큐에 대해 조회합니다.
passall	모든 컨텍스트를 전달합니다.
passid	ID 컨텍스트를 전달합니다.
put	MQPUT 호출을 발행하여 특정 큐에 메시지를 넣습니다(put).
set	MQSET 호출을 발행하여 MQI에서 큐의 속성을 설정합니다.
setall	큐의 모든 컨텍스트를 설정합니다.
setid	큐의 ID 컨텍스트를 설정합니다.

지원되는 관리 조작에 대한 권한은 다음 명령 세트에 적용됩니다.

- 제어 명령
- MQSC 명령
- PCF 명령

리턴 코드

0	조작 성공
36	올바르지 않은 인수가 제공됨
40	큐 관리자 사용불가능

dspmqaut

- 49 큐 관리자 정지 중
- 69 저장영역 사용 불가능
- 71 예상치 못한 오류
- 72 큐 관리자 이름 오류
- 133 알 수 없는 오브젝트 이름
- 145 예상치 못한 오브젝트 이름
- 146 오브젝트 이름 누락
- 147 오브젝트 유형 누락
- 148 올바르지 않은 오브젝트 유형
- 149 엔티티 이름 누락

예제

다음 예는 사용자 그룹 `staff`와 연관되는 `saturn.queue.manager` 큐 관리자에 대한 권한을 표시하는 명령을 보여줍니다.

```
dspmqaut -m "saturn.queue.manager" -t qmgr -g staff
```

이 명령의 결과는 다음과 같습니다.

```
Entity staff has the following authorizations for object:  
  get  
  browse  
  put  
  inq  
  set  
  connect  
  altusr  
  passid  
  passall  
  setid
```

관련 명령

setmqaut 권한 설정 또는 재설정

dspmqsrv(명령 서버 표시)

목적

지정된 큐 관리자에 대한 명령 서버의 상태를 표시하려면 **dspmqsrv** 명령을 사용하십시오.

그 상태는 다음 중 하나가 될 수 있습니다.

- 시작 중
- 실행 중
- 가져오기(get)에 대해 SYSTEM.ADMIN.COMMAND.QUEUE가 사용되지 않는 상태에서 실행 중
- 종료 중
- 정지 중

구문

```
▶▶ dspmqsrv <QMgrName> ▶▶
```

선택적 매개변수

QMgrName

명령 서버 상태가 요청되는 로컬 큐 관리자의 이름을 지정합니다.

리턴 코드

- 0** 명령이 정상적으로 완료됨
- 10** 명령이 예상치 못한 결과로 완료됨
- 20** 처리 중에 오류가 발생함

예제

다음 명령은 `venus.q.mgr`과 연관되는 명령 서버의 상태를 표시합니다.

```
dspmqsrv "venus.q.mgr"
```

관련 명령

strmqsv 명령 서버 시작

dspmqsv

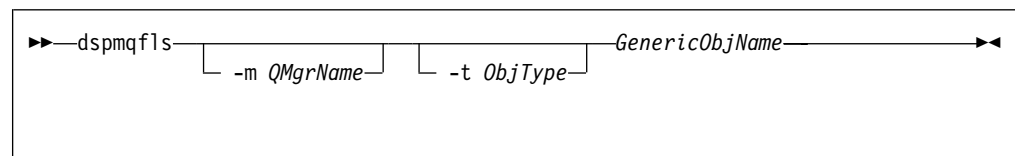
endmqsv 명령 서버 종료

dspmqls(MQSeries 파일 표시)

목적

지정된 기준과 일치하는 모든 MQSeries 오브젝트에 대해 실제 파일 시스템 이름을 표시하려면 **dspmqls** 명령을 사용하십시오. 이 명령을 사용하여 특정 MQSeries 오브젝트와 연관되는 파일을 식별할 수 있습니다. 이것은 특정 오브젝트의 백업에 유용합니다. 이름 변환에 대한 추가 정보는 25 페이지의 『MQSeries 파일 이름 이해』를 참조하십시오.

구문



필수 매개변수

GenericObjName

MQSeries 오브젝트의 이름을 지정합니다. 이름은 프래그가 없는 문자열로, 필수 매개변수입니다. 이름을 생략하면 오류가 리턴됩니다.

이 매개변수는 문자열 끝에서 와일드카드 문자 *를 지원합니다.

선택적 매개변수

-m *QMgrName*

파일이 조사될 큐 관리자의 이름을 지정합니다. 생략할 경우, 명령은 디폴트 큐 관리자에서 작동됩니다.

-t *ObjType*

MQSeries 오브젝트 유형을 지정합니다. 다음 목록은 올바른 오브젝트 유형을 보여줍니다. 축약된 이름이 먼저 표시되고 그 다음에 전체 이름이 표시됩니다.

***** 또는 **all** 모든 오브젝트 유형. 이것이 디폴트입니다.

q 또는 **queue** 오브젝트 이름 매개변수와 일치하는 큐

ql 또는 **qlocal**

로컬 큐

qa 또는 **qalias**

알리아스 큐

qr 또는 **qremote**

리모트 큐

dspmqfls

qm 또는 qmodel

모델 큐

qmgr 큐 관리자 오브젝트

prcs 또는 process

프로세스

ctlg 또는 catalog

오브젝트 카탈로그

nl 또는 namelist

이름 목록

주: **dspmqfls** 명령은 큐 이름이 아닌 큐를 포함하는 디렉토리를 표시합니다.

리턴 코드

- 0 명령이 정상적으로 완료됨
- 10 명령이 완료되었으나 예상과 다름
- 20 처리 중에 오류가 발생함

예제

1. 다음 명령은 디폴트 큐 관리자에 정의된 SYSTEM.ADMIN으로 시작하는 이름을 갖는 모든 오브젝트의 세부사항을 표시합니다.

```
dspmqfls SYSTEM.ADMIN*
```

2. 다음 명령은 RADIUS 큐 관리자에 정의된 PROC로 시작하는 이름을 갖는 모든 프로세스의 파일 세부사항을 표시합니다.

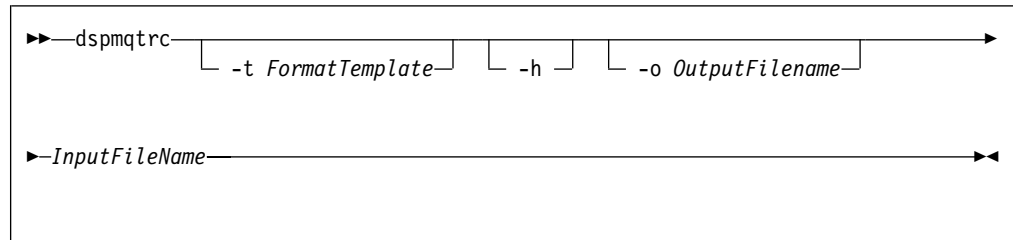
```
dspmqfls -m RADIUS -t prcs PROC*
```

dspmqtrc(MQSeries 형식화 추적 출력 표시)

목적

MQSeries 형식화 추적 출력을 표시하려면 **dspmqtrc** 명령을 사용하십시오.

구문



필수 매개변수

InputFileName

형식화되지 않은 추적을 포함하는 파일의 이름을 지정합니다(예: MQS_ROOT:[MQM.TRACE]AMQ20202345.TRC).

선택적 매개변수

-t *FormatTemplate*

추적을 표시하는 방법에 대한 세부사항을 포함하는 서식 파일의 이름을 지정합니다. 디폴트 값은 SYS\$SHARE:AMQTRC.FMT입니다.

-h 보고서에서 헤더 정보를 생략합니다.

-o *output_filename*

형식화된 데이터를 기록할 파일의 이름입니다.

예제

1. 다음 명령은 출력의 경로 재지정을 보여줍니다.

```
dspmqtrc mqs_root:[mqm.trace]amq20202345.trc > mqs_root:[mqm.trace] amq20202345.fmt
```

dspmqtrc

관련 명령

endmqtrc	MQSeries 추적 종료
strmqtrc	MQSeries 추적 시작

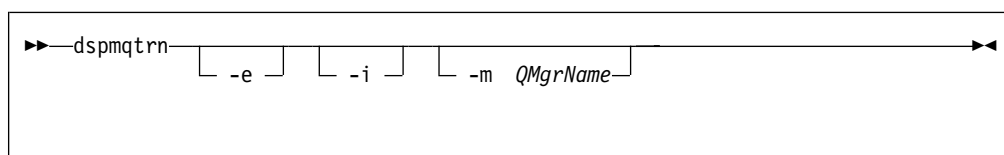
dspmqtrn(MQSeries 트랜잭션 표시)

목적

2단계 확약 프로시저에서 준비 상태에 있으며 큐 관리자에 알려진 트랜잭션을 표시하려면 **dspmqtrn** 명령을 사용하십시오(아래의 주의 참조).

각 트랜잭션은 트랜잭션 번호(사람이 읽을 수 있는 트랜잭션 ID), 트랜잭션 상태 및 트랜잭션 ID로 표시됩니다. 트랜잭션 ID의 길이는 최대 128자까지 가능합니다. 따라서 트랜잭션 번호가 필요합니다.

구문



주의: 이 명령은 사용자가 외부 트랜잭션 관리자를 사용 중이거나 2단계 확약 프로시저 단계에 있을 경우에만 사용하십시오. 2단계 확약을 사용하지 않으면, 이 명령을 사용하지 마십시오. 이 명령은 동기점 관리자가 트랜잭션을 해석하지 못한 경우에만 사용해야 합니다.

선택적 매개변수

-m *QMgrName*

트랜잭션이 검사될 큐 관리자의 이름을 지정합니다. 생략할 경우, 명령은 디폴트 큐 관리자에서 작동됩니다.

-e 외부적으로 통합된 인다우트(in-doubt) 트랜잭션의 세부사항을 요청합니다. 그러한 트랜잭션은 MQSeries가 확약 준비 요청을 받았으나 아직 트랜잭션 결과가 통지되지 않은 트랜잭션입니다.

-i 내부적으로 통합된 인다우트(in-doubt) 트랜잭션의 세부사항을 요청합니다. 그러한 트랜잭션은 각 자원 관리자가 확약 준비 요청을 받았으나 아직 트랜잭션 결과가 자원 관리자에 통지되지 않은 트랜잭션입니다.

참여하는 자원 관리자 각각에서 추론된 트랜잭션 상태에 대한 정보가 표시됩니다. 이 정보는 특정 자원 관리자에서의 실패 결과를 평가하는데 도움이 될 수 있습니다.

주: **-e**나 **-i**를 지정하지 않으면, 내부적으로 또는 외부적으로 통합된 인다우트(in-doubt) 트랜잭션 모두에 대해 세부사항이 표시됩니다.

dspmqtrn

리턴 코드

0	조작 성공
36	올바르지 않은 인수가 제공됨
40	큐 관리자 사용불가능
49	큐 관리자 정지 중
69	저장영역 사용 불가능
71	예상치 못한 오류
72	큐 관리자 이름 오류
102	트랜잭션을 찾을 수 없음

관련 명령

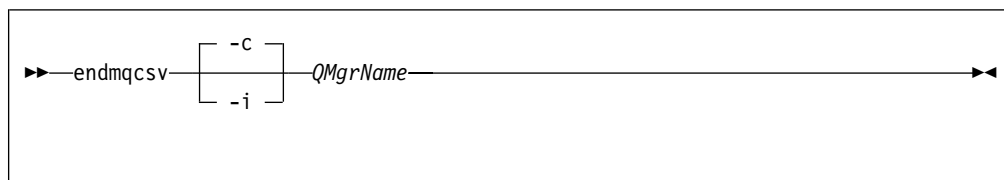
rsvmqtrn MQSeries 트랜잭션 해석

endmqcsv(명령 서버 종료)

목적

지정된 큐 관리자에 대해 명령 서버를 정지하려면 **endmqcsv** 명령을 사용하십시오.

구문



필수 매개변수

QMGrName

명령 서버가 종료될 큐 관리자의 이름을 지정합니다.

선택적 매개변수

-c 제어되는 방식으로 명령 서버가 정지됨을 지정합니다. 명령 서버에서는 이미 시작된 명령 메시지의 처리를 완료할 수 있습니다. 명령 큐의 어떤 새로운 메시지도 읽을 수 없습니다.

이것이 디폴트입니다.

-i 명령 서버가 즉시 정지됨을 지정합니다. 현재 처리 중인 명령 메시지와 연관되는 조치는 완료되지 않을 수도 있습니다.

리턴 코드

- 0** 명령이 정상적으로 완료됨
- 10** 명령이 예상치 못한 결과로 완료됨
- 20** 처리 중에 오류가 발생함

예제

1. 다음 명령은 큐 관리자 saturn.queue.manager에서 명령 서버를 정지합니다.

```
endmqcsv -c "saturn.queue.manager"
```

명령 서버는 정지되기 전에 이미 시작된 명령 처리를 완료할 수 있습니다. 수신된 명령은 명령 서버가 재시작될 때까지 명령 큐에서 처리되지 않은 상태로 남아 있습니다.

endmqsv

- 다음 명령은 큐 관리자 pluto에서 즉시 명령 서버를 정지합니다.

```
endmqsv -i "pluto"
```

관련 명령

- strmqsv 명령 서버 시작
- dspmqsv 명령 서버의 상태 표시

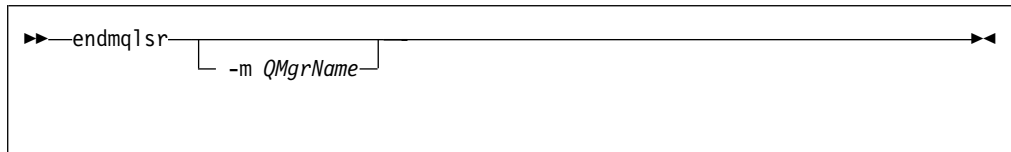
endmq1sr(리스너 종료)

목적

endmq1sr 명령은 지정된 큐 관리자에 대한 모든 리스너 프로세스를 종료합니다.

큐 관리자는 **endmq1sr** 명령이 발행되기 전에 정지해야 합니다.

구문



선택적 매개변수

-m QMgrName

큐 관리자의 이름을 지정합니다. 어떤 이름도 지정하지 않으면, 디폴트 큐 관리자에 대해 프로세스가 진행됩니다.

리턴 코드

- 0** 명령이 정상적으로 완료됨
- 10** 명령이 예상치 못한 결과로 완료됨
- 20** 처리 중에 오류가 발생함

endmqm(큐 관리자 종료)

목적

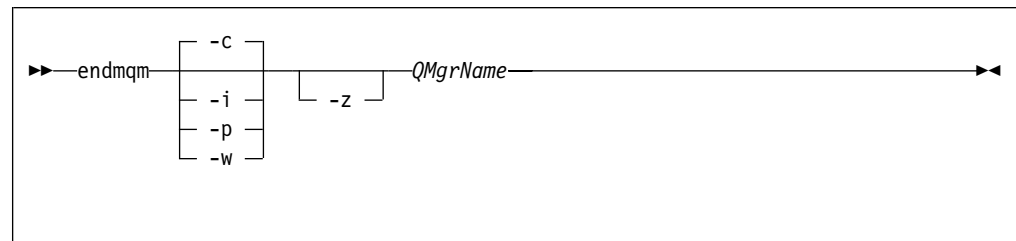
지정된 로컬 큐 관리자를 종료(정지)하려면 **endmqm** 명령을 사용하십시오. 이 명령은 세 가지 모드 중 하나에서 큐 관리자를 정지합니다.

- 정상 또는 정지 종료
- 즉시 종료
- 선제 종료

큐 관리자의 속성과 이에 연관되는 오브젝트는 영향을 받지 않습니다. **strmqm**(큐 관리자 시작) 명령을 사용하여 큐 관리자를 재시작할 수 있습니다.

큐 관리자를 삭제하려면, 이를 정지한 후 **dltmqm**(큐 관리자 삭제) 명령을 사용해야 합니다.

구문



필수 매개변수

QMGrName

정지할 메시지 큐 관리자의 이름을 지정합니다.

선택적 매개변수

-c 제어(또는 정지) 종료. 큐 관리자는 모든 응용프로그램이 연결을 끊은 후에만 정지합니다. 현재 처리 중인 MQI 호출은 완료됩니다. 이것이 디폴트입니다.

제어가 즉시 리턴되고 큐 관리자가 정지된 시기는 통지되지 않습니다.

-w 종료 대기

이 유형의 종료는 큐 관리자가 정지된 후에만 제어가 리턴된다는 것을 제외하고는 제어된 종료와 같습니다. 종료 처리 중에 "Waiting for queue manager *QMGrName* to end"라는 메시지가 표시됩니다.

-i 즉시 종료. 큐 관리자가 현재 처리 중인 모든 MQI 호출을 완료한 후에 정지합니다. 명령이 발행된 후에 발행된 MQI 요청은 실패합니다. 미완료 작업 단위는 큐 관리자가 다음에 시작될 때 롤백됩니다.

endmqm

-p 선제 종료.

예외 상황이 발생했을 경우에만 이 유형의 종료를 사용하십시오. 예를 들어, 큐 관리자가 정상적인 **endmqm** 명령을 실행해도 정지되지 않는 경우가 이에 해당합니다.

큐 관리자는 응용프로그램의 연결이 끊어지거나 MQI 호출이 완료되기를 기다리지 않고 정지합니다. 이는 MQSeries 응용프로그램에 대해 예측할 수 없는 결과를 제공할 수도 있습니다. 정지에 실패한 큐 관리자의 모든 프로세스는 명령이 발행된 후 30초에 종료됩니다.

주: 강제 또는 선제 종료를 수행한 후나 큐 관리자가 실패할 경우에는 큐 관리자가 소유하고 있는 공유 메모리를 클린업하지 않은 상태로 종료될 수도 있습니다. 이것은 재시작시 문제를 초래할 수 있습니다. 이러한 유형의 갑작스러운 종료 후에 MONMQ 유틸리티를 사용하여 클린업하는 방법에 대한 정보는 387 페이지의 『MONMQ를 사용하여 공유 메모리 관리』를 참조하십시오.

-z 명령에 대한 오류 메시지를 억제합니다.

리턴 코드

0	큐 관리자가 종료됨
3	큐 관리자가 작성됨
16	큐 관리자가 없음
40	큐 관리자 사용불가능
49	큐 관리자 정지 중
69	저장영역 사용 불가능
71	예상치 못한 오류
72	큐 관리자 이름 오류

예제

다음 예는 지정된 큐 관리자를 종료(정지)하는 명령을 보여줍니다.

1. 이 명령은 `mercury.queue.manager`라는 큐 관리자를 제어되는 방식으로 종료합니다. 현재 연결된 모든 응용프로그램의 연결을 끊을 수 있습니다.

```
endmqm "mercury.queue.manager"
```

2. 이 명령은 `saturn.queue.manager`라는 큐 관리자를 즉시 종료합니다. 현재 모든 MQI 호출은 완료되지만, 새로운 호출은 완료되지 않습니다.

```
endmqm -i "saturn.queue.manager"
```

endmqm

관련 명령

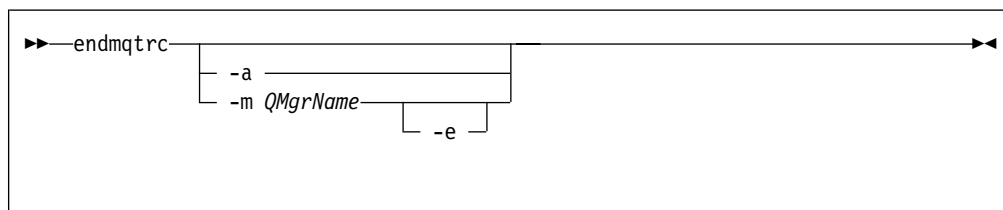
crtmqm	큐 관리자 작성
strmqm	큐 관리자 시작
dltmqm	큐 관리자 삭제

endmqtrc(MQSeries 추적 종료)

목적

지정한 엔티티나 모든 엔티티에 대한 추적을 종료하려면 **endmqtrc** 명령을 사용하십시오.

구문



선택적 매개변수

-m *QMgrName*

추적이 종료될 큐 관리자의 이름입니다.

명령에서 최대 하나의 **-m** 플래그와 연관되는 큐 관리자 이름을 제공할 수 있습니다.

큐 관리자 이름 및 **-m** 플래그는 **-e** 플래그와 같은 명령에서 지정할 수 있습니다.

-e 이 플래그를 지정할 경우, 초기 추적이 종료됩니다.

-a 이 플래그를 지정하면 모든 추적이 종료됩니다.

이 플래그는 반드시 단독으로 지정해야 합니다.

리턴 코드

AMQ5611 올바르지 않은 인수를 명령에 제공할 경우에 이 메시지가 발행됩니다.

예제

이 명령은 QM1이라고 하는 큐 관리자에 대한 데이터의 추적을 종료합니다.

```
endmqtrc -m QM1
```

관련 명령

dspmqtrc 형식화된 추적 출력 표시

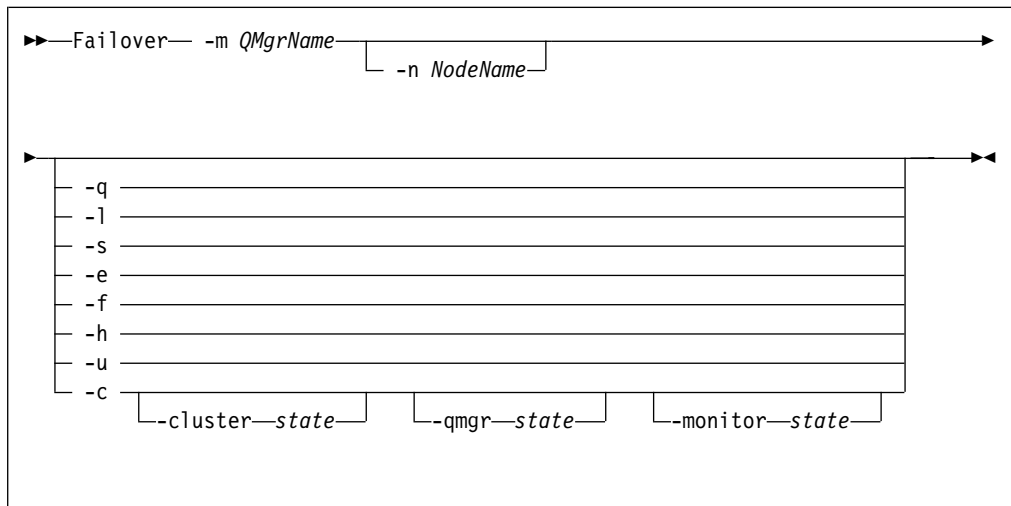
strmqtrc MQSeries 추적 시작

failover(Failover 세트 관리)

목적

Failover 세트를 관리하려면 **failover** 명령을 사용하십시오. **failover** 명령은 갱신 및 조회 매개변수 모두를 포함합니다. **failover** 명령은 Failover 세트의 모든 OpenVMS 노드에서 실행할 수 있습니다.

구문



필수 매개변수

-m QMgrName

failover 명령이 적용될 큐 관리자의 이름을 지정합니다. *QMgrName* 이름에 대해 지원되는 최대 길이는 25자입니다.

-n NodeName

명령이 적용될 OpenVMS 노드 이름을 지정합니다. 이 매개변수는 **-h** 및 **-c** 매개변수에 필요합니다.

선택적 매개변수

-q 는 Failover 세트의 상태를 조회하고 출력을 표시합니다.

-l Failover 세트의 상태를 조회하고 다음 DCL 기호를 설정합니다.

DCL 기호 이름	설명
MQ\$QMGR_NODE	실행 중인 큐 관리자가 없을 경우, 큐 관리자와 널(null) 문자열을 실행하는 OpenVMS 노드로 설정하십시오.
MQ\$AVAILABLE_NODES	큐 관리자 실행에 사용할 수 있는 OpenVMS 노드의 목록으로 설정하십시오. 이는 큐 관리자가 AVAILABLE 상태이고 Failover 모니터가 실행 중인 노드들입니다.

DCL 기호 이름	설명
MQS\$MONITOR_NODES	Failover 모니터가 실행 중인 OpenVMS 노드들의 목록으로 설정하십시오.

- s Failover 세트에서 큐 관리자를 시작합니다. -n 매개변수를 지정하면, 큐 관리자는 지정된 OpenVMS 노드에서 시작되고, 그렇지 않으면 최상위 우선순위의 사용가능한 노드에서 시작됩니다.
- e Failover 세트에서 큐 관리자를 종료합니다.
- f 큐 관리자를 Failover 세트의 다른 노드로 이동합니다. -n 매개변수를 지정하면, 큐 관리자는 지정된 노드로 이동되고, 그렇지 않으면 최상위 우선순위의 사용가능한 노드로 큐 관리자가 이동됩니다.
- h -n 매개변수와 함께 지정된 노드에서 실행 중인 Failover 모니터를 정지합니다.
- u 진행 중인 갱신 플래그를 지웁니다.
- c Failover 세트 상태를 변경합니다. 변경된 상태는 다음의 세 매개변수에 의해 판별됩니다. 변경사항은 Failover 세트의 실행 중 상태와 일관성이 있는 경우에만 적용됩니다.
 - cluster started | stopped**
전체 Failover 세트 상태를 변경하기 위해 -c 매개변수와 함께 사용됩니다.
 - qmgr available | running | excluded**
-n 매개변수로 지정된 노드에 대한 노드 큐 관리자 상태를 변경하기 위해 -c 매개변수와 함께 사용됩니다.
 - monitor started | stopped | watching**
-n 매개변수로 지정된 노드에 대한 노드 Failover 모니터 상태를 변경하기 위해 -c 매개변수와 함께 사용됩니다.

리턴 코드

- 0 명령이 정상적으로 완료됨
- 5 큐 관리자 실행 중
- 36 명령에 제공된 인수가 올바르지 않음
- 326 MQseries 큐 관리자가 실행 중이 아님
- 1925 큐 관리자에 대해 시작된 Failover 모니터가 없음
- 1926 Failover 세트 갱신 조작 진행 중
- 1937 큐 관리자를 시작할 사용가능한 노드가 없음
- 1939 큐 관리자 종료강제 실행됨
- 1940 큐 관리자 종료강제가 완료 이전에 시간 종료됨

OpenVMS 오류 코드

- 36 %SYSTEM-F-NOPRIV, 불충분한 특권 또는 오브젝트 보호 파기

Failover

652 %SYSTEM-F-NOSUCHNODE, 리모트 노드를 알 수 없음

660 %SYSTEM-F-REJECT, 네트워크 오브젝트에 대한 연결이 거부됨

예제

1. 이 예는 testqm 큐 관리자에 대한 Failover 세트의 상태를 조회합니다.

```
failover -m "testqm" -q
```

2. 이 예는 노드에서 testqm이라고 하는 큐 관리자를 시작합니다.

```
failover -m "testqm" -n batman -s
```

3. 이 예는 testqm이라고 하는 큐 관리자를 최상위 우선순위의 사용가능한 노드로 이 동합니다.

```
failover -m "testqm" -f
```

관련 명령

runmqfm Failover 모니터 시작

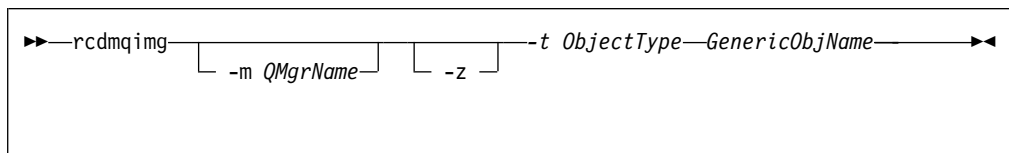
rcdmqimg(매체 이미지 기록)

목적

MQSeries 오브젝트 또는 오브젝트 그룹의 이미지를 매체 복원에 사용하기 위해 로그에 기록하려면 **rcdmqimg** 명령을 사용하십시오. 이미지에서 오브젝트를 재작성하려면 연관되는 명령인 **rcrmqobj**를 사용하십시오.

이 명령은 활성 큐 관리자에 대해 사용됩니다. 큐 관리자에 대한 추가 활동이 로그에 기록되므로, 이미지가 오래된 경우에도 로그 레코드는 오브젝트에 대한 변경사항을 반영합니다.

구문



필수 매개변수

GenericObjName

기록할 오브젝트의 이름을 지정합니다. 이 매개변수에는 별표 이전의 이름 부분과 일치하는 이름의 오브젝트가 기록됨을 나타내기 위해 후미 문자로 별표가 있을 수 있습니다.

이 매개변수는 큐 관리자 오브젝트나 채널 동기화 파일을 기록하고 있지 않은 경우 필수입니다. 채널 동기화 파일의 오브젝트 이름을 지정할 경우, 이것은 무시됩니다.

-t *ObjectType*

해당되는 이미지를 기록할 오브젝트의 유형을 지정합니다. 유효한 오브젝트 유형은 다음과 같습니다.

prcs 또는 **process**

프로세스

q 또는 **queue** 큐의 모든 유형

ql 또는 **qlocal**

로컬 큐

qa 또는 **qalias**

알리아스 큐

qr 또는 **qremote**

리모트 큐

qm 또는 qmodel	모델 큐
qmgr	큐 관리자 오브젝트
syncfile	채널 동기화 파일
nl 또는 namelist	이름 목록
ctlg 또는 catalog	오브젝트 카탈로그
* 또는 all	위의 것 모두

선택적 매개변수

-m *QMgrName*

이미지가 기록될 큐 관리자의 이름을 지정합니다. 생략할 경우, 명령은 디폴트 큐 관리자에서 작동됩니다.

-z 오류 메시지를 억제합니다.

리턴 코드

0	조작 성공
36	올바르지 않은 인수가 제공됨
40	큐 관리자 사용불가능
49	큐 관리자 정지 중
68	매체 복원이 지원되지 않음
69	저장영역 사용 불가능
71	예상치 못한 오류
72	큐 관리자 이름 오류
119	사용자에게 권한이 없음
128	처리된 오브젝트가 없음
131	자원 문제점
132	오브젝트 손상
135	기록할 수 없는 임시 오브젝트

예제

다음 명령은 큐 관리자 오브젝트 `saturn.queue.manager`의 이미지를 로그에 기록합니다.

```
rcdmqimg -t qmgr -m "saturn.queue.manager"
```

rcdmqing

관련 명령

rcrmqobj 큐 관리자 오브젝트 재작성

rcrmqobj(오브젝트 재작성)

목적

로그에 포함된 이미지로부터 오브젝트나 오브젝트 그룹을 재작성하려면 **rcrmqobj** 명령을 사용하십시오. 로그에 오브젝트 이미지를 기록하려면 연관된 명령 **rcdmqimg**를 사용하십시오.

이 명령은 실행 중인 큐 관리자에서 사용해야 합니다. 이미지가 기록된 후의 모든 활동이 로그에 기록됩니다. 오브젝트를 재작성하려면 로그를 재생하여 오브젝트 이미지가 캡처된 후에 발생한 이벤트를 재작성해야 합니다.

구문

```

rcrmqobj -m QMgrName -z -t ObjectType GenericObjName

```

필수 매개변수

GenericObjName

재작성할 오브젝트의 이름을 지정합니다. 이 매개변수에는 별표 이전의 이름 부분과 일치하는 오브젝트가 재작성됨을 나타내기 위해 후미 문자로 별표가 있을 수 있습니다.

이 매개변수는 오브젝트 유형이 채널 동기화 파일이 아닌 경우에는 필수입니다. 하지만, 오브젝트 이름이 이 오브젝트 유형에 대해 제공된 경우에는 무시됩니다.

-t *ObjectType*

재작성할 오브젝트의 유형을 지정합니다. 유효한 오브젝트 유형은 다음과 같습니다.

prcs 또는 **process**

프로세스

q 또는 **queue** 큐의 모든 유형

ql 또는 **qlocal**

로컬 큐

qa 또는 **qalias**

알리아스 큐

qr 또는 **qremote**

리모트 큐

qm 또는 qmodel

모델 큐

nl 또는 namelist

이름 목록

ctlg 또는 catalog

오브젝트 카탈로그

*** 또는 all**

위의 것 모두

syncfile

채널 동기화 파일

주: 이 플래그를 사용하면 지정된 큐 관리자에 대해 채널 동기화 파일이 다시 생성됩니다. 이것은 파일이 **rcdmqimg** 명령에 의해 저장되므로 필요한 것입니다.

선택적 매개변수

-m QMgrName

오브젝트가 재작성될 큐 관리자의 이름을 지정합니다. 생략할 경우, 명령은 디폴트 큐 관리자에서 작동됩니다.

-z 오류 메시지를 억제합니다.

리턴 코드

- 0** 조작 성공
- 36** 올바르지 않은 인수가 제공됨
- 40** 큐 관리자 사용불가능
- 49** 큐 관리자 정지 중
- 66** 매체 이미지 사용불가능
- 68** 매체 복원이 지원되지 않음
- 69** 저장영역 사용 불가능
- 71** 예상치 못한 오류
- 72** 큐 관리자 이름 오류
- 119** 사용자에게 권한이 없음
- 128** 처리된 오브젝트가 없음
- 135** 복구할 수 없는 임시 오브젝트
- 136** 오브젝트 사용 중

예제

1. 다음 명령은 디폴트 큐 관리자에 대해 모든 로컬 큐를 재작성합니다.

```
rcrmqobj -t ql *
```

2. 다음 명령은 큐 관리자 store와 연관되는 모든 리모트 큐를 재작성합니다.

```
rcrmqobj -m "store" -t qr *
```

rcrmqobj

관련 명령

rcdmqimg 로그에 MQSeries 오브젝트 기록

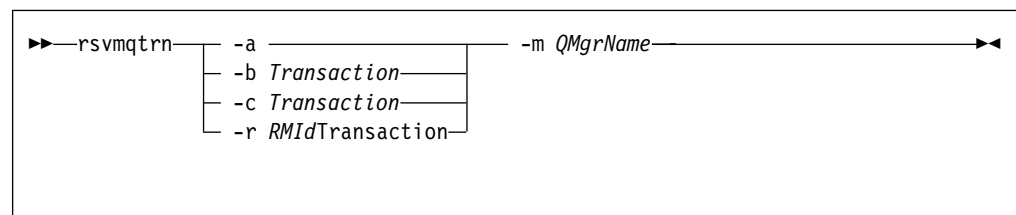
rsvmqtrn(MQSeries 트랜잭션 해석)

목적

내부적으로나 외부적으로 통합된 인다우트(in-doubt) 트랜잭션을 백아웃 또는 파악하려면 **rsvmqtrn** 명령을 사용하십시오.

일반 프로토콜로는 트랜잭션을 해석할 수 없는 것이 확실한 경우에만 이 명령을 사용하십시오. 이 명령을 발행하면 분산 트랜잭션에 대한 자원 관리자 사이에서 트랜잭션의 무결성이 손실될 수도 있습니다.

구문



필수 매개변수

-m *QMgrName*

큐 관리자의 이름을 지정합니다. 이 매개변수는 필수입니다.

선택적 매개변수

-a 큐 관리자가 내부적으로 통합된 모든 인다우트(in-doubt) 트랜잭션(즉, 모든 전역 작업 단위)을 해석하려고 시도해야 함을 지정합니다.

-b 이름 지정된 트랜잭션이 백아웃됨을 지정합니다. 이 플래그는 외부적으로 통합된 트랜잭션(즉, 외부 작업 단위)에 대해서만 유효합니다.

-c 이름 지정된 트랜잭션이 파악됨을 지정합니다. 이 플래그는 외부적으로 통합된 트랜잭션(즉, 외부 작업 단위)에 대해서만 유효합니다.

-r *RMId*

확약 또는 백아웃 결정이 적용될 자원 관리자를 식별합니다. 이 플래그는 내부적으로 통합된 트랜잭션과, 더이상 큐 관리자의 *qm.ini* 파일에 구성되어 있지 않은 자원 관리자에 대해서만 유효합니다. 전달되는 결과는 트랜잭션에 대해 MQSeries에서 도달되는 결정과 일관됩니다.

Transaction

확약되거나 백아웃될 트랜잭션의 트랜잭션 번호입니다. 관련 트랜잭션 번호를 발견하려면, **dspmqtrn** 명령을 사용하십시오. 이 매개변수는 **-b**, **-c** 및 **-r** *RMId* 매개변수에 대해 필수입니다.

rsvmqtrn

리턴 코드

0	조작 성공
32	트랜잭션을 해석할 수 없음
34	자원 관리자가 인식되지 않음
35	자원 관리자가 영구적으로 사용가능하지 않음
36	올바르지 않은 인수가 제공됨
40	큐 관리자 사용불가능
49	큐 관리자 정지 중
69	저장영역 사용 불가능
71	예상치 못한 오류
72	큐 관리자 이름 오류
85	트랜잭션을 알 수 없음

관련 명령

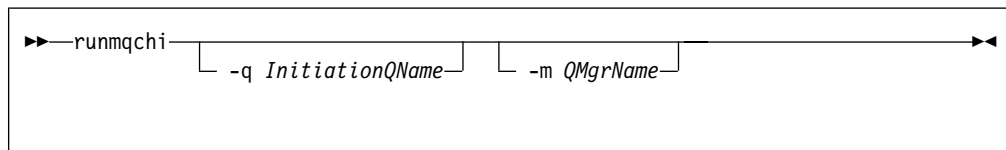
dspmqtrn 준비된 트랜잭션 목록 표시

runmqchi(채널 시작기 실행)

목적

채널 시작기 프로세스를 실행하려면 **runmqchi** 명령을 사용하십시오. 이 명령의 사용에 대한 자세한 정보는 *MQSeries* 상호통신을 참조하십시오.

구문



선택적 매개변수

-q *InitiationQName*

이 채널 시작기가 처리할 이니시에이션 큐의 이름을 지정합니다. 지정하지 않을 경우, SYSTEM.CHANNEL.INITQ가 사용됩니다.

-m *QMgrName*

이니시에이션 큐가 존재하는 큐 관리자의 이름을 지정합니다. 이름을 생략하면, 디폴트 큐 관리자가 사용됩니다.

리턴 코드

- 0** 명령이 정상적으로 완료됨
- 10** 명령이 예상치 못한 결과로 완료됨
- 20** 처리 중에 오류가 발생함

리턴 코드 10 또는 20을 야기하는 오류가 발생하면, 채널이 오류 메시지에 대해 연관되는 큐 관리자 오류 로그를 검토해야 합니다. 또한, \$SYSTEM 오류 로그도 검토해야 합니다. 채널이 큐 관리자와 연관되기 전에 발생하는 문제점은 이 로그에 기록되기 때문입니다. 오류 로그에 대한 자세한 정보는 217 페이지의 『오류 로그』를 참조하십시오.

runmqchl(채널 실행)

목적

송신자(SDR) 또는 요청자(RQSTR)를 실행하려면 **runmqchl** 명령을 사용하십시오.

채널은 동기식으로 수행됩니다. 채널을 정지하려면, MQSC 명령 STOP CHANNEL을 발행하십시오.

구문

```

▶▶—runmqchl— -c ChannelName —————▶▶
                        └-m QMgrName┘
  
```

필수 매개변수

-c ChannelName

실행할 채널의 이름을 지정합니다.

선택적 매개변수

-m QMgrName

이 채널이 연관되는 큐 관리자의 이름을 지정합니다. 이름을 지정하지 않으면, 디폴트 큐 관리자가 사용됩니다.

리턴 코드

- 0** 명령이 정상적으로 완료됨
- 10** 명령이 예상치 못한 결과로 완료됨
- 20** 처리 중에 오류가 발생함

리턴 코드 10 또는 20이 생성될 경우, 오류 메시지에 대해 연관된 큐 관리자의 오류 로그를 검토하십시오. 또한, \$SYSTEM 오류 로그도 검토해야 합니다. 채널이 큐 관리자와 연관되기 전에 발생하는 문제점은 이 로그에 기록되기 때문입니다.

runmqdlq(데드-레터 큐 핸들러 실행)

목적

데드-레터 큐에서 메시지를 모니터하고 핸들링할 수 있는 유틸리티인 데드-레터 큐(DLQ) 핸들러를 시작하려면 **runmqdlq** 명령을 사용하십시오.

데드-레터 큐 핸들러는 메시지를 선택하고 그 메시지에 대해 수행될 조치를 정의할 수 있는 규칙 세트를 지정하여 선택된 메시지에 대해 다양한 조치를 수행하는 데 사용할 수 있습니다.

runmqdlq 명령은 명령이 처리될 때 SYS\$INPUT에서 입력을 취하고, 결과와 요약은 보고서에 넣어져서 SYS\$OUTPUT으로 전송됩니다.

키보드에서 SYS\$INPUT을 취할 경우, 대화식으로 **runmqdlq** 규칙을 입력하면 됩니다.

