

7.5

IBM WebSphere MQ 계획

IBM

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [143 페이지의 『주의사항』](#)에 있는 정보를 확인하십시오.

This edition applies to version 7 release 5 of IBM® WebSphere® MQ and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM은 귀하가 IBM으로 보낸 정보를 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 사용하거나 배포할 수 있습니다.

© Copyright International Business Machines Corporation 2007년, 2024.

목차

계획 중.....	5
IBM MQ 및 IBM MQ 어플라이언스에서 GDPR 준비를 위한 온프레미스 고려사항.....	5
IBM WebSphere MQ 아키텍처 설계.....	13
단일 큐 관리자 아키텍처.....	14
단일 큐 관리자 아키텍처.....	14
단일 큐 관리자 아키텍처.....	14
단일 큐 관리자 아키텍처.....	14
다중 큐 관리자 아키텍처.....	14
포인트-투-포인트 메시징.....	18
발행/구독 소개.....	18
데드-레터 큐 핸들러를 사용하여 미배달 메시지 핸들링.....	99
다중 설치 계획.....	108
기본 설치 선택.....	109
스토리지 및 성능 요구사항 계획.....	113
디스크 공간 요구사항.....	113
파일 시스템 지원 계획.....	115
IBM WebSphere MQ 및 UNIX System V IPC 자원.....	139
AIX의 공유 메모리.....	140
UNIX 프로세스 우선순위 값 설정.....	140
HP Integrity NonStop Server 에서 IBM WebSphere MQ 클라이언트 환경 계획.....	140
HP Integrity NonStop Server 환경 준비.....	140
IBM WebSphere MQ 및 HP NonStop TMF.....	141
HP NonStop TMF 사용.....	141
주의사항.....	143
프로그래밍 인터페이스 정보.....	144
상표.....	144

계획 중

IBM WebSphere MQ 환경을 계획하는 경우 구성하려는 IBM WebSphere MQ 아키텍처, 자원 요구사항, 로깅에 대한 필요 및 백업 기능을 고려해야 합니다. 이 토픽의 링크를 사용하여 IBM WebSphere MQ가 실행되는 환경을 계획하십시오.

IBM WebSphere MQ 환경을 계획하기 전에 기본 IBM WebSphere MQ 개념에 익숙해지십시오. [기술 개요](#)의 토픽을 참조하십시오.

관련 개념

[가용성, 복구 및 재시작](#)

관련 태스크

[마이그레이션](#)

[설치](#)

[구성](#)

[WebSphere MQ 관리](#)

[메시지가 유실되지 않았는지 확인\(로깅\)](#)

IBM MQ 및 IBM MQ 어플라이언스에서 GDPR 준비를 위한 온프레미스 고려 사항

관련 PID:

- 5724-H72 IBM MQ
- z/OS® 용 5655-AV9 IBM MQ Advanced
- 5655-AV1 IBM MQ Advanced for z/OS, Value Unit Edition
- z/OS 용 5655-AM9 IBM MQ Advanced Message Security
- 5725-Z09 IBM MQ Appliance M2001
- 5725-S14 IBM MQ Appliance M2000
- z/OS 용 5655-MQ9 IBM MQ
- 5655-VU9 IBM MQ for z/OS Value Unit Edition
- 5639-L92 IBM MQ 인터넷 Pass-Thru
- z/OS 용 5655-MF9 IBM MQ Managed File Transfer
- 5655- z/OS 용 ADV IBM WebSphere MQ Advanced
- 5655-AMS IBM WebSphere MQ Advanced Message Security for z/OS
- 5724-R10 멀티플랫폼용 IBM WebSphere MQ File Transfer Edition
- HP NonStop Server용 5724-A39 IBM WebSphere MQ
- 5724-A38 HP OpenVMS용 IBM WebSphere MQ
- 5655-W97 IBM WebSphere MQ for z/OS
- z/OS Value Unit Edition용 5655-VU8 IBM WebSphere MQ
- 5655-VUE IBM WebSphere MQ for z/OS Value Unit Edition
- x86 용 5725-C79 IBM WebSphere MQ Hypervisor Edition for Red Hat Enterprise Linux®
- 5725-F22 IBM WebSphere MQ AIX®
- z/OS 용 5655-MFT IBM WebSphere MQ Managed File Transfer

주의사항:

이는 귀사의 GDPR 대비를 준비하도록 도와주기 위한 것입니다. 이는 사용자가 구성할 수 있는 IBM MQ 의 기능 및 GDPR 대비 과정에서 조직을 지원하기 위해 고려해야 하는 제품 사용의 측면에 대한 정보를 제공합니다. 고객이 기능을 선택하고 구성할 수 있는 여러 방법이 있고 본 제품을 그 자체로 그리고 제3자 애플리케이션 및 시스템과 병행하여 사용할 수 있는 다양한 방식이 있으므로 이 정보는 완전한 목록은 아닙니다.

고객은 유럽 연합의 GDPR을 포함한 다양한 법령과 규정을 준수해야 할 책임이 있습니다. 고객은 고객의 비즈니스에 영향을 줄 수 있는 관련 법령 및 규정에 대한 확인과 해석, 그러한 법령 및 규정의 준수를 위해 필요한 고객의 모든 조치와 관련하여 적절한 법률 자문을 받아야 할 단독 책임이 있습니다.