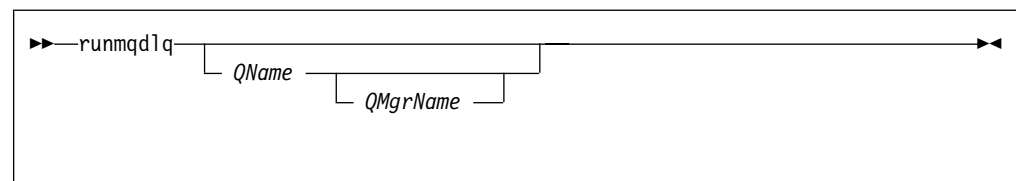
파일에서 입력의 경로를 재지정하여, 규칙 테이블을 지정된 큐에 적용할 수 있습니다. 규칙 테이블에는 최소한 하나의 규칙이 포함되어야 합니다.

파일에서 SYS\$INPUT 경로를 재지정하지 않고 포그라운드 모드에서 DLQ 핸들러를 사용하면, (규칙 테이블) DLQ 핸들러는 다음을 수행합니다.

- 키보드에서 입력을 읽습니다.
- end_of_file(ctrl-Z) 문자를 수신할 때까지 이름 지정된 큐의 처리를 시작하지 않습니다.

규칙 테이블과 이 테이블을 구성하는 방법에 대한 자세한 정보는 111 페이지의 『DLQ 핸들러 규칙 테이블』을 참조하십시오.

구문



선택적 매개변수

주석 줄과 줄 결합에 대한 MQSC 규칙은 DLQ 핸들러 입력 매개변수에도 적용됩니다.

QName

처리할 큐의 이름을 지정합니다.

runmqdlq

이름을 지정하지 않으면, 로컬 큐 관리자에 대해 정의된 데드-레터 큐가 사용됩니다. 하나 이상의 공백(' ')이 사용되면, 로컬 큐 관리자의 데드-레터 큐가 명확하게 지정됩니다.

DLQ 핸들러는 특수 처리를 위한 데드-레터 큐의 특정 메시지를 선택하는데 사용할 수 있습니다. 예를 들어, 메시지 경로를 다른 데드-레터 큐로 재지정할 수 있습니다. 그러면 DLQ 핸들러의 다른 인스턴스에 대한 후속 처리에서는 다른 규칙 테이블에 따라 메시지를 처리합니다.

QMgrName

처리될 큐를 소유하는 큐 관리자의 이름을 지정합니다.

이름을 지정하지 않으면, 설치시 디폴트 큐 관리자가 사용됩니다. 하나 이상의 공백(' ')이 사용되면, 이 설치에 대한 디폴트 큐 관리자가 명확하게 지정됩니다.

runmqfm(Failover 모니터 시작)

목적

OpenVMS 노드에서 Failover 모니터를 시작하려면 **runmqfm** 명령을 사용하십시오. Failover 모니터는 **runmqfm** 명령이 실행되는 OpenVMS 노드에서 실행됩니다.

구문

```
▶▶—runmqfm— -m QMgrName— [ -d ] ▶▶
```

필수 매개변수

-m *QMgrName*

runmqfm 명령이 시작될 큐 관리자의 이름을 지정합니다. *QMgrName* 이름에 대해 지원되는 최대 길이는 25자입니다.

선택적 매개변수

-d 추가 디버그 정보가 로그 파일에 기록될 것임을 지정합니다.

리턴 코드

- 0** 명령이 정상적으로 완료됨
- 10** 명령이 예상치 못한 결과로 완료됨
- 20** 처리 중에 오류가 발생함

예제

다음 예는 testqm 큐 관리자에 대해 Failover 모니터를 시작하고 디버그 정보를 test.log 로그 파일에 기록합니다.

```
runmqfm -m "testqm" -d > test.log
```

관련 명령

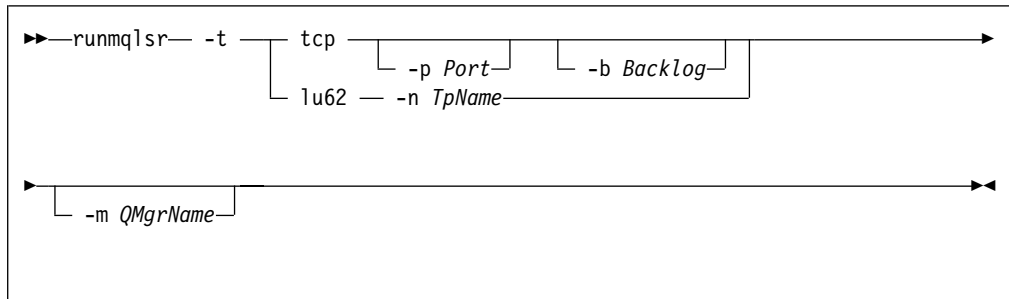
failover Failover 세트 관리

runmqlsr(리스너 실행)

목적

runmqlsr(리스너 실행) 명령은 리스너 프로세스를 실행합니다.

구문



필수 매개변수

-t 사용될 전송 프로토콜을 지정합니다.

tcp TCP/IP

lu62 SNA LU 6.2(이 매개변수를 사용하는 방법에 대한 최신 정보는 sys\$help:mqseries0510.release_notes의 릴리스 정보를 참조하십시오).

선택적 매개변수

-p Port

TCP/IP의 포트 번호. 이 플래그는 TCP 및 UDP에 대해 유효합니다. 값을 지정하지 않으면, 값은 큐 관리자 구성 파일이나 프로그램의 디폴트로부터 취해집니다. 디폴트 값은 1414입니다.

-n TpName

LU 6.2 트랜잭션 프로그램 이름. 이 플래그는 LU 6.2 전송 프로토콜에 대해서만 유효합니다. 값을 지정하지 않으면, 값은 큐 관리자 구성 파일로부터 취해집니다. 값이 제공되지 않으면, 명령은 실패합니다.

-m QMgrName

큐 관리자의 이름을 지정합니다. 어떤 이름도 지정하지 않으면, 명령은 디폴트 큐 관리자에 대해 작동합니다.

-b Backlog

리스너가 지원하는 동시 연결 요청 수를 지정합니다. 디폴트 값과 추가 정보에 대해서는 198 페이지의 『LU62 및 TCP 스탠자』를 참조하십시오.

리턴 코드

- 0 명령이 정상적으로 완료됨
- 10 명령이 예상치 못한 결과로 완료됨
- 20 처리 중에 오류가 발생함

예제

다음 명령은 TCP/IP 프로토콜을 사용하여 디폴트 큐 관리자에 대해 리스너를 실행합니다. 다음 명령은 리스너가 포트 4321을 사용해야 함을 지정합니다.

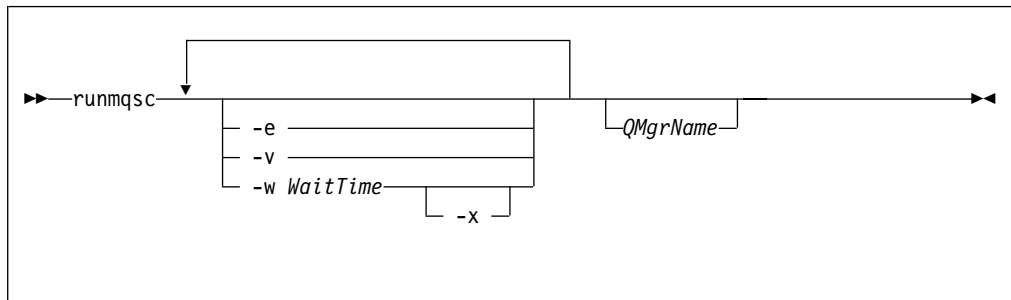
```
runmqtsr -t tcp -p 4321
```

runmqsc(MQSeries 명령 실행)

목적

큐 관리자에 대해 MQSC 명령을 발행하려면 **runmqsc** 명령을 사용하십시오. MQSC 명령은 로컬 큐 오브젝트 정의, 변경 또는 삭제와 같은 관리 작업을 수행할 수 있도록 합니다. MQSC 명령 및 구문은 *MQSeries Application Programming Guide*에 설명되어 있습니다.

구문



설명

runmqsc 명령은 다음 세 가지 모드에서 실행할 수 있습니다.

확인 모드

MQSC 명령이 확인되지만 실제로 실행되지는 않습니다. 각 명령의 성공 또는 실패를 나타내는 출력 보고서가 생성됩니다. 이 모드는 로컬 큐 관리자에서만 사용할 수 있습니다.

직접 모드

MQSC 명령이 로컬 큐 관리자에 직접 송신됩니다.

간접 모드

MQSC 명령은 리모트 큐 관리자에서 실행됩니다. 이 명령들은 리모트 큐 관리자의 명령 큐에 넣어지고(put) 그 순서대로 실행됩니다. 명령 보고서가 로컬 큐 관리자에 리턴됩니다.

주: 리모트 큐 관리자를 실행하는 사용자 ID는 로컬로 존재해야 하며 올바른 권한을 가지고 있어야 합니다.

runmqsc 명령은 SYS\$INPUT에서 입력을 취합니다. 명령이 처리될 때, 결과 및 요약이 보고서에 넣어져 SYS\$OUTPUT에 송신됩니다.

키보드에서 SYS\$INPUT을 취하여, 대화식으로 MQSC 명령을 입력할 수 있습니다.

파일에서 입력의 경로를 재지정하여, 파일에 있는 자주 사용되는 명령 순서를 실행할 수 있습니다. 또한 출력 보고서의 경로를 파일로 재지정할 수도 있습니다.

선택적 매개변수

- e MQSC 명령에 대한 소스 텍스트가 보고서에 복사되지 않도록 합니다. 이는 대화식으로 명령을 입력할 경우에 유용합니다.
- v 확인 모드를 지정합니다. 이는 조치를 수행하지 않고 지정된 명령을 확인합니다. 이 모드는 로컬로만 사용할 수 있습니다. -w 및 -x 플래그가 동시에 지정된 경우에는 무시됩니다.

-w WaitTime

간접 모드를 지정합니다. 즉 MQSC 명령이 다른 큐 관리자에서 실행되도록 합니다. 이에 대해 필수 채널 및 트랜스미션 큐를 설정해야 합니다.

WaitTime

runmqsc가 응답을 기다리는 시간을 초 단위로 지정합니다. 이후에 수신된 응답은 제거되고, MQSC 명령은 계속 실행됩니다. 1 - 999 999 초 사이의 시간을 지정하십시오.

각 명령은 이스케이프 PCF로서 타겟 큐 관리자의 명령 큐(SYSTEM.ADMIN.COMMAND.QUEUE)로 송신됩니다.

응답은 SYSTEM.MQSC.REPLY.QUEUE 큐에서 수신되고 결과는 보고서에 추가됩니다. 이것은 로컬 큐나 모델 큐로 정의됩니다.

디폴트 큐 관리자에서 간접 모드 조작이 수행됩니다.

이 플래그는 -v 플래그를 지정할 경우 무시됩니다.

- x 타겟 큐 관리자가 MVS/ESA 환경에서 실행됨을 지정합니다. 이 플래그는 간접 모드에서만 적용됩니다. -w 플래그도 지정해야 합니다. 간접 모드에서, MQSC 명령은 MVS/ESA용 MQSeries 명령 큐에 적합한 양식으로 작성됩니다.

QMgrName

MQSC 명령이 실행될 타겟 큐 관리자의 이름을 지정합니다. 생략할 경우, MQSC 명령은 디폴트 큐 관리자에서 실행됩니다.

리턴 코드

- 0 MQSC 명령 파일이 성공적으로 처리됨
- 10 MQSC 명령 파일이 오류와 함께 처리됨--보고서에 실패한 명령에 대한 원인이 있습니다.
- 20 오류--MQSC 명령 파일이 실행되지 않음

예제

1. OpenVMS 명령 프롬프트에 다음 명령을 입력하십시오.

```
runmqsc
```

이제, OpenVMS 명령 프롬프트에 직접 MQSC 명령을 입력할 수 있습니다. 큐 관리자 이름이 지정되지 않았습니다. 따라서 MQSC 명령은 디폴트 큐 관리자에서 처리됩니다.

2. MQSC 명령이 확인만 되도록 지정하려면 이 명령을 사용하십시오.

```
runmqsc -v BANK < DKA0:[USERS]COMMFILE.IN
```

이 명령은 DKA0:[USERS] 디렉토리에서 MQSC 명령 파일 COMMFILE.IN을 확인합니다. 큐 관리자 이름은 BANK입니다. 출력은 현재 창에 표시됩니다.

3. 이 명령은 MQSC 명령 파일 MQS_ROOT:[MQM.MQSC]MQSCFILE.IN을 디폴트 큐 관리자에 대해 실행합니다.

```
runmqsc < MQS_ROOT:[MQM.MQSC]MQSCFILE.IN > MQS_ROOT:[MQM.MQSC]MQSCFILE.OUT
```

이 예에서, 출력은 MQS_ROOT:[MQM.MQSC]MQSCFILE.OUT 파일로 보내집니다.

runmqtmc(클라이언트 트리거 모니터 시작)

목적

클라이언트에 대해 트리거 모니터를 실행하려면 **runmqtmc** 명령을 사용하십시오. 트리거 모니터 사용에 대한 추가 정보는 *MQSeries Application Programming Guide*를 참조하십시오.

주: 이 명령은 오직 OpenVMS, OS/2 및 AIX 클라이언트에서만 사용 가능합니다.

구문

```

▶▶—runmqtmc ————┬── -m QMgrName ┬── -q InitiationQName ───▶▶

```

선택적 매개변수

-m *QMgrName*

클라이언트 트리거 모니터가 작동하는 큐 관리자의 이름을 지정합니다. 생략할 경우, 클라이언트 트리거 모니터는 디폴트 큐 관리자에서 작동됩니다.

-q *InitiationQName*

처리될 이니시에이션 큐 이름을 지정합니다. 생략하면, SYSTEM.DEFAULT.INITIATION.QUEUE가 사용됩니다.

리턴 코드

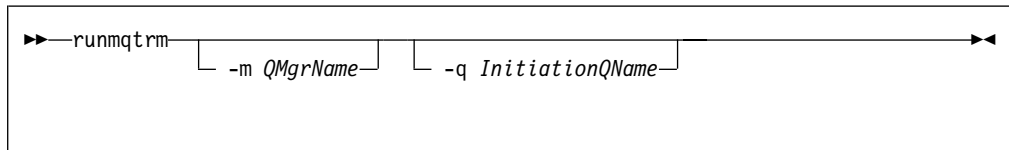
- 0** 사용되지 않습니다. 클라이언트 트리거 모니터는 연속으로 실행되도록 설계되어 있으므로 종료되지 않습니다. 값은 예약되어 있습니다.
- 10** 클라이언트 트리거 모니터가 오류로 인터럽트되었습니다.
- 20** 오류--클라이언트 트리거 모니터가 실행되지 않습니다.

runmqtrm(트리거 모니터 시작)

목적

트리거 모니터를 실행하려면 **runmqtrm** 명령을 사용하십시오. 트리거 모니터 사용에 대한 추가 정보는 *MQSeries Application Programming Guide*를 참조하십시오.

구문



선택적 매개변수

-m *QMgrName*

트리거 모니터가 작동하는 큐 관리자의 이름을 지정합니다. 생략할 경우, 트리거 모니터는 디폴트 큐 관리자에서 작동됩니다.

-q *InitiationQName*

처리될 이니시에이션 큐 이름을 지정합니다. 생략하면, SYSTEM.DEFAULT.INITIATION.QUEUE가 사용됩니다.

리턴 코드

- 0** 사용되지 않습니다. 트리거 모니터는 연속으로 실행되도록 설계되어 있으므로 종료되지 않습니다. 값은 예약되어 있습니다.
- 10** 트리거 모니터가 오류로 인터럽트되었습니다.
- 20** 오류--트리거 모니터가 실행되지 않습니다.

setmqaut(권한 설정/재설정)

목적

오브젝트나 오브젝트 클래스에 대한 권한을 변경하려면 **setmqaut** 명령을 사용하십시오. 권한은 여러 개의 기본 원칙이나 그룹에 대해 부여되거나 취소될 수 있습니다.

구문

```
setmqaut -m QMgrName -n ObjectName -t ObjectType
```

```

  -s ServiceComponent
  -p PrincipalName
  -g GroupName

```

```

MQI authorizations
Context authorizations
Administration authorizations
Generic authorizations

```

MQI 권한:

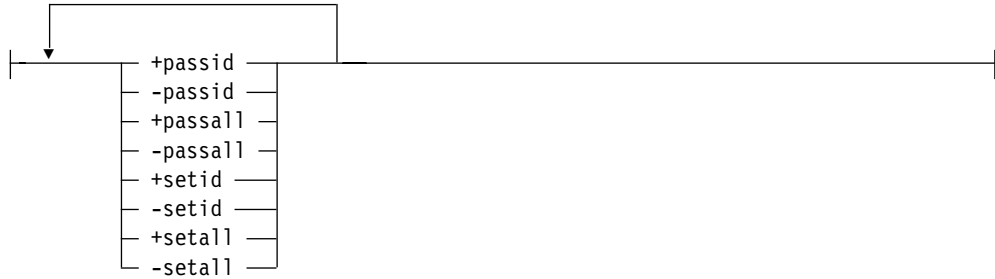
```

+get
-get
+browse
-browse
+put
-put
+inq
-inq
+set
-set
+connect
-connect
+altusr
-altusr

```

컨텍스트 권한:

setmqaut



관리 권한:



일반 권한:



설명

이 명령을 사용하여 권한을 설정(즉, 조작을 수행하기 위해 사용자 그룹 또는 기본 원칙 사용권한 제공)하고, 권한을 재설정(즉, 조작을 수행하기 위해 사용권한 제거)할 수 있습니다. 권한이 적용되는 사용자 그룹 및 기본 원칙과 큐 관리자, 오브젝트 유형 및 오브젝트의 이름을 지정해야 합니다. 하나의 명령으로 여러 개의 그룹 및 기본 원칙을 지정할 수 있습니다.

주의: 하나의 기본 원칙에 대해 권한 세트를 지정할 경우, 동일한 1차 그룹의 모든 기본 원칙에 대해 동일한 권한이 제공됩니다.

제공될 수 있는 권한은 다음과 같은 범주로 나누어집니다.

- MQI 호출을 발행하는 권한
- MQI 컨텍스트에 대한 권한

- 관리 작업에 대한 명령을 발행하는 권한
- 일반 권한

변경될 각 권한이 명령의 일부로 권한 목록에 지정됩니다. 목록에 있는 각 항목의 문자열 앞에는 '+' 또는 '-'가 붙습니다. 예를 들어 권한 목록에 +put이 포함되면 큐에 대해 MQPUT 호출을 발행할 수 있는 권한이 부여됩니다. 권한 목록에 -put이 포함되면 MQPUT 호출을 발행할 수 있는 권한이 제거됩니다.

권한은 충돌하지 않는 상태에서 임의 순서로 지정할 수 있습니다. 예를 들어, allmqi와 set을 함께 지정하면 충돌이 발생합니다.

사용자에게 필요한 수만큼 그룹 또는 권한을 하나의 명령으로 지정할 수 있습니다.

사용자 ID가 여러 그룹의 구성원일 경우, 적용되는 권한은 사용자 ID가 속한 각 그룹의 권한들을 합한 것입니다.

필수 매개변수

-m *QMGrName*

권한이 변경될 오브젝트의 큐 관리자 이름을 지정합니다. 이름은 48자 이하로 지정할 수 있습니다.

-n *ObjectName*

권한이 변경될 오브젝트의 이름을 지정합니다.

이것은 오브젝트가 큐 관리자 자체가 아닐 경우 필수 매개변수입니다. 큐 관리자, 큐 또는 프로세스의 이름을 지정해야 하며, 일반 이름을 사용해서는 안됩니다.

-t *ObjectType*

권한이 변경될 오브젝트의 유형을 지정합니다.

가능한 값은 다음과 같습니다.

- **q** 또는 **queue**
- **procs** 또는 **process**
- **qmgr**

선택적 매개변수

-p *PrincipalName*

권한이 변경될 기본 원칙의 이름을 지정합니다.

최소한 하나의 기본 원칙이나 그룹이 있어야 합니다.

-g *GroupName*

권한이 변경될 사용자 그룹을 나타내는 권한 ID의 이름을 지정합니다. 여러 개의 권한 ID 이름을 지정할 수 있으며, 각 이름 앞에는 -g 플래그를 붙여야 합니다.

-s ServiceComponent

이 매개변수는 설치가능 권한 서비스를 사용할 경우에만 적용되며, 그렇지 않을 경우에는 무시됩니다.

설치가능 권한 서비스가 지원될 경우, 이 매개변수는 권한이 적용되는 권한 부여 서비스의 이름을 지정합니다. 이 매개변수는 선택적입니다. 이를 지정하지 않으면, 서비스의 첫 번째 설치 가능 구성요소에 대한 권한이 갱신됩니다.

Authorizations

부여되거나 제거될 권한을 지정합니다. 목록의 각 항목 앞에는 권한 부여를 나타내는 '+'가 붙거나, 권한 제거를 나타내는 '-'가 붙습니다. 예를 들어, MQI에서 MQPUT 호출을 발행하는 권한을 부여하려면 목록에 +put을 지정하십시오. MQPUT 호출을 발행하는 권한을 제거하려면 -put을 지정하십시오.

표17에서는 여러 오브젝트 유형에 제공될 수 있는 권한을 보여줍니다.

표 17. 여러 오브젝트 유형에 대한 권한 지정

권한	큐	프로세스	Qmgr	이름 목록
all	✓	✓	✓	✓
alladm	✓	✓	✓	✓
allmqi	✓	✓	✓	✓
altusr			✓	
browse	✓			
chg	✓	✓	✓	✓
clr	✓			
connect			✓	
crt	✓	✓	✓	✓
dlt	✓	✓	✓	✓
dsp	✓	✓	✓	✓
get	✓			
inq	✓	✓	✓	✓
passall	✓			
passid	✓			
put	✓			
set	✓	✓	✓	
setall	✓		✓	
setid	✓		✓	

MQI 호출에 대한 권한

altusr 메시지에서 대체 사용자 ID를 사용하십시오.

대체 사용자 ID에 대한 자세한 정보는 *MQSeries Application Programming Guide*를 참조하십시오.

browse

BROWSE 옵션으로 MQGET 호출을 발행하여 큐에서 메시지를 검색합니다.

connect

MQCONN 호출을 발행하여 지정된 큐 관리자에 응용프로그램을 연결합니다.

get MQGET 호출을 발행하여 큐에서 메시지를 검색합니다.

inq MQINQ 호출을 발행하여 특정 큐에 대해 조회합니다.

put MQPUT 호출을 발행하여 특정 큐에 메시지를 넣습니다(put).

set MQSET 호출을 발행하여 MQI에서 큐의 속성을 설정합니다.

주: 여러 옵션에 대해 큐를 열 경우 각 옵션마다 권한이 부여되어야 합니다.

컨텍스트에 대한 권한**passall**

지정된 큐의 모든 컨텍스트를 전달합니다. 원래 요청에서 모든 컨텍스트 필드가 복사됩니다.

passid 지정된 큐의 ID 컨텍스트를 전달합니다. ID 컨텍스트는 요청의 ID 컨텍스트와 같습니다.

setall 지정된 큐의 모든 컨텍스트를 설정합니다. 이것은 특수 시스템 유틸리티에서 사용합니다.

setid 지정된 큐의 ID 컨텍스트를 설정합니다. 이것은 특수 시스템 유틸리티에서 사용합니다.

명령에 대한 권한

chg 지정된 오브젝트의 속성을 변경합니다.

clr 지정된 큐를 지웁니다(PCF 큐 지우기 명령만).

crt 지정된 유형의 오브젝트를 작성합니다.

dlt 지정된 오브젝트를 삭제합니다.

dsp 지정된 오브젝트의 속성을 표시합니다.

일반 조작에 대한 권한

all 오브젝트에 적용 가능한 모든 조작을 사용합니다.

alladm

오브젝트에 적용 가능한 모든 관리 조작을 수행합니다.

allmqi 오브젝트에 적용 가능한 모든 MQI 호출을 사용합니다.

setmqaut

리턴 코드

0	명령이 정상적으로 완료됨
36	올바르지 않은 인수가 제공됨
40	큐 관리자 사용불가능
49	큐 관리자 정지 중
69	저장영역 사용 불가능
71	예상치 못한 오류
72	큐 관리자 이름 오류
133	알 수 없는 오브젝트 이름
145	예상치 못한 오브젝트 이름
146	오브젝트 이름 누락
147	오브젝트 유형 누락
148	올바르지 않은 오브젝트 유형
149	엔티티 이름 누락
150	권한 스펙 누락
151	올바르지 않은 권한 스펙

예제

1. 다음 예는 큐 관리자 saturn.queue.manager의 orange.queue 큐를 권한이 부여되는 오브젝트로 지정하는 명령을 보여줍니다.

```
setmqaut -m "saturn.queue.manager" -n "orange.queue" -t queue -g "tango" +inq +alladm
```

사용자 그룹 tango에 권한이 부여되고 연관된 권한 목록은 사용자 그룹 tango를 다음과 같은 상태로 지정합니다.

- MQINQ 호출을 발행할 수 있습니다.
 - 해당 오브젝트에 대한 모든 관리 조작을 수행하는 권한이 있습니다.
2. 다음 예에서 권한 목록은 사용자 그룹 foxy를 아래와 같이 지정합니다.
 - MQI로부터 지정된 큐에 대한 호출을 발행할 수 없습니다.
 - 지정된 큐에 대한 모든 관리 조작을 수행하는 권한이 있습니다.

```
setmqaut -m "saturn.queue.manager" -n "orange.queue" -t queue -g "foxy" -allmqi +alladm
```

3. 다음 예에서 권한 목록은 사용자 그룹 waltz가 큐 관리자 saturn.queue.manager를 작성하고 삭제하는 권한을 가지도록 지정합니다.


```
setmqaut -m "saturn.queue.manager" -t qmgr -g "waltz" +crt +dlt
```

setmqaut

관련 명령

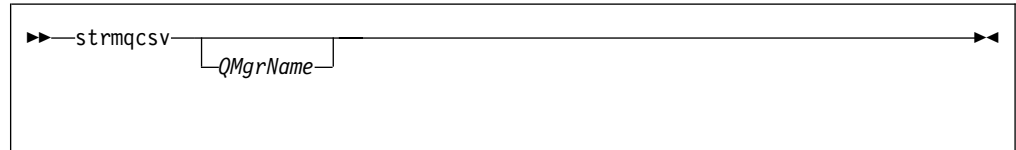
dspmqaut 권한 표시

strmqcsv(명령 서버 시작)

목적

지정된 큐 관리자에 대한 명령 서버를 시작하려면 **strmqcsv** 명령을 사용하십시오. 그러면 MQSeries가 명령 큐에 송신되는 명령을 처리할 수 있게 됩니다.

구문



선택적 매개변수

QMGrName

명령 서버가 시작될 큐 관리자의 이름을 지정합니다.

리턴 코드

- 0 명령이 정상적으로 완료됨
- 10 명령이 예상치 못한 결과로 완료됨
- 20 처리 중에 오류가 발생함

예제

다음 명령은 earth 큐 관리자에 대한 명령 서버를 시작합니다.

```
strmqcsv "earth"
```

관련 명령

- endmqcsv** 명령 서버 종료
- dspmqcsv** 명령 서버의 상태 표시

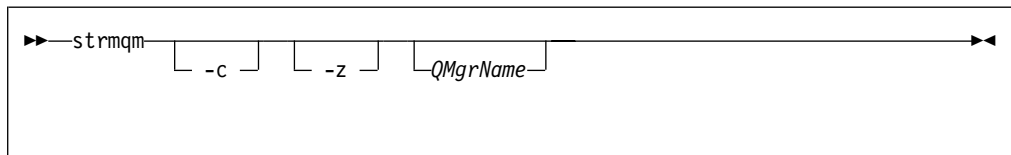
strmqm(큐 관리자 시작)

목적

로컬 큐 관리자를 시작하려면 **strmqm** 명령을 사용하십시오.

주: **strmqm** 명령을 사용하거나 다른 제어 명령을 사용하려면, 먼저 큐 관리자를 시작하여 성공적으로 수행될 수 있도록 그 이전에 최종 재시동 이후로 한번 **mqs_startup**이 실행되어야 합니다.

구문



선택적 매개변수

-c 큐 관리자를 시작하고, 디폴트 및 시스템 오브젝트를 재정의한 후 큐 관리자를 정지합니다(큐 관리자에 대한 디폴트 및 시스템 오브젝트는 **crtmqm** 명령에 의해 초기에 작성됩니다). 큐 관리자에 속하는 기존 시스템 및 디폴트 오브젝트는 이 플래그를 지정할 경우에 바뀝니다.

QMGrName

시작할 로컬 큐 관리자의 이름을 지정합니다. 생략하면, 디폴트 큐 관리자가 시작됩니다.

-z 오류 메시지를 억제합니다.

이 플래그는 MQSeries 내에서 원하지 않는 오류 메시지를 억제하기 위해 사용됩니다. 이 플래그를 사용하면 정보가 유실되므로, 명령행에 명령을 입력할 경우에는 이를 사용하지 말아야 합니다.

리턴 코드

- 0 큐 관리자가 시작됨
- 3 큐 관리자가 작성됨
- 5 큐 관리자 실행 중
- 16 큐 관리자가 없음
- 23 로그 사용불가능
- 49 큐 관리자 정지 중
- 69 저장영역 사용 불가능
- 71 예상치 못한 오류
- 72 큐 관리자 이름 오류

100 로그 위치가 올바르지 않음

예제

다음 명령은 큐 관리자 account에서 시작됩니다.

```
strmqm "account"
```

관련 명령

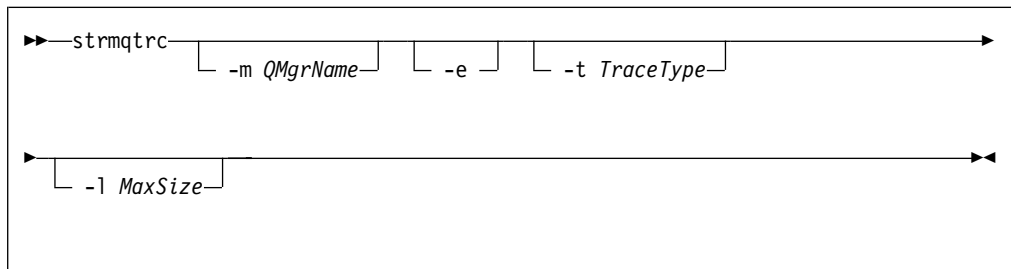
crtmqm	큐 관리자 작성
dltmqm	큐 관리자 삭제
endmqm	큐 관리자 종료

strmqtrc(MQSeries 추적 시작)

목적

추적을 사용할 수 있게 하려면 **strmqtrc** 명령을 사용하십시오. 이 명령은 추적 사용 여부에 따라 실행될 수 있습니다. 추적이 이미 사용 가능하면, 사실상 추적 옵션은 최근 명령 실행에 지정된 옵션으로 수정됩니다.

구문



설명

서로 다른 레벨의 추적 세부사항을 요청할 수 있습니다. 지정하는 각 플로우 추적 유형 값에 대해(-t all을 포함하여), 특정 추적 유형에 대해 -t params 또는 -t detail을 지정하십시오. 해당되는 추적 유형에 대해 디폴트 세부사항 추적이 생성됩니다.

이 명령에 의해 생성되는 추적 데이터의 예는 222 페이지의 『MQSeries 추적 사용』을 참조하십시오.

선택적 매개변수

-m *QMgrName*

추적될 큐 관리자의 이름을 지정합니다.

큐 관리자 이름 및 -m 플래그는 -e 플래그와 같은 명령에서 지정할 수 있습니다. 여러 개의 추적 스펙이 주어진 추적 엔티티에 적용될 경우, 실제 추적에는 지정된 모든 옵션이 포함됩니다.

-e 플래그를 지정하지 않는 한 -m 플래그와 큐 관리자 이름을 생략하면 오류가 발생합니다.

-e 이 플래그를 지정할 경우, 초기 추적이 요청됩니다. 결국, 큐 관리자 작성 또는 시동을 추적할 수 있습니다. 여기에는 속해 있는 MQSeries 구성요소를 프로세스에서 인식하기 전에 기록되는 추적 정보가 포함됩니다. 이 플래그를 지정하면 임의 큐 관리자의 임의 구성요소에 속하는 임의 프로세스는 해당 초기 처리를 추적합니다. 이 플래그를 지정하지 않을 경우 초기 추적을 수행하지 않는 것이 디폴트가 됩니다.

-t TraceType

처리 중에 추적할 수 있는 지점을 정의합니다. 이 플래그를 생략하면, 모든 추적 지점이 사용되어 전체 추적이 생성됩니다.

또한, 다음 목록에서 하나 이상의 옵션을 제공할 수도 있습니다.

주: 여러 추적 유형을 제공할 경우, 각각은 반드시 해당되는 고유 -t 플래그를 가지고 있어야 합니다. 각각 연관되는 올바른 추적 유형을 수반하고 있으면 몇 개의 -t 플래그도 지정할 수 있습니다.

여러 -t 플래그에 대해 같은 추적 유형을 지정하는 것은 오류가 아닙니다.

all	시스템의 모든 추적 지점에 대해 데이터를 출력합니다. 이것은 또한 -t 플래그를 지정하지 않을 경우에 디폴트이기도 합니다.
api	MQI 및 주요 큐 관리자 구성요소와 연관되는 추적 지점에 대한 데이터를 출력합니다.
comms	통신 네트워크를 거쳐 이동하는 데이터와 연관되는 추적 지점에 대한 데이터를 출력합니다.
csflows	공동 서비스에서 처리 플로우와 연관되는 추적 지점에 대한 데이터를 출력합니다.
lqmflows	로컬 큐 관리자에서 처리 플로우와 연관되는 추적 지점에 대한 데이터를 출력합니다.
remoteflows	통신 구성요소에서 처리 플로우와 연관되는 추적 지점에 대한 데이터를 출력합니다.
otherflows	기타 구성요소에서 처리 플로우와 연관되는 추적 지점에 대한 데이터를 출력합니다.
csdata	공동 서비스에서 내부 데이터 버퍼와 연관되는 추적 지점에 대한 데이터를 출력합니다.
lqmdata	로컬 큐 관리자에서 내부 데이터 버퍼와 연관되는 추적 지점에 대한 데이터를 출력합니다.
remotedata	통신 구성요소에서 내부 데이터 버퍼와 연관되는 추적 지점에 대한 데이터를 출력합니다.
otherdata	기타 구성요소에서 내부 데이터 버퍼와 연관되는 추적 지점에 대한 데이터를 출력합니다.
versiondata	실행 중인 MQSeries 버전과 연관되는 추적 지점에 대한 데이터를 출력합니다.
commentary	MQSeries 구성요소에서 주석과 연관되는 추적 지점에 대한 데이터를 출력합니다.

-l MaxSize

MaxSize 값은 최대 추적 파일(AMQnnnnn.TRC) 크기(백만 바이트 단위)를 나타냅니다. 예를 들어, MaxSize 1을 지정할 경우, 추적 크기는 1백만 바이트로 제한됩니다.

strmqtrc

추적 파일이 지정된 최대값에 도달하면, AQnnnn.TRC에서 AMQnnnn.TRS로 이름이 바뀌고 새로운 AMQnnnn.TRC 파일이 시작됩니다. 모든 추적 파일은 최대 한계에 도달할 때 시작됩니다. AMQnnnn.TRS 파일의 이전 사본이 존재할 경우, 이것은 삭제됩니다.

리턴 코드

AMQ7024 올바르지 않은 인수를 명령에 제공할 경우에 이 메시지가 발행됩니다.
AMQ8304 최대 9개의 동시 추적이 이미 실행되고 있습니다.

예제

이 명령을 사용하면 QM1이라는 큐 관리자에 대해 공통 서비스와 로컬 큐 관리자에서 데이터 추적이 가능합니다.

```
strmqtrc -m QM1 -t csdata -t lqmdata
```

관련 명령

dspmqtrc 형식화된 추적 출력 표시
endmqtrc MQSeries 추적 종료

제3부 부록

부록A. Compaq OpenVMS용 MQSeries 요약

프로그램 및 부품 번호

- 5724-A38 Compaq OpenVMS용 MQSeries, Alpha 버전 5 릴리스 1, 부품 번호 0790997.

하드웨어 요구사항

최소 128 MB의 시스템 디스크 공간을 갖춘 Compaq Alpha 시스템을 MQSeries 서버로 사용할 수 있습니다.

소프트웨어 요구 사항

소프트웨어 요구 사항은 별도로 명시하지 않는 한 서버 및 클라이언트 Compaq OpenVMS 환경이 동일합니다.

최소 지원 레벨은 다음과 같습니다.

- Compaq OpenVMS Alpha 버전 7.2-1

연결

Compaq OpenVMS용 MQSeries는 다음과 같은 네트워크 프로토콜 및 하드웨어를 지원합니다.

네트워크 프로토콜:

- SNA LU6.2
- TCP/IP
- DECnet Phase V

이 외에 DECnet, TCP/IP, DIGITAL DECnet/SNA Gateway를 지원하는 모든 통신 하드웨어

DECnet 연결의 경우:

- OpenVMS 버전 7.2-1용 DECnetPLUS 버전 7.1

TCP/IP 연결의 경우:

- OpenVMS용 DIGITAL TCP/IP 서비스 AlphaV5.0.a 및 V5.1
- Process Software's TCPWare V5.4
- Process Software's Multinet V4.3

소프트웨어 요구 사항

SNA 연결의 경우: SNA APPC LU6.2 소프트웨어와 사용권을 설치해야 합니다. 알맞게 구성된 SNA 게이트웨이를 액세스할 수 있어야 합니다.

- DECnet SNA Gateway ST V1.3
- DECnet SNA LU6.2 API V2.4

보안

Compaq OpenVMS용 MQSeries는 Compaq OpenVMS용 MQSeries용 OAM(Object Authority Manager) 보안 기능을 사용합니다.

모든 MQSeries 자원은 VMS 권한 ID MQM을 사용하여 실행됩니다. 이 권한 ID는 MQSeries 설치시 작성되며 MQSeries 자원을 제어해야 하는 모든 사용자에게 이 자원 속성을 가진 권한 ID를 부여해야 합니다.

유지보수 기능

다음 요소를 갖춘 MQSeries 기능:

- **runmqsc** 명령행 인터페이스

호환성

Compaq OpenVMS Alpha용 MQSeries V5.1용 MQI는 버전 2.2.1.1을 실행하는 기존 응용프로그램과 호환가능합니다.

지원 컴파일러

프로그램은 C, C++, COBOL, Java를 사용하여 작성할 수 있습니다.

- C 프로그램에는 DEC C 컴파일러를 사용할 수 있습니다.
- C++ 프로그램에는 DEC C++ 컴파일러를 사용할 수 있습니다.
- COBOL 프로그램에는 DEC COBOL 컴파일러를 사용할 수 있습니다.
- Java 프로그램에는 Java 컴파일러를 사용할 수 있습니다.

언어 선택

제공되는 메시지 텍스트 파일은 OpenVMS 운영 체제에 고유한 7 비트 문자 세트로 인코딩되어 있습니다.

국제화

Compaq OpenVMS용 MQSeries는 큐 관리자 인스턴스가 작성될 때 CCSID를 지정할 수 있도록 합니다. 큐 관리자 CCSID의 디폴트는 819로 설정됩니다. Compaq OpenVMS용 MQSeries에서는 문자 세트가 큐 관리자의 구성된 CCSID로 변환될 수 있도록 지원합니다. Euro 문자를 지원하는 CCSID를 포함하여 Compaq OpenVMS용 MQSeries 큐 관리자에 지정할 수 있는 CCSID에 대한 자세한 정보는 *MQSeries Application Programming Reference*를 참조하십시오.

부록B. 시스템 디폴트

crtmqm 제어 명령을 사용하여 큐 관리자를 작성하면 시스템 오브젝트와 디폴트 오브젝트가 자동으로 작성됩니다.

- 시스템 오브젝트는 큐 관리자나 채널 조작에 필요한 MQSeries 오브젝트입니다.
- 디폴트 오브젝트는 한 오브젝트의 모든 속성을 정의합니다. 로컬 큐와 같은 오브젝트를 작성할 때 명확하게 지정하지 않은 모든 속성은 디폴트 오브젝트에서 상속됩니다.