본 콘텐츠에 설명된 제품, 서비스 및 기타 기능은 일부 고객의 상황에 적합하지 않거나 사용 가능성이 제한될 수 있습니다. IBM은 법률, 회계 또는 감사 관련 자문을 제공하지 않으며, IBM의 제품이나 서비스가 고객의 관련 법령 및 규정을 준수한다는 진술이나 보증을 제공하지 않습니다.

목차

1. [GDPR](#)
2. [GDPR을 위한 제품 구성](#)
3. [데이터 수명 주기](#)
4. [데이터 수집](#)
5. [데이터 스토리지](#)
6. [데이터 액세스](#)
7. [데이터 처리](#)
8. [데이터 삭제](#)
9. [데이터 모니터링](#)
10. [개인정보 사용 제한 기능](#)
11. [파일 핸들링](#)

GDPR

일반 개인정보 보호법률(General Data Protection Regulation, "GDPR")은 유럽 연합("EU")에 의해 채택되어 2018년 5월 25일부터 적용됩니다.

GDPR이 중요한 이유

GDPR은 개인정보 처리를 위해 보다 엄격한 정보 보호 규제 체계를 규정하고 있습니다. GDPR 규정 사항은 다음과 같습니다.

- 개인을 위한 새롭고 강화된 권리
- 개인정보의 정의 확장
- 처리자에 대한 새로운 의무
- 미준수에 대한 상당한 금전적 제재 가능성
- 데이터 유출(breach)에 대한 의무적 고지

GDPR에 대해 자세히 알아보기:

- [EU GDPR 정보 포털](#)
- ibm.com/GDPR 웹 사이트

제품 구성 - GDPR 대비를 위한 고려사항

다음 절에서는 GDPR을 구성하는 데 도움이 되도록 IBM MQ 구성에 대한 고려사항을 제공합니다.

데이터 수명 주기

IBM MQ는 애플리케이션이 애플리케이션에서 제공한 데이터를 비동기적으로 교환할 수 있도록 하는 트랜잭션 메시지 지향 미들웨어 제품입니다. IBM MQ는 애플리케이션 연결을 위해 메시징 API, 프로토콜 및 브릿지의 범위를 지원합니다. 이와 같이, IBM MQ는 많은 형태의 데이터를 교환하기 위해 사용될 수 있으며, 이들 중 일부는 잠재적으로 GDPR에 영향을 받을 수 있습니다. IBM MQ가 데이터를 교환할 수 있는 몇 가지 써드파티 제품이 있습니다. 이들 중 일부는 IBM-소유이지만, 기타 여러 제품은 다른 기술 공급자에 의해 제공됩니다. [소프트웨어 제품 호환성 보고서 웹 사이트](#)에서는 연관된 소프트웨어의 목록을 제공합니다. 써드파티 제품의 GDPR 대비 관련 고려사항은 제품 문서를 참조해야 합니다. IBM MQ 관리자는 IBM MQ가 큐, 토픽 및 구독의 정의를 사용하여, 전달되는 데이터와 상호작용하는 방식으로 제어합니다.

IBM MQ를 통해 전달되는 유형의 데이터

IBM MQ는 애플리케이션 데이터에 대해 비동기 메시징 서비스를 제공하므로, 애플리케이션 배치를 통해 유스케이스가 다양해지기 때문에 이 질문에 대한 하나의 명확한 응답은 없습니다. 애플리케이션 메시지 데이터는 큐 파일(z/OS에서의 커플링 기능 또는 데이터 세트)에 지속되며, 로그와 아카이브 및 메시지 자체는 GDPR이 관리하는 데이터를 포함할 수 있습니다. 애플리케이션 제공 메시지 데이터도 오류 로그, 추적 파일 및 FFST와 같이 문제점 판별 용도로 수집된 파일에 포함될 수 있습니다. 또한 z/OS 애플리케이션 제공 메시지 데이터가 주소 공간 또는 커플링 기능 덤프에 포함될 수도 있습니다.

다음은 IBM MQ를 사용하여 교환할 수 있는 개인 데이터의 일반적인 예입니다.

- 고객의 직원(예: IBM MQ는 고객의 급여 또는 HR 시스템을 연결하는 데 사용될 수 있음)
- 고객이 소유한 고객의 개인 데이터(예: IBM MQ는 고객과 관련된 애플리케이션 사이에서 데이터를 교환하기 위해 사용할 수 있습니다. 예를 들어, 판매 리드를 가져오고 해당 고객의 CRM 시스템 내에 데이터를 저장할 수 있습니다.)
- 고객이 보유한 고객의 민감한 개인 데이터 (예: IBM MQ는 개인 데이터를 교환해야 하는 산업 컨텍스트 내에서 사용될 수 있습니다(예: 임상 애플리케이션 통합 시 HL7 기반 의료 레코드).)

애플리케이션 제공 메시지 데이터 외에 IBM MQ는 다음 유형의 데이터를 처리합니다.

- 인증 신임 정보(예: 사용자 이름 및 비밀번호, API 키 등)
- 일반적으로 식별 가능한 개인 정보(예: 디바이스 ID, 사용 기반 ID, IP 주소 등 - 개인으로 링크될 때)

IBM과의 온라인 문의에 사용되는 개인 데이터

IBM MQ 클라이언트는 온라인 댓글/피드백/요청을 제출하여 IBM에 IBM MQ 주제에 대해 주로 다음과 같은 다양한 방법으로 문의할 수 있습니다.

- [IBM Developer](#)의 IBM MQ 영역에 있는 페이지의 공개 댓글 영역
- [IBM Documentation](#)의 IBM MQ 제품 정보 페이지에 있는 공개 댓글 영역
- [IBM 지원 센터 포럼](#)의 공개 댓글
- [IBM Developer](#)의 IBM RFE 커뮤니티에 있는 공개 댓글

일반적으로 사용자 문의사항에 대한 개별적인 답변에는 고객 이름 및 이메일 주소만 사용되며, 이러한 개인정보 사용은 IBM 온라인 개인정보처리방침을 준수합니다.

데이터 수집

IBM MQ는 개인 데이터를 수집하는 데 사용될 수 있습니다. IBM MQ의 사용을 평가하고 GDPR 요구에 맞출 필요가 있는 경우, 사용자 환경에서 IBM을 통해 전달되는 개인 데이터 유형을 고려해야 합니다. 다음과 같은 측면을 고려할 수 있습니다.

- 데이터가 어떤 방식으로 큐 관리자에 도달합니까? (어떤 프로토콜을 거칩니까? 데이터가 암호화되어 있습니까? 데이터가 서명되어 있습니까?)
- 큐 관리자에서 데이터를 송신하는 방법은 무엇입니까? (어떤 프로토콜을 거칩니까? 데이터가 암호화되어 있습니까? 데이터가 서명되어 있습니까?)
- 데이터가 큐 관리자를 통해 전달될 때 어떻게 저장됩니까? (메시지가 비지속적인 경우에도 모든 메시징 애플리케이션은 Stateful 매체에 메시지 데이터를 기록할 수 있습니다. 메시징 기능이 제품을 통해 전달되는 애플리케이션 메시지 데이터의 측면을 잠재적으로 노출하는 방식을 알고 있습니까?)

- IBM MQ가 써드파티 애플리케이션에 액세스하는 데 필요한 경우 신임 정보 수집 및 저장 방법

IBM MQ는 인증이 필요한 다른 시스템 및 서비스와 통신할 필요가 있을 수 있습니다(예: LDAP). 필요한 경우, 인증 데이터(사용자 ID, 비밀번호)는 이러한 통신으로 사용하도록 IBM MQ에서 구성되고 저장됩니다. 가능한 경우, IBM MQ 인증에 개인 신임 정보를 사용하지 않아야 합니다. 인증 데이터에 사용되는 스토리지 보호를 고려하십시오. (다음 데이터 스토리지를 참조하십시오.)

데이터 스토리지

메시지 데이터가 큐 관리자를 통해 이동할 때, IBM MQ는 상태 저장 매체에 데이터를 직접적으로(여러 개의 사본으로) 지속합니다. IBM MQ 사용자는 메시지가 쉬는 동안 메시지 데이터를 보호하는 것을 고려할 수 있습니다.

다음 항목은 IBM MQ가 애플리케이션 제공 데이터를 지속하는 영역을 강조표시하며, 사용자가 GDPR 준수를 확인할 때 고려해야 할 수도 있습니다.

- 애플리케이션 메시지 큐:

IBM MQ는 메시지 큐를 제공하여 애플리케이션 사이에서 비동기 데이터 교환을 허용합니다. 큐에 저장된 비지속적 메시지 및 지속적 메시지는 **stateful** 매체에 기록됩니다.

- 파일 전송 에이전트 큐:

IBM MQ 관리 파일 전송은 파일 데이터의 신뢰할 수 있는 전송을 조정하기 위해 메시지 큐를 사용하며, 개인 데이터를 포함하는 파일 및 전송 레코드가 이러한 큐에 저장됩니다.

- 전송 큐:

큐 관리자 간에 안전하게 메시지를 전송하기 위해 메시지를 전송 큐에 임시 저장합니다.

- 데드 레터 큐:

메시지를 목적지 큐에 넣을 수 없어 데드-레터 큐(큐 관리자에 구성되어 있는 경우)에 저장하는 상황이 있습니다.

- 백아웃 큐:

JMS 및 XMS 메시징 인터페이스는 다른 올바른 메시지를 처리할 수 있도록 많은 백아웃이 발생한 후 포이즌 메시지를 백아웃 큐로 이동할 수 있게 하는 기능을 제공합니다.

- AMS 오류 큐:

IBM MQ Advanced Message Security 는 보안 정책을 준수하지 않는 메시지를 SYSTEM.PROTECTION.ERROR.QUEUE 오류 큐.

- 보유된 발행:

IBM MQ는 보유된 발행 기능을 제공하여 구독 애플리케이션이 이전 발행물을 다시 호출할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- [로깅: 메시지가 유실되지 않았는지 확인](#)
- [MFT 에이전트 큐 설정](#)
- [전송 큐 정의](#)
- [데드-레터 큐 사용](#)
- [JMS용 IBM MQ 클래스에서 포이즌 메시지 처리](#)
- [AMS 오류 처리](#)
- [보유된 발행물](#)

다음 항목은 IBM MQ가 GDPR 준수를 확인할 때 사용자가 고려할 수도 있는 애플리케이션 제공 데이터를 간접적으로 유지할 수 있는 영역을 강조표시합니다.

- 추적 라우트 메시징:

IBM MQ는 추적 라우트 기능을 제공하여, 메시지를 라우트하는 레코드가 애플리케이션에 위치합니다. 생성된 이벤트 메시지에는 IP 주소와 같은 기술적으로 식별 가능한 개인 정보가 포함될 수 있습니다.