표 18. 큐의 시스템 오브젝트 및 디폴트 오브젝트

오브젝트 이름	설명
SYSTEM.DEFAULT.ALIAS.QUEUE	디폴트 알리아스 큐
SYSTEM.DEFAULT.LOCAL.QUEUE	디폴트 로컬 큐
SYSTEM.DEFAULT.MODEL.QUEUE	디폴트 모델 큐
SYSTEM.DEFAULT.REMOTE.QUEUE	디폴트 리모트 큐
SYSTEM.DEAD.LETTER.QUEUE	샘플 데드-레터(미배달 메시지) 큐
SYSTEM.DEFAULT.INITIATION.QUEUE	디폴트 이니시에이션 큐
SYSTEM.CICS.INITIATION.QUEUE	디폴트 CICS 이니시에이션 큐
SYSTEM.ADMIN.COMMAND.QUEUE	관리 명령 큐. 리모트 MQSC 명령과 PCF 명령에 사용됩니다.
SYSTEM.MQSC.REPLY.QUEUE	MQSC 리플라이-투 큐. 리모트 MQSC 명령에 대한 응답으로 템포러리 다이나믹 큐를 작성하는 모델 큐.
SYSTEM.ADMIN.QMGR.EVENT	큐 관리자 이벤트에 대한 이벤트 큐
SYSTEM.ADMIN.PERFM.EVENT	성능 이벤트에 대한 이벤트 큐
SYSTEM.ADMIN.CHANNEL.EVENT	채널 이벤트에 대한 이벤트 큐
SYSTEM.CHANNEL.INITQ	채널 이니시에이션 큐
SYSTEM.CHANNEL.SYNCQ	채널의 동기화 데이터를 보유하는 큐
SYSTEM.CLUSTER.COMMAND.QUEUE	메시지를 저장소 큐 관리자로 전달하는 데 사용되는 큐
SYSTEM.CLUSTER.REPOSITORY.QUEUE	모든 저장소 정보를 저장하는 데 사용되는 큐
SYSTEM.CLUSTER.TRANSMIT.QUEUE	클러스터에 대한 모든 메시지의 트랜스미션 큐

표 19. 채널의 시스템 오브젝트 및 디폴트 오브젝트

오브젝트 이름	설명
SYSTEM.DEF.SENDER	디폴트 송신자 채널
SYSTEM.DEF.SERVER	디폴트 서버 채널
SYSTEM.DEF.RECEIVER	디폴트 수신자 채널
SYSTEM.DEF.REQUESTER	디폴트 요청자 채널
SYSTEM.DEF.SVRCONN	디폴트 서버 연결 채널

디폴트

표 19. 채널의 시스템 오브젝트 및 디폴트 오브젝트 (계속)

오브젝트 이름	설명
SYSTEM.DEF.CLNTCONN	디폴트 클라이언트 연결 채널
SYSTEM.AUTO.RECEIVER	동적 수신자 채널
SYSTEM.AUTO.SVRCONN	동적 서버 연결 채널
SYSTEM.DEF.CLUSRCVR	클러스터의 큐 관리자에 CLUSRCVR 채널을 작성할 때 지정하지 않은 모든 속성에 대해 디폴트 값을 제공하는 데 사용되는 클러스터의 디폴트 수신자 채널
SYSTEM.DEF.CLUSSDR	클러스터의 큐 관리자에 CLUSSDR 채널을 작성할 때 지정하지 않은 모든 속성에 대해 디폴트 값을 제공하는 데 사용되는 클러스터의 디폴트 송신자 채널

표 20. 이름 목록의 시스템 오브젝트 및 디폴트 오브젝트

오브젝트 이름	설명
SYSTEM.DEFAULT.NAMELIST	디폴트 이름 목록

표 21. 프로세스의 시스템 오브젝트 및 디폴트 오브젝트

오브젝트 이름	설명
SYSTEM.DEFAULT.PROCESS	디폴트 프로세스 정의

부록C. 디렉토리 구조

그림26에서는 특정 큐 관리자와 연관된 데이터 및 로그 디렉토리의 일반 레이아웃을 보여 줍니다. 표시된 디렉토리는 디폴트로 설치한 경우에 해당합니다. 설치 위치를 변경하면 파일과 디렉토리의 위치도 그에 따라 변경됩니다.

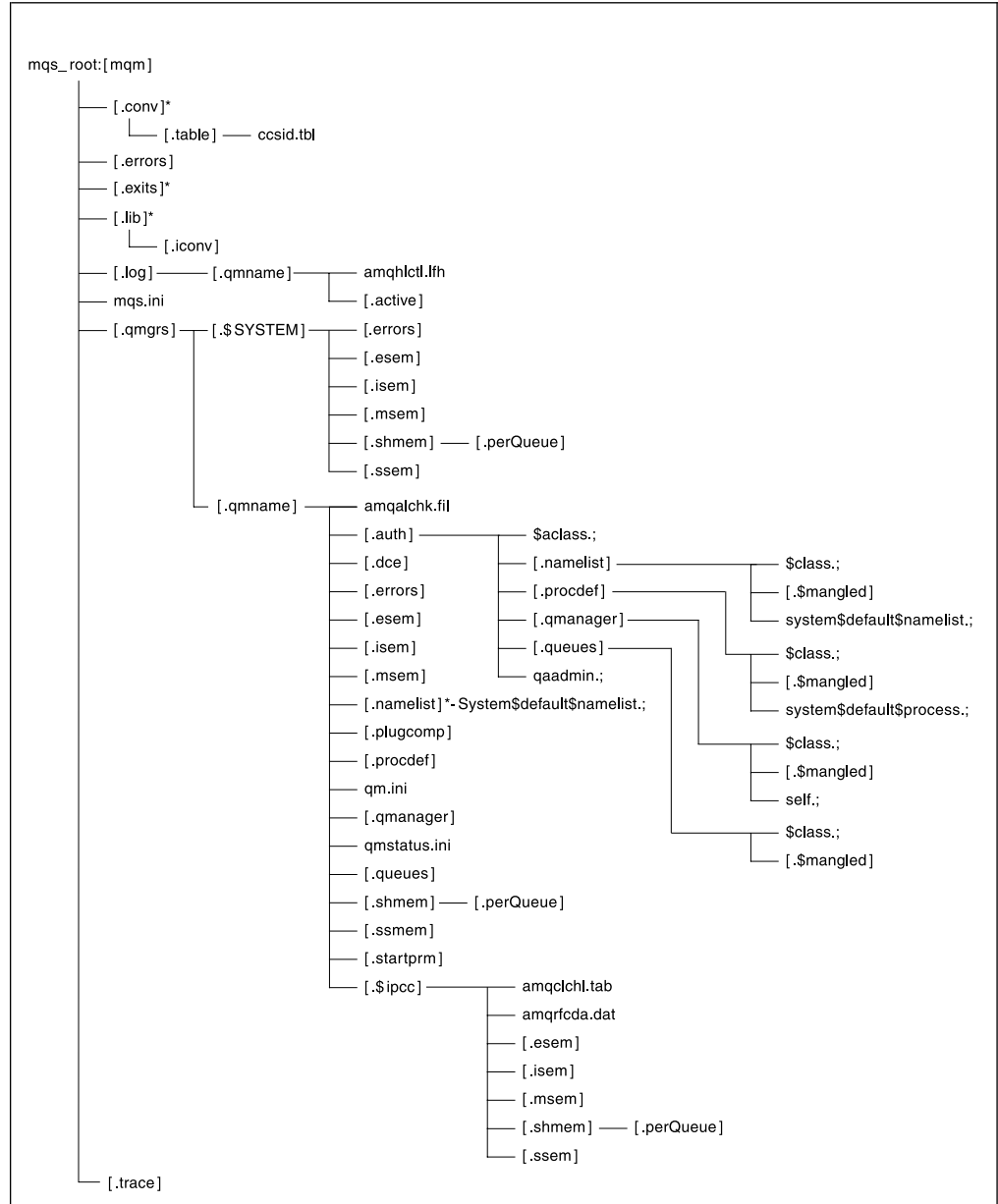


그림 26. 큐 관리자를 시작한 후의 디폴트 디렉토리 구조

그림26의 레이아웃은 큐 관리자를 일정 기간 동안 사용한 후의 MQSeries를 나타낸 것입니다. 실제 구조는 큐 관리자에 수행한 조작에 따라 다릅니다.

MQS_ROOT:[MQM]의 디렉토리와 파일

디폴트로 MQS_ROOT:[MQM]: 디렉토리에는 다음 디렉토리와 파일이 있습니다.

.conv 이 디렉토리에는 데이터 변환에 사용되는 모든 파일이 있습니다.

.table 이 디렉토리에는 ccsid.tbl 파일이 있습니다.

.errors

이 디렉토리에는 연산자 메시지 파일이 최근 날짜부터 오래된 날짜의 순서로 있습니다.

AMQERR01.LOG

AMQERR02.LOG

AMQERR03.LOG

.exits 사용자가 작성한 종료를 포함하기 위한 빈 디렉토리입니다.

.lib 이 디렉토리에는 .iconv 서브디렉토리가 있습니다. 서브디렉토리에는 모든 코드 세트 변환 테이블이 있습니다.

.iconv 코드 세트 변환 테이블(예: 002501B5.TBL을 44B031A8.TBL로 변환)을 포함하는 디렉토리

.log MQSeries를 설치하고 큐 관리자를 작성하여 시작한 후 일정 기간 동안 큐 관리자를 사용하면 이 디렉토리에 다음 서브디렉토리와 파일이 포함됩니다.

amqhlctl.lfh

로그 제어 파일

active 이 디렉토리에는 다음과 같이 번호가 매겨지는 로그 파일이 있습니다.

S0000000.LOG

S0000001.LOG

S0000002.LOG

... 등

mqs.ini

MQSeries 구성 파일

.qmgrs

각 큐 관리자의 이 디렉토리에는 `.$system` 서브디렉토리와 `.qmname` 서브디렉토리가 있습니다. `.$system` 디렉토리에는 MQSeries가 내부적으로 사용하는 디렉토리와 파일이 있습니다. `.qmname` 서브디렉토리에 대한 자세한 정보는 345 페이지의 『MQS_ROOT:[MQM.QMGRS.QMNAME] 서브디렉토리의 디렉토리와 파일』을 참조하십시오.

.trace 이 디렉토리에는 `strmqtrc` 명령으로 작성한 추적 파일이 있습니다.

MQS_ROOT:[MQM.QMGRS.QMNAME] 서브디렉토리의 디렉토리와 파일

디폴트로 MQS_ROOT:[MQM.QMGRS.QMNAME] 디렉토리에는 다음 디렉토리와 파일이 있습니다. .QMNAME은 시스템에서 작성하여 실행하는 모든 큐 관리자에 대해 작성됩니다.

amqalchk.fil

마지막 체크포인트에 대한 정보를 포함하는 체크포인트 파일

.auth 이 디렉토리에는 권한과 연관된 서브디렉토리와 파일이 있습니다.

\$aclass.;

이 파일에는 모든 클래스에 대한 권한 스탠자가 있습니다.

.namelist

이 디렉토리에는 각 이름 목록에 대한 파일이 있습니다. 각 파일에는 연관된 이름 목록에 대한 권한 스탠자가 있습니다.

\$class.;

이 파일에는 이름 목록 클래스에 대한 권한 스탠자가 있습니다.

.\$mangled

이름 목록 이름에 올바르지 않은 OpenVMS 문자가 있으면 자동으로 올바른 OpenVMS 이름으로 변환됩니다. 이 파일에 올바른 OpenVMS 이름이 있습니다. 25 페이지의 『MQSeries 파일 이름 이해』 부분을 참조하십시오.

system\$default\$namelist

이 파일에는 시스템 디폴트 이름 목록에 대한 권한 스탠자가 있습니다.

.procdef

각 MQSeries 프로세스 정의는 이 디렉토리의 파일과 연관됩니다.

\$class.;

이 파일에는 프로세스 정의 클래스에 대한 권한 스탠자가 있습니다.

.\$mangled

프로세스 정의 이름에 올바르지 않은 OpenVMS 문자가 있으면 자동으로 올바른 OpenVMS 이름으로 변환됩니다. 이 파일에 올바른 OpenVMS 이름이 있습니다. 25 페이지의 『MQSeries 파일 이름 이해』 부분을 참조하십시오.

.system\$default\$process.;

이 파일에는 시스템 디폴트 프로세스에 대한 권한 스탠자가 있습니다.

.qmanager

이 디렉토리에는 각 큐 관리자에 대한 파일이 있습니다. 각 파일에는 연관된 큐 관리자에 대한 권한 스탠자가 있습니다.

\$class.;

이 파일에는 큐 관리자 클래스에 대한 권한 스탠자가 있습니다.

.\$mangled

큐 관리자 정의 이름에 올바르지 않은 OpenVMS 문자가 있으면 자동으로 올바른 OpenVMS 이름으로 변환됩니다. 이 파일에 올바른 OpenVMS 이름이 있습니다. 25 페이지의 『MQSeries 파일 이름 이해』 부분을 참조하십시오.

self.; 이 파일에는 큐 관리자 오브젝트에 대한 권한 스탠자가 있습니다.

.queues

이 디렉토리에는 각 큐에 대한 파일이 있습니다. 각 파일에는 연관된 큐에 대한 권한 스탠자가 있습니다.

\$CLASS

이 파일에는 큐 클래스에 대한 권한 스탠자가 있습니다.

.\$mangled

큐 이름에 올바르지 않은 OpenVMS 문자가 있으면 자동으로 올바른 OpenVMS 이름으로 변환됩니다. 이 파일에 올바른 OpenVMS 이름이 있습니다. 25 페이지의 『MQSeries 파일 이름 이해』 부분을 참조하십시오.

큐의 정의 파일

각 파일은 큐 관리자에 사전정의된 오브젝트에 해당합니다.

- system\$admin\$channel\$event.;
- system\$admin\$command\$queue.;
- system\$admin\$perfm\$event.;
- system\$admin\$qmgr\$event.;
- system\$channel\$initq.;
- system\$channel\$syncq.;
- system\$cics\$initiation\$queue.;
- system\$cluster\$command\$queue.;
- system\$cluster\$repository\$queue.;
- system\$cluster\$transmit\$queue.;
- system\$dead\$letter\$queue.;
- system\$default\$alias\$queue.;
- system\$default\$initiation\$queue.;

```
system$default$local$queue.;
system$default$model$queue.;
system$default$remote$queue.;
system$mqsc$reply$queue.;
```

.qaadmin.;

권한 부여를 제어하기 위해 내부적으로 사용되는 파일

.dce DCE 지원이 사용할 수 있도록 예약된 빈 디렉토리

.errors

이 디렉토리에는 연산자 메시지 파일이 최근 날짜부터 오래된 날짜의 순서로 있습니다.

```
amqerr01.log
amqerr02.log
amqerr03.log
```

.esem 내부적으로 사용되는 파일이 있는 디렉토리

.isem 내부적으로 사용되는 파일이 있는 디렉토리

.msem 내부적으로 사용되는 파일이 있는 디렉토리

.namelist

이 디렉토리에는 각 큐 관리자에 대한 이름 목록이 있습니다.

.plugcomp

설치가능 서비스가 사용할 수 있도록 예약된 빈 디렉토리

.procdef

각 MQSeries 프로세스 정의는 이 디렉토리의 파일과 연관됩니다. 파일 이름은 프로세스 정의 이름과 일치합니다.

qm.ini

큐 관리자 구성 파일

.qmanager

큐 관리자 오브젝트

qmstatus.ini

이 파일에는 큐 관리자의 상태를 설명하는 텍스트가 있습니다.

.queues

각 큐에는 여기에서 'q'라고 하는 하나의 파일이 포함된 디렉토리가 있습니다. 파일 이름은 큐 이름과 일치하지만 몇 가지 제한이 있습니다. 25 페이지의 『MQSeries 파일 이름 이해』를 참조하십시오.

디렉토리 구조

.shmem

.perQueue

내부적으로 사용되는 파일이 있는 디렉토리

.ssem 내부적으로 사용되는 파일이 있는 디렉토리

.startprm

내부적으로 사용되는 임시 파일이 있는 디렉토리

.\$ipcc

amqclchl.tab

클라이언트 채널 테이블 파일

amqrfcda.dat

채널 테이블 파일

.esem 내부적으로 사용되는 파일이 있는 디렉토리

.isem 내부적으로 사용되는 파일이 있는 디렉토리

.msem 내부적으로 사용되는 파일이 있는 디렉토리

.shmem

.perQueue 내부적으로 사용되는 파일이 있는 디렉토리

.ssem 내부적으로 사용되는 파일이 있는 디렉토리

부록D. 명령 세트 비교

다음 표에서는 여러 관리 명령 세트에서 사용 가능한 기능을 비교합니다.

- 『큐 관리자의 관리에 사용하는 명령』
- 『명령 서버 관리에 사용하는 명령』
- 350 페이지의 『큐 관리에 사용하는 명령』
- 350 페이지의 『프로세스 관리에 사용하는 명령』
- 351 페이지의 『채널 관리에 사용하는 명령』
- 351 페이지의 『기타 제어 명령』

주: Compaq OpenVMS용 MQSeries에 적용되는 MQSC 명령만 표시됩니다.

큐 관리자의 관리에 사용하는 명령

표 22. 큐 관리자의 관리에 사용하는 명령

PCF	MQSC	제어
큐 관리자 변경	ALTER QMGR	-
(큐 관리자 작성)*	-	crtmqm
(큐 관리자 삭제)*	-	dltmqm
큐 관리자 조회	DISPLAY QMGR	-
(큐 관리자 정지)*	-	endmqm
큐 관리자 ping	PING QMGR	-
(큐 관리자 시작)*	-	strmqm
주: * PCF 명령으로 사용할 수 없음		

명령 서버 관리에 사용하는 명령

표 23. 명령 서버 관리에 사용하는 명령

설명	제어
명령 서버 표시	dspmqcsv
명령 서버 시작	strmqcsv
명령 서버 정지	endmqcsv
주: 이 그룹에 있는 기능은 제어 명령으로만 사용 가능합니다. 이 그룹에는 상응하는 MQSC 명령이나 PCF 명령이 없습니다.	

큐 관리에 사용하는 명령

표 24. 큐 관리에 사용하는 명령

PCF	MQSC
큐 변경	ALTER QLOCAL ALTER QALIAS ALTER QMODEL ALTER QREMOTE
큐 지우기	CLEAR QUEUE
큐 복사	DEFINE QLOCAL(x) LIKE(y) DEFINE QALIAS(x) LIKE(y) DEFINE QMODEL(x) LIKE(y) DEFINE QREMOTE(x) LIKE(y)
큐 작성	DEFINE QLOCAL DEFINE QALIAS DEFINE QMODEL DEFINE QREMOTE
큐 삭제	DELETE QLOCAL DELETE QALIAS DELETE QMODEL DELETE QREMOTE
큐 조회	DISPLAY QUEUE
큐 이름 조회	DISPLAY QUEUE
주: 이 그룹에는 상응하는 제어 명령이 없습니다.	

프로세스 관리에 사용하는 명령

표 25. 프로세스 관리에 사용하는 명령

PCF	MQSC
프로세스 변경	ALTER PROCESS
프로세스 복사	DEFINE PROCESS(x) LIKE(y)
프로세스 작성	DEFINE PROCESS
프로세스 삭제	DELETE PROCESS
프로세스 조회	DISPLAY PROCESS
프로세스 이름 조회	DISPLAY PROCESS
주: 이 그룹에는 상응하는 제어 명령이 없습니다.	

채널 관리에 사용하는 명령

표 26. 채널 관리에 사용하는 명령

PCF	MQSC	제어
채널 변경	ALTER CHANNEL	-
채널 복사	DEFINE CHANNEL(x) LIKE(y)	-
채널 작성	DEFINE CHANNEL	-
채널 삭제	DELETE CHANNEL	-
채널 조회	DISPLAY CHANNEL	-
채널 이름 조회	DISPLAY CHANNEL	-
채널 ping	PING CHANNEL	-
채널 재설정	RESET CHANNEL	-
채널 해석	RESOLVE CHANNEL	-
채널 시작	START CHANNEL	runmqchl
채널 시작기 시작	START CHINIT	runmqchi
채널 리스너 시작	-	runmqlsr
채널 정지	STOP CHANNEL	-

기타 제어 명령

표 27. 기타 제어 명령

설명	제어
MQSeries 변환 종료 작성	crtmqcvx
권한 표시	dspmqaout
오브젝트가 사용한 파일 표시	dspmqlfs
MQSeries 형식의 추적 출력 표시	dspmqtrc
MQSeries 추적 종료	endmqtrc
Failover 세트 관리	Failover
매체 이미지 레코드	rcdmqimg
매체 오브젝트 재작성	rcrmobj
MQSeries 트랜잭션 해석	rsvmqtrn
MQSC 명령 실행	runmqsc
트리거 모니터 실행	runmqtrm
클라이언트 트리거 모니터 실행	runmqtrmc
권한 설정 또는 재설정	setmqaut
Failover 모니터 시작	runmqfm
MQSeries 추적 시작	strmqtrc

주: 이 그룹에 있는 기능은 제어 명령으로만 사용 가능합니다. 직접적으로 상응하는 PCF 명령이나 MQSC 명령은 없습니다.

명령 세트 비교

부록E. 샘플 MQI 프로그램 및 MQSC 파일

Compaq OpenVMS용 MQSeries는 실험적으로 사용할 수 있는 일련의 간략한 샘플 MQI 프로그램과 MQSC 명령 파일을 제공합니다. 이러한 샘플 프로그램과 명령 파일은 다음 절에서 설명됩니다.

- 『MQSC 명령 파일 샘플』
- 『C 및 COBOL 프로그램 샘플』
- 354 페이지의 『기타 도구』

MQSC 명령 파일 샘플

표28에는 MQSC 명령 파일 샘플 목록이 나옵니다. 이 파일은 MQSC 명령이 포함된 단순한 ASCII 텍스트 파일입니다. 각 파일에 순서대로 **runmqsc** 명령을 실행하여 파일에 지정된 오브젝트를 작성할 수 있습니다. 47 페이지의 『제공된 MQSC 명령 파일의 실행』을 참조하십시오.

이러한 파일의 디폴트 위치는 **MQS_EXAMPLES**: 디렉토리입니다.

표 28. MQSC 명령 파일

파일 이름	목적
AMQSCOS0.TST	C 및 COBOL 프로그램 샘플에 사용할 일련의 MQI 오브젝트를 작성합니다.

C 및 COBOL 프로그램 샘플

표29에는 샘플 MQI 소스 파일 목록이 나옵니다. 소스 파일의 디폴트 위치는 **MQS_EXAMPLES**: 디렉토리이며 컴파일된 버전의 디폴트 위치는 **MQS_EXAMPLES**: 아래의 **[.BIN]** 디렉토리입니다. 프로그램이 수행하는 작업 및 프로그램 사용 방법에 대한 자세한 정보는 *MQSeries Application Programming Guide*를 참조하십시오.

표 29. 샘플 프로그램 - 소스 파일

C	COBOL	목적
AMQSBCG0.C	-	지정된 큐에 있는 모든 메시지의 메시지 컨텍스트 필드와 메시지 설명자를 읽고 출력합니다.
AMQSECHA.C	AMQVECHX.COB	메시지를 메시지 큐에서 리플라이-투 큐로 반향합니다. 트리거된 응용프로그램으로 실행할 수 있습니다.
AMQSGBR0.C	AMQ0GBR0.COB	메시지 자체는 큐에 남겨 두고 큐의 메시지를 SYSSOUTPUT에 씁니다. 찾아보기 옵션으로 MQGET을 사용합니다.
AMQSGET0.C	AMQ0GET0.COB	MQGET을 사용하여 이름 지정된 큐에서 메시지를 제거하고 해당 메시지를 \$OUTPUT에 씁니다.

샘플

표 29. 샘플 프로그램 - 소스 파일 (계속)

C	COBOL	목적
AMQSINQA.C	AMQVINQX.COB	트리거된 큐를 읽습니다. 각 요청을 큐 이름으로 읽고 그 큐에 대한 정보로 응답합니다.
AMQSPUT0.C	AMQ0PUT0.COB	SYSS\$INPUT을 메시지에 복사한 다음 이 메시지를 지정한 큐에 넣습니다(put).
AMQSREQ0.C	AMQ0REQ0.COB	요청 메시지를 지정한 큐에 넣은(put) 다음 응답 메시지를 표시합니다.
AMQSSETA.C	AMQVSETX.COB	이름 지정된 큐에 대한 메시지 입력(put)을 금지하고, 결과 명령문으로 응답합니다. 트리거된 응용프로그램으로 실행합니다.
AMQSTRG0.C	-	이름 지정된 이니시에이션 큐를 읽은 다음 각 트리거 메시지와 연관된 프로그램을 시작하는 트리거 모니터. 제공된 runmqtrm 명령의 전체 트리거 기능 서브세트를 제공합니다.
AMQSVFCX.C	-	데이터 변환 종료 루틴의 샘플 C 구조

기타 도구

다음 도구 파일은 형식기와 코드 변환을 지원하기 위해 제공됩니다.

표 30. 기타 파일

파일 이름	위치	목적
AMQTRC.FMT	SYSS\$LIBRARY	MQSeries 추적 형식을 정의합니다.
CCSID.TBL	MQS_ROOT:[MQM.CONV.TABLE]	이 파일을 편집하여 새로 지원되는 CCSID 값을 MQSeries 시스템에 추가할 수 있습니다. CCSID에 대한 자세한 정보는 CDRA (Character Data Representation Architecture) 문서를 참조하십시오.

부록F. OpenVMS 클러스터 Failover 세트 템플리트

이 부록에는 다음과 같은 Failover 세트 템플리트가 있습니다.

- 『템플리트 구성 파일 FAILOVER.TEMPLATE』
- 357 페이지의 『템플리트 StartCommand 프로시저 START_QM.TEMPLATE』
- 359 페이지의 『템플리트 EndCommand 프로시저 END_QM.TEMPLATE』
- 362 페이지의 『템플리트 TidyCommand 프로시저 TIDY_QM.TEMPLATE』

템플리트 구성 파일 FAILOVER.TEMPLATE

```
*****#
#*                                           *#
#* Statement:      Licensed Materials - Property of IBM      *#
#*                                           *#
#*                33H2205, 5622-908                          *#
#*                33H2267, 5765-623                          *#
#*                29H0990, 5697-176                          *#
#*                (C) Copyright IBM Corp. 2000, 2001         *#
#*                                           *#
*****#
#
# FAILOVER.TEMPLATE
# Template for creating a FAILOVER.INI configuration file
# All lines beginning with a '#' are treated as comments
#
# OpenVMS Cluster Failover Set Configuration information
# -----
#
# The TCP/IP address used by the OpenVMS Cluster Failover Set
#
IpAddress=n.n.n.n
#
# The TCP/IP port number used by the MQSeries Queue Manager
#
PortNumber=1414
#
# The timeout used by the EndCommand command procedure
#
TimeOut=30
#
# The command procedure used to start the Queue Manager
#
StartCommand=@sys$manager:start_qm
#
# The command procedure used to end the Queue Manager
#
EndCommand=@sys$manager:end_qm
#
# The command procedure used to tidy up on a node after a
# Queue Manager failure but the OpenVMS node did not fail
```

FAILOVER.TEMPLATE

```
#
TidyCommand=@sys$manager:tidy_qm
#
# The directory in which the log files for the start, end and
# tidy commands are written
#
LogDirectory=mqs_root:[mqm.errors]
#
# The number of nodes in the OpenVMS Cluster Failover Set. The
# number of nodes defined below must agree with this number
#
NodeCount=2
#
# The Name of the OpenVMS node
#
NodeName=BATMAN
#
# The TCP/IP interface name for the node
#
Interface=we0
#
# The priority of the node
#
Priority=1
#
# The Name of the OpenVMS node
#
NodeName=ROBIN
#
# The TCP/IP interface name for the node
#
Interface=we0
#
# The priority of the node
#
Priority=2
```

그림 27. 템플릿 구성 파일: *failover.template*

템플릿 StartCommand 프로시저 START_QM.TEMPLATE

```

$ on error then exit
$!*****
$!* Statement:      Licensed Materials - Property of IBM      *
$!*                33H2205, 5622-908                          *
$!*                33H2267, 5765-623                          *
$!*                29H0990, 5697-176                          *
$!*                (C) Copyright IBM Corp. 2000, 2001        *
$!*****
$! Template command procedure used by Failover Sets to start the
$! queue manager
$! Parameters :
$! P1 = Queue Manager Name
$! P2 = Queue Manager Directory Name
$! P3 = TCP/IP address
$! P4 = TCP/IP interface name
$! P5 = Listener port number
$!
$ @sys$startup:mqs_symbols
$ set def mqs_root:[mqm.qmgrs.'p2'.errors]
$ define sys$scratch mqs_root:[mqm.qmgrs.'p2'.errors]
$!
$! Digital TCP/IP Services for OpenVMS commands
$!
$ @sys$startup:tcpip$define_commands
$!
$! Configure the IP address
$!
$ ifconfig 'p4' alias 'p3'
$!
$! TCPware for OpenVMS commands
$!
$! @tcpware:tcpware_commands
$!
$! Configure the IP address
$!
$! netcu add secondary 'p3'
$!
$! MultiNet for OpenVMS commands
$!
$! Configure the IP address
$!
$! define/sys/exec multinet_ip_cluster_aliases "'p3'"
$!
$! Restart the Multinet server
$!
$! @multinet:start_server
$!$! Start the queue manager
$!
$ strmqm 'p1'
$!
$! Start the listener
$!
$! runmq1sr -t tcp -p 'p5' -m 'p1'

```

START_QM.TEMPLATE

```
$!  
$! Insert commands to start any applications  
$!  
$exit
```

그림 28. 템플릿 *StartCommand* 프로시저: *Start_QM.template*

템플릿 EndCommand 프로시저 END_QM.TEMPLATE

```

$ on error then exit
$!
$!*****
$!*
$!* Statement:      Licensed Materials - Property of IBM      *
$!*
$!*              33H2205, 5622-908                            *
$!*              33H2267, 5765-623                            *
$!*              29H0990, 5697-176                            *
$!*              (C) Copyright IBM Corp. 2000, 2001          *
$!*
$!*****
$!
$! Template Command procedure used by Failover Sets to end the
$! queue manager
$!
$! Parameters :
$!
$! P1 = Queue Manager Name
$! P2 = Queue Manager Directory Name
$! P3 = TCP/IP address
$! P4 = TCP/IP interface name
$! P5 = Listener port number
$! P6 = End Queue Manager Timeout
$!
$ @sys$startup:mqs_symbols
$ check_qm:==$sys$system:mqcheckqm
$ set def mqs_root:[mqm.qmgrs.'p2'.errors]
$ define sys$scratch mqs_root:[mqm.qmgrs.'p2'.errors]
$ SS$_NORMAL=1
$ SS$_ABORT=44
$ SS$_TIMEOUT=556
$!
$! Insert commands to shutdown any applications prior to ending MQSeries
$!
$! Get the timeout period for each operation seconds
$!
$ timeout = 'p6'
$!
$! Initialise the outer loop
$!
$ out_count = 0
$!
$! Initialise the complete flag
$!
$ complete = 0
$!
$ out_next:
$ if (out_count .gt. 2) .or. (complete .eq. 1) then goto out_finish
$!
$ if out_count .eq. 0
$ then
$!
$! End the queue manager gracefully first
$!
$ spawn/nowait $endmqm -i 'p1'

```

END_QM.TEMPLATE

```
$ else
$ if out_count .eq. 1
$ then
$!
$! End the queue manager abruptly
$!
$ spawn/nowait $endmqm -p 'p1'
$ else
$!
$! Stop/id the execution controller
$!
$ check_qm -m 'p1'
$ if ( mqs$ec_pid .nes. "" ) then $stop/id='mqs$ec_pid'
$ endif
$ endif
$!
$ in_start:
$!
$! Initialise the outer loop
$!
$ in_count = 0
$!
$ in_next:
$!
$! Inner loop
$!
$ if ( ( in_count .ge. timeout ) .and. ( timeout .ne. 0 ) ) -
    .or. ( complete .eq. 1 ) then goto in_finish
$!
$! Check if the execution controller is still running
$!
$ check_qm -m 'p1'
$ if mqs$ec_pid .eqs. ""
$ then
$!
$! The Execution controller is no longer running so we are finished
$!
$ complete = 1
$ goto in_finish
$ endif
$!
$! Wait a second and go round again
$!
$ wait 00:00:01
$ in_count = in_count + 1
$ goto in_next
$ in_finish:
$!
$! End of the inner loop
$!
$ out_count = out_count + 1
$ goto out_next
$ out_finish:
$!
$! End of the outer loop
$!
$! Digital TCP/IP Services for OpenVMS commands
$!
```

```

$ @sys$startup:tcpip$define_commands
$!
$! De-configure the IP address
$!
$ ifconfig 'p4' -alias 'p3'
$!
$! TCPware for OpenVMS commands
$!
$! @tcpware:tcpware_commands
$!
$! De-configure the IP address
$!
$! netcu remove secondary 'p3'
$!
$! MultiNet for OpenVMS commands
$!
$! De-configure the IP address
$!
$! deass/sys/exec multinet_ip_cluster_aliases
$!
$! Restart the Multinet server
$!
$! @multinet:start_server
$!
$!
$! If the Queue Manager was shutdown successfully set the status
$! to SS$_NORMAL. If it was necessary to STOP/ID the Execution
$! controller set the status to SS$_ABORT and if the Execution
$! controller is still running set the status to SS$_TIMEOUT to
$! indicate an error
$!
$ if ( complete .eq. 1 )
$then
$!
$! End the listener process
$!
$! endmq1sr -m 'p1'
$!
$ if ( out_count .eq. 3 )
$ then
$ exit SS$_ABORT
$ else
$ exit SS$_NORMAL
$ endif
$else
$ exit SS$_TIMEOUT
$endif

```

그림 29. 템플릿 EndCommand 프로시저: END_QM.template

템플릿 TidyCommand 프로시저 TIDY_QM.TEMPLATE

```

$ on error then exit
$!*****
$!* Statement:      Licensed Materials - Property of IBM      *
$!*               *                                           *
$!*               33H2205, 5622-908                          *
$!*               33H2267, 5765-623                          *
$!*               29H0990, 5697-176                          *
$!*               (C) Copyright IBM Corp. 2000, 2001        *
$!*****
$! Template Command procedure used by Failover Sets to tidy up after
$! a queue manager failure
$!
$! Parameters :
$! P1 = Queue Manager Name
$! P2 = Queue Manager Directory Name
$! P3 = TCP/IP address
$! P4 = TCP/IP interface name
$! P5 = Listener port number
$!
$ @sys$startup:mqs_symbols
$ set def mqs_root:[mqm.qmgrs.'p2'.errors]
$ define sys$scratch mqs_root:[mqm.qmgrs.'p2'.errors]
$!
$! Insert commands to do any tidying up after a queue manager has failed
$!
$! Digital TCP/IP Services for OpenVMS commands
$!
$ @sys$startup:tcpip$define_commands
$!
$! De-configure the IP address
$!
$ ifconfig 'p4' -alias 'p3'
$!
$! TCPware for OpenVMS commands
$!
$! @tcpware:tcpware_commands
$!
$! De-configure the IP address
$!
$! netcu remove secondary 'p3'
$!
$! MultiNet for OpenVMS commands
$!
$! De-configure the IP address
$!
$! deass/sys/exec multinet_ip_cluster_aliases
$!
$! Restart the Multinet server
$!
$! @multinet:start_server
$!
$exit

```

그림 30. Template TidyCommand 프로시저: TIDY_QM.template

부록G. Compaq OpenVMS용 MQSeries에서의 코드 세트 지원

Compaq OpenVMS용 MQSeries는 Compaq OpenVMS용 MQSeries에서 표준으로 제공되는 로케일(특정 문화에 대한 규칙을 정의하는 사용자 환경 서브세트)에서 사용하는 대부분의 코드세트를 지원합니다.

로케일이 설정되어 있지 않은 경우 사용되는 CCSID는 819(ISO8859-1 코드 세트)입니다.

MQSeries에서 메시지 및 메시지 헤더 데이터에 사용된 코드세트를 식별하기 위해 사용되는 CCSID(Coded Character Set Identifier)는 로케일 구성의 LC_CTYPE 범주를 분석하여 확보할 수 있습니다.

표31은 로케일에서 사용하는 코드 세트에 등록된 로케일과 CCSID를 표시합니다.

표 31. 로케일 및 CCSID

로케일	언어	코드 세트	CCSID
C	영어	ISO8859-1	819
CS_CZ_ISO8859-2	체코어	ISO8859-2	912
DA_DK_ISO8859-1	덴마크어	ISO8859-1	819
DE_DE_ISO8859-1	독일어	ISO8859-1	819
DE_CH_ISO8859-1	독일어(스위스)	ISO8859-1	819
EL_GR_ISO8859-7	그리스어	ISO8859-7	813
EN_GB_ISO8859-1	영어(영국)	ISO8859-1	819
EN_US_ISO8859-1	영어(미국)	ISO8859-1	819
ES_ES_ISO8859-1	스페인어	ISO8859-1	819
FI_FI_ISO8859-1	핀란드어	ISO8859-1	819
FR_FR_ISO8859-1	프랑스어(프랑스)	ISO8859-1	819
FR_BE_ISO8859-1	프랑스어(벨기에)	ISO8859-1	819
FR_CA_ISO8859-1	프랑스어(캐나다)	ISO8859-1	819
FR_CH_ISO8859-1	프랑스어(스위스)	ISO8859-1	819
HU_HU_ISO8859-2	헝가리어	ISO8859-2	912
IS_IS_ISO8859-1	아이슬란드어	ISO8859-1	819
IT_IT_ISO8859-1	이탈리아어(이탈리아)	ISO8859-1	819
IW_IL_ISO8859-8	히브리어	ISO8859-8	916
JA_JP_EUCJP	일본어	eucJP	954
JA_JP_SDECKANJI	일본어	SDECKANJI	954**
JA_JP_SJIS	일본어	SJIS	932
KO_KR_DECKOREAN	한국어	DECKOREAN	970**
NL_NL_ISO8859-1	네덜란드어(네덜란드)	ISO8859-1	819

지원되는 코드 세트

표 31. 로케일 및 CCSID (계속)

로케일	언어	코드 세트	CCSID
NL_BE_ISO8859-1	네덜란드어(벨기에)	ISO8859-1	819
NO_NO_ISO8859-1	노르웨이어	ISO8859-1	819
PL_PL_ISO8859-2	폴란드어	ISO8859-2	912
PT_PT_ISO8859-1	포르투갈어	ISO8859-1	819
SK_SK_ISO8859-2	슬로바키아어	ISO8859-2	912
RU_RU_ISO8859-5	키릴어	ISO8859-5	915
SV_SE_ISO8859-1	스웨덴어	ISO8859-1	819
TR_TR_ISO8859-9	터키어	ISO8859-9	920
ZH_CN_DECHANZI	중국어	DECHANZI	1383**
ZH_HK_DECHANZI	중국어	DECHANZI	1383**
ZH_HK_EUCTW	대만어	eucTW	964
ZH_HK_EUCTW	대만어	eucTW	964
ZH_HK_DECHANYU	대만어	DECHANYU	964**
ZH_TW_DECHANYU	대만어	DECHANYU	964**
ZH_HK_BIG5	대만어	big5	950
ZH_TW_BIG5	대만어	big5	950

주:

** 등록된 IBM CCSID 중 가장 근접한 CCSID가 사용됩니다.

이러한 로케일의 플랫폼간 상호 지원 목록에 대한 추가 정보는 *MQSeries Application Programming Reference*를 참조하십시오.

부록H. MONMQ 진단 유틸리티

MONMQ 유틸리티는 Compaq OpenVMS용 MQSeries 문제점 진단 및 해결을 지원하기 위한 도구입니다. MONMQ 유틸리티는 명령행이나 DCL 스크립트 내에서 대화식으로 사용할 수 있습니다.

MONMQ 유틸리티는 주로 다음 작업에서 사용됩니다.

- 공유 메모리 관리
- OpenVMS 자원 사용량 정보 수집
- 실행 중인 큐 관리자에서 추적 출력 확보

MONMQ에는 매개변수를 지원하고 MONMQ 명령 스크립트를 실행하기 위한 도움말 시스템이 있습니다. MONMQ 시작하면 디폴트 스크립트 `sys$manager:mqs_trace_startup.mqt`가 실행되어 초기 구성을 제공합니다.

```
$monmq
ok - trace mailbox 0 opened as default

MQT> help
Help can be used to display information about available commands or parameters
Help [ <verb> || <parameter/variable name> || commands || parameters || examples]

Valid trace commands are in the format:
Verb [<parameters>] [<variable = expression>][;][optional second command]
```

개요

OpenVMS에서의 MQSeries 추적은 전역 섹션 및 편지함을 사용하여 구현됩니다. MQSeries가 설치된 노드 하나에 최대 10개의 추적 섹션(LU)이 공존할 수 있습니다. 그러나 한 추적 세션에서는 LU를 한 번에 하나만 사용하는 것이 좋습니다. 뿐만 아니라 여러 사용자가 동시에 같은 LU를 열지 않는 것이 좋습니다. 그렇지 않은 경우 결과를 예측할 수 없습니다.

각 공유 섹션(LU)은 채널 정의와 LU 정의 자체를 포함합니다. 각 채널 정의는 연결된 스레드의 세부사항과 스레드의 개인용 스택 및 스레드 순환 버퍼를 포함합니다. 또한 공유 섹션은 MONMQ와 연결된 스레드 간의 상호프로세스 통신에 사용되는 일련의 플래그를 포함합니다.