- 애플리케이션 활동 추적:

IBM MQ는 애플리케이션 및 채널의 메시징 API 활동을 기록하는 애플리케이션 활동 추적을 제공하며, 애플리케이션 활동 추적은 애플리케이션 제공 메시지 데이터의 콘텐츠를 이벤트 메시지에 기록할 수 있습니다.

- 서비스 추적:

IBM MQ는 메시지 데이터가 플로우되는 내부 코드 경로를 기록하는 서비스 추적 기능을 제공합니다. 이러한 기능의 일부로서, IBM MQ는 애플리케이션 제공 메시지의 콘텐츠를 디스크에 저장된 추적 파일에 기록할 수 있습니다.

- 큐 관리자 이벤트:

IBM MQ는 권한, 명령 및 구성 이벤트와 같은 개인 데이터를 포함할 수 있는 이벤트 메시지를 생성할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- [라우트 추적 메시징](#)
- [추적 사용](#)
- [이벤트 모니터링](#)
- [큐 관리자 이벤트](#)

애플리케이션 제공 메시지 데이터의 사본에 대한 액세스를 보호하려면 다음 조치를 고려하십시오.

- 파일 시스템의 IBM MQ 데이터에 대한 권한 부여된 사용자 액세스를 제한합니다. 예를 들어, UNIX 플랫폼에서 'mqm' 그룹의 사용자 멤버십을 제한하십시오.
- 전용 큐 및 액세스 제어를 통해 애플리케이션 액세스를 IBM MQ 데이터로 제한합니다. 애플리케이션 간 큐와 같은 자원의 불필요한 공유를 피하고 큐 및 토픽 자원에 대한 세부 단위의 액세스 제어를 제공하십시오.
- IBM MQ Advanced Message Security를 사용하여 메시지 데이터의 엔드-투-엔드 서명 및/또는 암호화를 제공합니다.
- 파일 레벨 또는 볼륨 레벨 암호화를 사용하여 추적 로그를 저장하는 데 사용된 디렉토리의 콘텐츠를 보호하십시오.
- 서비스 추적을 IBM으로 업로드한 후, 잠재적으로 개인 데이터를 포함하고 있는 콘텐츠를 고려하는 경우 서비스 추적 파일 및 FFST 데이터를 삭제할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

- [권한이 있는 사용자](#)
- [멀티플랫폼에서 파일 시스템 지원 계획](#)

IBM MQ 관리자는 신임 정보(사용자 이름과 암호, API 키 등)로 3rd 당사자 서비스(예: LDAP, IBM Cloud® Product Insights, Salesforce 등) 이 데이터는 일반적으로 파일 시스템 권한을 통해 보호되는 큐 관리자 데이터 디렉토리에 저장됩니다.

IBM MQ 큐 관리자가 작성되면, 데이터 디렉토리는 IBM MQ가 구성 파일을 읽고 신임 정보를 사용하여 이 시스템에 연결할 수 있도록 그룹 기반 액세스로 설정됩니다. IBM MQ 관리자는 권한있는 사용자로 간주되며 이 그룹의 구성원은 파일에 대한 읽기 액세스 권한을 가지고 있습니다. 일부 파일은 변조되지만 암호화되지는 않습니다. 이러한 이유로 신임 정보에 대한 액세스를 완전히 보호하려면 다음 조치를 고려해야 합니다.

- IBM MQ 데이터에 대한 권한 부여된 사용자 액세스를 제한하십시오. 예를 들어, UNIX 플랫폼에서 'mqm' 그룹의 멤버십을 제한합니다.
- 파일 레벨 또는 볼륨 레벨 암호화를 사용하여 큐 관리자 데이터 디렉토리의 콘텐츠를 보호하십시오.
- 프로덕션 구성 디렉토리의 백업을 암호화하고 적절한 액세스 제어를 사용하여 이를 저장하십시오.
- 인증 실패에 대한 감사 추적, 액세스 제어 및 보안에 대한 구성 변경, 명령 및 구성 이벤트 제공을 고려하십시오.

자세한 내용은 다음을 참조하십시오.

- [IBM MQ 보안](#)

데이터 액세스

IBM MQ 큐 관리자 데이터는 다음 제품 인터페이스를 통해 액세스할 수 있으며, 이들 중 일부는 원격 연결을 통해 액세스하도록 설계되었으며 다른 일부는 로컬 연결을 통해 액세스할 수 있습니다.

- IBM MQ Console [원격 전용]
- IBM MQ REST API [원격 전용]
- MQI[로컬 및 원격]
- JMS[로컬 및 원격]
- XMS[로컬 및 원격]
- IBM MQ Telemetry (MQTT) [원격 전용]
- IBM MQ Light (AMQP) [원격 전용]
- IBM MQ IMS 브릿지 [로컬 전용]
- IBM MQ CICS 브릿지 [로컬 전용]
- HTTP용 IBM MQ 브릿지 [원격 전용]
- IBM MQ MFT 프로토콜 브릿지 [원격 전용]
- IBM MQ Connect:Direct 브릿지 [원격 전용]
- IBM MQ Bridge to Salesforce [원격 전용]
- IBM MQ Bridge to Blockchain [원격 전용]
- IBM MQ MQAI [로컬 및 원격]
- IBM MQ PCF 명령 [로컬 및 원격]
- IBM MQ MQSC 명령 [로컬 및 원격]
- IBM MQ 탐색기 [로컬 및 원격]

이 인터페이스는 사용자가 IBM MQ 큐 관리자와 이에 저장된 메시지를 변경할 수 있도록 설계되었습니다. 관리 및 메시징 조작은 요청이 이루어질 때 세 단계를 거치도록 보안됩니다.

- 인증
- 역할 매핑
- 권한 부여

인증:

메시지 또는 관리 조작이 로컬 연결에서 요청되면 이 연결의 소스가 동일한 시스템에서 프로세스를 실행합니다. 프로세스를 실행하는 사용자는 운영 체제에서 제공하는 인증 단계를 거쳐야 합니다. 연결이 이루어지는 프로세스 소유자의 사용자 이름은 ID로 확인됩니다. 예를 들어, 이는 애플리케이션이 시작된 셸을 실행하는 사용자의 이름이 될 수 있습니다. 로컬 연결에 대해 가능한 인증 양식은 다음과 같습니다.

1. 확인된 사용자 이름(로컬 OS)
2. 선택적 사용자 이름 및 비밀번호(OS, LDAP 또는 사용자 정의 써드파티 저장소)

원격 연결에서 관리 조치가 요청된 경우 IBM MQ와의 통신은 네트워크 인터페이스를 통해 작성됩니다. 다음 ID 양식은 네트워크 연결을 통해 인증에 제공될 수 있습니다.

1. 확인된 사용자 이름(원격 OS에서)
2. 사용자 이름 및 비밀번호(OS, LDAP 또는 사용자 정의 써드파티 저장소)
3. 소스 네트워크 주소(예: IP 주소)
4. X.509 디지털 인증서(상호 SSL/TLS 인증)
5. 보안 토큰(예: LTPA2 토큰)
6. 기타 사용자 정의 보안(써드파티 엑시트로 제공되는 기능)

역할 매핑:

역할 매핑 단계에서 인증 단계에 제공된 신임 정보는 대체 사용자 ID에 매핑될 수 있습니다. 매핑된 사용자 ID가 진행되도록 허용되는 경우(예를 들어, 관리 사용자가 채널 인증 규칙에 의해 차단될 수 있음), 매핑된 사용자 ID는 IBM MQ 자원에 대해 활동을 인가할 때 최종 단계로 전달됩니다.

권한 부여:

IBM MQ는 다른 사용자가 큐, 토픽 및 기타 큐 관리자 오브젝트와 같은 다른 메시징 자원에 대해 서로 다른 권한을 가질 수 있는 기능을 제공합니다.

로그 기록 활동:

IBM MQ의 일부 사용자는 MQ 자원에 대한 액세스의 감사 레코드를 작성해야 할 수도 있습니다. 원하는 감사 로그 예에는 요청한 사용자 외에 변경사항에 대한 정보가 포함된 구성 변경사항이 포함될 수 있습니다.

다음 정보 소스를 사용하여 이 요구사항을 구현할 수 있습니다.

1. 관리 명령이 성공적으로 실행되었을 때 IBM MQ 큐 관리자가 명령 이벤트를 생성하도록 구성할 수 있습니다.
2. 큐 관리자 자원이 작성, 변경되거나 삭제된 경우 구성 이벤트를 생성하도록 IBM MQ 큐 관리자를 구성할 수 있습니다.
3. 자원에 대한 인증 확인이 실패할 때 권한 이벤트를 생성하도록 IBM MQ 큐 관리자를 구성할 수 있습니다.
4. 실패한 권한 검사를 표시하는 오류 메시지는 큐 관리자 오류 로그에 기록됩니다.
5. 인증, 권한 확인이 실패하거나 큐 관리자가 작성, 시작, 중지 또는 삭제될 때 IBM MQ 웹 콘솔은 감사 메시지를 로그에 작성합니다.

이러한 종류의 솔루션을 고려할 때 IBM MQ 사용자는 다음 사항을 고려할 수 있습니다.

- 이벤트 메시지는 비지속적이므로 큐 관리자가 재시작될 때 정보는 손실됩니다. 이벤트 모니터는 사용 가능한 메시지를 지속적으로 이용하고 콘텐츠를 지속적 매체로 전송할 수 있도록 구성해야 합니다.
- IBM MQ 권한이 있는 사용자는 이벤트를 사용 불가능하게 하거나 로그를 지우거나 큐 관리자를 삭제할 수 있는 충분한 권한을 가지고 있습니다.

IBM MQ 데이터에 대한 액세스 확보 및 감사 추적 제공에 대한 자세한 정보는 다음 주제를 참조하십시오.

- [IBM MQ 보안 메커니즘](#)
- [구성 이벤트](#)
- [명령 이벤트](#)
- [오류 로그](#)

데이터 처리

공개 키 인프라를 사용하는 암호화:

IBM MQ에 대한 네트워크 연결을 보안하여 TLS를 사용할 수 있으며, 연결의 시작 측의 상호 인증을 제공할 수도 있습니다.

전송 메커니즘에서 제공하는 PKI 보안 기능 사용은 IBM MQ를 사용한 데이터 처리 보안을 위한 첫 번째 단계입니다. 하지만 추가 보안 기능을 사용하지 않으면 처리 애플리케이션이 메시지 원본을 유효성 검증하지 않거나 메시지가 전송 시 대체되었는지를 유효성 검증하지 않은 채 전달된 모든 메시지를 처리합니다.