각 LU에는 실시간 추적 메시지를 수신하는 데 사용되는 연관된 편지함이 있습니다. 실시간 추적을 수행하려면 **TRACE START** 명령을 사용하여 클라이언트 프로세스를 시작해야 합니다. 분리된 이 전용 프로세스는 메시지가 LU의 편지함에 도착하면 각 메시지를 읽은 후 형식화하여 표시합니다. 연결된 각 스레드는 메시지를 같은 편지함에 기록하므로 MQSeries 프로세스/스레드 간 상호통신을 물리적으로 볼 수 있습니다.

MONMQ 개요

올바르게 작동하면, MONMQ는 상호프로세스 시기 문제점, 운영 체제 자원 고갈이나 코드 문제점 등의 문제점을 진단하기 위한 광범위한 방법을 제공합니다.

이 부록에서는 MONMQ 명령에 대해 설명합니다.

MONMQ의 변수

MONMQ에서 대부분의 명령은 변수를 사용합니다. 명령 내에 변수를 지정하지 않으면 **set** 명령이 정의한 디폴트 값이 사용됩니다. set 이외의 명령에 변수를 사용하면 해당 변수의 디폴트 값이 변경되지 않습니다.

변수는 다음 항목이 포함됩니다.

- 정수 변수(10진수 또는 16진수)
16진수 값은 앞에 0x를 넣어 입력하거나 16진수 값이 들어갈 자리에 a에서 f까지의 문자를 사용하여 입력할 수 있습니다.
- 따옴표로 묶인 텍스트
- 범위는 최대:최소 값을 넣어 지정합니다.
범위는, 예를 들어, 채널 범위에 명령을 적용할 수 있도록 하기 위해 사용됩니다.

예를 들면 다음과 같습니다.

```
MQT> set lu=2
MQT> set pid=0x223
MQT> set pid=2fa
MQT> set buffile="filename.buf"
MQT> set chl=0:20
```

변수의 현재 디폴트 값은 variables 명령을 사용하여 표시할 수 있습니다.

```
MQT> variables
defined variables
lu=0:0                nochls=20                buffer=1000
chl=0:20             component=0(HEX)       line=0
mask=0(HEX)         pid=0(HEX)            node=(null)
function=0(HEX)     div=0                 depth=32
resource=0          wait=1(BOOL)          timestamp=0(BOOL)
listfile=(null)    buffile=(null)        step=0(BOOL)
active=0(BOOL)     fname=(null)          delay=100
post=0(BOOL)

defined constants
fent=1(HEX)         fout=2(HEX)           ferr=4(HEX)
fxxx=8(HEX)        dgn=10(HEX)          shm=20(HEX)
spl=40(HEX)        evt=80(HEX)          mtx=100(HEX)
prc=200(HEX)       msc=400(HEX)         inf=800(HEX)
log=2000(HEX)     shl=4000(HEX)        memory=3
mutex=4            mailbox=5             nanoseconds=1
microseconds=2    milliseconds=3       seconds=4
```


디폴트 값을 설정하여 명령을 단순화하는 것이 좋습니다. 예를 들어, 다음 명령 순서는 동일하게 작동합니다.

```
MQT> open lu=0 buffer=1000 nochls=20
MQT> open lu=1 buffer=1000 nochls=20
MQT> show channels lu=0 chl=1:10
MQT> show channels lu=1 chl=1:10
```

또는

```
MQT> set nochls=20 chl=0:10 buffer=1000 lu=0
MQT> open
MQT> open lu=1
MQT> show channels
MQT> show channels lu=1
```

MONMQ 명령은 고유한 명령이 되기 위해 필요한 최소 문자 수까지 줄일 수 있습니다. 다음과 같은 일련의 명령은 좀 더 줄일 수 있습니다.

```
MQT> se noc=20 ch=0:10 buffe=1000 lu=0
MQT> op
MQT> op lu=1
MQT> sh ch
MQT> sh ch lu=1
```

MONMQ는 변수를 사용하여 간단한 산술 연산을 수행할 수 있으므로 `set lu=lu+1`과 같은 명령을 사용할 수 있습니다.

새 변수는 **declare** 명령을 사용하여 선언할 수 있습니다. 매개변수는 다음과 같습니다.

- 변수 이름
- 변수 유형
- 도움말 텍스트. 도움말 명령으로 도움말 텍스트를 검색할 수 있습니다.

```
MQT> declare ec int "channel number for execution controller"
MQT> set ec = 4
MQT> show channel chl=ec
Chl  Pid  Mailbox  Stack  Active  Post  Time  Mask  Process Name
 4   2c1f  7ee70290  4      0      0    0   ffffffff  AMQZXA0.EXE
MQT> help ec

VARIABLE ec:
channel number for execution controller

MQT>
```

디폴트 값 지정

DEFAULT variable=<expression> [variable=expression] ...

이 명령은 MONMQ 내에 정의된 모든 변수에 디폴트 값을 지정하도록 합니다. 디폴트 값을 설정한 후에는 명령행에서 디폴트 변수 이름을 생략할 수 있습니다. 예를 들면 다음과 같습니다.

```
MQT>default lu=2 ch1=3:6
```

이 명령은 디폴트 값 2를 lu 변수에 설정하고 3부터 6까지의 값이 채널 변수에 포함되도록 합니다. 이후부터는 **show channels**와 같은 일반 명령을 사용할 때 lu 2에 채널 3-6이 포함되어 표시됩니다.

명령행에서 변수를 생략한 경우에만 디폴트 값이 사용됩니다. 모든 디폴트 값은 시동 스크립트 파일 MQS_TRACE_STARTUP.MQT에 설정됩니다. 필요에 맞게 이 파일을 편집할 수 있습니다.

추적 섹션 및 연관된 편지함을 열거나 작성

OPEN [lu=number] [nochls=number] [buffer=number]

이 명령은 추적 섹션 및 연관된 편지함을 열거나 작성합니다. **open** 명령은 MQSeries 프로세스 추적에 필요한 기본 자원을 작성합니다. 각 LU에는 MQSeries 프로세스와 통신하기 위해 필요한 연관된 공유 섹션 및 편지함이 있습니다.

이 명령은 세 가지 선택적 매개변수를 사용합니다. 첫 번째 매개변수 [LU]는 추적 섹션/편지함에 지정된 번호로, 대부분의 다른 MONMQ 명령이 참조로 사용합니다. 노트 하나에 LU를 최대 10개까지 작성할 수 있습니다. 디폴트 값은 0입니다. 지정한 LU가 있으면 MONMQ가 기존 추적 섹션에 연결합니다. 섹션이 없으면 새 섹션이 작성됩니다.

두 번째 매개변수 [nochls]는 이 LU에서 사용할 채널 개수를 지정합니다. 각 채널은 하나의 MQSeries 프로세스/스레드 연결을 나타냅니다. 디폴트 값은 20입니다.

세 번째 매개변수 [buffer]는 각 채널에 대한 추적 기록 버퍼의 최대 크기를 지정합니다. 디폴트는 1000입니다.

다른 MONMQ 명령을 수행하려면 적어도 하나의 LU가 열려 있어야 합니다.

논리 단위 정의 표시

SHOW SEGMENT [lu=range]

SHOW SEGMENT

이 명령은 논리 단위 정의를 표시합니다. **show segment lu=0** 명령을 입력하면 아래의 예와 같이 각 필드 옆에 간략한 설명과 함께 출력이 나타납니다.

```
Trace LU           : 0           /* The LU number as specified in the OPEN [lu] parameter.
Mailbox name      : MQS_TRC_MBX_0 /* The permanent mailbox name assigned to this LU
Device name      : MBAI065:      /* The device name of the mailbox
Status           : Disabled      /* The current status of the mailbox ie.
Mailbox channel  : 352          /* The mailbox channel number assigned to the MONMQ process
History buffer size: 1000       /* The maximum number of message entries in the history
/* circular buffer (as specified by the OPEN [buffer] parameter)
Threads mapped #  : 1           /* The number of processes/threads mapped to this LUs global
/* section (MONMQ always attached)
Time stamping    : Enabled      /* Global timestamp flag (not yet implemented)
Max channels #   : 20          /* Number of channels defined for this LU as specified by the
/* OPEN [nochls] parameter.
Display depth    : 0           /* The stack display depth. Default (0) is to display all stack entries.
Text filename    :             /* The client text trace file
Binary filename  :             /* The client binary trace file
Last status      : 1           /* Last status of mailbox Qio activity (useful if VMS low on
/* resources and MONMQ fails)
Connection map[0] : 0           /* A bit map of all connected channels (maximum no. of channels is 128)
```

LU 닫기 및 삭제

CLOSE [lu=number]

이 명령은 **OPEN** 명령이 수행하는 것과 반대 작업을 수행하며 지정한 LU를 닫고 삭제합니다. LU는 먼저 연결된 각 프로세스에 연결을 끊으라는 신호를 보낸 후 각 채널을 재설정된 다음 마지막으로 추적 편지함 지정을 취소하고 공유 섹션을 삭제하는 제어된 순서대로 닫힙니다. 이 명령은 추적 세션이 완료되었을 때만 수행할 수 있습니다.

채널 세부사항 표시

SHOW CHANNELS [full] [connected] [chl=range]

이 명령은 지정한 채널에 대한 세부사항을 표시합니다. [connected] 매개변수를 사용하면 연결된 스레드가 있는 채널만 표시됩니다. 예를 들어, **show channels connected** 명령을 실행하면 다음과 같이 표시됩니다.

Chl	Pid/Tid	Mailbox	Stack	History	RTime	Time	Mask	Process Name
0	00000245/1	800b0330	4	0	0	0	fffffff	AMQZLAA0.EXE
1	00000244/1	800b0200	8	0	0	0	fffffff	RUNMQCHI.EXE
2	00000243/1	800b01e0	10	0	0	0	fffffff	AMQRRMFA.EXE
3	00000242/1	800b01c0	4	0	0	0	fffffff	AMQZLLP0.EXE
4	00000241/1	800b0220	5	0	0	0	fffffff	AMQHASM0.EXE
5	00000240/1	800b03f0	4	0	0	0	fffffff	AMQZXMA0.EXE

[full] 매개변수는 지정한 채널에 대한 전체 정의를 표시합니다. 예를 들어, **show channels full connected chl=0:3** 명령을 실행하면 다음과 같이 표시됩니다.

SHOW MASK

```
Pid/Tid      : 0000024b/1      /* Connected threads process id and thread sequence number
Status       : *** Connected *** /* Current status of channel (thread is connected)
Process name  : AMQRRMFA.EXE   /* Process name of connected thread
Assigned LU   : 0              /* This channels associated LU
Channel no.   : 2              /* Allocated channle number within the LU
Mailbox channel : 800b01e0     /* Connected threads mailbox channel number for the trace mailbox
Current stack depth : 10      /* Threads current stack depth
Circular logging : Disabled    /* History enabled flag
Next log entry : 0            /* Next history buffer slot number
Realtime tracing : Disabled    /* Real time enable flag (needs client to read messages)
Time stamping : Disabled      /* Enables timestamping for this threads messages
Trace mask    : ffffffff      /* Hexadecimal format of trace mask for this thread (see show mask command)
Step mode     : Off           /* Not yet implemented
No Wait       : On           /* Forces threads qio activity to wait for a resource if not available
Last QIO status : 0          /* Threads last qio call status
Mapped address : 9a2000-c9ffff /* The virtual mapped address range of the LU global section for this thread
```

채널의 현재 추적 마스크 표시

SHOW MASK [chl=*range*]

이 명령은 채널의 현재 추적 마스크를 표시합니다. 강조표시된 줄은 btrace 마스크 비트가 사용 가능함을 나타냅니다. 예를 들어, **show mask chl=1** 명령을 실행하면 다음과 같이 표시됩니다.

```
Trace Mask for Channel 1
Bit 00 - (fent) function entry      Function entry messages
Bit 01 - (fout) function exit       Function exit messages
Bit 02 - (ferr) function exit with error Function exit with error return status
Bit 03 - (fxx) missing function exit Unbalanced function entry/exit message (see note below)
Bit 04 - (dgn) diagnostic messages  Diagnostic messages
Bit 05 - (shm) shared memory         OVMS shared memory messages
Bit 06 - (spl) spinlocks            OVMS spinlock messages
Bit 07 - (evt) events               OVMS event messages
Bit 08 - (mtx) mutexes              OVMS mutex messages
Bit 09 - (prc) process msgs         OVMS thread messages
Bit 10 - (msc) miscellaneous        OVMS kernal niscellaneous messages
Bit 11 - (inf) informational        Internal data messages as requested by show command
Bit 12 -                             Reserved for user defined messages
```

이 출력은 이 스택에 대해 함수 입력항목, 함수 종료, 스핀 잠금 및 이벤트 메시지가 추적됨을 표시합니다. 다른 모든 메시지 유형은 차단됩니다.

대상 스레드 스택의 내용 표시

SHOW STACK [chl=*range*]

이 명령은 대상 스레드 스택의 내용을 표시합니다. 예를 들어, **show stack chl=0:1** 명령을 실행하면 다음과 같이 표시됩니다.

```
0001- 00:00:00.00 03 - 01 -->| ExecCtrlrMain
0002 - 12:36:20.18 03 - 02 --->| zcpReceiveOnLink
0003 - 12:36:20.81 03 - 03 ---->| xcsWaitEventSem
0004 - 12:36:20.83 03 - 04 ----->| vms_evt
0001- 00:00:00.00 03 - 01 -->| ExecCtrlrMain
0002 - 12:36:20.18 03 - 02 --->| zcpSendOnLink
0003 - 12:36:20.81 03 - 03 ---->| xcsPostEventSem
0004 - 12:36:20.83 03 - 04 ----->| vms_evt
```

활성 MQSeries 관련 프로세스 및 메모리 사용량 표시

SHOW PROCESSES

이 명령은 현재 노드에 있는 모든 활성 MQSeries 관련 프로세스를 프로세스의 메모리 사용량과 함께 표시합니다. 예를 들어, **show process** 명령을 실행하면 다음과 같이 표시됩니다.

PID	Proc_Name	Image	Process	WS Size	WS Peak	Virt Peak	Gbl Pg_Cnt	Prc Pg_Cnt	Total Mem
0000023D	BKM3_AG	AMQZLAA0	Agent	23152	16576	203776	3616	12960	16576
0000023C	BKM3_CI	RUNMQCHI	Run Chan Init	8752	6208	180832	1840	4368	6208
0000023B	BKM3_RM	AMQRRMFA	Repository Mgr	11152	8144	185360	2224	5920	8144
0000023A	BKM3_CP	AMQZLLP0	Checkpoint	8752	6384	185952	1920	4464	6384
00000239	BKM3_LG	AMQHASMx	Logger	8752	6288	182016	2080	4208	6288
00000238	BKM3_EC	AMQZMA0	EC	20752	15232	203792	3536	11680	15216
00000128	_FTA4:	MONMQ	MONMQ Utility	8400	8528	198736	2224	3584	5808

채널이 보유한 모든 메시지 표시

SHOW HISTORY [chl=range]

이 명령은 채널의 순환 실행 기록 버퍼에 보유한 모든 메시지를 표시합니다. 생성되었을 때의 스택 용량에 따라 각 메시지가 형식화되고 출력이 들여쓰기됩니다. 예를 들어, **show history chl=3** 명령을 실행하면 다음과 같이 표시됩니다.

```

0215 - 12:35:44.52 03 - 02 ---->| zxcProcessChildren
0216 - 12:35:44.55 03 - 02 ---->| zxcStartWLMserver
0217 - 12:35:44.57 03 - 02 ---->| zxcStartWLMserver
0218 - 12:35:44.59 03 - 02 ---->| zcpReceiveOnLink
0219 - 12:35:44.61 03 - 03 ---->| xcsRequestMutexSem
0220 - 12:35:44.63 03 - 04 ---->| xllSemReq
0221 - 12:35:44.66 03 - 05 ---->| vms_mtx
0222 - 12:35:44.66 03 - 05 .....| vms_mtx :- Locking BKM3/@ipcc_m_1_10 - timeout: -1
0223 - 12:35:44.70 03 - 06 ---->| vms_get_lock
0224 - 12:35:44.72 03 - 06 ---->| vms_get_lock
0225 - 12:35:44.74 03 - 05 ---->| vms_mtx
0226 - 12:35:44.76 03 - 04 ---->| xllSemReq
0227 - 12:35:44.79 03 - 03 ---->| xcsRequestMutexSem
0228 - 12:35:44.81 03 - 03 ---->| xcsResetEventSem
0229 - 12:35:44.83 03 - 04 ---->| vms_evt
0230 - 12:35:44.83 03 - 04 .....| vms_evt Reset on mailbox BKM3/@ipcc_e_1_2 : tout = -1
0231 - 12:35:44.87 03 - 05 ---->| vms_get_mbx_chan
0232 - 12:35:44.87 03 - 05 .....| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0233 - 12:35:44.87 03 - 05 .....| vms_get_mbx_chan Returning key 1a0
0234 - 12:35:44.94 03 - 05 ---->| vms_get_mbx_chan
0235 - 12:35:44.83 03 - 04 .....| vms_evt rc = 0
0236 - 12:35:44.98 03 - 04 ---->| vms_evt
0237 - 12:35:45.00 03 - 03 ---->| xcsResetEventSem
0238 - 12:35:45.03 03 - 03 ---->| xcsReleaseMutexSem
0239 - 12:35:45.05 03 - 04 ---->| xllSemRel
0240 - 12:35:45.07 03 - 05 ---->| vms_mtx
0241 - 12:35:45.07 03 - 05 .....| vms_mtx :- Unlocking BKM3/@ipcc_m_1_10 - timeout: -1
0242 - 12:35:45.11 03 - 06 ---->| vms_get_lock
0243 - 12:35:45.13 03 - 06 ---->| vms_get_lock
0244 - 12:35:45.16 03 - 05 ---->| vms_mtx

```

이 샘플 출력은 실행 기록 버퍼 내의 줄 번호와 메시지가 생성된 시간, 채널 번호, 메시지가 생성되었을 때의 스택 용량 및 함수 이름을 표시합니다. LU가 열렸을 경우에는 실행 기록 메시지 입력 항목의 최대 개수가 정의됩니다. 이 버퍼가 꽉 찰 경우, MONMQ는 첫 번째 입력 항목으로 되돌아가서 첫 번째 메시지 및 그 이후의 메시지를 겹쳐줍니다. 추적 중 FFST가 생성되면 실패한 스레드에 대해 실패 지점에서 추적이 불가능하게

SHOW HISTORY

됩니다. 이는 오류 루틴이 생성한 추적 메시지가 버퍼를 채우는 것을 방지하기 위한 것입니다. 따라서 실행 기록 버퍼에 표시된 마지막 메시지가 FFST가 생성된 지점이 됩니다.

현재 노드에 있는 모든 MQSeries 관련 전역 섹션 표시

SHOW GLOBALS

이 명령은 현재 노드에 있는 모든 MQSeries 관련 전역 섹션을 표시합니다.

```
MQS1_shm_00000000 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=6128/383
MQS1_shm_01300010 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=1904/119
MQS1_shm_012c000f (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=464/87
MQS1_shm_012c000e (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=4112/514
MQS1_shm_012c000d (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=240/30
MQS1_shm_012c000c (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=528/132
MQS1_shm_012c000b (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=272/51
MQS1_shm_012c000a (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=4112/771
MQS1_shm_012c0009 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=272/51
MQS1_shm_012c0008 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=144/27
MQS1_shm_012c0007 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=528/99
MQS1_shm_012c0006 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=16/2
MQS1_shm_012c0005 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=128/24
MQS1_shm_012c0004 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=1968/492
MQS1_shm_012c0003 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=1904/476
MQS1_shm_012c0002 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=304/76
MQS1_shm_012c0001 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=1904/595
MQS1_shm_01280000 (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=16/6
MQS1_shm_fffffffe (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=16/0
MQS1_shm_fffffff (00000000) WRT DZRO PRM SYS Pgltcnt/Refcnt=144/63
```

mutex 테이블을 클라이언트 추적 프로세스로 송신하라는 신호를 대상 스레드에 보내기

SHOW MUTEX [chl=range]

이 명령은 스레드의 내부 mutex 테이블의 내용을 클라이언트 추적 프로세스로 송신하라는 신호를 대상 스레드에 보냅니다. 올바른 추적 마스크 비트를 설정해야 클라이언트가 이런 유형의 정보 데이터를 표시할 수 있다는 점에 유의하십시오. INF와 DGN 비트가 이 채널의 추적 마스크에서 사용 가능해야 합니다(381 페이지의 『마스크 비트 사

용/사용 안함』 참조). 예를 들어, **show mutex chl=2** 명령을 실행하면 다음과 같이 표시됩니다.

```
Mutex Utilisation for Process AMQZXA0.EXE - Pid 248 ***
0960 - Lock ID: 0100013c - Name: BKM3/@ipcc_m_1_24
0961 - Lock ID: 020006ca - Name: BKM3/@ipcc_m_1_23
0962 - Lock ID: 0b00061e - Name: BKM3/@ipcc_m_1_22
0963 - Lock ID: 0900068e - Name: BKM3/@ipcc_m_1_21
0964 - Lock ID: 0b00032f - Name: BKM3/@ipcc_m_1_20
0965 - Lock ID: 210006eb - Name: BKM3/@ipcc_m_1_19
0966 - Lock ID: 07000742 - Name: BKM3/@ipcc_m_1_18
0967 - Lock ID: 1e000075 - Name: BKM3/@ipcc_m_1_17
0968 - Lock ID: 0c0004dd - Name: BKM3_m_1_45
0969 - Lock ID: 0d00035a - Name: BKM3/@ipcc_m_1_16
0970 - Lock ID: 190000a1 - Name: BKM3/@ipcc_m_1_15
0971 - Lock ID: 1a0005a3 - Name: BKM3/@ipcc_m_1_14
0972 - Lock ID: 14000628 - Name: BKM3/@ipcc_m_1_13
0973 - Lock ID: 130005f3 - Name: BKM3/@ipcc_m_1_12
0974 - Lock ID: 0f0000dc - Name: BKM3_m_1_43
0975 - Lock ID: 02000095 - Name: BKM3_m_1_42
0976 - Lock ID: 2200053e - Name: BKM3_m_1_41
0977 - Lock ID: 020000fc - Name: BKM3_m_1_40
0978 - Lock ID: 31000113 - Name: BKM3_m_1_39
0979 - Lock ID: 02000555 - Name: BKM3_m_1_38
0980 - Lock ID: 2e000389 - Name: BKM3_m_1_37
0981 - Lock ID: 2300011f - Name: BKM3_m_1_36
0982 - Lock ID: 02000109 - Name: BKM3_m_1_35
0983 - Lock ID: 02000327 - Name: BKM3_m_1_34
0984 - Lock ID: 020004a8 - Name: BKM3_m_1_33
0985 - Lock ID: 02000453 - Name: BKM3_m_1_32
0986 - Lock ID: 260007ad - Name: BKM3_m_1_31
0987 - Lock ID: 0200060c - Name: BKM3_m_1_30
```

이 데이터는 실행 기록 파일의 줄 번호와 mutex 이름 및 시스템 잠금 ID를 표시합니다.

내부 이벤트 테이블을 클라이언트 추적 프로세스로 송신하라는 신호를 대상 스레드에 보내기

SHOW EVENTS [chl=range]

이 명령은 스레드의 내부 이벤트 테이블의 내용을 클라이언트 추적 프로세스로 송신하라는 신호를 대상 스레드에 보냅니다. 올바른 추적 마스크 비트를 설정해야 클라이언트가 이런 유형의 정보 데이터를 표시할 수 있다는 점에 유의하십시오. INF와 DGN 비트가 이 채널의 추적 마스크에서 사용 가능해야 합니다(381 페이지의 『마스크 비트 사용/사용 안함』 참조). 예를 들어, **show events chl=2** 명령을 실행하면 다음과 같이 표시됩니다.

SHOW EVENTS

```
Event Utilisation for Process AMQZXMA0.EXE - Pid 248 ***
1037 - Channel: 000003e0 - Name: BKM3/@ipcc_e_1_19
1038 - Channel: 000003d0 - Name: BKM3/@ipcc_e_1_18
1039 - Channel: 000003c0 - Name: BKM3/@ipcc_e_1_17
1040 - Channel: 000003b0 - Name: BKM3/@ipcc_e_1_14
1041 - Channel: 000003a0 - Name: BKM3/@ipcc_e_1_13
1042 - Channel: 00000390 - Name: BKM3/@ipcc_e_1_12
1043 - Channel: 00000380 - Name: BKM3/@ipcc_e_1_10
1044 - Channel: 00000370 - Name: BKM3/@ipcc_e_1_9
1045 - Channel: 00000360 - Name: BKM3/@ipcc_e_1_8
1046 - Channel: 00000330 - Name: BKM3/@ipcc_e_1_7
1047 - Channel: 00000300 - Name: BKM3/@ipcc_e_1_6
1048 - Channel: 000002f0 - Name: BKM3/@ipcc_e_1_5
1049 - Channel: 000002b0 - Name: BKM3_e_1_11
1050 - Channel: 000002a0 - Name: BKM3_e_1_10
1051 - Channel: 00000290 - Name: BKM3_e_1_9
1052 - Channel: 00000280 - Name: BKM3_e_1_8
1053 - Channel: 00000270 - Name: BKM3_e_1_7
1054 - Channel: 00000260 - Name: BKM3_e_1_6
1055 - Channel: 00000250 - Name: BKM3_e_1_5
1056 - Channel: 00000240 - Name: BKM3_e_1_4
1057 - Channel: 00000230 - Name: BKM3_e_1_3
1058 - Channel: 00000220 - Name: BKM3_e_1_2
1059 - Channel: 00000210 - Name: BKM3_e_1_1
1060 - Channel: 00000200 - Name: BKM3_e_1_0
1061 - Channel: 000001c0 - Name: BKM3/@ipcc_e_1_4
1062 - Channel: 000001b0 - Name: BKM3/@ipcc_e_1_3
1063 - Channel: 000001a0 - Name: BKM3/@ipcc_e_1_2
1064 - Channel: 00000190 - Name: BKM3/@ipcc_e_1_1
1065 - Channel: 00000180 - Name: BKM3/@ipcc_e_1_0
1066 - *** End of data ***
```

이 데이터는 실행 기록 파일의 줄 번호와 편지함 채널 번호 및 이벤트 이름을 표시합니다.

맵핑된 내부 공유 메모리 테이블을 클라이언트 추적 프로세스로 송신하라는 신호를 대상 스레드에 보내기

SHOW MEMORY [chl=range]

이 명령은 스레드의 맵핑된 내부 공유 메모리 테이블의 내용을 클라이언트 추적 프로세스로 송신하라는 신호를 대상 스레드에 보냅니다. 올바른 추적 마스크 비트를 설정해야 클라이언트가 이런 유형의 정보 데이터를 수신할 수 있다는 점에 유의하십시오. INF와 DGN 비트가 이 채널의 추적 마스크에서 사용 가능해야 합니다(381 페이지의 『마스크 비트 사용/사용 안함』 참조). 예를 들어, **show memory chl=2** 명령을 실행하면 다음과 같이 표시됩니다.

```
*** Shared Memory Utilisation for Process AMQZXMA0.EXE - pid/tid 248-1 ***
0942 - ShmId: 0248000f - Addr: 011d8000/01211fff - Perm: 950 - Size: 00038530 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/AMQ
0943 - ShmId: 0248000e - Addr: 00fbc000/011bdfff - Perm: 950 - Size: 002005f8 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/AMQ
0944 - ShmId: 0248000d - Addr: 00f1c000/00f39fff - Perm: 950 - Size: 0001d478 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/WLM
0945 - ShmId: 0248000c - Addr: 00cba000/00cfbfff - Perm: 944 - Size: 000405f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/HMEMSET.0
0946 - ShmId: 0248000b - Addr: 00c98000/00cb9fff - Perm: 944 - Size: 000205f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon005.0
0947 - ShmId: 0248000a - Addr: 00a96000/00c97fff - Perm: 944 - Size: 00200584 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon004.0
0948 - ShmId: 02480009 - Addr: 00a74000/00a95fff - Perm: 944 - Size: 000205f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon003.0
0949 - ShmId: 02480008 - Addr: 00a62000/00a73fff - Perm: 944 - Size: 000105f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon002.0
0950 - ShmId: 02480007 - Addr: 00a20000/00a61fff - Perm: 944 - Size: 000405f0 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/Anon001.0
0951 - ShmId: 02480006 - Addr: 00908000/00909fff - Perm: 950 - Size: 00001664 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/PLU
0952 - ShmId: 02480005 - Addr: 008f8000/00907fff - Perm: 950 - Size: 0000fff8 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/IPC
0953 - ShmId: 02480004 - Addr: 00802000/008f7fff - Perm: 950 - Size: 000f4838 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/IPC
0954 - ShmId: 02480003 - Addr: 00714000/00801fff - Perm: 950 - Size: 000ec718 Name: /mqc_root/mqm/qmgrs/BKM3/@ipcc/shmem/SUB
0955 - ShmId: 02480002 - Addr: 006ee000/00713fff - Perm: 944 - Size: 000253a8 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/zutSESSAN
0956 - ShmId: 02480001 - Addr: 00600000/006edfff - Perm: 944 - Size: 000ec710 Name: /mqc_root/mqm/qmgrs/BKM3/shmem/SUBPOOL.0
0957 - ShmId: 01280000 - Addr: 005f8000/005f9fff - Perm: 950 - Size: 00000454 Name: /var/mqm/errors
0958 - *** End of data ***
```


이 데이터는 실행 기록 파일의 줄 번호와 공유 메모리 ID, 가상 맵핑 주소 범위, 섹션 작성/맵핑에 사용된 플래그 및 섹션에 지정된 내부 MQSeries 이름을 표시합니다.

이름 및 16진수 ID별로 활성 MQSeries 구성요소 표시

SHOW COMPONENTS

이 명령은 모든 활성 MQSeries 구성요소를 이름 및 연관된 16진수 ID별로 표시합니다. 함수 표시 및 구성요소 선택과 같은 다른 MONMQ 표시 명령에서 이러한 16진수 ID를 사용합니다. 예를 들어, **show components** 명령을 실행하면 다음과 같이 표시됩니다.

```
00000001 - Data hardening
00000002 - Log management
00000003 - Object Catalogue
00000004 - Queue management
00000005 - Transaction Management
00000006 - Mobile Component
00000007 - Mobile Component
00000008 - Communications
0000000a - Object Authority Manager
0000000b - Logger
0000000d - LQM Kernal
0000000f - Administration App
00000010 - Administration App
00000013 - Command Server
00000014 - Remote queue processor
00000015 - XA Transaction Manager
00000016 - Data Conversion
00000017 - Common Services
00000018 - Common Services (overflow)
00000019 - Application Interface
0000001a - IPCC
0000001b - DCE Support
0000001c - Pluggable Services
0000001d - Agent
0000001e - XA Transaction Manager
0000001f - C++ Layer
00000020 - CLI
00000021 - Z Utilities
00000022 - Execution Controller
00000023 - App. Bindings
00000024 - Service Component
00000025 - Publish/Subscribe
00000026 - MMC Snap-in for Admin
00000027 - Web Administration
00000028 - KYG Services
00000029 - OVMS MQ kernel
```

지정한 구성요소 내의 함수 표시

SHOW FUNCTIONS [comp=hex]

이 명령은 지정한 구성요소 내의 모든 함수를 표시합니다. 구성요소는 16진수로 입력해야 합니다. 모든 활성 MQSeries 구성요소를 표시하려면 **SHOW COMPONENT**를 사용하십시오. 예를 들어, **show functions component=0x1f** 명령을 실행하면 다음과 같이 표시됩니다.

ONSTARTUP 시작

```
00000000 - ImqBinary::copyOut
00000001 - ImqBinary::pasteIn
00000002 - ImqCache::operator =
00000003 - ImqCache::moreBytes
00000004 - ImqCache::read
00000005 - ImqCache::resizeBuffer
00000006 - ImqCache::setDataOffset
00000007 - ImqCache::setMessageLength
00000008 - ImqCache::useEmptyBuffer
00000009 - ImqCache::write
0000000a - ImqDeadLetterHeader::pasteIn
0000000b - ImqDistributionList::openInfoPrepare
0000000c - ImqItem::structureIdIs
0000000d - ImqQueueManager::backout
0000000e - ImqQueueManager::begin
0000000f - ImqQueueManager::commit
00000010 - ImqQueueManager::connect
00000011 - ImqQueueManager::disconnect
00000012 - ImqMessageTracker::setAccountingToken
00000013 - ImqMessageTracker::setCorrelationId
00000014 - ImqMessageTracker::setGroupId
00000015 - ImqMessageTracker::setMessageId
00000016 - ImqObject::close
00000017 - ImqObject::closeTemporarily
00000018 - ImqObject::inquire
00000019 - ImqObject::open
0000001a - ImqObject::openFor
.....
```

프로세스가 시작된 지점에서 추적 활성화

```
ONSTARTUP [start] [lu=number] [chl=range]
```

이 명령은 프로세스가 시작되는 지점에서 추적이 활성화 되도록 합니다. 이 명령을 실행하면 시스템의 논리 이름 테이블에 논리 이름 MQS_DEF_TRACE가 정의되고 *lu channel* 형식으로 된 동등한 이름을 갖게 됩니다. MQSeries 프로세스가 시작되면 프로세스 초기화 루틴 내에서 이 논리 이름이 점검되고, 이름이 있으면 지정한 LU 및 채널 번호에 연결합니다. 채널 ID가 이미 할당된 경우에는 사용 가능한 다음 채널이 사용됩니다. 이 명령은 MQSeries 프로세스/스레드 작성의 초기 단계에서 추적이 필요할 때 유용하게 사용할 수 있습니다.

시동 후 즉시 MQSeries 프로세스가 추적되지 않도록 방지

```
ONSTARTUP [stop]
```

이 명령은 시스템 논리 이름 테이블에서 MQS_DEF_TRACE 논리 지정을 취소하여 시동 후 즉시 MQSeries 프로세스가 추적되지 않도록 합니다.

대상 스레드를 지정한 채널에 연결

```
CONNECT pid number [tid=number] [chl=range]
```

이 명령은 지정한 채널에 연결하라는 신호를 대상 스레드에 보냅니다. 채널을 지정하지 않으면 사용 가능한 첫 번째 채널이 사용됩니다.

지정한 채널에서 대상 스레드 연결 끊기

DISCONNECT [chl=*range*]

이 명령은 지정한 채널에서 연결을 끊으라는 신호를 대상 스레드에 보냅니다.

LU 추적 편지함에 기록된 실시간 추적 메시지 표시

TRACE START [node=*string*]

이 명령은 클라이언트 추적 프로세스를 시작하여 LU 추적 편지함에 기록된 실시간 추적 메시지를 표시합니다. 선택적 노드 매개변수는 지정한 노드에 창을 만들어 이 창으로 출력을 내보냅니다.

현재 클라이언트 프로세스 분리 및 종료

TRACE STOP

이 명령은 현재 클라이언트 프로세스가 추적 편지함에서 분리되어 종료되도록 합니다. 현재 이 편지함에 기록 중인 모든 스레드는 더 이상 메시지를 기록할 수 없습니다.

```
MQT> trace stop
Circular buffering has been disabled for process 24d thread 1
Circular buffering has been disabled for process 24c thread 1
Circular buffering has been disabled for process 24b thread 1
Circular buffering has been disabled for process 248 thread 1
Disconnecting thread pid : 24d, tid : 1 from channel 0 ..... OK
Disconnecting thread pid : 24c, tid : 1 from channel 1 ..... OK
Disconnecting thread pid : 24b, tid : 1 from channel 2 ..... OK
Disconnecting thread pid : 248, tid : 1 from channel 3 .....OK
*** Trace ended - no processes connected ***
```

위의 출력 내용이 추적 클라이언트 창에 나타납니다.

추적 데이터 지정

SELECT [component] AND/OR [function] OR [*fname*]

이 명령은 추적할 구성요소/함수 조합을 8개까지 지정할 수 있도록 합니다. 다른 모든 추적 데이터는 필터되어 제외됩니다. 함수 이름, 구성요소, 구성요소/함수 중 하나를 지정할 수 있습니다. 선택한 구성요소/함수가 올바르면 필터 테이블에 기록됩니다. 입력 항목이 없으면 모든 구성요소/함수가 디폴트로 추적됩니다.

SELECT

매개변수 없이 **SELECT** 명령을 입력하면 필터 테이블의 내용이 표시됩니다. 출력된 각 줄에는 16진수로 된 테이블 색인 항목, 구성요소, 함수 및 함수에 대한 텍스트 이름이 표시됩니다. 구성요소만 입력한 경우에는 이 구성요소 내의 모든 함수가 추적됩니다. 이는 함수 값에 대해 0xffff로 표시됩니다.

예를 들어, **SELECT** 명령만 실행하면 다음과 같이 표시됩니다.

```
Ch1:0 - Cmp/fnc selection criteria
ALL component/functions
```

다음 명령 세트를 실행하면,

```
MQT>select fname="kill"
MQT>select comp=0x1f
MQT>select comp=0x20 func=0x3
MQT>select
```

아래와 같이 표시됩니다.

```
Ch1:0 - Cmp/fnc selection criteria
Idx: 0 - Cmp: 00000029 - Fnc: 00000005 - Name - kill
Idx: 1 - Cmp: 00000019 - Fnc: 0000ffff - Name -
Idx: 2 - Cmp: 00000016 - Fnc: 00000003 - Name - vqiAddCacheEntry
```

추적 필터 테이블에서 단일 입력 항목 제거

```
DESELECT INDEX=<0:7>
```

이 명령은 테이블 색인 매개변수로 지정한 단일 입력 항목을 추적 필터 테이블에서 제거합니다. 8개의 입력 항목이 모두 비어 있으면 모든 구성요소/함수가 추적됩니다. 예를 들어, **select** 명령을 실행하면 다음 입력 항목이 색인과 함께 표시됩니다.

```
Ch1:0 - Cmp/fnc selection criteria
Idx: 0 - Cmp: 00000029 - Fnc: 00000005 - Name - kill
Idx: 1 - Cmp: 00000019 - Fnc: 0000ffff - Name -
Idx: 2 - Cmp: 00000016 - Fnc: 00000003 - Name - vqiAddCacheEntry
```

다음 선택 취소 명령은 지정한 프로세스나 함수를 제거합니다.

```

MQT> dese1 index=0
MQT> dese1 index=2
MQT>select
Ch1:0 - Cmp/fnc selection criteria
Idx: 1 - Cmp: 00000019 - Fnc: 0000ffff - Name -
-----
MQT> deselect index=1
MQT> select
Ch1:0 - Cmp/fnc selection criteria
ALL component/functions
-----

```

2진 파일에 추적 메시지를 기록하는 클라이언트 프로세스

OPEN BINARY [filename=*string*]

이 명령은 추적 메시지 2진 파일을 열고 클라이언트 프로세스가 이 파일에 실시간 추적 메시지를 기록하도록 합니다. 나중에 이 파일을 사용하여 MQSeries 응용프로그램 성능을 분석할 수 있습니다. 디폴트 파일 이름은 `mqs_root:[mqm.errors]mqs_buffer_xx.bin`이고, 여기에서 `xx`는 LU 번호입니다.

2진 추적 메시지 파일 닫기

CLOSE BINARY

이 명령은 지정한 LU 2진 추적 파일을 닫습니다.

텍스트 파일에 추적 메시지를 기록하는 클라이언트 프로세스

OPEN TEXT [filename=*string*]

이 명령은 읽을 수 있는 텍스트 파일을 열고 클라이언트 프로세스가 이 파일에 형식화된 2진 추적 메시지를 기록하도록 합니다. 나중에 간단히 DCL 입력 명령이나 편집 명령을 사용하여 이 파일을 볼 수 있습니다. 디폴트 파일 이름은 `mqs_root:[mqm.errors]mqs_buffer_xx.lis`이고, 여기에서 `xx`는 LU 번호입니다. 이 유형의 출력 파일을 사용하면 파일을 읽기 위한 전처리 작업이 필요하지 않으므로 편리합니다. 그러나 2진 파일보다 디스크 공간을 많이 사용하는 단점이 있습니다.

텍스트 추적 메시지 파일 닫기

CLOSE TEXT

이 명령은 지정한 LU 텍스트 추적 파일을 닫습니다.

시간 소인 메시지

ENABLE TIMESTAMP [chl=*range*]

ENABLE TIMESTAMP

이 명령은 채널 정의 테이블에 Timestamp 플래그를 설정합니다. 이 명령을 사용하면 MQSeries 프로세스가 각 메시지에 현재 시간을 소인할 수 있습니다. 성능 분석을 위해 2진 추적 파일을 사용할 때는 이 플래그를 설정해야 합니다.

메시지 시간 소인 정지

DISABLE TIMESTAMP [chl=*range*]

이 명령은 채널 정의 테이블에 설정한 Timestamp 플래그 설정을 취소합니다(379 페이지의 『시간 소인 메시지』 참조).