AMS(Advanced Message Security) 기능을 사용하는 라이선스가 있는 IBM MQ의 사용자는 보안 정책의 정의와 구성을 통해 애플리케이션이 메시지에 있는 개인 데이터를 처리하는 방식으로 제어할 수 있습니다. 보안 정책을 사용하면 디지털 서버 및/또는 암호화를 애플리케이션 간 메시지 데이터에 적용할 수 있습니다.

메시지를 이용하여 메시지가 진짜인지 확인할 때 보안 정책을 사용하여 디지털 서명을 요구하고 유효성 검증할 수 있습니다. AMS 암호화는 메시지 데이터가 의도된 수신인 또는 메시지이고 올바른 복호화 키에 대한 액세스 권한이 있는 경우 읽기 가능한 양식에서 다른 애플리케이션이 디코딩만 할 수 있는 인코딩된 버전으로 변환하는 방법을 제공합니다.

네트워크 연결을 보안하기 위해 SSL 및 인증서를 사용하는 방법에 대한 자세한 정보는 IBM MQ V9 제품 문서에서 다음 주제를 참조하십시오.

- [IBM MQ의 TLS 보안 구성](#)
- [AMS 개요](#)

데이터 삭제

IBM MQ는 제품에 제공된 데이터를 삭제하기 위한 명령 및 사용자 인터페이스 조치를 제공합니다. 이를 사용하여 IBM MQ의 사용자는 필요한 경우 특정 개인과 관련된 데이터를 삭제할 수 있습니다.

- GDPR 클라이언트 데이터 삭제를 준수하기 위해 고려할 수 있는 IBM MQ 작동의 영역
 - 다음을 수행하여 애플리케이션 큐에 저장된 메시지 데이터를 삭제합니다.
 - 메시지 만기를 사용하거나 메시지 API 또는 도구로 개별 메시지를 제거합니다.
 - 비지속적 메시지 클래스가 정상인 큐에 보관된 메시지가 비지속적임을 지정하고 큐 관리자를 재시작합니다.
 - 관리 면에서 큐를 지웁니다.
 - 큐를 삭제합니다.
 - 다음을 수행하여 토픽에 저장한 보유된 발행 데이터를 삭제합니다.
 - 메시지가 비지속적임을 지정하고 큐 관리자를 재시작합니다.
 - 메시지 만기를 사용하거나 보유된 데이터를 새 데이터로 바꿉니다.
 - 관리 면에서 토픽 문자열을 지웁니다.
 - 전체 큐 관리자를 삭제하여 큐 관리자에 저장된 데이터를 삭제합니다.
 - 추적 디렉토리에서 파일을 삭제하여 서비스 추적 명령으로 저장한 데이터를 삭제합니다.
 - 오류 디렉토리에서 파일을 삭제하여 저장된 FFST 데이터를 삭제합니다.
 - 주소 공간 및 커플링 기능 덤프(z/OS에서)를 삭제하십시오.
 - 이와 같은 데이터의 아카이브, 백업 또는 다른 사본을 삭제합니다.
- GDPR 계정 데이터 삭제를 준수하기 위해 고려할 수 있는 IBM MQ 작동의 영역
 - 다음을 제거하여(아카이브, 백업 또는 복제된 복사본) 써드파티 서비스 및 큐 관리자에 연결하기 위해 IBM MQ에서 저장한 계정 데이터와 환경 설정을 제거할 수 있습니다.
 - 신임 정보를 저장한 큐 관리자 인증 정보 오브젝트.
 - 사용자 ID를 참조하는 큐 관리자 권한 레코드.
 - 특정 IP 주소, 인증서 DN 또는 사용자 ID를 맵핑하거나 차단하는 큐 관리자 채널 인증 규칙.
 - 큐 관리자 및 파일 서버를 사용하여 인증하기 위해 IBM MQ 관리 파일 전송 에이전트, 로거 및 MQ 탐색기 MFT 플러그인에서 사용되는 신임 정보 파일.
 - SSL/TLS 연결 또는 IBM MQ AMS(Advanced Message Security)에서 사용될 수 있는 키 저장소에서 개인에 대한 정보를 나타내거나 포함하는 X.509 디지털 인증서.
 - 시스템 로그 파일에 있는 계정에 대한 참조를 포함하여 IBM MQ Appliance의 개별 사용자 계정.
 - IBM MQ 탐색기 작업공간 메타데이터 및 Eclipse 설정.
 - **비밀번호 환경 설정에 지정된 IBM MQ Explorer 비밀번호 저장소입니다.**
 - IBM MQ 콘솔 및 mqweb 서버 구성 파일.
 - Salesforce 연결 데이터 구성 파일.
 - 블록체인 연결 데이터 구성 파일.
 - IBM Cloud qm.ini 및 APIKeyFile의 ReportingService 스탠자 아래에 있는 Product Insights 연결 데이터.

자세한 내용은 다음을 참조하십시오.

- [IBM MQ Bridge to Salesforce 구성](#)
- [블록체인에 사용할 IBM MQ 구성](#)
- [MFT 및 IBM MQ 연결 인증](#)
- [ProtocolBridgeCredentials.xml 파일을 사용하여 파일 서버의 신임 정보 맵핑](#)
- [IBM MQ 콘솔 사용자 및 역할 구성](#)

데이터 모니터링

IBM MQ는 사용자가 이용할 수 있는 모니터링 기능 범위를 제공하여 애플리케이션 및 큐 관리자가 수행하는 방법을 보다 잘 이해할 수 있도록 합니다.

IBM MQ는 큐 관리자 오류 로그를 관리하는 데 도움이 되는 여러 기능도 제공합니다.

자세한 내용은 다음을 참조하십시오.

- [IBM MQ 네트워크 모니터링](#)
- [진단 메시지 서비스](#)
- [QMErrorLog 서비스](#)

IBM MQ는 IBM MQ 사용자가 큐 관리자 시동 및 사용 정보를 볼 수 있도록 사용자가 IBM Cloud Product Insights 서비스에 정보를 공개할 수 있도록 하는 기능을 제공합니다.

자세한 내용은 다음을 참조하십시오.

- [IBM Cloud에서 IBM Cloud Product Insights 서비스에 사용할 IBM MQ 구성](#)

개인정보 사용 제한 기능

이 문서에서 요약된 기능을 사용하여, IBM MQ는 일반 사용자가 개인 데이터 사용을 제한할 수 있도록 합니다.

IBM MQ 메시지 큐는 데이터베이스와 동일한 방식으로 영구 데이터 저장소로 사용해서는 안 됩니다. 특히 GDPR에 적용되는 애플리케이션 데이터를 처리할 때 사용해서는 안 됩니다.

검색 조회를 통해 데이터를 찾을 수 있는 데이터베이스와 달리 큐, 메시지 및 메시지의 상관 ID를 모르면 메시지 데이터를 찾기 어렵습니다.

개인의 데이터를 포함하는 메시지를 쉽게 식별하여 찾을 수 있는 경우, 표준 IBM MQ 메시징 기능을 사용하여 데이터에 액세스하거나 수정할 수 있습니다.

파일 핸들링

1. IBM MQ Managed File Transfer는 전송된 파일에 대해 악성코드 스캔을 수행하지 않습니다. 파일 데이터를 전송 중 수정하지 않도록 파일은 있는 그대로 전송되며 무결성 검사를 수행합니다. 소스와 대상 체크섬은 전송 상태 발행의 일부로 발행됩니다. 일반 사용자는 MFT가 파일을 전송하기 전과 MFT가 원격 엔드포인트로 파일을 전달한 후에 환경에 맞게 악성코드 스캐닝을 구현하는 것이 좋습니다.
2. IBM MQ Managed File Transfer는 MIME 유형 또는 파일 확장자에 따라 조치를 수행하지 않습니다. MFT는 파일을 읽고 입력 파일에서 읽은 그대로 바이트를 전송합니다.

IBM WebSphere MQ 아키텍처 설계

포인트-투-포인트 및 Pub/Sub 메시징 스타일에 대해 IBM WebSphere MQ에서 지원하는 서로 다른 아키텍처에 대해 확인하십시오.

IBM WebSphere MQ 아키텍처를 계획하기 전에, 기본 IBM WebSphere MQ 개념을 숙지하십시오. [IBM WebSphere MQ 기술 개요](#)에 있는 토픽을 참조하십시오.

IBM WebSphere MQ 아키텍처 범위는 단일 큐 관리자를 사용하는 단순 아키텍처에서 상호 연결된 큐 관리자의 보다 복잡한 네트워크에 이릅니다. 다중 큐 관리자가 분산 큐 기술을 사용하여 함께 연결되어 있습니다. 단일 큐 관리자 및 다중 큐 관리자 아키텍처에 대한 자세한 정보는 다음 주제를 참조하십시오.

- [14 페이지의 『단일 큐 관리자에 기반한 아키텍처』](#)
- [14 페이지의 『다중 큐 관리자에 기반한 아키텍처』](#)
- [15 페이지의 『네트워크 및 네트워크 계획』](#)
- [WebSphere MQ 분산 메시징 기술](#)

논리적으로 관련되고 데이터 및 애플리케이션을 공유해야 하는 다중 큐 관리자가 필요한 경우 클러스터에서 함께 그룹화할 수 있습니다. 클러스터를 사용하면 구성 및 관리를 단순화하며 리모트 큐 정의 또는 추가 채널 정의를 설정하지 않고도 큐 관리자가 서로 통신할 수 있습니다. 클러스터 사용에 대한 자세한 정보는 [클러스터 작동 방법](#)을 참조하십시오.

관련 개념

[5 페이지의 『계획 중』](#)

IBM WebSphere MQ 환경을 계획하는 경우 구성하려는 IBM WebSphere MQ 아키텍처, 자원 요구사항, 로깅에 대한 필요 및 백업 기능을 고려해야 합니다. 이 토픽의 링크를 사용하여 IBM WebSphere MQ가 실행되는 환경을 계획하십시오.

관련 태스크

[구성](#)

단일 큐 관리자에 기반한 아키텍처

가장 단순한 IBM WebSphere MQ 아키텍처는 단일 큐 관리자 구성 및 사용입니다.

사용자의 IBM WebSphere MQ 아키텍처를 계획하기 전에, 기본 IBM WebSphere MQ 개념에 익숙해야 하며, [IBM WebSphere MQ 소개](#)를 참조하십시오.

단일 큐 관리자를 사용하는 가능한 많은 아키텍처는 다음 절에서 설명됩니다.