추적 사용

ENABLE TRACE [chl=*range*]

이 명령은 채널 정의 테이블에 RTime 추적 플래그를 설정합니다. 이 플래그를 사용하면 추적 메시지가 추적 클라이언트로 송신되거나 송신되지 않도록 할 수 있습니다. 스레드가 연결되어 있고 추적 클라이언트가 있으면 이 스레드의 연결을 끊는 대신, 예를 들어, TRACE START를 사용하여 채널을 연결하거나 끊을 수 있습니다.

추적 사용 안함

DISABLE TRACE [chl=*range*]

이 명령은 채널 정의 테이블에 설정한 RTime 추적 플래그 설정을 취소합니다(『추적 사용』 참조).

메시지 실행 기록 저장

ENABLE HISTORY [chl=*range*]

이 명령은 지정한 채널의 채널 정의 테이블에 설정한 History 플래그를 설정합니다. 이 명령은 추적 메시지를 LU 순환 버퍼에 기록하라는 신호를 연결된 스레드에 보냅니다. 추적된 프로세스가 메시지를 기록하므로 클라이언트 프로세스가 필요하지 않습니다. 추적 순환 버퍼의 크기는 **open** 명령으로 LU를 작성할 때 정의됩니다. 이 버퍼는 마지막 입력 항목이 기록되면 처음으로 되돌아갑니다. LU 정의 테이블의 Next Log 레코드 필드는 버퍼에서 다음 레코드가 기록될 위치를 지정합니다.

메시지 실행 기록 사용 안함

DISABLE HISTORY [chl=*range*]

이 명령은 지정한 채널의 채널 정의 테이블에 설정한 History 플래그 설정을 취소합니다(380 페이지의 『메시지 실행 기록 저장』 참조).

메시지 실행 기록 삭제

DELETE HISTORY [chl=*range*]

이 명령은 순환 실행 기록 버퍼에 있는 모든 메시지를 삭제합니다. 이 명령은 버퍼에 기록 중인 프로세스가 있을 때도 수행할 수 있기 때문에 메시지를 삭제하기 전에 실행 기록을 사용 중지할 필요가 없습니다.

실행 기록 용량 설정

SET [depth]

이 명령은 추적 클라이언트 창에 출력되는 최대 스택 용량을 제어합니다. 이 값보다 큰 메시지는 출력되지 않지만, 사용할 경우 2진 추적 파일과 실행 기록 버퍼에 나타납니다. 디폴트 값 0을 사용하면 스택 용량에 상관 없이 모든 메시지가 출력됩니다. 2진 파일에 분석 데이터를 기록할 때는 이 값을 아주 낮게(예: 1) 설정하는 것이 좋습니다. 전체 스택을 표시하면 클라이언트 프로세스의 성능에 좋지 않은 영향을 줄 수 있습니다.

채널의 스택 및 실행 기록 데이터 재설정

SET [free] [chl=*range*]

이 명령은 지정한 채널을 재설정합니다. 기존의 모든 스택과 실행 기록 데이터는 삭제되고 채널 할당이 해제되어 다시 사용할 수 있는 상태가 됩니다.

마스킹 비트 사용/사용 안함

SET [mask=*var*] [chl=*range*]

이 명령은 연결된 스레드 비트 마스크 필드 내에서 마스크 비트를 사용하거나 사용하지 않습니다. 각 비트는 MQSeries 프로세스가 생성한 메시지 유형을 나타냅니다. 이 명령을 사용하여 추적할 메시지 유형을 필터할 수 있습니다. 메시지 유형은 다음과 같습니다.

```
MQT>set mask = 0xffffffff chl=1
MQT>show mask chl=1
```

SET mask

```
Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

이러한 메시지 유형을 여러 개 결합하려면 다음과 같이 OR 기호를 사용하여 각 마스크 유형을 구분하십시오.

```
MQT>set mask =0x0 chl=1 MQT>show mask chl=1
```

```
Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

```
MQT>set mask = mtx | evt | fent chl=1
MQT>show mask chl=1
```



```
Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

각 마스크는 8개의 마스크 유형으로 구성되며, 이 중 특정 메시지 유형에 대해 사용/사용 안함을 전환할 수 있습니다. 예를 들어, 함수 시작점만 고려할 때는 **set mask = fent** 명령을 입력하십시오.

채널에 색상 설정

```
SET COLOR [chl=range]
```

이 명령은 지정한 채널에 색상을 연관시킵니다. 이 채널과 관련된 모든 출력은 색상을 변경하거나 채널을 재설정하기 전까지는 지정한 색상으로 표시됩니다. 이 명령은 한 출력 스트림 내의 서로 다른 스레드 메시지를 구별하거나 강조표시할 때 유용합니다. 예를 들어, 다음 명령

```
MQT> set color=yellow chl=2
MQT> set color=blue chl=0
MQT> sho chan chl=0:3 connected
```

아래와 같이 표시됩니다.

Chl	Pid/Tid	Mailbox	Stack	History	RTime	Time	Mask	Process Name
0	00000245/1	800b0330	4	0	0	0	fffffff	AMQZLAA0.EXE
1	00000244/1	800b0200	8	0	0	0	fffffff	RUNMQCHI.EXE
2	00000243/1	800b01e0	10	0	0	0	fffffff	AMQRRMFA.EXE
3	00000242/1	800b01c0	4	0	0	0	fffffff	AMQZLLP0.EXE

여기에서 Channel 0은 파란색이고 Channel 2는 노란색입니다.

출력의 경로를 파일에 재지정

```
SET OUTPUT [filename=string]
```

SET OUTPUT

이 명령은 모든 출력을 지정한 파일로 전달하고 출력이 표시되지 않도록 합니다. **output** 명령을 다른 명령에서 매개변수로 사용한 경우에는 해당 명령에만 적용됩니다. 오류는 이전과 마찬가지로 파일이 아닌 표시 장치에 보고된다는 점을 주의하십시오. 올바른 추적 데이터만 지정한 파일에 기록됩니다.

추적 2진 파일 분석

ANALYSE [component] [function] [unit=xx]

이 명령은 이전에 추적 세션에서 사용된 추적 2진 파일의 내용을 분석합니다. 분석할 구성요소나 함수를 지정할 수는 있지만, 해당 파일에 이 구성요소나 함수와 관련된 데이터가 있을 때만 적용됩니다. 예를 들어, 파일이 생성되었을 때 추적 마스크를 지정하거나 특정 구성요소를 선택하면 이렇게 선택한 항목만 2진 파일에서 찾고 이 기준에 맞지 않는 구성요소나 함수는 분석할 수 없습니다.

주: 전체 명령을 사용하려면 *S*를 사용하여 ANALYSE라고 입력하십시오. 이 명령을 OpenVMS의 ANALYZE 명령과 혼동하지 마십시오. 이 두 명령은 서로 다릅니다.

단위 매개변수는 분석의 시간 단위를 지정하는 데 사용되고, 다음 값(xx) - (초, 밀리초, 마이크로초, 나노초) 중 하나를 사용할 수 있습니다. 디폴트는 밀리초입니다.

예를 들어, **analyse unit=micro:** 명령을 사용하여 출력을 마이크로초 단위로 표시할 수 있습니다. 다음은 이 명령의 샘플 출력입니다.

```

=====
                        COMPONENT :- Common Services
=====
Calls  Minimum    Average    Maximum    Total    Function
42     0.00      66.41     311.78    2789.15  xcsRequestMutexSem
42     0.00      96.98     222.64    4072.98  xcsReleaseMutexSem
6      0.00     138.50     286.11     831.00   xcsResetEventSem
5      0.00    10112.60   10159.51   50563.01 xcsWaitEventSem
6      0.00     564.74    1029.23    3388.45  xcsCheckExtendMemory
42     0.00     51.32     266.86    2155.41  xllSemReq
42     0.00     73.05     159.17    3068.16  xllSemRel
=====
                        COMPONENT :- Common Services (overflow)
=====
Calls  Minimum    Average    Maximum    Total    Function
36     0.00     41.15     122.06    1481.35  xcsCheckProcess
6      0.00     37.92     68.35     227.52   xihGetConnSPDetailsFromList
6      0.00     65.26    114.25     391.58   xihHANDLEtoSUBPOOLFn
6      0.00     12.69     23.44    1267.49  xihGetNextSetConnDetailsFromList
6      0.00     12.53     22.46     75.19   xcsRequestThreadMutexSem
6      0.00     12.37     22.46     74.21   xcsReleaseThreadMutexSem
=====
                        COMPONENT :- IPCC
=====
Calls  Minimum    Average    Maximum    Total    Function
5      0.00   10589.97   11029.84   52949.85 xcpReceiveOnLink
=====
                        COMPONENT :- CLI
=====
Calls  Minimum    Average    Maximum    Total    Function
6      0.00     1.95     1.95     11.72    zapInquireStatus
=====
                        COMPONENT :- Execution Controller
=====
Calls  Minimum    Average    Maximum    Total    Function
6      0.00     954.46   3275.18   11726.79 zxcProcessChildren
6      0.00     12.69     22.46     76.17   zxcStartWLMserver
=====
                        COMPONENT :- OVMS MQ kernel
=====
Calls  Minimum    Average    Maximum    Total    Function
36     0.00     15.49     99.60     557.58   kill
6      0.00     0.65     3.91       3.91    vms_mapgbl
84     0.00     11.17     88.16     938.68   vms_get_lock
84     0.00     42.41    221.94    3562.54  vms_mtx
12     0.00     44.51    149.40     534.14  vms_get_mbx_chan
11     0.00    4651.77  10137.05  51169.42 vms_evt
6      0.00     1.63     4.88       9.76    vms_check_health
=====

```

열은 다음과 같습니다.

Calls	추적 세션 중 함수에 대한 입력 항목 개수입니다.
Minimum	함수 내에서 소요된 가장 짧은 시간입니다.
Average	함수를 호출한 총 시간을 호출 개수로 나눈 값입니다.
Maximum	함수 내에서 소요된 가장 긴 시간입니다.
Total	이 함수에 이루어진 모든 호출에 소요된 총 시간입니다.

ANALYSE

Function 함수 이름입니다.

주: 분석 범위가 2진 추적 파일의 내용입니다. 추적 2진 파일을 열고 원하는 시간에 추적 사용/사용 안함으로 설정하여 분석 경계를 정의하는 것은 사용자가 선택할 수 있습니다.

MQSeries 스레드의 현재 상태 표시

FFST [chl=*range*]

이 명령은 대상 채널에 연결된 스레드가 FFST를 강제 실행하도록 합니다. 이 명령은 대상 스레드의 실행 경로에는 영향을 주지 않습니다. 이 명령을 사용하여 MQSeries 스레드의 현재 상태를 스냅샷으로 볼 수 있습니다. 잘라낸 FFST에는 스레드 자원 사용량, 특권 및 기타 유용한 시스템 정보가 포함됩니다. FFST의 헤더에는 MONMQ에서 FFST가 작성되었으며(아래 참조) 실패의 결과가 아님이 분명하게 표시됩니다.

다음은 샘플 출력의 일부입니다.

MQSeries First Failure Symptom Report

=====

```
Date/Time      :- Wednesday November 12  10:59:38 GMT 2000
Host Name      :- CATWMN (Unknown)
PIDS           :- 5697175
LVLS           :- 510
Product Long Name :- MQSeries for OpenVMS Alpha
Vendor         :- IBM
Probe Id       :- VM026000
Application Name :- MQM
Component      :- vms_evt
Build Date     :- Oct 22 2000 (Collector)
Userid         :- [400,400] (SYSTEM)
Program Name   :- AMQZXMA0.EXE
Process        :- 00000248
Thread         :- 00000001
QueueManager   :- BKM3
Major Errorcode :- xecF_E_UNEXPECTED_SYSTEM_RC
Minor Errorcode :- OK
Probe Type     :- MSGAMQ6119
Probe Severity  :- 2
Probe Description :- AMQ6119: An internal MQSeries error has occurred
                  (***) FORCED FFST BY USER (***)
Comment1       :- *** FORCED FFST BY USER ***
Comment2       :- -SYSTEM-S-NORMAL, normal successful completion
```

etc.....

추적 닫기 및 MONMQ 종료

EXIT

이 명령은 CLOSE 명령을 수행하고 MONMQ를 종료합니다.

추적을 닫지 않고 MONMQ 종료

QUIT

이 명령은 CLOSE 명령을 수행하지 않고 MONMQ를 종료합니다. 이 명령은 추적을 계속 실행하면서 MONMQ만 종료하려고 할 때 유용합니다. 이후에 MONMQ를 활성화하면 이전 추적 세션이 계속됩니다.

MONMQ를 사용하여 공유 메모리 관리

큐 관리자에 오류가 발생하거나 OpenVMS stop/id 명령을 사용하여 강제 종료하는 것과 같은 일반적이지 않은 상황에서는 큐 관리자가 MQSeries 공유 메모리 세그먼트를 자동으로 삭제하지 못하는 경우가 발생할 수 있습니다. 이러한 상황이 발생하면, 큐 관리자가 이미 실행 중임을 **strmqm**에서 보고하기 때문에 큐 관리자를 재시작할 수 없습니다.

MONMQ는 현재 존재하는 MQ 공유 메모리(전역 섹션)를 나열하고 이러한 공유 메모리 섹션을 삭제할 수 있습니다.

주: 따라서 MONMQ 유틸리티를 사용하여 공유 메모리 세그먼트를 삭제하기 전에 큐 관리자를 모두 종료했는지 확인해야 합니다. 실행 중인 큐 관리자의 공유 메모리를 삭제하면 큐 관리자에 오류가 발생하고 큐 파일이 손상될 수 있습니다.

MONMQ SHOW PROCESS 명령을 사용하여 실행 중인 MQ 프로세스가 있는지 확인할 수 있습니다. 오류가 발생한 큐 관리자에서 실행 중인 프로세스가 있고 **endmqm** 명령으로 이 프로세스를 정지할 수 없을 때는 OpenVMS DCL 명령 **stop /id=<pid>**를 사용할 수 있습니다.

다음 명령을 사용하여 실행 중인 MQSeries 프로세스가 있는지 확인하십시오.

```
MQT> show process
```

QUIT

MQ Processes

PID	Proc Name	Image	Process	WS Size	WS Peak	Virt Peak	Gbl Pg Cnt	Prc Pg Cnt	Total Mem
-----	-----------	-------	---------	---------	---------	-----------	------------	------------	-----------

List the shared memory global sections that currently exist

```
MQT> show globals
MQS1_shm_2695000a (00000000) WRT   DZRO PRM SYS Pgltcnt/Refcnt=4112/514
MQS1_shm_26950009 (00000000) WRT   DZRO PRM SYS Pgltcnt/Refcnt=272/34
MQS1_shm_ffffffff (00000000) WRT   DZRO TMP SYS Pgltcnt/Refcnt=144/63
MQS1_shm_00000000 (00000000) WRT   DZRO TMP SYS Pgltcnt/Refcnt=6304/394
```

다음 명령을 사용하여 현재 존재하는 공유 메모리 전역 섹션을 나열하십시오.

```
MQT> show globals
```

이러한 전역 섹션을 삭제하십시오.

```
MQT> delete
Deleted global section: MQS1_shm_2695000a
Deleted global section: MQS1_shm_26950009
Deleted global section: MQS1_shm_ffffffff
sys$delgbl - unable to delete section MQS1_shm_00000000
```

MONMQ가 이 섹션을 사용하기 때문에 MQS_shm_00000000 섹션 삭제 중 오류가 발생할 것입니다. 이제 다음 명령을 발행하여 MONMQ를 종료할 수 있습니다.

```
MQT> exit
```

모든 큐 관리자가 프로세스를 정지했다고 확인하는 경우에는 스크립트 내에서 삭제 명령을 사용하십시오.

```
$ monmq delete
Deleted global section: MQS1_shm_2695000a
Deleted global section: MQS1_shm_26950009
Deleted global section: MQS1_shm_ffffffff
sys$delgbl - unable to delete section MQS1_shm_00000000
```

MONMQ의 스크립트 및 매크로

MONMQ 내에서 또는 명령 프롬프트에서 MONMQ 명령 스크립트를 실행할 수 있습니다. 스크립트는 일련의 데이터를 수집하거나 MONMQ 환경을 구성할 때 유용하게 사용할 수 있습니다. MONMQ 시작 SYS\$MANAGER:MQS_TRACE_STARTUP.MQT에서 스크립트가 실행되어 MONMQ 추적 변수를 구성합니다.

주: 스크립트가 현재 디렉토리에 없으면 스크립트의 전체 경로 이름을 따옴표로 묶어야 합니다. 예를 들면 다음과 같습니다.

```
MQT> ! "sys$manager:test.mqt"
```

일반적인 작업이나 반복적인 작업을 간단하게 하는 매크로를 정의할 수도 있습니다. 매크로 선언은 다음 세부분으로 이루어 집니다.

1. 첫 번째 부분은 매크로 이름이며, 이 이름은 고유한 명령 이름이어야 합니다.
2. 두 번째 부분은 매크로 본문입니다. 매크로 본문은 여러 줄에 걸쳐 입력할 수 있으며 MQSeries 명령 목록으로 구성됩니다. 매크로 본문은 { 및 }로 구분합니다. 여러 줄로 된 매크로 본문을 선언하면 MONMQ 프롬프트가 ****MACRO>**로 변경됩니다. 모든 \$n(여기에서 n은 한 자리 숫자)은 매크로 명령행에서 매개변수 n으로 바뀝니다.
3. 매크로 정의의 세 번째 부분은 도움말 <macroname>을 사용하면 표시되는 간단한 도움말 텍스트 설명입니다. 도움말 텍스트는 다음표로 묶어야 합니다.

매크로를 선언할 때는 선언할 시기를 잘 고려해야 합니다. 왜냐하면 매크로는 아주 빨리 처리되지만 일부 MONMQ 명령이 작업을 수행하라는 신호를 리모트 프로세스에 보내면 이 작업이 완료되어야만 다음 매크로 명령을 시작할 수 있기 때문입니다.

이런 이유로 어떤 경우에는 선언을 잠시 지연할 필요가 있습니다. 이를 수행하려면, 10분의 1초 단위로 지정된 지연 매개변수가 있는 **sleep** 명령을 사용하십시오.

다음 명령을 입력하면 채널의 연결을 끊고 추적 마스크를 재설정하여 채널 사용을 중지하는 매크로를 작성할 수 있습니다.

주: “;”을 분리 문자로 사용하면 한 줄에 여러 개의 MONMQ 명령을 넣을 수 있습니다.

```
MQT> declare tmpchl intrange "variable to hold a chl range temporarily"
MQT> macro remove { set tmpchl = chl ; dis chl= $1 ; sleep delay=5
**MACRO> set mask=0xffffffff chl= $1 ; set free chl = $1
**MACRO> set chl=tmpchl
**MACRO> } "A macro to disconnect and free channels Param: chl number"
MQT> help remove

VERB remove:
A macro to disconnect and free channels Param: chl number

Macro text:
set tmpchl = chl ; dis chl= $1 ; sleep delay=5 ; set mask=0xffffffff chl=$1
; set free chl = $1 ; set chl=tmpchl
MQT>remove 4
ok - process disconnected process 282 from channel 4
```

샘플 추적 세션

이 절에서는 각 MONMQ 명령을 순서대로 표시하는 일반적인 추적 세션에 대해 설명합니다. 이 샘플은 실행 중인 큐 관리자의 실행 제어기와 관련 에이전트의 기본 스레드를 추적합니다.

추적을 시작하려면 다음 조건이 충족되어야 합니다.

- 추적할 큐 관리자 시작 - STRMQM BKM3
- sys\$manager:mqs_trace_startup.mqt에 미리 설치된 디폴트 명령 이외의 추가 명령이 없는지 점검합니다.
- MQS_DEF_TRACE 논리가 정의되어 있지 않은지 점검합니다. 이 논리가 정의되어 있으면 MONMQ에서 **ONSTARTUP END**를 수행하십시오.

momq를 시작하십시오.

```
>mommq
```

MONMQ 프롬프트가 표시됩니다.

```
MQT>
```

채널이 10개이고 ID가 0인 LU 하나와 100개의 메시지가 있는 실행 기록 버퍼를 여십시오(주: 일반적인 추적에서 메시지 100개는 너무 적을 수 있습니다. 보통 1000개가 적당합니다).

```
MQT> open lu=0 nochls=10 buffer=100  
ok - LU:0 opened
```

LU1 정의를 표시하십시오.

```
MQT> show seg lu=1
```



```
Trace LU          : 1
Mailbox name     : MQS_TRC_MBX_1
Device name     : MBA431:
Status          : Disabled
Mailbox channel  : 384
History buffer size : 100
Threads mapped # : 1
Time stamping   : Enabled
Max channels #  : 10
Display depth   : 0
Text filename   :
Binary filename :
Last status     : 1
Connection map[0] : 0
=====
```

MQSeries 프로세스를 표시하십시오.

```
MQT> show process
```

PID	Proc_Name	Image	Process	WS_Size	WS_Peak	Virt_Peak	Gbl_Pg_Cnt	Prc_Pg_Cnt	Total_Mem
2A00023D	BKM1_AG	AMQZLAA0	Agent	23152	16576	203776	3616	12960	16576
2A00023C	BKM1_CI	RUNMQCHI	Run Chan Init	8752	6208	180832	1840	4368	6208
2A00023B	BKM1_RM	AMQRRMFA	Repository Mgr	11152	8144	185360	2224	5920	8144
2A00023A	BKM1_CP	AMQZLLP0	Checkpoint	8752	6384	185952	1920	4464	6384
2A000239	BKM1_LG	AMQHASMx	Logger	8752	6288	182016	2080	4208	6288
2A000238	BKM1_EC	AMQZXMA0	EC	20752	15232	203792	3536	11680	15216
2A000128	_FTA4:	MONMQ	MONMQ Utility	8400	8528	198736	2224	3584	5808

									52112

실행 제어기와 에이전트 프로세스를 식별한 다음 이를 각각 채널 1과 2에 연결하십시오.

```
MQT>connect pid=0x238 tid=1 lu=1 chl=1
MQT>connect pid=0x23D tid=1 lu=1 chl=2
```

연결 세부사항을 점검하십시오.

```
MQT>show channel full connected lu=1
```

샘플 추적 세션

```
Pid/Tid           : 2a000bc/1
Status            : *** Connected ***
Process name      : AMQZXMA0.EXE
Assigned LU       : 1
Channel no.       : 1
Mailbox channel   : 800c03f0
Current stack depth : 4
Circular logging  : Disabled
Next log entry    : 0
Realtime tracing  : Disabled
Time stamping     : Disabled
Trace mask        : ffffffff
Step mode         : Off
No Wait           : On
Last QIO status   : 0
Mapped address    : 1242000-126bfff
=====
Pid/Tid           : 2a000c1/1
Status            : *** Connected ***
Process name      : AMQZLAA0.EXE
Assigned LU       : 1
Channel no.       : 2
Mailbox channel   : 800c03f0
Current stack depth : 4
Circular logging  : Disabled
Next log entry    : 0
Realtime tracing  : Disabled
Time stamping     : Disabled
Trace mask        : ffffffff
Step mode         : Off
No Wait           : On
Last QIO status   : 0
Mapped address    : 1376000-139ffff
=====
```

chl, lu, tid에 디폴트 값을 설정하면 후속 명령을 실행할 때마다 항목을 입력하는 데 소요되는 시간을 줄일 수 있습니다.

```
MQT>default chl=1:2 lu=1 tid=1
```

서로 다른 추적 메시지를 구별할 수 있도록 채널 색상을 설정하십시오. 다음 두 명령에는 chl 매개변수가 지정되어 있는데, 이는 디폴트 값(1:2)을 사용하면 두 채널이 노란색과 청록색으로 설정되기 때문입니다.

```
MQT>set chl=1 color=yellow
MQT>set chl=2 color=cyan
```

이제 채널을 표시하십시오.

```
MQT>show channels
```

Ch1	Pid/Tid	Mailbox	Stack	History	RTime	Time	Mask	Process Name
1	2a0000bc/1	800c03f0	4	0	0	0	ffffffff	AMQZXMA.EXE
2	2a0000c1/1	800c0360	4	0	0	0	ffffffff	AMQLAA0.EXE

이제 두 프로세스 모두 채널에 연결되었으므로 스택을 조사할 수 있습니다.

```
MQT>show stacks
```

```
0001- 00:00:00.00 03 - 01 -->| ExecCtrlrMain
0002 - 00:00:00.00 03 - 02 ---->| zcpReceiveOnLink
0003 - 00:00:00.00 03 - 03 ---->| xcswaitEventSem
0004 - 00:00:00.00 03 - 04 ---->| vms_evt
0001- 00:00:00.00 03 - 01 -->| zlaMain
0002 - 00:00:00.00 03 - 02 ---->| zcpReceiveOnLink
0003 - 00:00:00.00 03 - 03 ---->| xcswaitEventSem
0004 - 00:00:00.00 03 - 04 ---->| vms_evt
```

이 두 채널에 시간 소인을 사용하면 두 프로세스 중 하나가 보류되고 있는지 여부를 확인할 수 있습니다.

```
MQT>enable timestamp
MQT>show stack
```

```
0001- 00:00:00.00 03 - 01 -->| ExecCtrlrMain
0002 - 12:36:20.18 03 - 02 ---->| zcpReceiveOnLink
0003 - 12:36:20.81 03 - 03 ---->| xcswaitEventSem
0004 - 12:36:20.83 03 - 04 ---->| vms_evt
0001- 00:00:00.00 03 - 01 -->| zlaMain
0002 - 12:36:20.18 03 - 02 ---->| zcpReceiveOnLink
0003 - 12:36:20.81 03 - 03 ---->| xcswaitEventSem
0004 - 12:36:20.83 03 - 04 ---->| vms_evt
```

이제 올바른 시간 소인이 있는 몇몇 메시지가 있음을 볼 수 있습니다. 이는 두 프로세스가 활성 프로세스임을 표시합니다. 이 경우 두 프로세스는 시간 종료가 10초인 이벤트 루프에 있습니다. 이 시간 종료는 시간 소인 데이터가 변경될 때까지 **show stack** 명령을 수행하여 메시지 시간 소인을 통해 점검할 수 있습니다.

이제 실행 기록을 사용하여 각 프로세스가 순환 버퍼에 해당 추적 메시지를 기록하도록 강제 실행할 수 있습니다.

샘플 추적 세션

```
MQT>enable history
MQT> show history
```

이 때는 모든 추적 메시지가 버퍼에 기록됩니다. 추적 마스크와 구성요소/함수 테이블을 표시하여 이를 점검할 수 있습니다.

```
MQT>show mask
```

```
Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
Trace Mask for Channel 2
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
```

```
MQT> select
```

```
Ch1:1 - Cmp/fnc selection criteria
ALL component/functions
-----
```

```
Ch1:2 - Cmp/fnc selection criteria
ALL component/functions
-----
```

```

Trace Mask for Channel 1
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
Trace Mask for Channel 2
Bit 00 - (fent) function entry
Bit 01 - (fout) function exit
Bit 02 - (ferr) function exit with error
Bit 03 - (fxx) missing function exit
Bit 04 - (dgn) diagnostic messages
Bit 05 - (shm) shared memory
Bit 06 - (spl) spinlocks
Bit 07 - (evt) events
Bit 08 - (mtx) mutexes
Bit 09 - (prc) process msgs
Bit 10 - (msc) miscellaneous
Bit 11 - (inf) informational
Bit 12 -
    
```

이제 특정 메시지 유형에 대해 중점적으로 살펴보도록 합니다. 예를 들어, 공유 메모리 진단 메시지만 고려한다고 가정합니다.

```

MQT>set mask=shm
MQT>show mask
    
```

이제 두 프로세스 모두 진단 메모리 유형의 메시지만 버퍼에 기록하게 됩니다. 버퍼를 삭제하고 몇 초 동안 대기한 다음 버퍼의 내용을 다시 조사합니다.

```

MQT>clear history
(wait a few seconds)
MQT> show history
    
```

샘플 추적 세션

```
*** Trace History Ch1:1 ***
0990 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0991 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0992 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0993 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0994 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0995 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0996 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0997 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0998 - 00:00:00.00 00 - 04 ..... | vms_mapgbl key : ffffffff - addr : 0/0
0999 - 00:00:00.00 00 - 04 ..... | vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
*** End of buffer ***
*** Trace History Ch1: ***
*** End of buffer ***
```

이제 추적 출력에 있는 이벤트 유형 진단 메시지를 표시하십시오. 마스크를 설정한 후 몇 초 동안 대기해야 합니다.

```
MQT>set mask=evt | shm
(wait a few seconds)
MQT>show history
```

```

*** Trace History Ch1:1 ***
0977 - 00:00:00.00 00 - 04 .....| vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0978 - 00:00:00.00 00 - 04 .....| vms_evt Event BKM3/@ipcc_e_1_2 TIMEOUT
0979 - 00:00:00.00 00 - 04 .....| vms_evt rc = 1
0980 - 00:00:00.00 00 - 04 .....| vms_mapgbl key : ffffffff - addr : 0/0
0981 - 00:00:00.00 00 - 04 .....| vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0982 - 00:00:00.00 00 - 04 .....| vms_evt Reset on mailbox BKM3/@ipcc_e_1_2 : tout = -1
0983 - 00:00:00.00 00 - 05 .....| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0984 - 00:00:00.00 00 - 05 .....| vms_get_mbx_chan Returning key 1a0
0985 - 00:00:00.00 00 - 04 .....| vms_evt rc = 0
0986 - 00:00:00.00 00 - 04 .....| vms_evt Wait on mailbox BKM3/@ipcc_e_1_2 : tout = 10000
0987 - 00:00:00.00 00 - 05 .....| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0988 - 00:00:00.00 00 - 05 .....| vms_get_mbx_chan Returning key 1a0
0989 - 00:00:00.00 00 - 04 .....| vms_evt Event BKM3/@ipcc_e_1_2 TIMEOUT
0990 - 00:00:00.00 00 - 04 .....| vms_evt rc = 1
0991 - 00:00:00.00 00 - 04 .....| vms_mapgbl key : ffffffff - addr : 0/0
0992 - 00:00:00.00 00 - 04 .....| vms_mapgbl Section MQS1_shm_ffffffff mapped at 1510000-1511fff
0993 - 00:00:00.00 00 - 04 .....| vms_evt Reset on mailbox BKM3/@ipcc_e_1_2 : tout = -1
0994 - 00:00:00.00 00 - 05 .....| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0995 - 00:00:00.00 00 - 05 .....| vms_get_mbx_chan Returning key 1a0
0996 - 00:00:00.00 00 - 04 .....| vms_evt rc = 0
0997 - 00:00:00.00 00 - 04 .....| vms_evt Wait on mailbox BKM3/@ipcc_e_1_2 : tout = 10000
0998 - 00:00:00.00 00 - 05 .....| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_2
0999 - 00:00:00.00 00 - 05 .....| vms_get_mbx_chan Returning key 1a0
*** End of buffer ***
*** Trace History Ch1:2 ***
0982 - 00:00:00.00 01 - 04 .....| vms_evt Event BKM3/@ipcc_e_1_7 TIMEOUT
0983 - 00:00:00.00 01 - 04 .....| vms_evt rc = 1
0984 - 00:00:00.00 01 - 04 .....| vms_evt Reset on mailbox BKM3/@ipcc_e_1_7 : tout = -1
0985 - 00:00:00.00 01 - 05 .....| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_7
0986 - 00:00:00.00 01 - 05 .....| vms_get_mbx_chan Returning key 1c0
0987 - 00:00:00.00 01 - 04 .....| vms_evt rc = 0
0988 - 00:00:00.00 01 - 04 .....| vms_evt Wait on mailbox BKM3/@ipcc_e_1_7 : tout = 10000
0989 - 00:00:00.00 01 - 05 .....| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_7
0990 - 00:00:00.00 01 - 05 .....| vms_get_mbx_chan Returning key 1c0
0991 - 00:00:00.00 01 - 04 .....| vms_evt Event BKM3/@ipcc_e_1_7 TIMEOUT
0992 - 00:00:00.00 01 - 04 .....| vms_evt rc = 1
0993 - 00:00:00.00 01 - 04 .....| vms_evt Reset on mailbox BKM3/@ipcc_e_1_7 : tout = -1
0994 - 00:00:00.00 01 - 05 .....| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_7
0995 - 00:00:00.00 01 - 05 .....| vms_get_mbx_chan Returning key 1c0
0996 - 00:00:00.00 01 - 04 .....| vms_evt rc = 0
0997 - 00:00:00.00 01 - 04 .....| vms_evt Wait on mailbox BKM3/@ipcc_e_1_7 : tout = 10000
0998 - 00:00:00.00 01 - 05 .....| vms_get_mbx_chan Getting mbx BKM3/@ipcc_e_1_7
0999 - 00:00:00.00 01 - 05 .....| vms_get_mbx_chan Returning key 1c0
*** End of buffer ***

```

```

MQT>disable history
MQT>clear history

```

이제 특정 함수를 추적하는 것에 대해 중점적으로 살펴보도록 합니다. **SHOW COMPONENT**와 **SHOW FUNCTION**을 사용하여 추적할 특정 영역을 식별할 수 있습니다. 이 예에서는 공통 서비스 함수인 'xcsRequestMutexSem'을 추적합니다. 구

샘플 추적 세션

성요소는 0x17이고 함수 코드는 0x1b입니다. 다음 두 방법 중 하나를 사용하여 이를 설정할 수 있습니다.

```
MQT>select comp=0x17 func=0x1b
```

또는

```
MQT>select fname="xcsRequestMutexSem"
```

이제 실행 기록을 사용 가능으로 설정하면 버퍼에 출력이 나타나지 않음을 알 수 있습니다. 이는 모든 항목에 대해 추적 마스크 비트를 재설정해야 하기 때문입니다.

```
MQT>enable history  
MQT>show history
```

```
*** Trace History Ch1:1 ***  
*** End of buffer ***  
*** Trace History Ch:2 ***  
*** End of buffer ***
```

```
MQT>set mask=0xffffffff  
MQT>show history
```



```

*** Trace History Ch1:1 ***
0973 - 00:00:00.00 01 - 02 ---->| xcsRequestMutexSem
0974 - 00:00:00.00 01 - 03 ---->| xllSemReq
0975 - 00:00:00.00 01 - 04 ---->| vms_mtx
0976 - 00:00:00.00 01 - 04 .....| vms_mtx :- Locking BKM3_m_1_45 - timeout: -1
0977 - 00:00:00.00 01 - 05 ---->| vms_get_lock
0978 - 00:00:00.00 01 - 05 ----<| vms_get_lock
0979 - 00:00:00.00 01 - 04 ----<| vms_mtx
0980 - 00:00:00.00 01 - 03 ----<| xllSemReq
0981 - 00:00:00.00 01 - 02 ----<| xcsRequestMutexSem
0982 - 00:00:00.00 01 - 03 ---->| xcsRequestMutexSem
0983 - 00:00:00.00 01 - 04 ---->| xllSemReq
0984 - 00:00:00.00 01 - 05 ---->| vms_mtx
0985 - 00:00:00.00 01 - 05 .....| vms_mtx :- Locking BKM3_m_1_6 - timeout: -1
0986 - 00:00:00.00 01 - 06 ---->| vms_get_lock
0987 - 00:00:00.00 01 - 06 ----<| vms_get_lock
0988 - 00:00:00.00 01 - 05 ----<| vms_mtx
0989 - 00:00:00.00 01 - 04 ----<| xllSemReq
0990 - 00:00:00.00 01 - 03 ----<| xcsRequestMutexSem
0991 - 00:00:00.00 01 - 03 ---->| xcsRequestMutexSem
0992 - 00:00:00.00 01 - 04 ---->| xllSemReq
0993 - 00:00:00.00 01 - 05 ---->| vms_mtx
0994 - 00:00:00.00 01 - 05 .....| vms_mtx :- Locking BKM3/@ipcc_m_1_18 - timeout: -1
0995 - 00:00:00.00 01 - 06 ---->| vms_get_lock
0996 - 00:00:00.00 01 - 06 ----<| vms_get_lock
0997 - 00:00:00.00 01 - 05 ----<| vms_mtx
0998 - 00:00:00.00 01 - 04 ----<| xllSemReq
0999 - 00:00:00.00 01 - 03 ----<| xcsRequestMutexSem
*** End of buffer ***

```

지정한 함수와 하위 함수만 두 프로세스에서 추적됨을 알 수 있습니다. **select** 명령을 사용하면 구성요소와 함수를 8개까지 추적할 수 있습니다. 실시간으로 추적을 사용하려면 특정 LU에 대한 메시지를 표시하는 클라이언트 프로세스를 만들어야 합니다. 이를 수행하려면 **TRACE** 명령을 실행하십시오.

이 명령을 실행하면 지정한 노드에서 클라이언트 프로세스가 시작되고 수신될 추적 메시지를 기다립니다. 클라이언트 창의 추적 세션은 MONMQ를 사용하여 계속 제어할 수 있습니다.

```
MQT>trace start node="mihell"
```

이제 클라이언트 프로세스를 사용하고 메시지가 도착할 때의 상태 그대로 메시지를 표시하십시오.

```
MQT>enable trace
```

샘플 추적 세션

필요하면 추적 출력에서 MQSeries 스톱드를 추가하거나 제거할 수 있습니다. 스톱드를 연결된 상태로 둘 수는 있지만 그 스톱드의 추적 데이터를 사용하지 않게 설정하여 추적이 성능에 부담을 주지 않도록 할 수 있습니다.

프로세스나 스톱드가 시작될 때 추적을 시작할 수 있습니다. **ONSTARTUP** 명령이 사용되고 결과적으로 새로운 모든 MQSeries 프로세스는 시동시부터 추적됩니다.

추적 세션을 종료하려면 모든 활성 채널의 사용을 중지하고 클라이언트 프로세스를 종료해야 합니다. **close** 명령을 실행하면 이러한 모든 작업이 자동으로 수행됩니다.

추적을 계속하려면 MONMQ에서 **quit**를 사용한 다음 나중에 추적을 다시 시작하십시오.

추적 마스크 비트와 구성요소/함수 선택은 매우 다릅니다. 추적 마스크 비트는 추적 메시지 유형의 출력을 제어합니다. 예를 들어, 추적 입력 항목과 추적 출력은 메시지 유형입니다. 구성요소/함수 선택은 이러한 마스크 비트의 설정 사항에 의존하므로 마스크 비트 설정 사항을 사용하지 않으면 **select** 명령을 사용하여 설정된 사항이 모두 적용되지 않습니다.

부록. 사용자 종료

Compaq OpenVMS용 MQSeries는 채널 종료 프로그램과 데이터 변환 종료 프로그램을 모두 지원합니다. 채널 종료에 대한 자세한 정보는 *MQSeries 상호통신*을 참조하십시오. 데이터 변환 종료에 대한 자세한 정보는 *MQSeries Application Programming Guide* 및 *MQSeries Application Programming Reference*를 참조하십시오.

이 부록에서는 Compaq OpenVMS용 MQSeries에서의 종료 프로그램을 사용과 관련한 정보를 제공합니다.

채널 및 워크로드 종료

Compaq OpenVMS용 MQSeries에서는 별도의 스레드된 종료 프로그램을 연결하는 요구사항을 적용할 수 없습니다.

MQSeries 클러스터 워크로드 종료

OpenVMS에서 워크로드 종료를 연결할 때는 연결자 옵션 파일에 다음을 지정해야 합니다.

```
sys$share:mqm/share  
sys$share:mqutl/share  
SYMBOL_VECTOR=(clw|Function=PROCEDURE,MQStart=PROCEDURE)
```

종료 이미지를 참조하려면 시스템 전역 실행파일 논리 이름이 필요합니다. 예를 들어, 종료 이름이 SYSS\$SHARE:AMQSWLM.EXE이면 다음 논리 이름을 정의해야 합니다.

```
$DEFINE/SYSTEM/EXEC AMQSWLM SYSS$SHARE:AMQSWLM
```

논리 이름 정의에는 .EXE 파일 확장자를 지정하지 마십시오.

시스템 시동 시 이 논리 이름이 정의되도록 하려면

```
SYSS$MANAGER:MQS_SYSTARTUP.COM에 정의하십시오.
```

부록J. 트러스트 응용프로그램

사용자 환경에서 성능이 중요한 고려사항이며 환경이 안정적일 때는 사용자 응용프로그램, 채널, 리스너를 "트러스트"된 것으로 즉, 빠른 경로 바인딩을 사용한다고 정의할 수 있습니다. 그러면 OpenVMS 시스템에서 비지속 메시지의 MQPUT과 MQGET 호출을 처리하는 데 걸리는 시간을 최고 400%까지 줄일 수 있습니다.

트러스트 응용프로그램에서 MQSeries 응용프로그램과 로컬 큐 관리자 에이전트는 같은 프로세스가 됩니다. 응용프로그램은 큐 관리자 자원에 직접 연결되어 큐 관리자의 기능을 확장합니다. 그러나 저장영역을 겹쳐쓰지 못하도록 하는 보호장치가 없으므로 이 옵션을 사용할 경우 큐 관리자의 무결성이 손상될 수 있습니다.

또한 트러스트 응용프로그램이 공유 메모리와 같은 특정 자원을 작성해야 할 경우도 있습니다. 다른 큐 관리자의 프로세스에서 이러한 자원을 액세스해야 할 수도 있으므로, 같은 UIC가 자원을 소유해야 합니다. 큐 관리자 프로세스는 모두 MQM 계정에서 실행되므로 트러스트 응용프로그램도 이 계정에서 실행되어야 합니다.

트러스트 응용프로그램을 사용하기 전에 반드시 위에 설명된 사항을 고려해야 합니다.

사용자 응용프로그램

MQM 계정에서 직접 응용프로그램을 실행하지 않아도 됩니다. 큐 관리자에 성공적으로 연결되면 MQSeries가 활성 스레드의 보안 프로파일을 자동으로 수정하여 스레드가 MQM 계정 ID를 사용하도록 만듭니다. 큐 관리자로부터 연결을 끊으라는 호출을 받으면 스레드의 원래 ID가 다시 사용됩니다.

트러스트 응용프로그램이 큐 관리자에 연결되어 있는 동안에는 응용프로그램이 MQM 계정에서 실행된다는 점을 주목하십시오. 큐 관리자에 연결되어 있는 동안 스레드 ID를 다른 UIC로 변경해야 할 경우에는 다음 MQI 호출을 발행하기 전에 이를 MQM으로 변경해야 합니다.

트러스트 응용프로그램 설정

OpenVMS용 MQSeries에서 트러스트 응용프로그램을 실행하려면 MQCONNX 호출의 Options 필드에서 바인딩 유형을 MQCNO_FASTPATH_BINDING으로 지정해야 합니다(표준 바인딩에는 MQCNO_STANDARD_BINDING 옵션을 사용합니다). 옵션을 지정하지 않으면(MQCNO_NONE) 디폴트로 STANDARD_BINDING이 사용됩니다.

또한 논리 이름 MMQ_CONNECT_TYPE을 사용하여 QCONNX 호출에 지정된 바인딩 유형을 대체할 수 있습니다. 논리 이름을 정의할 경우 논리 이름에 FASTPATH나

트러스트 응용프로그램 설정

STANDARD 값을 지정하여 필요한 바인딩 유형을 선택하도록 해야 합니다. 그러나 FASTPATH 바인딩은 MQCONNX 호출에 연결 옵션이 올바르게 지정되어 있을 때만 사용됩니다. 이 논리 이름을 사용하면, FASTPATH_BINDING에 문제가 발생할 경우 응용프로그램을 다시 빌드할 필요 없이 STANDARD_BINDING을 사용하여 응용 프로그램을 실행할 수 있습니다.

즉, 트러스트 응용프로그램을 실행하려면 다음 중 하나를 수행하십시오.

- MQCONNX 호출에 MQCNO_FASTPATH_BINDING 옵션을 지정하고 MQ_CONNECT_TYPE 논리 이름을 FASTPATH로 정의합니다.

또는

- MQCONNX 호출에 MQCNO_FASTPATH_BINDING 옵션을 지정하고 MQ_CONNECT_TYPE 논리 이름을 정의하지 않습니다.

트러스트 응용프로그램 사용에 대한 자세한 정보는 *MQSeries 상호통신*을 참조하십시오.

채널과 리스너를 트러스트 응용프로그램으로 실행

runmqsc start channel 명령을 사용하여 시작한 채널 프로그램은 MQM 계정에서 실행됩니다. 수신되는 TCP(또는 DECnet 연결) 요청으로 시작한 채널 수신자 프로그램 또한 MQM 계정에서 실행됩니다.

runmqchl과 **runmqslr** 명령은 MQM 계정에서 실행되는 분리된 프로세스를 작성합니다. 큐 관리자의 `qm.ini` 파일의 채널 스탠자에 있는 `MQ_CONNECT_TYPE` 논리 이름과 `MQIBindType` 결합은 채널이나 리스너를 트러스트된 것으로 실행할지의 여부를 정의합니다.

트러스트 채널이나 트러스트 리스너를 설정하려면 다음 중 하나를 수행하십시오.

- `qm.ini` 파일에 `MQIBindType=FASTPATH`를 지정하고 논리 이름을 FASTPATH로 설정합니다.

또는

- `qm.ini` 파일에 `MQIBindType=FASTPATH`를 지정하고 논리 이름을 정의하지 않습니다.

빠른 비지속 메시지

비지속 메시지 속도(NPMSPEED) 채널 속성을 사용하여 비지속 메시지의 송신 속도를 지정할 수 있습니다. 일반 또는 고속으로 지정할 수 있습니다. 디폴트는 고속입니다. 송신 속도를 고속으로 지정하면, 동기점까지 대기하지 않고도 채널의 비지속 메시지를 검색에 사용할 수 있습니다. 이러한 메시지는 훨씬 빨리 검색에 사용할 수 있게 되지만, 전송이 실패하거나 메시지 전송 중 채널이 정지할 경우 메시지가 손실될 수 있습니다.

빠른 비지속 메시지

채널과 리스너를 트러스트 응용프로그램으로 실행하는 것에 대한 자세한 정보나 빠른 비지속 메시지에 대한 자세한 정보는 *MQSeries* 상호통신을 참조하십시오.

부록K. 추가 정보

이 부록에서는 Compaq OpenVMS용 MQSeries 설치에 필요한 추가 정보를 제공합니다.

이 부록에 포함된 정보는 다음에 이 서적을 갱신할 때 해당 서적에 서적에 포함됩니다.

응용프로그램 프로그래밍 안내

MQI(Message Queue Interface) 호출이 AST 루틴 내에서 작성되지 않는다는 점을 고지하기 위해 OpenVMS에서의 프로그래밍 정보가 개정되었습니다.

이는 MQSeries에서 AST 루틴이 사용되며 다른 AST 루틴이 활성화되었을 때는 이러한 루틴이 실행될 수 없기 때문입니다.

응용프로그램 트리거

MQTRIGGER.COM 명령 파일은 MQSeries 트리거 모니터(RUNMQTRM)가 제공하는 매개변수를 사용하고 MQTMC2 구조에서 필드를 분리할 수 있도록 만들어진 명령 파일의 예로 제공됩니다.

이 명령 파일의 첫 번째 매개변수는 MQTMC2 구조에서 선택한 필드로 호출할 수 있는 이미지 또는 명령 파일이어야 합니다.

MQTRIGGER는 MQTMC2 구조의 다음 필드를 실행된 이미지나 명령 파일로 전달합니다.

매개변수	MQTMC2 필드
1	QName
2	ProcessName
3	TriggerData
4	ApplType
5	UserData
6	QMgrName

예

1. amqsech 이미지를 트리거하려는 경우:

트리거 프로세스 정의의 ApplId 필드는 다음과 같이 지정됩니다.

```
APPLICID('@mqseries:mqtrigger $mqbin:amqsech')
```

추가 정보

이 예에서는 MQBIN 논리 디렉토리가 다음과 같이 정의되었다고 가정합니다.

```
SYS$SYSROOT:[SYSHLP.EXAMPLES.MQSERIES.BIN]
```

2. dka200:[user]cmd.com 명령 파일을 호출하려는 경우:
트리거 프로세스 정의의 ApplicId 필드는 다음과 같이 지정됩니다.

```
APPLICID('@mqs_examples:mqtrigger @dka200:[user]cmd')
```

부록L. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. IBM은 다른 국가에서는 이 책에 기술된 제품, 서비스 또는 부속 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 IBM 영업 담당자에게 문의하십시오. 이 책에서는 IBM 제품, 프로그램 또는 서비스에 관해 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스 만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 사용권까지 부여하는 것은 아닙니다. 특허 사용권에 대한 의문 사항은 다음으로 문의하십시오.

135-270

서울특별시 강남구 도곡동 47-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2 바이트(DBCS) 정보에 관한 사용권 문의는 한국 IBM 고객만족센터에 문의 하거나 다음 주소로 문의하시기 바랍니다.

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함 한(단, 이에 한하지 않음) 어떠한 종류의 묵시적 또는 명시적 보증없이 이 책을 “현상 태대로” 제공합니다. 일부 국가에서는 특정 거래에 있어서 명시적 또는 묵시적 보증의 포기를 허용하지 않으므로 이 문장이 사용자에게 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 여기에 수록된 정보는 주기적으로 변경되며, 그러한 변경 사항은 책의 최신판에 통합됩니다. IBM은 언제든지 이 책에서 설명한 제품 및/또는 프로그램을 개선 및/또는 변경할 수 있습니다.

주의사항

이 정보에서 비IBM의 웹 사이트를 언급한 것을 편의를 위한 것뿐이며 어떤 방식으로든 해당 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로, 해당 웹 사이트의 사용으로 인한 위험은 사용자 본인의 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독자적으로 작성된 프로그램 및 기타 프로그램(이 프로그램 포함) 사이의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 정보를 원하는 이 프로그램의 사용권자는 다음주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조항 및 조건에 따라(예를 들면, 사용료 지불 등) 사용할 수 있습니다.

이 정보에 기술된 사용권 프로그램 및 사용 가능한 모든 사용권 자료는 IBM 기본 계약, IBM 프로그램 사용권 계약(IPE) 또는 동등한 계약에 따라 제공한 것입니다.

비IBM 제품에 대한 정보는 해당 제품의 공급업체, 공개 자료 또는 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트 하지 않았으므로, IBM 제품과 관련한 성능의 정확성, 호환성 또는 배상청구에 대해서는 확신할 수 없습니다. 호환성에 대한 의문 사항은 해당 제품의 공급업체에 문의 해야 합니다.

저작권 사용 정보:

이 정보에는 다양한 운영 플랫폼에서의 프로그래밍 기술을 설명하는 소스 언어로 된 샘플 응용프로그램이 있습니다. 이러한 샘플 프로그램은, 샘플 프로그램이 작성된 운영 체제의 응용프로그램 프로그래밍 인터페이스(API)를 준수하는 응용프로그램의 개발, 사용, 홍보 또는 분배할 목적을 위해서라면 IBM에 비용을 지불하지 않고도 어떠한 형식으로든 복사, 수정, 분배할 수 있습니다. 이러한 샘플 프로그램은 모든 조건에서 완전히 테스트되지 않았습니다. 따라서 IBM은 샘플 프로그램의 안정성, 서비스, 기능을 보증하지 않습니다. 이러한 샘플 프로그램은 IBM의 응용프로그램 프로그래밍 인터페이스(API)를 준수하는 응용프로그램의 개발, 사용, 홍보 또는 분배할 목적을 위해서라면 IBM에 비용을 지불하지 않고도 어떠한 형식으로든 복사, 수정, 분배할 수 있습니다.

등록상표 및 서비스표

다음 용어는 미국 및 기타 국가에서 IBM Corporation의 등록상표입니다.

AIX	IBM
MQSeries	AS/400
MVS/ESA	NetView
CICS	OS/2
First Failure Support Technology	VSE/ESA
OS/390	BookManager

UNIX는 미국 및 기타 국가에서 X/Open Company Limited를 통해 독점적 사용권을 부여 받은 등록상표입니다.

DIGITAL, OpenVMS, Compaq, DecNet, Alpha는 Compaq Corporation의 등록상표입니다.

Intel은 미국 및 기타 국가에서 Intel Corporation의 등록상표입니다.

Microsoft, Windows, Windows Logo는 Microsoft Corporation의 등록상표입니다.

MultiNet과 TCPware는 Process Software의 등록상표입니다.

Java와 모든 Java-기준 등록상표와 로고는 미국 및/또는 기타 국가에서 Sun Microsystems, Inc.의 등록상표입니다.

기타 회사 이름, 제품 이름, 서비스 이름은 해당 소유자의 등록상표나 서비스표입니다.

참고 문헌

이 절에서는 현재의 모든 MQSeries 제품에 사용 가능한 문서에 대해 설명합니다.

MQSeries의 모든 플랫폼에 적용되는 서적

MQSeries 『제품군』 서적이라고도 불리는 아래 서적들의 대부분은 모든 MQSeries 레벨 2 제품에 적용됩니다. 최신 MQSeries 레벨 2 제품은 다음과 같습니다.

- AIX용 MQSeries V5.2
- AS/400용 MQSeries V5.2
- MQSeries for AT&T GIS UNIX V2.2
- Compaq OpenVMS Alpha용 MQSeries V5.1
- Compaq Tru64 UNIX용 MQSeries V5.1
- HP-UX용 MQSeries V5.2
- Linux용 MQSeries V5.2
- OS/2 Warp용 MQSeries V5.1
- OS/390용 MQSeries V5.2
- MQSeries for SINIX and DC/OSx V2.2
- Sun Solaris용 MQSeries, V5.2
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- Tandem NonStop Kernel용 MQSeries V2.2.0.1
- MQSeries for VSE/ESA, V2.1.1
- Windows용 MQSeries V2.0
- Windows용 MQSeries V2.1
- Windows NT 및 Windows 2000용 MQSeries V5.2

MQSeries의 모든 플랫폼에 적용되는 서적은 다음과 같습니다.

- *MQSeries Brochure*, G511-1908
- *An Introduction to Messaging and Queuing*, GC33-0805
- *MQSeries 상호통신*, SA30-0272

- *MQSeries Queue Manager Clusters*, SC34-5349
- *MQSeries 클라이언트*, GA30-0274
- *MQSeries 시스템 관리*, SA30-0273
- *MQSeries 명령 참조서*, SA30-0582
- *MQSeries Event Monitoring*, SC34-5760
- *MQSeries Programmable System Management*, SC33-1482
- *MQSeries Administration Interface Programming Guide and Reference*, SC34-5390
- *MQSeries Messages*, GC33-1876
- *MQSeries Application Programming Guide*, SC33-0807
- *MQSeries Application Programming Reference*, SC33-1673
- *MQSeries Programming Interfaces Reference Summary*, SX33-6095
- *MQSeries Using C++*, SC33-1877
- *MQSeries Using Java*, SC34-5456
- *MQSeries Application Messaging Interface*, SC34-5604

MQSeries 플랫폼별 서적

각 MQSeries 제품은 MQSeries 제품군 서적 외에 적어도 하나의 플랫폼별 서적에 문서화되어 있습니다.

AIX용 MQSeries V5.2

AIX용 MQSeries V5.0 빠른 시작, GA30-0267

AS/400용 MQSeries V5.2

AS/400용 MQSeries V5.1 빠른 시작, GA30-0839

MQSeries for AS/400 V5.1 System Administration, SC34-5558

참고 문헌

MQSeries for AS/400 V5.1 Application Programming Reference(RPG), SC34-5559

MQSeries for AT&T GIS UNIX V2.2

MQSeries for AT&T GIS UNIX System Management Guide, SC33-1642

Compaq OpenVMS Alpha용 MQSeries V5.1

Compaq OpenVMS Alpha용 MQSeries 빠른 시작, GA30-1334

Compaq OpenVMS Alpha용 MQSeries 시스템 관리 안내서, SA30-1334

Compaq Tru64 UNIX용 MQSeries V5.1

Compaq Tru64 UNIX용 MQSeries V5.1 빠른 시작, GA30-0958

HP-UX용 MQSeries V5.2

HP-UX용 MQSeries V5.0 빠른 시작, GA30-0269

Linux용 MQSeries V5.2

Linux용 MQSeries 빠른 시작, GA30-0889

OS/2 Warp용 MQSeries V5.1

OS/2 Warp용 MQSeries V5.0 빠른 시작, GA30-0268

OS/390용 MQSeries V5.2

MQSeries for OS/390 Concepts and Planning Guide, GC34-5650

MQSeries for OS/390 System Setup Guide, SC34-5651

MQSeries for OS/390 System Administration Guide, SC34-5652

MQSeries for OS/390 System Administration Guide, GC34-5892

MQSeries for OS/390 Messages and Codes, GC34-5891

MQSeries for OS/390 Licensed Program Specifications, GC34-5893

MQSeries for OS/390 Program Directory

R/3용 MQSeries 링크 버전 1.2

R/3용 MQSeries 링크 버전 1.2 사용자 안내서, GA30-0371

MQSeries for SINIX and DC/OSx V2.2

MQSeries for SINIX and DC/OSx System Management Guide, GC33-1768

Sun Solaris용 MQSeries, V5.2

Sun Solaris용 MQSeries V5.0 빠른 시작, GA30-0270

MQSeries for Sun Solaris, Intel Platform Edition, V5.1

MQSeries for Sun Solaris, Intel Platform Edition Quick Beginnings, GC34-5851

Tandem NonStop Kernel용 MQSeries V2.2.0.1

MQSeries for Tandem NonStop Kernel System Management Guide, GC33-1893

MQSeries for VSE/ESA, V2.1.1

MQSeries for VSE/ESA Licensed Program Specifications, GC34-5365

MQSeries for VSE/ESA System Management Guide, GC34-5364

Windows용 MQSeries V2.0

Windows용 MQSeries V2.0 사용자 안내서, GA30-0170

Windows용 MQSeries V2.1

Windows용 MQSeries V2.1 사용자 안내서, GA30-0371

Windows NT 및 Windows 2000용 MQSeries V5.2

Windows NT용 MQSeries V5.0 빠른 시작, GA30-0583

MQSeries for Windows NT Using the Component Object Model Interface, SC34-5387

MQSeries LotusScript Extension, SC34-5404

소프트카피 서적

대부분의 MQSeries 서적은 하드카피와 소프트카피 두 형식으로 제공됩니다.

HTML 형식

다음 MQSeries 제품에는 HTML 형식으로 된 관련 MQSeries 문서가 제공됩니다.

- AIX용 MQSeries V5.2
- AS/400용 MQSeries V5.2
- Compaq OpenVMS Alpha용 MQSeries V5.1
- Compaq Tru64 UNIX용 MQSeries V5.1
- HP-UX용 MQSeries V5.2
- Linux용 MQSeries V5.2
- OS/2 Warp용 MQSeries V5.1
- OS/390용 MQSeries V5.2
- Sun Solaris용 MQSeries, V5.2
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- Windows NT 및 Windows 2000용 MQSeries V5.2(컴파일된 HTML)
- R/3용 MQSeries 링크 V1.2

HTML 형식의 MQSeries 서적은 다음 위치에 있는 MQSeries 제품군 웹 사이트에서도 참조할 수 있습니다.

<http://www.ibm.com/software/mqseries/>

PDF(Portable Document Format)

Adobe Acrobat Reader를 사용하여 PDF 파일을 보거나 인쇄할 수 있습니다.

Adobe Acrobat Reader를 구하거나 Acrobat Reader가 지원되는 플랫폼에 대한 최신 정보를 보려면 다음 위치에 있는 Adobe Systems Inc. 웹 사이트를 방문하십시오.

<http://www.adobe.com/>

다음 MQSeries 제품에는 관련 MQSeries 서적의 PDF 버전이 제공됩니다.

- AIX용 MQSeries V5.2
- AS/400용 MQSeries V5.2
- Compaq OpenVMS Alpha용 MQSeries V5.1
- Compaq Tru64 UNIX용 MQSeries V5.1
- HP-UX용 MQSeries V5.2
- Linux용 MQSeries V5.2
- OS/2 Warp용 MQSeries V5.1
- OS/390용 MQSeries V5.2
- Sun Solaris용 MQSeries, V5.2
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- Windows NT 및 Windows 2000용 MQSeries V5.2
- R/3용 MQSeries 링크 V1.2

현재 모든 MQSeries 서적의 PDF 버전은 다음 위치에 있는 MQSeries 제품군 웹 사이트에서도 참조할 수 있습니다.

<http://www.ibm.com/software/mqseries/>

BookManager 형식

MQSeries 라이브러리는 다양한 온라인 라이브러리 콜렉션 킷에서 IBM BookManager 형식으로 제공됩니다. *Transaction Processing and Data* 콜렉션 킷, SK2T-0730도 이러한 온라인 라이브러리 콜렉션 킷 중 하나입니다. 다음과 같은 사용 허가된 IBM 프로그램을 사용하여 소프트카피 서적을 IBM BookManager 형식으로 볼 수 있습니다.

BookManager READ/2
 BookManager READ/6000
 BookManager READ/DOS
 BookManager READ/MVS

참고 문헌

BookManager READ/VM
BookManager READ for Windows

포스트스크립트 형식

MQSeries 라이브러리는 대부분의 MQSeries Version 2 제품에 포스트스크립트(.PS) 형식으로 제공됩니다. 포스트스크립트 형식으로 된 서적은 포스트스크립트 프린터에서 인쇄하거나 적절한 표시기를 사용하여 볼 수 있습니다.

Windows 도움말 형식

Windows용 MQSeries 사용자 안내서는 Windows용 MQSeries 버전 2.0과 Windows용 MQSeries 버전 2.1에 Windows 도움말 형식으로 제공됩니다.

인터넷에서 사용 가능한 MQSeries 정보

MQSeries 제품군 웹 사이트는 다음과 같습니다.

<http://www.ibm.com/software/mqseries/>

이 웹 사이트의 링크를 따라가면 다음을 수행할 수 있습니다.

- MQSeries 제품군에 대한 최신 정보를 볼 수 있습니다.
- HTML과 PDF 형식으로 된 MQSeries 서적에 액세스할 수 있습니다.
- MQSeries SupportPac을 다운로드할 수 있습니다.

관련 서적

- *Compaq OpenVMS Performance Management* (1999년 1월)
이 서적은 OpenVMS 시스템의 성능을 최적화하는데 도움이 되는 정보를 제공합니다.
- *Compaq OpenVMS System Management Utilities*(2권, 1999년 1월)
이 서적은 OpenVMS에 사용하는 시스템 관리 유틸리티에 대한 참조 정보를 제공합니다.

- *Character Data Representation Library, Character Data Representation Architecture, Reference and Registry*, SC09-2190-00

이 문서는 CDRA(Character Data Representation Architecture)에 대한 개요를 제공하며 구조의 요소를 참조 매뉴얼 형식으로 정의합니다.

- *DecNet SNA Gateway for Synchronous Transport Installation(OpenVMS)*(1993년 11월)
이 안내서는 DecNet SNA Gateway를 설치하고 구성하는 방법을 설명합니다.
- *Digital SNA APPC/LU6.2 Programming Interface for OpenVMS*(1996년 5월)
이 안내서는 SNA APPC/LU6.2를 설치하고 구성하는 방법을 설명합니다.
- *Digital TCP/IP Services for OpenVMS Installation and Configuration*(1999년 1월)
이 안내서는 Digital TCP/IP 설치 및 구성에 대해 설명합니다.
- *Guidelines for OpenVMS Cluster Configurations*(1999년 1월)
이 안내서는 OpenVMS 클러스터 가용성과 확장성을 최대화하는 방법을 설명합니다.
- *Introduction to Compaq Networking and Data Communications*(Compaq Part No. 093148)
이 안내서는 Compaq 네트워킹과 데이터 통신 개념, 작업, 제품, 매뉴얼에 대한 개요를 제공합니다.

용어집

이 용어집은 이 서적에 사용된 MQSeries 용어 및 약어를 정의합니다. 찾는 용어가 없으면 색인이나 *IBM Dictionary of Computing*(New York: McGraw-Hill, 1994)를 참조하십시오.

이 용어집에는 *American National Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI)의 용어와 정의가 포함되어 있습니다. 사본은 American National Standards Institute, 11 West 42 Street, New York, New York 10036에서 구입하실 수 있습니다. 정의는 정의 뒤에 붙는 (A) 기호로 식별됩니다.

가

강제 종료. 현재 활성 작업의 상태에 상관 없이 어댑터가 OS/390용 MQSeries와의 연결을 즉시 끊는 CICS 어댑터 종료 유형. 정지 종료와 반대 개념입니다.

경고. 현재의 문제점이나 곧 발생할 수 있는 문제점을 식별하기 위해 네트워크의 관리 서비스 초점으로 보내지는 메시지.

경고 모니터. OS/390용 MQSeries에서 OS/390용 MQSeries에 연결을 요청한 결과로 발생하는 스케줄되지 않은 이벤트를 처리하는 CICS 어댑터 구성요소.

관리자 명령. 큐, 프로세스, 이름 목록과 같은 MQSeries 오브젝트를 관리하는 데 사용하는 MQSeries 명령.

구성 파일. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 로그, 통신 또는 설치가능 서비스 등과 관련된 구성 정보가 있는 파일. *.INI* 파일과 동의어입니다. 스탠자를 참조하십시오.

권한 서비스. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 명령이나 호출과 연관된 사용자 ID에 대해 명령 및 MQI 호출에 권한 점검을 제공하는 서비스.

권한 점검. 예를 들어, 사용자가 큐를 열거나 큐 관리자에 연결하기 위해 오브젝트에 명령을 발행하려고 할 때 수행되는 보안 점검.

권한 파일. UNIX 시스템에 설치된 MQSeries에서, 오브젝트나 단일 오브젝트 클래스 또는 모든 오브젝트 클래스에 보안 정의를 제공하는 파일.

규칙 테이블. 데드-레터 큐 핸들러가 DLQ의 메시지에 적용하는 하나 이상의 규칙을 포함하는 제어 파일.

기록된 메시지. 시스템에 오류가 발생할 경우 손실되지 않도록 보조 디스크 저장영역 장치에 기록되는 메시지. 지속 메시지를 참조하십시오.

기본 원칙. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 사용자 ID에 사용되는 용어. 오브젝트 권한 관리자가 시스템 자원에 대한 권한 부여를 점검할 때 사용됩니다.

나

널(null) 문자. X'00'으로 나타나는 문자.

논리적 작업 단위(LUW). 작업 단위를 참조하십시오.

다

다이나믹 큐. 프로그램에서 모델 큐 오브젝트를 열 때 작성된 로컬 큐. 퍼머넌트 다이나믹 큐 및 템포러리 다이나믹 큐를 참조하십시오.

단일 로깅. 각 변경사항이 데이터 세트 하나에만 기록될 경우 OS/390용 MQSeries 작업을 기록하는 방법. 이중 로깅과 반대 개념입니다.

대체 사용자 보안. 예를 들어 MQSeries 오브젝트를 여는 경우와 같이 사용자 ID가 가진 권한을 다른 사용자 ID로 사용할 수 있는 보안 기능.

데드-레터 큐 핸들러. 데드-레터 큐(DLQ)를 모니터링하고 사용자가 작성한 규칙 테이블에 따라 큐의 메시지를 처리하는 MQSeries 제공 유틸리티.

데드-레터 큐(DLQ). 큐 관리자나 응용프로그램이 올바른 목적 지로 전달할 수 없는 메시지를 송신하는 큐.

데이터그램. MQSeries에서 지원하는 가장 단순한 메시지. 이 메시지 유형에는 응답이 필요하지 않습니다.

도구 이벤트. MQSeries 시스템 네트워크에서 큐 관리자의 작업을 모니터링하는 데 사용할 수 있는 기능. MQSeries는 큐 관리자 지원 정의, 성능 상태 및 채널 상태를 모니터링하는 데 사용할 수 있는 도구 이벤트를 제공합니다. 시스템 운영자에게 이벤트를 표시하는 관리 응용프로그램의 사용자 작성 보고 메커니즘이 도구 이벤트를 사용할 수 있습니다. 도구 이벤트를 사용하면 다른 관리 네트워크의 에이전트로 작동하는 응용프로그램이 보고서를 모니터링하거나 해당 경고 메시지를 작성할 수 있습니다.

동기 메시징. 메시지 큐에 메시지를 위치시키는 프로그램 간의 통신 방법. 동기 메시징을 사용하면 송신 프로그램이 자체 처리를 시작하기 전에 메시지에 대한 응답을 기다립니다. *비동기 메시징*과 반대 개념입니다.

동기점. 트랜잭션 처리에서 트랜잭션의 보호 자원이 일관되게 되는 중간 지점 또는 엔드 포인트. 동기점에서는 자원의 변경사항을 안전하게 요약하거나 이전 동기점으로 백아웃할 수 있습니다.

디폴트 오브젝트. 모든 속성이 정의된 오브젝트(예: 큐) 정의. 사용자가 오브젝트를 정의한 후 이 오브젝트에 가능한 모든 속성을 지정하지 않으면 큐 관리자는 지정되지 않은 속성에 디폴트 속성을 사용합니다.

라

레지스트리. Windows NT에서 시스템 및 응용프로그램 구성 데이터를 저장하기 위한 단일 소스를 제공하는 보안 데이터베이스.

레지스트리 구조. Windows NT에서 레지스트리에 저장된 데이터 구조.

레지스트리 편집기. Windows NT에서 사용자가 레지스트리를 편집할 수 있는 프로그램 항목.

로그. MQSeries에서 큐 관리자가 메시지를 수신하고, 전달하고, 전송하는 중 오류가 발생하면 메시지를 복구하기 위해 수행한 작업을 기록하는 파일.

로그 제어 파일. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 크기, 위치, 사용 가능한 다음 파일의 이름 등 로그 파일 사용을 모니터링하는 데 필요한 정보를 포함하는 파일.

로그 파일. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 큐 관리자가 제어하는 데이터에 발생한 모든 중요한 변경사항을 기록하는 파일. 1차 로그 파일이 꽉 차면 MQSeries에서 2차 로그 파일을 할당합니다.

로컬 정의. 로컬 큐 관리자에 속하는 MQSeries 오브젝트.

로컬 큐. 로컬 큐 관리자에 속하는 큐. 로컬 큐는 처리 대기 중인 메시지 목록을 포함할 수 있습니다. *리모트 큐*와 반대 개념입니다.

로컬 큐 관리자. 프로그램에 연결되어 있고 그 프로그램에 메시지 큐잉 서비스를 제공하는 큐 관리자. 프로그램과 동일한 시스템에서 실행되더라도 프로그램에 연결되어 있지 않은 큐 관리자를 *리모트 큐 관리자*라고 합니다.

로케일. UNIX 시스템에서 시간, 숫자, 통화 형식 및 문자 분류, 병합 또는 변환과 같은 특정 환경에 대한 규칙을 정의하는 사용자 환경 서브세트. 큐 관리자 CCSID는 이 큐 관리자를 작성한 사용자 ID의 로케일에서 도출됩니다.

롤백. 백아웃과 동의어입니다.

리모트 큐. 리모트 큐 관리자에 속하는 큐. 프로그램이 리모트 큐에 메시지를 넣을(put) 수는 있지만 리모트 큐에서 메시지를 가져올(get) 수는 없습니다. *로컬 큐*와 반대 개념입니다.

리모트 큐 관리자. 프로그램 측면에서 보았을 때 프로그램에 연결되어 있지 않은 큐 관리자.

리모트 큐 오브젝트. *리모트 큐의 로컬 정의를 참조하십시오.*

리모트 큐의 로컬 정의. 로컬 큐 관리자에 속하는 MQSeries 오브젝트. 이 오브젝트는 다른 큐 관리자가 소유한 큐의 속성을 정의합니다. 또한 큐 관리자 별명 및 리플라이-투 큐 별명을 지정하는 데에도 사용됩니다.

리모트 큐잉. 메시지 큐잉에서 응용프로그램이 다른 큐 관리자에 속하는 큐에 메시지를 넣을(put) 수 있도록 하는 예비 서비스.

리스너. MQSeries 분산 큐잉에서, 수신되는 네트워크 연결을 모니터링하는 프로그램.

리턴 코드. 완료 코드 및 이유 코드를 총칭하는 이름.

리플라이-투 큐. MQPUT 호출을 발행한 프로그램이 응답 메시지나 보고 메시지를 송신하려는 큐 이름.

마

매체 이미지. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 오브젝트 이미지를 포함하는 로그 레코드 순서. 이 이미지로 오브젝트를 다시 작성할 수 있습니다.

메시지. 메시지 큐잉 응용프로그램에서 프로그램 간에 송신된 정보. *지속 메시지* 및 *비지속 메시지*를 참조하십시오. 시스템 프로그래밍에서 터미널 운영자나 시스템 관리자를 위한 정보.

메시지 설명자. MQSeries 메시지의 일부로 전송되는 프레젠테이션 및 메시지 형식을 설명하는 제어 정보. 메시지 설명자의 형식은 MQMD 구조에서 정의합니다.

메시지 순서 매기기. 통신 링크를 통해 메시지를 전송할 때 메시지에 고유한 번호를 지정하는 프로그래밍 기술. 메시지 순서 매기기를 사용하면 수신 프로세스에서 모든 메시지를 수신할 것인지 여부를 점검할 수 있으며, 원래 순서대로 큐에 메시지를 위치시키거나 중복 메시지를 제거할 수 있습니다.

메시지 우선순위. MQSeries에서 큐의 메시지가 검색되는 순서 및 트리거 이벤트 생성 여부에 영향을 주는 메시지 속성.

메시지 채널. 분산 메시지 큐잉에서, 한 큐 관리자에서 다른 큐 관리자로의 메시지 이동을 위한 메커니즘. 메시지 채널은 두 메시지 채널 에이전트(양 끝에 송신자와 수신자가 위치)와 통신 링크로 이루어집니다. *MQI 채널*과 반대 개념입니다.

메시지 큐. *큐*와 동의어입니다.

메시지 큐잉. 응용프로그램 내의 각 프로그램이 큐에 메시지를 넣어(put) 다른 프로그램과 통신하는 프로그래밍 기술

메시징. 동기 메시징 및 비동기 메시징을 참조하십시오.

명령. MQSeries에서 큐 관리자가 수행할 수 있는 관리 명령.

명령 서버. 시스템 명령 입력 큐의 명령을 읽고 확인한 다음 올바른 명령을 명령 프로세서로 전달하는 MQSeries 구성요소.

명령 프로세서. 명령을 처리하는 MQSeries 구성요소.

모델 큐 오브젝트. 프로그램에서 다이내믹 큐를 작성할 때 템플릿으로 작동하는 일련의 큐 속성.

미배달 메시지 큐. *데드-레터 큐*를 참조하십시오.

바

백아웃. 현재 복구 단위나 작업 단위에서 이루어진 모든 변경 사항을 취소하는 작업. 작업이 완료되면 새 복구 단위나 작업 단위가 시작됩니다. *확약*과 반대 개념입니다.

버퍼 풀. OS/390용 MQSeries 큐, 메시지 및 오브젝트 정의에 사용되는 기본 저장영역 장치의 영역. *페이지 세트*를 참조하십시오.

보고 메시지. 다른 메시지에 대한 정보를 제공하는 메시지 유형. 보고 메시지는 메시지가 전달되었거나, 목적지에 도착했거나 또는 어떤 이유로 처리되지 않았음을 표시합니다. *응답 메시지* 및 *요청 메시지*와 반대 개념입니다.

보류 이벤트. CICS 어댑터로부터의 연결 요청 결과로 발생하는 스케줄되지 않은 이벤트.

복구 단위. 단일 자원 관리자 내의 복구 가능한 조각 순서. 작업 단위와 반대 개념입니다.

복구 로그. OS/390용 MQSeries에서 메시지, 큐, MQSeries 서브시스템을 복구하는 데 필요한 정보를 가지고 있는 데이터 세트. OS/390용 MQSeries은 각 레코드를 *활성 로그*라고 하는 데이터 세트에 기록합니다. *활성 로그*가 꼭 차면 *로그* 내용이 DASD나 *아카이브 로그*라고 하는 테이프 데이터로 오프로드됩니다. *로그*와 동의어입니다.

복구 인다우트 단위. MQSeries에서 동기점이 요청되었지만 확인되지 않는 복구 단위의 상태.

복구점. OS/390용 MQSeries에서는 OS/390용 MQSeries 페이지 세트의 일련의 백업 사본과 해당 페이지 세트를 복구할 때 필요한 해당 로그 데이터 세트를 설명할 때 복구점이라는 용어를 사용합니다. 이러한 백업 사본은 페이지 세트가 손실되었을 때(예: 페이지 세트 I/O 오류) 가능한 재시작 지점을 제공합니다.

분산 응용프로그램. 메시지 큐잉에서 각각 다른 큐 관리자에 연결할 수 있지만 집합적으로 단일 응용프로그램을 구성하는 응용프로그램의 프로그램 세트.

분산 컴퓨팅 환경(DCE). 몇몇 기본 서비스를 제공하여 분산 응용프로그램을 쉽게 개발할 수 있도록 하는 미들웨어. DCE는 OSF(Open Software Foundation)에서 정의합니다.

분산 큐 관리(DQM). 메시지 큐잉에서 다른 시스템에 있는 큐 관리자에 대한 메시지 채널 설정 및 제어.

비동기 메시지. 메시지 큐에 메시지를 위치시키는 프로그램 간의 통신 방법. 비동기 메시지를 사용하면 송신 프로그램이 메시지에 대한 응답을 기다리지 않고 처리를 계속합니다. 동기 메시징과 반대 개념입니다.

비지속 메시지. 큐 관리자를 재시작하면 손실되는 메시지. 지속 메시지와 반대 개념입니다.

사

서버. (1) MQSeries에서 리모트 워크스테이션에서 실행되는 클라이언트 응용프로그램에 큐 서비스를 제공하는 큐 관리자. (2) 클라이언트/서버라는 특정 두 프로그램으로 이루어진 정보 흐름 모델에서 정보 요청에 응답하는 프로그램. 클라이언트를 참조하십시오.

서버 연결 채널 유형. 큐 관리자를 실행하는 서버와 연관된 MQI 채널 정의 유형. 클라이언트 연결 채널 유형을 참조하십시오.

서버 채널. 메시지 큐잉에서, 요청자 채널에 응답하거나 트랜스미션 큐에서 메시지를 제거하고 또는 통신 링크를 통해 요청자 채널로 메시지를 이동하는 채널.

서브시스템. OS/390에서 OS/390에 의존하는 기능을 제공하는 모듈 그룹. 예를 들어, OS/390용 MQSeries은 OS/390 서브시스템입니다.

서비스 간격. 서비스 간격 이벤트 조건이 충족되었는지 여부를 결정할 때 큐 관리자가 넣기(put)나 가져오기(get) 또는 후속 가져오기(get) 사이의 경과 시간을 비교하는 기준 시간 간격. 큐의 서비스 간격은 큐 속성으로 지정합니다.

서비스 간격 이벤트. 서비스 간격과 관련된 이벤트.

선제 종료. MQSeries에서, 연결된 응용프로그램이 연결을 끊을 때까지 대기하지 않거나 현재 MQI 호출이 완료될 때까지 대기하지 않는 큐 관리자 종료. 즉시 종료 및 정지 종료와 반대 개념입니다.

선형 로깅. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 재시작 데이터를 연속된 파일에 관리하는 프로세스. 필요할 경우 새 파일이 추가됩니다. 큐 관리자를 재시작하기 전에는 데이터가 작성된 공간이 다시 사용되지 않습니다. 순환 로깅과 반대 개념입니다.

설치가능 서비스. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 독립 구성요소로 제공되는 추가 기능. 각 구성요

소 설치는 선택적입니다. In-house 구성요소나 Third-Party 구성요소를 대신 사용할 수도 있습니다. 권한 서비스, 이름 서비스 및 사용자 ID 서비스를 참조하십시오.

성능 이벤트. 한계 상태가 발생했음을 표시하는 이벤트 범주.

성능 추적. 성능 분석 및 성능 조정을 위해 추적 데이터를 사용하는 MQSeries 추적 옵션.

세션 ID. OS/390용 MQSeries에서 메시지를 트랜스미션 큐에서 링크로 이동할 때 메시지 채널 에이전트가 사용할 통신 링크를 정의하는 CICS 고유 ID.

속성. MQSeries 오브젝트의 특성을 정의하는 일련의 등록 정보 중 하나.

송신자 채널. 메시지 큐잉에서, 전송을 시작하거나 트랜스미션 큐에서 메시지를 제거하며 통신 링크를 통해 수신자 채널이나 요청자 채널로 메시지를 이동하는 채널

수신자 채널. 메시지 큐잉에서 송신자 채널에 응답하고, 통신 링크에서 메시지를 가져와 로컬 큐에 넣는(put) 채널.

순차 번호 줄바꿈 값. MQSeries에서 통신 링크의 양 끝에서 동시에 현재 메시지의 순차 번호가 재설정되도록 하는 방법. 순차 번호를 사용하여 메시지를 전송하면 수신 채널에서 메시지를 저장할 때 메시지 순서를 재설정할 수 있습니다.

순차적 전달. MQSeries에서 순차 번호를 사용하여 메시지를 전송하고 수신 채널에서 메시지를 저장할 때 메시지 순서를 재설정하는 방법. 메시지를 올바른 순서로 한 번만 전달해야 할 때는 순차적 전달을 사용해야 합니다.

순환 로깅. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 모든 재시작 데이터를 일련의 로그 파일에 보관하는 프로세스. 로그 기록은 첫 번째 파일부터 시작해서 모든 파일에 순서대로 채워집니다. 이 시점에서 공간이 비거나 공간이 더 이상 필요하지 않으면 첫 번째 파일로 돌아가서 다시 시작됩니다. 순환 로깅은 재시작 복구 중 사용되는데, 로그를 사용하여 시스템이 정지했을 때 처리 중이었던 트랜잭션을 롤백합니다. 선형 로깅과 반대 개념입니다.

스레드. MQSeries에서 운영 체제 플랫폼에서 사용 가능한 병렬 실행의 최하위 레벨.

스위치 프로파일. OS/390용 MQSeries에서 MQSeries가 시작되거나 새로 고침 보안 명령이 발행될 때 사용되는 RACF 프로파일. MQSeries가 감지하는 각 스위치 프로파일은 지정된 자원에 대해 점검 기능을 끕니다.

스탠자. 구성 파일에서 큐 관리자, 클라이언트 또는 채널의 작동을 수정하는 매개변수에 값을 지정하는 일련의 행. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서는 구성(.ini) 파일에 여러 개의 스탠자가 포함될 수 있습니다.

시간 독립적 메시징. 비동기 메시징을 참조하십시오.

시스템 제어 명령. 버퍼 풀, 저장영역 클래스, 페이지 세트와 같은 플랫폼별 엔티티를 처리하는 데 사용하는 명령.

시스템 체크 인터럽트. 장비 기능 결함이나 오류로 인해 발생하는 인터럽트. 시스템 체크 인터럽트는 하드웨어 복구 가능, 소프트웨어 복구 가능 또는 복구 불가능 중 하나입니다.

신호보내기. OS/390용 MQSeries과 Windows 2.1용 MQSeries에서 예상 메시지가 큐에 도착했을 때 운영 체제가 이를 프로그램에 알리는 기능.

실행 취소/다시 실행 레코드. 복구에 사용되는 로그 레코드. 레코드의 다시 실행 부분은 MQSeries에 만들 변경사항을 설명합니다. 실행 취소 부분은 작업이 예약되지 않은 경우 변경사항의 백아웃 방법을 설명합니다.

아

아카이브 로그. 복구 로그를 참조하십시오.

알리아스 큐 오브젝트. MQSeries 오브젝트로, 이 오브젝트의 이름은 로컬 큐 관리자에 정의된 기본 큐의 별명입니다. 응용프로그램이나 큐 관리자가 알리아스 큐를 사용하면 별명 이름이 해석되고 연관된 기본 큐에서 요청된 조작이 수행됩니다.

어댑터. OS/390용 MQSeries와 TSO, IMS, CICS 또는 배치 주소 공간 사이의 인터페이스. 어댑터는 응용프로그램이 MQSeries 서비스에 액세스할 수 있도록 하는 부속 기능입니다.

여과. 오류 복구에서 미리 구축된 제어 경로를 복구 루틴에서 상위 복구 루틴으로 전달하는 것.

연결 주소 공간. 연결 주소 공간(ALLY)을 참조하십시오.

연결 주소 공간(ALLY). OS/390용 MQSeries에 연결된 OS/390 주소 공간.

연결 핸들. 프로그램이 프로그램에 연결된 큐 관리자를 액세스할 때 사용하는 ID 또는 토큰.

오브젝트. MQSeries에서 오브젝트는 큐 관리자, 큐, 프로세스 정의, 채널, 이름 목록 또는 저장영역 클래스(OS/390에만 해당)입니다.

오브젝트 설명자. 특정 MQSeries 오브젝트를 식별하는 데이터 구조. 이 설명자에는 오브젝트 이름과 오브젝트 유형이 포함됩니다.

오브젝트 핸들. 프로그램이 작업 중인 MQSeries 오브젝트에 액세스할 때 사용하는 ID 또는 토큰.

오프-로딩. OS/390용 MQSeries에서 큐 관리자의 활성 로그가 아카이브 로그로 전송되는 자동 프로세스.

완료 코드. MQI 호출이 종료된 방법을 표시하는 리턴 코드.

요청 메시지. 다른 프로그램으로부터 응답을 요청하는 메시지 유형. 응답 메시지 및 보고 메시지와 반대 개념입니다.

요청자 채널. 메시지 큐잉에서 송신자 채널이 리모트로 시작하는 채널. 요청자 채널은 통신 링크를 통해 송신자 채널에서 메시지를 받아 메시지에 지정된 로컬 큐에 넣습니다(put). 서버 채널을 참조하십시오.

유틸리티. MQSeries에서 시스템 운영자나 시스템 관리자에게 MQSeries 명령이 제공하는 기능 외의 다른 기능을 제공하는 제공된 프로그램 세트. 몇몇 유틸리티는 기능을 하나 이상 실행합니다.

응답 메시지. 요청 메시지에 응답할 때 사용하는 메시지 유형. 요청 메시지 및 보고 메시지와 반대 개념입니다.

응답자. 분산 큐잉에서 다른 시스템으로부터의 네트워크 연결 요청에 응답하는 프로그램.

응용프로그램 로그. Windows NT에서 중요한 응용프로그램 이벤트를 기록하는 로그.

응용프로그램 큐. 응용프로그램이 사용하는 큐

응용프로그램 환경. 응용프로그램이 액세스할 수 있는 소프트웨어 기능. OS/390 플랫폼의 경우, 응용프로그램 환경의 예로 CICS 및 IMS를 들 수 있습니다.

이니시에이션 큐. 큐 관리자가 트리거 메시지를 넣는(put) 로컬 큐.

이름 목록. 예를 들어 큐 이름 등 일련의 이름을 포함하는 MQSeries 오브젝트.

이름 변환. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서, 사용할 시스템에 큐 관리자 이름이 고유하고 올바르게 되도록 큐 관리자 이름을 변경하는 내부 프로세스. 외부에서는 큐 관리자 이름이 변경되지 않습니다.

이름 서비스. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서, 지정한 큐를 소유한 큐 관리자를 판별하는 기능.

이벤트. 채널 이벤트, 도구 이벤트, 성능 이벤트 및 큐 관리자 이벤트를 참조하십시오.

이벤트 데이터. 이벤트 메시지에서, 이벤트를 발생시킨 응용프로그램 또는 큐 관리자 이름 등과 같이 이벤트에 대한 정보를 포함하는 메시지 데이터 부분. 이벤트 헤더를 참조하십시오.

이벤트 로그. 응용프로그램 로그를 참조하십시오.

이벤트 메시지. MQSeries 시스템 네트워크에서 도구 이벤트 원본과 관련된 정보(예: 이벤트 범주, 이벤트를 발생시킨 응용프로그램 이름, 큐 관리자 통계)를 포함합니다.

이벤트 큐. 큐 관리자가 이벤트를 감지한 후 이벤트 메시지를 넣는(put) 큐. 큐 관리자, 성능 또는 채널 이벤트와 같은 각 이벤트 범주에는 자체의 이벤트 큐가 있습니다.

이벤트 표시기. 로그 파일 조사 및 관리를 위해 Windows NT에서 제공하는 도구.

이벤트 헤더. 이벤트 메시지에서 이벤트 이유 코드의 이벤트 유형을 식별하는 메시지 데이터 부분.

이상 종료 이유 코드. OS/390용 MQSeries에 발생한 문제점을 고유하게 식별하는 4 바이트 16진 코드. OS/390용 MQSeries 이상종료 이유 코드의 전체 목록 및 코드에 대한 설명은 *MQSeries for OS/390 Messages and Codes*를 참조하십시오.

이유 코드. MAI 호출의 실패나 부분적 성공의 이유를 설명하는 리턴 코드.

이중 로깅. OS/390용 MQSeries 작업을 기록하는 방법으로, 각각의 변경사항을 두 데이터 세트에 기록하여 재시작을 해야 할 때 한 데이터 세트를 읽을 수 없는 경우 다른 데이터 세트를 사용할 수 있습니다. 단일 로깅과 반대 개념입니다.

이중 모드. 이중 로깅을 참조하십시오.

입력 매개변수. 호출할 때 사용자가 정보를 제공하는 MQI 호출의 매개변수.

입/출력 매개변수. 호출할 때 사용자가 정보를 제공하고, 호출이 완료되거나 실패하면 큐 관리자가 정보를 변경하는 MQI 호출의 매개변수.

자

자원. 작업에서 필요로 하는 계산 시스템이나 운영 체제의 모든 기능. OS/390용 MQSeries에서는 버퍼 풀, 페이지 세트, 로그 데이터 세트, 큐, 메시지 등이 자원입니다.

자원 관리자. 메모리 버퍼와 데이터 세트 등의 공유 자원에 대한 액세스를 관리하고 제어하는 응용프로그램, 프로그램 또는 트랜잭션. MQSeries, CICS, IMS가 자원 관리자입니다.

작업 단위. 두 일관성 지점 사이에서 응용프로그램이 수행하는 복구 가능한 조각 순서. 트랜잭션이 시작되거나 사용자가 요청한 동기점 후 작업 단위가 시작됩니다. 작업 단위는 사용자가 요청한 동기점에서 또는 트랜잭션이 종료될 때 종료됩니다. 복구 단위와 반대 개념입니다.

작업 전환. 여러 작업 간의 I/O 조각 및 처리를 겹쳐서 수행하는 것. OS/390용 MQSeries에서 작업 전환기는 일부 MQI 호출을 기본 CICS TCB가 아닌 하위 작업에서 실행되도록 함으로써 성능을 최적화합니다.

저장 및 전달. 패킷, 메시지, 프레임은 목적지로 재전송하기 전에 데이터 네트워크에 임시 저장하는 것.

저장영역 클래스. OS/390용 MQSeries에서 저장영역 클래스는 특정 큐의 메시지를 보유할 페이지 세트를 정의합니다. 저장영역 클래스는 큐를 정의할 때 지정합니다.

전송 프로그램. 메시지 채널 에이전트를 참조하십시오.

전역 추적. 추적 데이터가 전체 OS/390용 MQSeries 서브시스템의 데이터가 되는 OS/390용 MQSeries 추적 옵션.

정지 종료. MQSeries에서 연결된 모든 응용프로그램이 연결을 끊도록 하는 큐 관리자 종료. 즉시 종료 및 선제 종료와 반대 개념입니다. 현재 활성 작업이 모두 완료된 후에만 어댑터가 MQSeries에서 연결을 끊는 CICS 어댑터 종료 유형. 강제 종료와 반대 개념입니다.

정지중 상태. MQSeries에서, 정지되기 전의 큐 관리자 상태. 이 상태에서는 프로그램에서 처리를 완료할 수는 있지만 새 프로그램을 시작할 수는 없습니다.

제어 명령. UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 운영 체제의 명령행으로부터 대화식으로 입력할 수 있는 명령. 이러한 명령을 실행하려면 MQSeries 제품만 설치하면 되며 별도의 유틸리티나 프로그램은 필요하지 않습니다.

제어된 종료. 정지 종료를 참조하십시오.

종료. 즉시 종료, 선제 종료 및 정지 종료를 참조하십시오.

주소 공간. 특정 작업에 사용 가능한 가상 저장영역 공간.

중단 알림. CICS 서브시스템이 OS/390용 MQSeries에 성공적으로 연결되면 활성화되는 보류 이벤트.

즉시 종료. MQSeries에서 응용프로그램이 연결을 끊을 때까지 대기하지 않는 큐 관리자 종료. 즉시 종료가 요청된 후 현재 MQI 호출은 완료할 수 있지만, 새 MQI 호출은 실패합니다. 정지 종료 및 선제 종료와 반대 개념입니다.

증상 문자열. IBM 소프트웨어 지원 데이터베이스를 검색하기 위해 만들어진 구조화된 형식으로 표시되는 진단 정보.

지속 메시지. 큐 관리자를 재시작한 후에도 남아 있는 메시지. 비지속 메시지와 반대 개념입니다.

지원된 연결. OS/390용 MQSeries를 시작하기 전에 CICS 서브시스템이 OS/390용 MQSeries에 연결하려고 시도하면 활성화되는 보류 이벤트.

차

찾아보기 커서. 메시지 큐잉에서 다음 순서에 있는 메시지를 식별하기 위해 큐를 찾아볼 때 사용되는 표시기.

채널. 메시지 채널을 참조하십시오.

채널 이벤트. 채널 인스턴스의 사용 가능 또는 사용 불가능 여부를 표시하는 이벤트. 채널 이벤트는 큐 관리자의 채널 양 끝에서 생성됩니다.

체크포인트. 중요한 정보가 로그에 기록되는 시간. 동기점과 반대 개념입니다. UNIX 시스템에 설치된 MQSeries에서는 로그에 설정된 데이터 레코드가 큐의 데이터 레코드와 같아지는 시점입니다. 체크포인트는 자동으로 생성되며 시스템을 다시 시작할 때 사용됩니다.

초기화 입력 데이터 세트. OS/390용 MQSeries 시작시 사용하는 데이터 세트.

추적. MQSeries에서 MQSeries 작업을 기록하기 위한 기능. 추적 입력 항목의 목적지에 GTF와 SMF(System Management Facility)를 포함할 수 있습니다. 전역 추적 및 성능 추적을 참조하십시오.

출력 로그-버퍼. OS/390용 MQSeries에서 아카이브 로그에 기록되기 전에 복구 로그 레코드를 보유하는 버퍼.

출력 매개변수. 호출이 완료되거나 실패하면 큐 관리자가 정보를 리턴하는 MQI 호출의 매개변수.

카

컨텍스트. 메시지 원본에 대한 정보.

컨텍스트 보안. MQSeries에서 메시지가 메시지 설명자에 있는 원본 세부사항을 수행하도록 보안을 처리하는 방법.

콜백. MQSeries에서, 요청자 메시지 채널은 먼저 송신자를 호출한 다음 종료하고 콜백을 기다리면서 송신자 채널에서 전송을 시작합니다.

큐. MQSeries 오브젝트. 메시지 큐잉 응용프로그램은 큐에 메시지를 넣거나(put) 큐에서 메시지를 가져올(get) 수 있습니다. 큐는 큐 관리자가 소유하고 관리합니다. 로컬 큐는 처리 대기 중인 메시지 목록을 포함할 수 있습니다. 다른 유형의 큐는 다른 큐를 가리키거나 다이내믹 큐의 모델로 사용될 수 있으므로 메시지를 포함할 수 없습니다.

큐 관리자. 응용프로그램에 큐잉 서비스를 제공하는 시스템 프로그램. 큐 관리자는 응용프로그램 프로그래밍 인터페이스(API)를 제공하여 큐 관리자가 소유하는 큐의 메시지에 프로그램이 액세스할 수 있도록 합니다. 로컬 큐 관리자 및 리모트 큐 관리자를 참조하십시오. 특정 큐 관리자의 속성을 정의하는 MQSeries 오브젝트.

큐 관리자 이벤트. 큐 관리자 이벤트는 다음을 나타냅니다.

- 큐 관리자가 사용하는 자원과 관련된 오류 상태가 발생했음을 나타냅니다(예: 큐가 사용 불가능한 경우).
- 큐 관리자에 중요한 변경사항이 발생했음을 나타냅니다(예: 큐 관리자 정지 또는 시작).

큐잉. 메시지 큐잉을 참조하십시오.

클라이언트. 로컬 사용자 응용프로그램에 서버의 큐잉 서비스에 대한 액세스를 제공하는 런타임 구성요소. 응용프로그램이 사용하는 큐는 서버에 상주합니다. MQSeries 클라이언트를 참조하십시오.

클라이언트 연결 채널 유형. MQSeries 클라이언트와 연관된 MQI 채널 정의 유형. 서버 연결 채널 유형을 참조하십시오.

클라이언트 응용프로그램. 워크스테이션에서 실행되고 클라이언트에 연결된 응용프로그램으로, 서버의 큐잉 서비스에 액세스할 수 있도록 하는 응용프로그램.

클러스터. 어떤 점에서 논리적으로 연관된 큐 관리자의 네트워크.

타

템포러리 다이내믹 큐. 큐를 닫을 때 삭제되는 다이내믹 큐. 큐 관리자에 오류가 발생하면 비지속 메시지만 포함할 수 있도록 템포러리 다이내믹 큐가 복구되지 않습니다. 퍼머넌트 다이내믹 큐와 반대 개념입니다.

트랜스미션 큐. 목적이 리모트 큐 관리자인 준비된 메시지가 임시로 저장되는 로컬 큐.

트랜잭션 ID. CICS에서, 트랜잭션을 정의할 때 지정되는 이름으로, 트랜잭션을 실행하는 데 사용됩니다.

트리거. MQSeries에서 큐에 미리 결정된 조건이 충족될 때 큐 관리자가 자동으로 응용프로그램을 시작하도록 하는 기능.

트리거 메시지. 트리거 모니터가 시작할 프로그램에 대한 정보를 포함하는 메시지.

트리거 모니터. 하나 이상의 이니시에이션 큐를 담당하면서 지속적으로 실행되는 응용프로그램. 이니시에이션 큐에 트리거 메시지가 도착하면 트리거 모니터가 메시지를 검색합니다. 트리거 모니터는 트리거 메시지의 정보를 사용하여 트리거 이벤트가 발생한 큐를 처리하는 프로세스를 시작합니다.

트리거 이벤트. 큐 관리자가 이니시에이션 큐에 트리거 메시지를 작성하도록 하는 이벤트(예: 큐에 메시지 도착).

과

퍼머넌트 다이내믹 큐. 삭제를 명확하게 요청한 경우에 한해 큐를 닫을 때 삭제되는 다이내믹 큐. 큐 관리자에 오류가 발생하면 지속 메시지를 포함할 수 있도록 퍼머넌트 다이내믹 큐가 복구됩니다. 템포러리 다이내믹 큐와 반대 개념입니다.

페이지 세트. OS/390용 MQSeries에서, 큐 또는 메시지 등의 데이터를 기본 저장영역 장치의 버퍼에서 DASD(지속 백업 저장영역 장치)로 이동할 때 사용되는 VSAM 데이터 세트.

프로그래밍 가능 명령 형식(PCF). 다음에서 사용하는 MQSeries 메시지 유형.

- 사용자 관리 응용프로그램, 지정된 큐 관리자의 시스템 명령 입력 큐에 PCF 명령을 넣기(put) 위해 사용
 - 사용자 관리 응용프로그램, 지정된 큐 관리자에서 PCF 명령 결과를 가져오기(get) 위해 사용
 - 큐 관리자, 이벤트가 발생했음을 표시하는 알림으로 사용
- MQSC와 반대 개념입니다.

프로세스 정의 오브젝트. MQSeries 응용프로그램 정의를 포함하는 MQSeries 오브젝트. 예를 들어, 큐 관리자가 트리거 메시지로 작업할 때 이 정의를 사용합니다.

플랫폼. MQSeries에서 큐 관리자가 실행되는 운영 체제.

하

해석 경로. 응용프로그램이 별명을 지정하거나 MQOPEN 호출에 대한 입력에 리모트 큐를 지정할 때 열리는 큐 세트.

핸들. 연결 핸들 및 오브젝트 핸들을 참조하십시오.

확약. 현재 복구 단위나 작업 단위에서 이루어진 모든 변경사항을 적용하는 작업. 작업이 완료되면 새 복구 단위나 작업 단위가 시작됩니다. 백아웃과 반대 개념입니다.

환경. 응용프로그램 환경을 참조하십시오.

활성 로그. 복구 로그를 참조하십시오.

A

APAR. APAR(Authorized Program Analysis Report)

APAR(Authorized Program Analysis Report). 변경되지 않은 현재 릴리스 프로그램에서의 의심되는 결함으로 인해 발생한 문제점 보고서.

ASID(Address Space Identifier). 주소 공간 ID.

ASID(Adress Space Identifier). 시스템이 주소 공간에 지정한 고유한 ID.

B

BMS(Basic Mapping Support). 기본 맵핑 지원.

BMS(Basic Mapping Support). 다양한 터미널에서 사용하는 제어 문자를 고려하지 않고 입/출력 디스플레이 데이터를 형식화하고 여러 페이지로 된 출력 메시지를 라우트하는 CICS와 응용 프로그램 간 인터페이스.

browse. 메시지 큐잉에서 메시지를 큐에서 제거하지 않고 복사하기 위해 MQGET 호출을 사용하는 것. 가져오기(*get*)를 참조하십시오.

BSDS(Bootstrap Data Set). 부트 스트랩 데이터 세트.

BSDS(Bootstrap Data Set). 다음을 포함하는 VSAM 데이터 세트.

- OS/390용 MQSeries에 알려진 모든 활성 로그 및 아카이브 로그 데이터 세트 인벤토리
- 최근의 모든 OS/390용 MQSeries 작업이 저장된 광범위한 인벤토리

OS/390용 MQSeries 서브시스템을 재시작해야 할 때 BSDS가 필요합니다.

C

CCF(Channel Control Function). MQSeries에서 연산자 패널 인터페이스와 함께 메시지를 트랜스미션 큐에서 통신 링크로, 통신 링크에서 로컬 큐로 이동하여 채널을 설정하고 제어하도록 하는 프로그램.

CCF(Channel Control Function). 채널 제어 기능.

CCSID(Coded Character Set Identifier). 코드화 문자 세트 이름 및 코드 포인트 할당.

CCSID(Coded Character Set Identifier). 코드화 문자 세트 ID.

CDF(Channel Definition File). MQSeries에서 트랜스미션 큐를 통신 링크와 연결하는 통신 채널 정의를 포함하는 파일.

CDF(Channel Definition File). 채널 정의 파일.

CI(Control Interval). VSAM이 레코드를 저장하고 분산된 여유 공간을 만드는 직접 액세스 저장영역 장치의 고정 길이 영역. 제어 간격은 VSAM이 직접 액세스 저장영역 장치로 전송하거나 직접 액세스 저장영역 장치에서 전송 받는 정보 단위입니다.

CI(Control Interval). 제어 간격.

CL(Contrl Language). 제어 언어(CL).

CL(Control Language). AS/400용 MQSeries에서, CL 프로그램을 작성하여 또는 명령행에서 명령을 발행하는 데 사용할 수 있는 언어.

connect. 응용프로그램이 후속 MQI 호출에 사용하는 큐 관리자 연결 핸들을 제공합니다. 연결은 MQCONN 호출을 통해 이루어지거나 MQOPEN 호출을 통해 자동으로 이루어집니다.

CPF. 명령 접두부.

CPF(Command Prefix). OS/390용 MQSeries에서 OS/390용 MQSeries 명령이 지정되고 OS/390용 MQSeries 연산자 메시지를 송신하는 큐 관리자를 식별하는 문자열.

D

DAE(Dump Analysis and Elimination). 이전에 기록된 덤프와 중복되어 필요 없게 된 SVC 덤프 및 SYSUDUMP 덤프를 설치 프로그램이 설치하지 않도록 하는 OS/390 서비스.

DAE(Dump Analysis and Elimination). 덤프 분석 및 제거.

DCE(Distributed Computing Environment). 분산 컴퓨팅 환경.

DCI(Data Conversion Interface). 데이터 변환 인터페이스.

DCI(Data Conversion Interface). 서로 다른 시스템 인코딩과 CCSID 간에 응용프로그램 데이터를 변환하는 사용자 작성 또는 공급업체 작성 프로그램이 준수해야 하는 MQSeries 인터페이스. MQSeries Framework의 일부입니다.

DLQ. 데드-레터 큐.

DQM. 분산 큐 관리.

E

ESM(External Security Manager). OS/390 시스템 권한 부여 기능이 실행하는 보안 제품. RACF가 ESM의 한 예입니다.

ESM(External Security Manager). 외부 보안 관리자.

ESTAE(Extended Specify Task Abnormal Exit). 복구 기능을 제공하며, 이상종료 처리 및 진단 또는 재시도 주소 지정에 대해 지정된 종료 루틴을 제어하는 OS/390 매크로.

ESTAE(Extended Specify Task Abnormal Exit). 작업 비정상 종료 확장 지정.

F

FFST(First Failure Support Technology). UNIX 시스템, OS/2 Warp용 MQSeries, Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries 및 AS/400용 MQSeries에서 소프트웨어 문제점을 감지하고 보고하는 데 사용됨.

FFST(First Failure Support Technology). 1차 실패 지원 기술.

FIFO(First-In-First-Out). 가장 오래 큐에 있었던 항목을 다음으로 검색하는 큐잉 기술. (A)

FIFO(First-In-First-Out). 선입선출.

Framework. MQSeries에서, 공급업체나 사용자가 MQSeries 제품의 특정 기능을 확장하거나 바꾸는 프로그램을 작성할 수 있도록 하는 프로그래밍 인터페이스 콜렉션. 다음 인터페이스가 해당됩니다.

- MQSeries DCI(Data Conversion Interface)
- MQSeries MCI(Message Channel Interface)
- MQSeries NSI(Name Service Interface)
- MQSeries SEI(Security Enabling Interface)
- MQSeries TMI(Trigger Monitor Interface)

FRR(Functional Recovery Routine). 프로그램 인터럽트가 발생할 경우 복구 루틴에서 제어할 수 있도록 하는 OS/390 복구/종료 관리자 기능.

FRR(Functional Recovery Routine). 기능 복구 루틴.

G

GCPC(Generalized Command Preprocessor). MQSeries 명령을 처리하고 실행하는 OS/390용 MQSeries 구성요소.

GCPC(Generalized Command Preprocessor). 일반 명령 프리프로세서.

get. 메시지 큐잉에서, MQGET 호출을 사용하여 큐에서 메시지를 제거하는 것. 찾아보기를 참조하십시오.

GTF. GTF(Generalized Trace Facility).

GTF(Generalized Trace Facility). 문제점을 확인하기 위해 SVC(Supervisor Call)와 같은 중요 시스템 이벤트를 기록하고 I/O 작업을 시작하는 OS/390 서비스 프로그램.

I

IPCS(Interactive Problem Control System). 대화식 문제점 제어 시스템.

IPCS(Interactive Problem Control System). 온라인 문제점 관리, 대화식 문제점 진단, 디스크가 상주하는 이상종료 덤프의 온라인 디버깅, 문제점 추적 및 문제점 보고를 허용하는 OS/390 구성요소

ISPF(Interactive System Productivity Facility). 대화식 시스템 생산성 기능.

ISPF(Interactive System Productivity Facility). 전체 화면 편집기 및 대화 상자 관리자 기능을 가진 IBM 사용 허가 제품. 응용프로그램 작성에 사용되며, 응용프로그램 프로그래머와 최종 사용자 간에 대화식 의사 전달 및 표준 화면 패널을 생성할 수 있도록 합니다.

M

MCA(Message Channel Agent). 준비된 메시지를 트랜스미션 큐에서 통신 링크로, 또는 통신 링크에서 테스트네이션 큐로 전송하는 프로그램. 메시지 큐 인터페이스를 참조하십시오.

MCA(Message Channel Agent). 메시지 채널 에이전트.

MCI(Message Channel Interface). MQSeries 큐 관리자와 다른 메시징 시스템 간에 메시지를 전송하는 사용자 작성 또는 공급업체 작성 프로그램이 준수해야 하는 MQSeries 인터페이스. MQSeries Framework의 일부입니다.

MCI(Message Channel Interface). 메시지 채널 인터페이스.

MQAI(MQSeries Administration Interface). MQSeries의 프로그래밍 인터페이스.

MQAI(MQSeries Administration Interface). MQSeries 관리 인터페이스.

MQI 채널. MQSeries 클라이언트를 서버 시스템의 큐 관리자에 연결하고 MQI 호출과 응답만을 양방향으로 전송합니다. 메시지 채널과 반대 개념입니다.

MQI(Message Queue Interface). MQSeries 큐 관리자가 제공하는 프로그래밍 인터페이스. 이 프로그래밍 인터페이스를 사용하면 응용프로그램이 메시지 큐잉 서비스에 액세스할 수 있습니다.

MQ(Message Queue Interface). 메시지 큐 인터페이스.

MQSC. MQSeries 명령.

MQSC(MQSeries Command). MQSeries 오브젝트 조작에 사용되는 읽을 수 있고 모든 플랫폼에서 규칙적인 명령. *PCF(Programmable Command Format)*와 반대 개념입니다.

MQSeries. 메시지 큐잉 서비스를 제공하는 IBM 사용 허가 제품군.

MQSeries Client. 전체 큐 관리자를 설치하지 않고 시스템에 설치할 수 있는 MQSeries 제품의 일부. MQSeries 클라이언트는 응용프로그램에서 MQI 호출을 받아 서버 시스템의 큐 관리자 와 통신합니다.

N

NSI(Name Service Interface). 이름 서비스 인터페이스.

NSI(Name Service Interface). 큐 이름 소유권을 해석하는 사용자 작성 또는 공급업체 작성 프로그램이 준수해야 하는 MQSeries 인터페이스. MQSeries Framework의 일부입니다.

NTFS(New Technology File System). 신 기술 파일 시스템.

NTFS(New Technology File System). 파일에 보안을 제공하는 Windows NT의 복구 가능한 파일 시스템.

O

OAM(Object Authority Manager). UNIX 시스템, AS/400용 MQSeries Windows NT 및 Windows 2000용 MQSeries에 설치된 MQSeries에서 명령 및 오브젝트 관리에 대한 디폴트 권한 서비스. OAM을 사용자 제공 보안 서비스로 바꾸거나 사용자 제공 서비스와 결합하여 실행할 수 있습니다.

OAM(Object Authority Manager). 오브젝트 권한 관리자.

P

PCF 명령. 프로그래밍 가능 명령 형식(PCF)을 참조하십시오.

PCF(Programmable Command Format). 프로그래밍 가능 명령 형식.

ping. 분산 큐잉에서 테스트 메시지 교환을 통해 메시지 채널이나 TCP/IP 연결이 작동하고 있음을 확인하는 진단 기능.

PTF(Program Temporary Fix). 프로그램 임시 수정.

PTF(Program Temporary Fix). 변경되지 않은 현재 릴리스 프로그램의 결함으로 인해 IBM 엔지니어가 진단한 문제점을 해결하거나 무시하는 것.

R

RBA(Relative Byte Address). 저장 레코드의 바이트 수 전 위 또는 저장영역 공간이 속하는 데이터 세트에 할당된 저장영역 공간 시작 부분의 제어 간격.

RBA(Relative Byte Address). 상대 바이트 주소.

Reslevel. OS/390용 MQSeries에서 OS/390용 MQSeries의 API 자원 보안을 위해 점검된 CICS 사용자 ID 개수를 제어하는 옵션.

Resynch. MQSeries에서 메시지 전송을 재시작하지 않고 인다우트(in-doubt) 상태 메시지를 시작하고 해석하기 위해 채널을 전달하는 옵션.

RRS(Resource Recovery Service). 자원 복구 서비스.

RRS(Resource Recovery Service). 참가하는 여러 자원 관리자에 대해 2단계 동기점 지원을 제공하는 OS/390 기능.

RTM(Recovery Termination Manager). 복구 종료 관리자.

RTM(Recovery Termination Manager). 종료 기능과 연관된 복구 루틴으로 제어를 전달함으로써 모든 정상 종료 및 비정상 종료를 처리하는 프로그램.

S

SAF(System Authorization Facility). OS/390용 MQSeries가 RACF와 같은 외부 보안 관리자와 통신할 수 있도록 하는 OS/390 기능.

SAF(System Authorization Facility). 시스템 권한 부여 기능.

SDWA(System Diagnostic Work Area). 프로그램 오류나 하드웨어 오류를 설명하는 SYS1.LOGREC 입력 항목에 기록된 데이터.

SDWA(System Diagnostic Work Area). 시스템 진단 작업 공간.

SEI(Security Enabling Interface). 보안 사용 인터페이스

SEI(Security Enabling Interface). 권한을 점검하거나 사용자 ID를 제공하며 또는 인증을 수행하는 사용자 작성 또는 공급업체 작성 프로그램이 준수해야 하는 MQSeries 인터페이스. MQSeries Framework의 일부입니다.

SIT(System Initialization Table). CICS가 시작할 때 사용하는 매개변수를 포함하는 테이블.

SIT(System Initialization Table). 시스템 초기화 테이블.

SVC(Supervisor Call). 감독관 호출.

SVC(Supervisor Call). 실행 중인 프로그램을 인터럽트하고 제어할 감독관에게 전달하여 명령에서 지시하는 특정 서비스를 수행하도록 하는 OS/390 명령.

SYS1.LOGREC. 프로그램 오류나 하드웨어 오류에 대한 정보를 포함하는 서비스 기능.

SYSTEM.COMMAND.INPUT 큐. 응용프로그램의 프로그램이 MQSeries 명령을 넣을(put) 수 있는 로컬 큐. 명령 서버가 큐에서 명령을 검색하고 검증한 다음 명령 프로세서로 전달하여 실행하도록 합니다.

T

TACL(Tandem Advanced Command Language). Tandem 고급 명령어.

TCB. TCB(Task Control Block)

TCB(Task Control Block). OS/390용 MQSeries나 CICS와 같은 OS/390 서브시스템에 연결된 주소 공간 내의 작업에 대한 정보를 전달하는 데 사용되는 OS/390 제어 블록.

THLQUALA(Target Library High-Level Qualifier). OS/390 대상 데이터 세트 이름에 대한 상위 규정자.

THLQUAL(Target Library High-Level Qualifier). 대상 라이브러리 상위 규정자.

TMI(Trigger Monitor Interface). 사용자 작성 또는 공급업체 작성 트리거 모니터 프로그램이 준수해야 하는 MQSeries 인터페이스. MQSeries Framework의 일부입니다.

TMI(Trigger Monitor Interface). 트리거 모니터 인터페이스

TRANID. 트랜잭션 ID를 참조하십시오.

U

UIS(User Identifier Service). OS/2 Warp용 MQSeries에서 MQI 응용프로그램이 사용자 ID를 디폴트 사용자 ID가 아닌 MQSeries 메시지와 연결하도록 하는 기능.

UIS(User Identifier Service). 사용자 ID 서비스.

숫자

1단계 백아웃. 진행 중인 작업을 완료할 수 없도록 하고 작업의 일부가 되는 모든 변경사항을 취소하는 방법.

1단계 확약. 프로그램이 큐의 갱신을 다른 자원 관리자가 제어하는 자원에 발생한 갱신과 통합하지 않고 큐에 업데이트를 확약할 수 있는 방법. 2단계 확약과 반대 개념입니다.

2단계 확약. 단일 트랜잭션에서 하나 이상의 자원 관리자를 사용할 때 복구 가능한 자원에 발생한 변경사항을 통합하기 위한 프로토콜. 1단계 확약과 반대 개념입니다.

특수 문자

.ini 파일. 구성 파일을 참조하십시오.

색인

[가]

감시자 Failover 모니터

설명 237

정지 247

공유 메모리 관리 387

공유 큐

구성 작업 181

관련 서적 416

관리

권한 100

로컬 39

리모트 71

오브젝트 69

채널 72

트랜스미션 큐 72

명령 세트 23

제어 명령 24

프로그램 기능 명령 형식(PCF) 25

MQSeries 명령(MQSC) 24

구문

다이아그램, 읽는 방법 260

도움말 261

오류, MQSC 명령에서 42

구문에 대한 도움말 261

구성

데이터베이스 관리자 132

데이터베이스, qm.ini 195

로그 193

변경사항 구현 184

우선순위 184

큐 관리자 구성 파일, qm.ini 185

편집 183

AllQueueManagers 스탠자, mqs.ini 185

Channels 스탠자, qm.ini 196

ClientExitPath 스탠자, mqs.ini 187

DefaultQueueManager 스탠자,

mqs.ini 187

Exitpath 스탠자, qm.ini 200

ExitProperties 스탠자, mqs.ini 187

failover.ini 240

Log 스탠자, qm.ini 193

LogDefaults 스탠자, mqs.ini 188

LU62 스탠자, qm.ini 198

구성 (계속)

mqs.ini, 설명 184

OpenVMS 클러스터 Failover 세트 240

Oracle 134

qm.ini 파일 예 201

QueueManager 스탠자, mqs.ini 190

Service 스탠자, qm.ini 191

ServiceComponent 스탠자, qm.ini 192

TCP 스탠자, qm.ini 198

XAResourceManager 스탠자,

qm.ini 195

구성 파일

큐 관리자(qm.ini)

내용 185, 201

오브젝트 권한 관리자 사용 안함 89

채널 스탠자 196

ExitPath 스탠자 200

Log 스탠자 193

LU62 및 TCP 스탠자 198

Service Component 스탠자 192

Service 스탠자 191

XARsourceManager 스탠자 195

편집 183

failover.template 355

MQSeries (mqs.ini)

경로 48

내용 184

AllQueueManagers 스탠자 185

ClientExitPath 스탠자 187

DefaultQueueManager 스탠자 187

ExitProperties 스탠자 187

LogDefaults 스탠자 188

QueueManager 스탠자 190

OpenVMS 클러스터 Failover 세트 240

구성요소

함수 표시(MONMQ) 375

구성요소 표시 명령(MONMQ) 375

권한

관리 100

권한 ID 89

대체 사용자 93

명령들 91

목록 90

설정/재설정 명령 321

권한 (계속)

설치가능 서비스 91

컨텍스트 94

dspmqaout 명령 92

MQI 97

setmqaut 명령 92

권한 파일

경로 103

관리 106

권한 부여 106

내용 104

디렉토리 104

모든 클래스 106

이해 103

클래스 106

권한 ID

권한 디폴트 88

디폴트, nobody 88

MQM 85

권한 ID, 권한에 대한 88

권한이 없는 액세스, 보호 85

규칙 테이블, DLQ 핸들러 111

구문 116

예 119

제어 데이터 항목 111

INPUTQ 키워드 111

INPUTQM 키워드 111

RETRYINT 키워드 112

WAIT 키워드 112

처리 117

패턴 및 조치(규칙) 112

ACTION 키워드 114

APPLIDAT 키워드 113

APPLNAME 키워드 113

APPLTYPE 키워드 113

DESTQ 키워드 113

DESTQM 키워드 113

FEEDBACK 키워드 113

FORMAT 키워드 113

FWDQ 키워드 114

FWDQM 키워드 115

HEADER 키워드 115

MSGTYPE 키워드 113

PERSIST 키워드 113

규칙 테이블, DLQ 핸들러 111 (계속)
 PUTAUT 키워드 115
 REASON 키워드 114
 REPLYQ 키워드 114
 REPLYQM 키워드 114
 RETRY 키워드 115
 USERID 키워드 114

기본

권한 ID 88
 시스템 오브젝트 341
 오브젝트 17
 오브젝트의 속성 51
 큐 관리자 31
 변경 35, 44
 실수로 변경 35
 실수로 삭제 267
 처리된 명령 42
 트랜스미션 큐 33
 기본 원칙
 액세스 관리 87
 여러 권한 ID를 보유할 경우 88

[나]

논리 단위

달기(MONMQ) 369
 세그먼트 표시 사용(MONMQ) 368
 논리 이름, 보안 사용 안함 89

[다]

다이나믹 큐 9

권한 부여 93
 설명 9
 대소문자 구분 27
 제어 명령 27
 큐 관리자 이름 26
 MQSC 명령 28

대체 사용자 권한 93

대화식 MQSC

사용 41
 종료 43
 피드백 42
 대화식 MQSC 명령 종료 43

덤프

로그 레코드 덤프(dmpmqlog 명령) 161
 복구 로그의 내용 덤프 161
 형식화된 시스템 로그(dmpmqlog) 명령 273

덤프 로그

매개변수 273
 목적 273
 형식 273

데드-레터 큐(DLQ)

데드-레터 큐 핸들러 실행(runmqdlq) 311
 설명 15
 지정 32
 핸들러, DLQ 핸들러 참조 109

데드-레터 헤더, MQDLH 109

테스타네이션 큐

미배달 메시지 109
 소유하는 큐 관리자의 이름 79
 정의 79

테스타네이션 큐 관리자

별명 지정 81
 정의 79
 채널 시작 75
 채널 작성 73
 클러스터에 속하지 않음 14
 트랜스미션 큐 작성 73

데이터 변환 82

디폴트 데이터 변환 83
 사용자 정의 메시지 형식 변환 83
 ConvEBCDICNewline 속성,
 AllQueueManagers 스탠자 186
 crtmqcvx 명령 264
 EBCDIC NL 문자를 ASCII로 변환 186

데이터베이스 관리자

구성 132
 구성 정보 변경 143
 데이터베이스 관리자 인스턴스, 제거 144
 스위치 로드 파일, 작성 132
 연결 131
 인다우트(in-doubt) 작업 단위 140
 제한사항, 데이터베이스 통합 지원 131
 통합 130
 dspmqtrn 명령, 미해결 UOW 점검 140
 qm.ini에서 데이터베이스 관리자 정의 132
 rsvmqtrn 명령, UOW의 명확한 재동기화 141

도구 이벤트

메세지 127
 사용 127
 사용하는 이유 125
 설명 123
 유형 125
 동기점, 성능 고려사항 213

디렉토리

권한 104
 큐 관리자 93
 디렉토리 구조 343
 디버깅
 공통 프로그래밍 오류 209
 기본 점검사항 205
 2차 점검사항 209
 디폴트 데이터 변환 83
 디폴트 변수 명령(MONMQ) 368
 디폴트 트랜스미션 큐 80

[라]

레이로드 다이어그램, 읽는 방법 260

로그

관리 152
 구성 193
 내용 덤프 161
 로그 레코드 덤프(dmpmqlog 명령) 161
 매개변수 33
 복구를 위해 사용 155
 오류 217
 오류, 예 218
 큐 관리자 145
 파일
 위치 154
 재사용 148
 제어 146
 형식 146
 dmpmqlog 명령의 출력 162
 Log 스탠자, qm.ini 193
 로그 기록
 매체 복원 156
 선형 147
 순환 146
 유형 146
 체크포인트 148
 로그 파일 217
 로그 파일 관리 153
 로그 형식 146
 로컬 관리 39
 로컬 작업 단위
 정의 129
 로컬 큐 지우기 53
 로컬 큐ocal queues
 데드-레터 15
 명령 15
 미배달 메시지 15

- 로컬 큐ocal queues (계속)
 - 삭제 53
 - 설명 11, 12
 - 이니시에이션 14
 - 정의 복사 52
 - 지우기 53
 - 트랜스미션 14
 - 한 개 정의 50
- 로케일 363
- 리모트
 - 관리 71
 - 보안 고려사항 94
 - 큐
 - 리플라이-투 큐 별명으로 81
 - 큐 관리자 별명으로 81
 - 큐 오브젝트, 작업 81
 - 큐 정의, 작성 78
 - 큐잉
 - 권장사항 77
 - MQSC 명령 발행 75
 - 리모트 관리
 - 명령 서버 67
 - 오브젝트의 69
 - 초기 문제점 77
 - 리모트 큐
 - 권한 부여 93
 - 설명 11, 13
 - 리모트 큐잉 69
 - 리스너
 - 리스너 실행(runmqtsr) 명령 사용 314
 - 리스너 종료(endmqtsr) 명령 290
 - 트러스트 404
- 리턴 코드 206
 - 값 해석 263
 - crtmqcvx 명령 264
 - crtmqm 명령 268
 - dltmqm 명령 270
 - dspmqaout 명령 277
 - dspmqcsv 명령 279
 - dspmqfls 명령 282
 - dspmqtrn 명령 286
 - endmqcsv 명령 287
 - endmqtsr 명령 290
 - endmqm 명령 292
 - endmqtrc 명령 295
 - failover 명령 297
 - rcdmqimg 명령 301
 - rcrmqobj 명령 304

- 리턴 코드 206 (계속)
 - rsvmqtrn 명령 308
 - runmqchi 명령 309
 - runmqchl 명령 310
 - runmqfm 명령 313
 - runmqtsr 명령 315
 - runmqsc 명령 317
 - runmqtrc 명령 319
 - runmqtrm 명령 320
 - setmqaut 명령 326
 - strmqcsv 명령 329
 - strmqm 명령 330
 - strmqtrc 명령 334
- 리플라이-투 큐 15
- 리플라이-투 큐 별명 81

[마]

- 마스크
 - 마스크 설정(MONMQ) 381
 - 마스크 설정 명령(MONMQ) 381
 - 마스크 표시 명령(MONMQ) 370
- 매체 이미지
 - 기록 명령 300
 - 레코드 156
 - 복구 156
 - 설명 155
- 매체 이미지 복구 156
- 마크로
 - MONMQ 388
- 메모리 표시 명령(MONMQ) 374
- 메시지
 - 검색 알고리즘 9
 - 도구 이벤트 127
 - 미배달 221
 - 변수 길이 213
 - 비지속 메시지 속도 404
 - 설명 8
 - 설명자 8, 66
 - 성능 고려사항
 - 길이 212
 - 지속 213
 - 연산자 218
 - 예상치 못한 정보를 포함하는 215
 - 최대 길이 8
 - 큐에 나타나지 않음 214
 - 큐잉 7
 - 특정 메시지 검색 213

- 메시지 (계속)
 - DOS 및 Windows 클라이언트의 오류 229
- 메시지 검색 알고리즘 9
- 메시지 길이, 감소 53
- 메시지 채널 에이전트(MCA)
 - 설명 70
 - AdoptNewMCA 속성 197
 - RETRY 상태의 채널 199
- 메시지 큐 인터페이스(MQI) 7
- 메시지 큐잉 7
- 메시지에 의한 처리 7
- 명령 서버
 - 리모트 관리 67
 - 명령 서버 시작 67
 - 명령 서버 정지 68
 - 명령 종료 287
 - 상태 표시 68
 - 시작 명령 329
 - 표시 명령 279
- 명령 세트
 - 관리 23
 - 비교 349
- 명령 큐 15
- 명령 파일 45
- 명령 프로시저 241
 - 수정 251
 - 예 251
- 명령들
 - 구문에 대한 도움말 261
 - 권한 설정/재설정(setmqaut) 90, 321
 - 권한 표시(dspmqaut) 275
 - 데드-레터 큐 핸들러 실행 311
 - 데이터 변환(crtmqcvx) 264
 - 로그 덤프(dmpmqlog) 명령 273
 - 리스너 실행(runmqtsr) 명령 314
 - 리스너 종료(endmqtsr) 명령 290
 - 매체 이미지 기록(rcdmqimg) 300
 - 명령 서버 시작(strmqcsv) 329
 - 명령 서버 종료(endmqcsv) 287
 - 명령 서버 표시(dspmqcsv) 279
 - 명령 세트 비교 349
 - 보안 명령
 - dspmqaout 92
 - setmqaut 89
 - 실패 모니터 시작(runmqfm) 313
 - 오브젝트 재작성(rcrmqobj) 303
 - 제어 24

명령들 (계속)

- 채널 시작기 실행(runmqchi) 309
- 채널 실행(runmqchl) 310
- 큐 관리자 삭제(dltmqm) 270
- 큐 관리자 시작(strmqm) 330
- 큐 관리자 작성(crtmqm) 266
- 큐 관리자 종료(endmqm) 291
- 클라이언트 트리거 모니터 시작 (runmqmtmc) 319
- 트리거 모니터 시작(runmqtrm) 320
- 프로그램 가능 명령 형식(PCF) 25
- DLQ 핸들러 실행(runmqdlq) 109
- Failover 관리(failover) 296
- MQSC
 - ALTER QLOCAL 53
 - ALTER QREMOTE 80
 - DEFINE CHANNEL 73
 - DEFINE QALIAS 58
 - DEFINE QLOCAL 52
 - DEFINE QLOCAL LIKE 52
 - DEFINE QLOCAL REPLACE 53
 - DEFINE QMODEL 60
 - DEFINE QREMOTE 78
 - DELETE QLOCAL 53
 - DELETE QREMOTE 80
 - DISPLAY QREMOTE 80
- MQSC 명령 파일
 - 입력 45
 - 출력 보고서 46
- MQSeries 명령 실행(runmqsc) 316
- MQSeries 명령(MQSC) 24
- MQSeries 추적 시작(strmqtrc) 332
- MQSeries 추적 종료(endmqtrc) 295
- MQSeries 트랜잭션 표시(dspmqtrn) 285
- MQSeries 트랜잭션 해석(rsvmqtrn) 307
- MQSeries 파일 표시(dspmqfls) 281
- MQSeries 형식화 추적 표시 (dspmqtrc) 283
- MQSeries(MQSC)
 - 사용 24
 - 확인 47
- runmqsc 41

모델 큐

- 설명 13
- 작업 60
- 정의 60

문제점

- 로컬로 MQSC 사용 48

문제점 (계속)

- 리모트로 MQSC 사용 77
- 복구 155
- MQSC 명령 실행 48
- 문제점 판별 205
 - 구성 파일 221
 - 먼저 점검해야 할 사항 205, 209
 - 명령에서 응답 없음 210
 - 올바르지 않은 출력
 - 분산 큐잉 216
 - 예상치 못한 정보를 포함하는 메시지 215
 - 큐에 나타나지 않는 메시지 214
 - 추가 점검사항 209
 - 추적 222
 - 클라이언트 228
 - 프로그래밍 오류 209
- 미배달 메시지 큐 109

[바]

- 바인딩
 - 트러스트 응용프로그램 403
- 별명
 - 리플라이-투 큐 81
 - 큐 관리자 81
- 보안 85
 - 리모트 94
 - 명령 사용 89, 92
 - 사용 89
 - Failover 세트 모니터 248
- 보호 자원 88
- 복구 시나리오
 - 디스크 드라이브 오류 159
 - 손상된 단일 오브젝트 160
 - 손상된 큐 관리자 오브젝트 160
- 분산 큐잉
 - 데드-레터 큐 15
 - 미배달 메시지 큐 15
 - 올바르지 않은 출력 216
- 비지속 메시지 404
- 빠른 경로 바인딩 403
- 빠른 경로 채널 198

[사]

- 사용
 - 보안 89
 - 이벤트 127

- 사용자 정의 메시지 형식 83
- 사용자 종료 401
 - 데이터 변환 종료 19
 - 설명 19
 - 채널 종료 19
 - 클러스터 워크로드 19, 401
- 사용자 ID
 - 권한 85, 92
 - 권한에 대한 92
 - 그룹 nobody에 속함 88
 - 기본 원칙 87
 - OpenVMS 로그인 사용자 92
- 사전정의된 큐 9
- 삭제
 - 로컬 큐 53
 - 큐 관리자 37
 - dltmqm 명령 270
- 색상 설정 명령(MONMQ) 383
- 색인 선택 취소 명령(MONMQ) 378
- 샘플
 - 추적 데이터 222
 - 프로그램, 사용 353
 - MQSC 파일 353
- 서버 연결 채널, 자동 정의 75
- 선택 명령(MONMQ) 377
- 선형 로깅 147
- 설치가능 구성요소
 - 권한 관리자(OAM) 87
 - 이름 서비스 181
- 설치가능 서비스
 - 오브젝트 권한 관리자 87
 - 오브젝트 권한 관리자 사용 안함
 - 사용 안함 89
 - 이름 서비스 181
- 성능 고려사항
 - 동기점 213
 - 메시지 길이 212
 - 메시지 지속성 213
 - 변수 메시지 길이 213
 - 추적을 사용할 때 222
 - CorrelId 213
 - MQPUT1의 이점 214
 - MsgId 213
- 성능 이벤트 125
- 성능 조정 231
- 세그먼트 표시 명령(MONMQ) 368
- 셀, DCE 및 큐 181
- 소프트카피 서적 415

속성

- 기본 51
- 대체 44
- 모든 속성 51
- 변경 53
- 큐 12
- 큐 관리자
 - 대체 44
 - 표시 43
- 큐 관리자 표시 43
- MQSC와 PCF 비교 25
- 송신자 채널 정의 70
- 수신자 채널 정의 70
- 수신자 채널, 자동 정의 75
- 순환 로깅 146
- 스위치 로드 파일, 작성 132
- 스크립트
 - MONMQ 388
- 스택 표시 명령(MONMQ) 370
- 스탠자
 - AllQueueManagers, mqs.ini 185
 - Channels, qm.ini 196
 - ClientExitPath, mqs.ini 187
 - DefaultQueueManager, mqs.ini 187
 - ExitPath, qm.ini 200
 - ExitProperties, mqs.ini 187
 - LogDefaults, mqs.ini 188
 - Log, qm.ini 193
 - LU62, qm.ini 198
 - QueueManager, mqs.ini 190
 - ServiceComponent, qm.ini 192
 - Service, qm.ini 191
 - TCP, qm.ini 198
 - XAResourceManager, qm.ini 195
- 시간 독립적 응용프로그램 7
- 시간 소인
 - 시간 소인 사용 안함(MONMQ) 380
 - 시간 소인 사용(MONMQ) 379
- 시간 소인 사용 명령(MONMQ) 379
- 시간 소인 사용 안함 명령(MONMQ) 380
- 시스템
 - 디폴트 오브젝트 17, 341
- 시작
 - 채널 74
 - 큐 관리자 34
 - Failover 세트 내에서 큐 관리자 243
- 시작 명령 프로시저 241

실행 기록

- 실행 기록 사용 안함(MONMQ) 380
- 실행 기록 사용(MONMQ) 380
- 실행 기록 삭제(MONMQ) 381
- 실행 기록 설정(MONMQ) 381
- 채널 재설정(MONMQ) 381
- 채널의 메시지 371
- 실행 기록 사용 명령(MONMQ) 380
- 실행 기록 사용 안함 명령(MONMQ) 380
- 실행 기록 삭제 명령(MONMQ) 381
- 실행 기록 표시 명령(MONMQ) 371

[아]

알리아스 큐 57

- 권한 부여 93
- 설명 13

액세스 관리 88

연결 끊기 명령(MONMQ) 377

연결 명령(MONMQ) 376

연산자 메시지 218

연산자 명령, 응답 없음 210

열기 명령(MONMQ) 368

예

- 오류 로그 218
- 트랜스미션 큐 작성 80
- 프로그래밍 오류 209
- crtmqcvx 명령 264
- crtmqm 명령 269
- dltmqm 명령 270
- dspmqaout 명령 278
- dspmqcsv 명령 279
- dspmqlfs 명령 282
- dspmqltrc 명령 283
- endmqcsv 명령 287
- endmqm 명령 292
- endmqtrc 명령 295
- failover 명령 298
- mqs.ini 파일, Compaq OpenVMS용
 - MQSeries 200
- qm.ini 파일 201
- redmqimg 명령 301
- rermqobj 명령 304
- runmqfm 명령 313
- runmqfs 명령 315
- runmqsc 명령 318
- setmqaut 명령 326
- strmqcsv 명령 329
- strmqm 명령 331

예 (계속)

- strmqtrc 명령 334

오류 로그

- 개요 217
- 설정 이전에 발생하는 오류 218
- 예 218

오류 메시지 42

오브젝트

- 관리 11
- 기본
 - 속성 51
 - 리모트 관리 69
 - 매체 이미지 155
 - 복구 156
 - 속성 11
 - 시스템 디폴트 17, 341
 - 액세스 85
 - 오브젝트를 나타내는 파일 260
 - 유형 10
 - 이름 10, 41
 - 이름 목록 16
 - 이름 변환 26
 - 이름 지정 규칙 259
 - 재작성 명령 303
 - 큐 12
 - 큐 관리자
 - MQI 호출 12
 - 트리거를 위한 61
 - 프로세스 정의 15
- 오브젝트 권한 관리자 87
 - 기본 원칙 87
 - 디폴트 권한 ID 88
 - 민감한 조작 92
 - 사용 안함 89
 - 작동 방법 87
 - dspmqaout 명령 92
 - setmqaut 명령 89, 90
- 오브젝트 권한 관리자 사용 안함 89
- 오브젝트 유형 10
- 올바르지 않은 출력 214
- 요구사항
 - 소프트웨어 337
 - 하드웨어 337
- 용량 설정 명령(MONMQ) 381
- 용어집 417
- 운영 체제 논리 이름, 보안 사용 안함 89
- 워크로드 밸런스 유지
 - 클러스터 사용 10

- 워크로드 밸런싱 유지 (계속)
 - MQI 호출 경로 재지정 58
- 응용프로그램
 - 데이터 8
 - 로컬 큐 관리자에 대한 연결 78
 - 설계 고려사항 212
 - 시간 독립적 7
 - 클라이언트-서버 환경 18
 - 트러스트 403
 - 프로그래밍 오류, 예 209
 - MQI 관리 지원 39
- 이 서적에 사용된 용어 417
- 이니시에이션 큐
 - 설명 14
 - 정의 62
- 이름
 - 오브젝트 10
 - 오브젝트에 대해 허용 259
 - 최대 문자 수 10
- 이름 목록
 - 설명 16
 - 큐 관리자 클러스터에서 사용 16
- 이름 서비스 181
- 이름 지정 규칙, 자국어 지원 259
- 이벤트
 - 도구
 - 개념 123
 - 메시지 127
 - 사용 및 사용 안함 127
 - 사용하는 이유 125
 - 설명 123
 - 유형 125
 - 유형 125
 - 이벤트 표시(MONMQ) 373
 - 채널 126
 - 큐 126
 - 트리거 126
- 이벤트 광고 126
- 이벤트 사용 안함 127
- 이벤트 유형 125
- 이벤트 큐 15
- 이벤트 표시 명령(MONMQ) 373
- 이벤트에 의한 처리 7
- 이스케이프 PCF 25, 66
- 이유 코드 11, 109, 114
- 인다우트(in-doubt) 트랜잭션
 - 데이터베이스 관리자 140

- 입력 및 출력 경로 재지정, MQSC 명령에서 41

[자]

- 자국어 지원
 - 이름 지정 규칙 259
 - EBCDIC NL 문자를 ASCII로 변환 186
- 자원
 - 보호 88
 - 보호하는 이유 85
- 작성
 - 오브젝트 11
 - 큐 12
 - 큐 관리자 30, 34
 - 프로세스 정의 62
 - crtmqm 명령 266
- 작업 단위
 - 명확한 재동기화(rsvmqtrn 명령) 141
 - 정의 129
 - 혼합 결과 142
- 재시작 복구, 순환 로깅 148
- 전역 메모리
 - sysgen 매개변수를 사용하여 제어 231
- 전역 섹션 372
- 전역 작업 단위
 - 정의 129
 - XAResourcemanager 스탠자를 qm.ini에 추가, Oracle 137
- 전역 표시 명령(MONMQ) 372
- 정리 명령 프로시저 241
- 정지 종료, 큐 관리자 35
- 제어 명령 24
 - 대소문자 구분 27
 - 사용 29
 - runmqsc 41
- 제어된 종료 35
- 제한사항
 - 데이터베이스 통합 지원 131
 - 오브젝트 이름 259
 - MQM 오브젝트에 액세스 85
- 조치 키워드, 규칙 테이블 114
- 종료
 - 사용자 종료 19
 - 채널 종료 19
 - 큐 관리자 35
 - 선제 36
 - 정지 35
 - 제어된 35, 37

- 종료 (계속)
 - 큐 관리자 35 (계속)
 - 즉시 36
 - 클러스터 워크로드 401
 - 클러스터 워크로드 종료 19
- 종료 명령 프로시저 241
- 종료 명령(MONMQ) 387
- 지정된 운영 환경 337

[차]

- 채널
 - 대상 스프레드 연결 376
 - 대상 스프레드 연결 끊기 377
 - 리모트 관리 72
 - 리모트 큐잉 69
 - 마스크 표시(MONMQ) 370
 - 메시지 실행 기록 표시 371
 - 명령 보안 요구사항 95
 - 명령들 95
 - 보안 95, 96
 - 빠른 경로 198
 - 설명 16, 69
 - 송신자 채널 정의 70
 - 수신자 채널 정의 70
 - 시작 74
 - 시작기 실행 명령 309
 - 이벤트 126
 - 이스케이프 명령 권한 100
 - 자동 정의 75
 - 정의 73
 - 채널 세부사항 표시(MONMQ) 369
 - 큐 관리자간의 정의 13
 - 트러스트 404
 - Channels 스탠자, qm.ini 196
 - run 명령 310
- 채널 표시 명령(MONMQ) 369
- 채널의 동적 정의 75
- 채널의 자동 정의 75
- 처리, 이벤트에 의한 7
- 최대
 - 메시지 길이 8, 11
 - 메시지 수 9, 12
 - 큐 용량 12
 - 큐의 크기 9
 - MQSeries 오브젝트의 이름 길이 10
- 추가 정보 407
- 추적
 - 구성요소 또는 함수 지정 377

추적 (계속)

- 데이터 샘플 222
- 샘플 MONMQ 세션 390
- 성능 고려사항 222
- 추적 사용 안함(MONMQ) 380
- 추적 사용(MONMQ) 380
- 추적 시작(MONMQ) 377
- 추적 정지(MONMQ) 377
- 추적 종료(MONMQ) 387
- 프로세스 시작 시 시작 376
- MONMQ 사용 365
- MONMQ 추적 종료 387
- 추적 분석 명령(MONMQ) 384
- 추적 사용 명령(MONMQ) 380
- 추적 사용 안함 명령(MONMQ) 380
- 추적 시작 명령(MONMQ) 377
- 추적 정지 명령(MONMQ) 377
- 출력
 - 경로 재지정 383
- 출력 설정 명령(MONMQ) 383

[카]

- 컨텍스트 권한 94
- 코드 세트 363
- 코드화 문자 세트 82
- ID 363
- 큐
 - 공유 구성 작업 181
 - 권한 부여 93
 - 다른 큐 관리자에서 공유 181
 - 다이나믹 9
 - 데드-레터 15
 - 지정 32
 - 로컬 11, 12
 - 기본 17
 - 복사 52
 - 삭제 53
 - 정의 50
 - 지우기 53
 - 리모트 11, 13
 - 작성 78
 - 작업 81
 - 큐 관리자 별명 81
 - 리플라이-투 15, 81
 - 명령 15
 - 모델 13
 - 작업 60
 - 정의 60

큐 (계속)

- 미배달 메시지 15
 - 지정 32
- 별명 13
- 분산, 올바르게 않은 출력 216
- 사전정의 9
- 설명 8
- 속성 12
 - 변경 53
- 실패 211
- 알리아스, 에 대한 작업 57
- 오브젝트
 - 로컬 12
 - 리모트 13
 - 모델 13
 - 별명 13
 - 응용프로그램, 트리거를 위한 정의 61
 - 이니시에이션
 - 정의 62
 - 트리거 메시지 14
- 이벤트 15
- 이벤트 공고 126
- 임시 9
- 작업 50
- 정의 12
 - 찾아보기 54
- 클러스터 트랜스미션 14
- 트랜스미션 14
 - 기본 33, 80
 - 리모트 관리 72
 - 작성 80
 - 정의 73
- MQSeries 응용프로그램 39
- 큐 공유 181
- 큐 관리자
 - 고유 이름 31
 - 구성 파일
 - 지정 33
 - 권한 93
 - 권한 디렉토리 104
 - 기본 31
 - 변경 35
 - 실수로 변경 35
 - 실수로 삭제 267
- 디렉토리 93
- 로그 145
- 로컬 관리 39
- 리모트 관리 69

큐 관리자 (계속)

- 리모트 큐를 사용하는 별명 81
- 매체 이미지 기록 156
- 메시지 변환 82
- 명령 서버 67
- 모니터링 125
- 복구 로그의 내용 덤프 161
- 삭제 37
- 선제 종료 36
- 선형 로깅 147
- 설명 11
- 수 31
- 순환 로깅, 재시작 복구 148
- 시작 34
- 오브젝트
 - MQI 호출 12
 - 오브젝트 권한 관리자
 - 사용 안함 89
 - 설명 87
 - 이름 변환 26
 - 이벤트 125
 - 작성 30, 34
 - 재시작 37
 - 정지 35
 - 종료
 - 정지 35
 - 제어된 35
 - 즉시 종료 36
 - 클러스터 14, 16
 - 형식화된 시스템 로그 덤프(dmpmqlog) 명령 273
 - CCSID로 변경 후 재시작 84
 - crtmqm 명령 266
 - dltmqm 명령 270
 - endmqm 명령 291
 - Failover 세트 내에서 시작 243
 - Failover 세트 내에서 이동 244
 - Failover 세트 내에서 종료 244
 - MVS/ESA에서 76
 - qm.ini 파일 185
 - runmqsc에 지정 44
- 큐 관리자 모니터 125
- 큐 관리자 선제 종료 36
- 큐 관리자 속성 대체 44
- 큐 관리자 시작 명령 330
- 큐 관리자 재시작 37
- 큐 관리자 종료 37

- 큐 관리자 클러스터
 - 클러스터 참조 16
- 큐 속성 변경 53
- 큐 용량
 - 판별 52
 - 현재 52
- 큐 찾아보기 54
- 큐잉 모드, runmqsc 76
- 클라이언트 17
 - 링크 응용프로그램 18
 - 문제점 판별 228
 - 채널 작성 18
 - 트리거 모니터 시작 명령 319
 - DOS 및 Windows의 오류 메시지 229
- 클러스터
 - 리모트 큐잉 69
 - 설명 70
 - 큐 관리자
 - 설명 16
 - 워크로드 종료 19, 401
 - 이름 목록 사용 16
 - 트랜스미션 큐 14
 - OpenVMS 클러스터와 다른 점 16
 - 큐 관리자의 10
 - ExitProperties 스탠자 속성 187
 - OpenVMS
 - 큐 관리자 클러스터와의 차이 235
 - Failover 세트 236
 - MQSeries 설치 235
- 클러스터 별명 서비스 250
- 클러스터 워크로드 종료 401
- 클러스터 트랜스미션 큐
 - 설명 14

[타]

- 테이블 표시 374
- 텍스트 닫기 명령(MONMQ) 379
- 텍스트 열기 명령(MONMQ) 379
- 템포러리 큐 9
- 트랜스미션 큐
 - 기본 14, 33, 80
 - 리모트 관리 72
 - 설명 14, 70
 - 이름 지정 79
 - 작성 80
 - 정의 73
 - 큐 관리자간의 정의 13

- 트랜스미션 큐 (계속)
 - 클러스터 14
- 트랜잭션
 - MQSeries 명령 해석 307
 - MQSeries 표시 명령 285
- 트랜잭션 지원
 - 트랜잭션 지원 129
- 트러스트 응용프로그램 403
 - 바인딩 403
- 트리거 407
 - 도구 이벤트와 비교되는 이벤트 126
 - 모니터
 - 설명 14
 - 시작 명령 320
 - 이름 목록 사용 16
 - 실패 모니터
 - 시작 명령 313
 - 오브젝트 관리 61
 - 응용프로그램 큐, 정의 61
 - 이나시메이션 큐에 있는 메시지 14
 - 이벤트 큐 127
 - 정의 7
 - 트리거를 위한 오브젝트 관리 61

[과]

- 파일
 - 구성
 - 문제점 판별시 221
 - 권한
 - 경로 103
 - 관리 106
 - 권한 부여 106
 - 내용 104
 - 모든 클래스 106
 - 이해 103
 - 클래스 106
 - 로그 제어 146
 - 이름 이해 25
 - 큐 관리자 구성 185
 - MQSeries 구성 184
 - 패턴 대조 키워드, 규칙 테이블 113
 - 퍼머넌트 큐 9
 - 포스트스크립트 형식 416
 - 표시
 - 구성요소의 16진수 ID 375
 - 권한 명령 275
 - 대상 스레드 스택 370
 - 명령 서버 명령 279

- 표시 (계속)
 - 명령 서버 상태 68
 - 큐 관리자 속성 43
 - 테이블 표시 374
 - 프로세스 정의 63
 - 활성 MQSeries 프로세스 371
 - MQSeries 트랜잭션 명령 285
 - MQSeries 파일 명령 281
 - MQSeries 형식화 추적 출력 명령 283
- 풀에 지정된 할당량 232
- 프로그래밍 오류, 예 209
- 프로그램 가능 명령 형식(PCF)
 - 관리 25
 - 권한 96
 - 명령, 설명 25
 - 보안 요구사항 95
 - 속성 25
 - 이스케이프 PCF 25
- 프로그램, 제공된 샘플 353
- 프로세스
 - 추적 시작 지점 376
- 프로세스 권한 88
- 프로세스 정의
 - 설명 15
 - 작성 62
 - 표시 63
- 프로세스 표시 명령(MONMQ) 371

[하]

- 하이퍼텍스트 표시 언어(HTML) 415
- 함수
 - 구성 요소 표시 375
- 함수 표시 명령(MONMQ) 375
- 현재의 큐 용량 52
- 환경 변수
 - MQSPREFIX 186
 - ORACLE_HOME, Oracle 134
 - ORACLE_SID, Oracle 134

[숫자]

- 16진수 ID
 - 구성요소 표시 375
- 1차 권한 ID, 권한에 대한 88
- 1차 그룹 권한 88
- 2진
 - 추적 2진 파일 닫기 379
 - 추적 2진 파일 열기 379

2진 닫기 명령(MONMQ) 379
2진 열기 명령(MONMQ) 379

A

AdoptNewMCA 속성 197
AllQueueManagers 스탠자, mqs.ini 185
amqsdlq, 샘플 DLQ 핸들러 110
APPLIDAT 키워드, 규칙 테이블 113
APPLNAME 키워드, 규칙 테이블 113
APPLTYPE 키워드, 규칙 테이블 113
AUTHORIZE 유틸리티 232

B

BookManager 415

C

ccsid.tbl 82
Channels 스탠자, qm.ini 196
ClientExitPath 스탠자, mqs.ini 187
Compaq OpenVMS용 MQSeries 개요 337
Compaq OpenVMS용 MQSeries에서 지원되
는 363
 데이터 변환 82
 큐 관리자 재시작 84
 Compaq OpenVMS용 MQSeries에서 지원
 하는 339
CorrelId, 사용시 성능 고려사항 213
crtmqcvx 명령
 리턴 코드 264
 매개변수 264
 예 264
crtmqm 명령 266
 관련 명령 269
 리턴 코드 268
 매개변수 266
 예 269

D

DCE
 구성 182
 보안 20
 셀 181
 큐 공유 181
DefaultQueueManager 스탠자, mqs.ini 187
DESTQ 키워드, 규칙 테이블 113

DESTQM 키워드, 규칙 테이블 113

DLQ 핸들러
 규칙 테이블 111
 샘플, amqsdlq 110
 실행 109
dlmqm 명령 270
 관련 명령 272
 리턴 코드 270
 매개변수 270
 예 270
DOS 클라이언트 오류 메시지 229
dspmqaut 명령 275
 관련 명령 278
 리턴 코드 277
 매개변수 275
 사용 90, 91
 예 278
dspmqcsv 명령 279
 관련 명령 279
 리턴 코드 279
 매개변수 279
 예 279
dspmqfls 명령 281
 리턴 코드 282
 매개변수 281
 예 282
dspmqtrc 명령 283
 관련 명령 284
 매개변수 283
 예 283
dspmqtrn 명령 285
 관련 명령 286
 리턴 코드 286
 매개변수 285

E

EBCDIC NL 문자를 ASCII로 변환 186
EndCommand 프로시저
 템플릿 359
endmqcsv 명령 287
 관련 명령 289
 리턴 코드 287
 매개변수 287
 예 287
endmqlsr(리스너 종료) 명령
 리턴 코드 290
 매개변수 290

endmqlsr(리스너 종료) 명령 (계속)

 목적 290
 형식 290
endmqm 명령 35, 291
 관련 명령 294
 리턴 코드 292
 매개변수 291
 예 292
endmqtrc 명령 295
 관련 명령 295
 리턴 코드 295
 매개변수 295
 예 295
Euro 지원 339
Exitpath 스탠자, qm.ini 200
ExitProperties 스탠자, mqs.ini 187

F

Failback
 설명 237
Failover
 설명 237
failover 명령 238, 241, 296
 관련 명령 299
 리턴 코드 297
 매개변수 296
 예 298
Failover 모니터
 감시자 237
 설명 237
 시작 243
 정지 247
Failover 세트
 관리 242
 구성 239
 구성 예 250
 구성 이전 단계 238
 구성 파일 238
명령 프로시저 241
문제점 해결 249
상태 변경 247
상태 표시 245
설명 236
큐 관리자 시작 243
큐 관리자 이동 244
큐 관리자 종료 244
ICC(Intra Cluster Communication) 사용
 248

Failover 세트 (계속)
 OpenVMS용 MultiNet 사용 250
 Failover 세트 관리 296
 Failover 세트 템플릿 355
 failover.ini 구성 파일 250
 편집 240
 failover.template 250
 FEEDBACK 키워드, 규칙 테이블 113
 FFST
 FFST(MONMQ) 386
 FFST 명령(MONMQ) 386
 FFST, 조사 223
 FORMAT 키워드, 규칙 테이블 113
 FWDQ 키워드, 규칙 테이블 114
 FWDQM 키워드, 규칙 테이블 115

H

HEADER 키워드, 규칙 테이블 115
 HTML(Hypertext Markup Language) 415

I

ICC(Intra Cluster Communication) 248
 INPUTQ 키워드, 규칙 테이블 111
 INPUTQM 키워드, 규칙 테이블 111

L

LIKE 속성 52
 ListenerBacklog 속성 200
 Log 스탠자, qm.ini 193
 LogDefaults 스탠자, mqs.ini 188
 LU 닫기 명령(MONMQ) 369
 LU62 스탠자, qm.ini 198

M

MAXMSGL 11
 MaxMsgLength 속성 11
 MODPARAMS.DAT 파일 232
 monmq 유틸리티

개요 365
 공유 메모리 관리 387
 명령들
 구성요소 표시 375
 디폴트 변수 368
 마스크 설정 381
 마스크 표시 370

monmq 유틸리티 (계속)
 명령들 (계속)
 메모리 표시 374
 색상 설정 383
 색인 선택 취소 378
 선택 377
 세그먼트 표시 368
 스택 표시 370
 시간 소인 사용 379
 시간 소인 사용 안함 380
 실행 기록 사용 380
 실행 기록 사용 안함 380
 실행 기록 삭제 381
 실행 기록 표시 371
 연결 끊기 377
 열기 368
 용량 설정 381
 이벤트 표시 373
 전역 표시 372
 종료 387
 채널 표시 369
 추적 분석 384
 추적 사용 380
 추적 사용 안함 380
 추적 시작 365, 377
 추적 정지 377
 출력 설정 383
 텍스트 닫기 379
 텍스트 열기 379
 프로세스 표시 371
 함수 표시 375
 2진 닫기 379
 2진 열기 379
 connect 376
 FFST 386
 lu 닫기 369
 mutex 표시 372
 onstartup 시작 376
 onstartup 정지 376
 set free 381
 MQSeries 프로세스 추적
 샘플 추적 세션 390
 MONMQ의 변수 366
 MONMQ의 스크립트 및 매크로 388

MQAI

설명 66
 MQDATA 229
 MQDLH, 데드-레터 헤더 109

MQI

권한 96, 97
 로컬 관리 지원 39
 설명 7
 큐 관리자 호출 12

MQM

권한 ID 85
 사용자 ID 92

MQOPEN 권한 97

MQPUT 권한 97

MQPUT 및 MQPUT1, 성능 고려사항 214

MQSC

대소문자 구분 28
 대화식 명령 발행 41
 대화식 입력 종료 43
 리모트로 발행 75
 명령 발행 방법 40
 명령 사용 44
 명령 파일
 실행 47
 입력 45
 출력 보고서 46
 명령 확인 47
 명령들 24
 문제점
 로컬 48
 리모트 77
 샘플 파일 353
 속성 25
 시간 종료 명령 응답 76
 이스케이프 PCF 25
 입력 및 출력 경로 재지정 41
 채널에 대한 보안 요구사항 96
 최대 줄 길이 46
 텍스트 파일에서 명령 실행 44

MQSC 명령

대소문자 구분 28
 대화식 발행 41
 사용 24
 최대 줄 길이 46
 ALTER QLOCAL 53
 ALTER QREMOTE 80
 DEFINE CHANNEL 73
 DEFINE QALIAS 58
 DEFINE QLOCAL 52
 DEFINE QLOCAL LIKE 52
 DEFINE QLOCAL REPLACE 53
 DEFINE QMODEL 60

MQSC 명령 (계속)

- DEFINE QREMOTE 78
- DELETE QLOCAL 53
- DELETE QREMOTE 80
- DISPLAY QREMOTE 80

MQSC 명령에서의 응답 시간 종료 76

MQSC 명령의 최대 줄 길이 46

MQSC 명령의 피드백 42

MQSC 명령의 확인 47

MQSeries 명령 24

MQSeries 명령의 발행 40

MQSeries 서적 413

MQSeries 추적 시작 명령 332

MQSeries 추적 종료 295

MQSNOAUT 논리 이름 89

MQSPREFIX, 환경 변수 186

mqs.ini

- 경로 48
- 우선순위 184
- 정의 183
- 편집 183
- AllQueueManagers 스탠자 185
- ClientExitPath 스탠자 187
- DefaultQueueManager 스탠자 187
- ExitProperties 스탠자 187
- LogDefaults 스탠자 188
- QueueManager 스탠자 190

MQZAO 상수 및 권한 97

MsgId, 사용시 성능 고려사항 213

MSGTYPE 키워드, 규칙 테이블 113

mutex 테이블 372

mutex 표시 명령(MONMQ) 372

MVS/ESA 큐 관리자 76

N

NL 문자, EBCDIC를 ASCII로 변환 186

nobody, 디폴트 권한 ID 88

O

OAM 87

onstartup 시작 명령(MONMQ) 376

onstartup 정지 명령(MONMQ) 376

OpenVMS

- 개요 337
- 소프트웨어 요구사항 337
- 하드웨어 요구사항 337

OpenVMS 권한 ID

- 디폴트, nobody 88

OpenVMS 권한 ID (계속)

- MQM 85

OpenVMS 로그인 사용자 ID 92

OpenVMS 클러스터 235

OpenVMS 클러스터 Failover 세트 239

OpenVMS용 Digital TCP/IP 서비스 242

OpenVMS용 MQSeries

- 권한 ID, MQM 85
- OpenVMS 개요 337

OpenVMS용 MultiNet 250

- 템플릿 파일 구성 242

Oracle

- 구성 134
- 구성 매개변수, 변경 138
- 스위치 로드 파일, 작성 134
- 최소 지원 레벨 134
- 환경 변수 설정, 점점 134
- Oracle XA 지원, 사용 134
- ORACLE_HOME, 환경 변수 134
- ORACLE_SID, 환경 변수 134
- XAResourceManager 스탠자, qm.ini에 추가 137

P

PCF 명령

- 이스케이프 PCF 66
- MQAI, 간편함을 위해 사용 66
- MQSC 및 PCF의 속성 66
- PCF를 사용한 관리 작업 자동화 65

PDF(Portable Document Format) 415

PDFPortable Document Format) 415

PERSIST 키워드, 규칙 테이블 113

PUTAUT 키워드, 규칙 테이블 115

Q

qm.ini 구성 파일

- 우선순위 184
- 정의 185
- 채널 스탠자 196
- 편집 183
- Exitpath 스탠자 200
- Log 스탠자 193
- LU62 스탠자 198
- Service 스탠자 191
- ServiceComponent 스탠자 192
- TCP 스탠자 198
- XAResourceManager 스탠자 195

QueueManager 스탠자, mqs.ini 190

R

rcdmqimg 명령 300

- 관련 명령 302
- 리턴 코드 301
- 매개변수 300
- 예 301

rcrmqobj 명령 303

- 관련 명령 306
- 리턴 코드 304
- 매개변수 303
- 예 304

REASON 키워드, 규칙 테이블 114

REPLACE 속성, DEFINE 명령 46

REPLYQ 키워드, 규칙 테이블 114

REPLYQM 키워드, 규칙 테이블 114

RETRY 키워드, 규칙 테이블 115

RETRYINT 키워드, 규칙 테이블 112

rsvmqtrn 명령 307

- 관련 명령 308
- 리턴 코드 308
- 매개변수 307

runmqchi 명령 309

- 리턴 코드 309
- 매개변수 309

runmqchl 명령 310

- 리턴 코드 310
- 매개변수 310

runmqdlq 명령 109

runmqfm 명령 237, 313

- 관련 명령 313
- 리턴 코드 313
- 매개변수 296, 313
- 예 313

runmqslr(리스너 실행) 명령

- 리턴 코드 315
- 매개변수 314
- 목적 314
- 예 315
- 형식 314

runmqsc

- 대화식 사용 41
- 문제점 48
- 사용 44
- 종료 43
- 큐 관리자 지정 44
- 큐잉 모드 76

runmqsc (계속)
 피드백 42
 확인 47
 MQSC 명령의 발행 40
 runmqsc 명령 316
 리턴 코드 317
 매개변수 317
 예 318
 입력 및 출력 경로 재지정 41
 runmqtmc 명령 319
 리턴 코드 319
 매개변수 319
 runmqtrm 명령 320
 리턴 코드 320
 매개변수 320

S

SAFEGUARD 338
 Service 스탠자, qm.ini 191
 ServiceComponent 스탠자, qm.ini 192
 set free 명령(MONMQ) 381
 setmqaut 명령 321
 관련 명령 328
 리턴 코드 326
 매개변수 323
 사용 90
 설치가능 서비스 91
 예 326
 StartCommand 프로시저
 템플리트 357
 strmqscv 명령 329
 관련 명령 329
 리턴 코드 329
 매개변수 329
 예 329
 strmqm 명령 330
 관련 명령 331
 리턴 코드 330
 매개변수 330
 예 331
 strmqtrc 명령 332
 관련 명령 334
 리턴 코드 334
 매개변수 332
 예 334
 SupportPac 416
 sysgen 매개변수 231

SYSS\$INPUT, runmqsc 44
 SYSS\$OUTPUT, runmqsc 44

T

TCP 스탠자, qm.ini 198
 TCP/IP
 구성 198
 리모트 관리 17
 리스너 처리에 대해 정의 314
 연결 거부 199
 지원 패키지 337
 채널 설정 72
 채널 시작 74
 OpenVMS 클러스터 조작에 필수 238
 OpenVMS 클러스터 지원 237
 OpenVMS용 Digital TCP/IP 서비스 242
 StartCommand 프로시저에 의해 구성
 241
 TidyCommand 프로시저
 템플리트 362

U

USERID 키워드, 규칙 테이블 114

W

WAIT 키워드, 규칙 테이블 112
 Windows 도움말 416
 Windows 클라이언트 오류 메시지 229

X

XA 준수 관계형 데이터베이스 131
 XAResourceManager 스탠자, qm.ini 195



부품 번호: CT8YNKO

Printed in Singapore

SA30-1334-00



CT8YNKO