- [14 페이지의 『서비스에 액세스하는 로컬 애플리케이션이 있는 단일 큐 관리자』](#)
- [14 페이지의 『클라이언트로서 서비스에 액세스하는 원격 애플리케이션이 있는 단일 큐 관리자』](#)
- [14 페이지의 『발행/구독 구성의 단일 큐 관리자』](#)

서비스에 액세스하는 로컬 애플리케이션이 있는 단일 큐 관리자

단일 큐 관리자에 기반한 첫 번째 아키텍처는 서비스에 액세스하는 애플리케이션이 서비스를 제공하는 애플리케이션과 같은 시스템에서 실행 중입니다. IBM WebSphere MQ 큐 관리자는 서비스를 요청하는 애플리케이션과 서비스를 제공하는 애플리케이션 간에 비동기 상호통신을 제공합니다. 이는 애플리케이션 중 하나가 장기간 오프라인 상태일 때에도 애플리케이션 간의 통신이 계속될 수 있음을 의미합니다.

클라이언트로서 서비스에 액세스하는 원격 애플리케이션이 있는 단일 큐 관리자

단일 큐 관리자에 기반한 두 번째 아키텍처에는 서비스를 제공하는 애플리케이션에서 원격으로 실행하는 여러 애플리케이션이 있습니다. 원격 애플리케이션은 서비스에 대한 각기 다른 시스템에서 실행 중입니다. 애플리케이션은 단일 큐 관리자에 클라이언트로서 연결합니다. 이는 단일 큐 관리자를 통해 여러 시스템에 한 서비스에 대한 액세스를 제공할 수 있음을 의미합니다.

이 아키텍처의 제한사항은 애플리케이션이 작동하려면 네트워크 연결이 사용 가능해야 한다는 점입니다. 네트워크에서 애플리케이션과 큐 관리자 간의 상호작용은 동기식입니다.

발행/구독 구성의 단일 큐 관리자

단일 큐 관리자를 사용한 대체 아키텍처는 발행/구독 구성을 사용하는 것입니다. 발행/구독 메시징에서는 해당 정보의 이용자로부터 정보 제공자를 분리할 수 있습니다. 이는 메시지를 넣을 큐 이름과 같이 대상 애플리케이션에 대한 정보를 애플리케이션이 알아야 하는 이전에 설명한 아키텍처의 포인트 투 포인트 메시징 스타일과 다릅니다. IBM WebSphere MQ 발행/구독을 사용하여, 송신 애플리케이션은 정보의 주제를 기반으로 지정된 토픽을 사용하여 메시지를 발행합니다. IBM WebSphere MQ는 구독을 통해 이 주제에 대한 관심을 등록한 애플리케이션으로 메시지 분배를 핸들링합니다. 수신 애플리케이션도 수신할 메시지의 소스에 대한 정보를 알고 있지 않아도 됩니다. 발행/구독 메시징에 대한 자세한 정보는 [WebSphere MQ 발행/구독 메시징 소개](#)를 참조하십시오. 단일 큐 관리자를 사용하는 발행/구독 메시징의 예는 [단일 큐 관리자 발행/구독 구성 예](#)를 참조하십시오.

관련 개념

[13 페이지의 『IBM WebSphere MQ 아키텍처 설계』](#)

포인트-투-포인트 및 Pub/Sub 메시징 스타일에 대해 IBM WebSphere MQ에서 지원하는 서로 다른 아키텍처에 대해 확인하십시오.

관련 정보

[WebSphere MQ 소개](#)

[큐 관리자 작성 및 관리](#)

다중 큐 관리자에 기반한 아키텍처

분산 메시징 큐잉 기술을 사용하여 다수의 큐 관리자를 구성하고 사용하는 IBM WebSphere MQ 아키텍처를 작성할 수 있습니다.

사용자의 IBM WebSphere MQ 아키텍처를 계획하기 전에, 기본 IBM WebSphere MQ 개념에 익숙해야 하며, [IBM WebSphere MQ 소개](#)를 참조하십시오.

IBM WebSphere MQ 아키텍처는 추가 큐 관리자를 추가해서 서비스를 제공하는 애플리케이션을 변경하지 않고도 변경이 가능합니다.

애플리케이션은 큐 관리자와 동일한 시스템에서 호스트될 수 있으며 다른 시스템의 또 다른 큐 관리자에서 호스트되는 서비스와 비동기식으로 통신할 수 있습니다. 또는 서비스에 액세스하는 애플리케이션이 클라이언트로서 큐 관리자에 연결한 후 다른 큐 관리자의 서비스에 대한 비동기 액세스를 제공할 수 있습니다.

다른 큐 관리자와 해당 큐에 연결하는 라우트는 분산 큐잉 기술을 사용하여 정의됩니다. 아키텍처 내의 큐 관리자는 채널을 사용하여 연결됩니다. 채널은 큐 관리자의 구성에 따라 한 큐 관리자에서 다른 큐 관리자에 한 방향으로, 메시지를 자동으로 이동시키는 데 사용됩니다.

IBM WebSphere MQ 네트워크 계획에 대한 상위 레벨 개요는 [15 페이지의 『네트워크 및 네트워크 계획』](#)의 내용을 참조하십시오.

IBM WebSphere MQ 아키텍처의 채널을 계획하는 방법에 대한 정보는 [WebSphere MQ 분산 메시징 기술의 내용](#)을 참조하십시오.

분산 큐 관리를 통해 큐 관리자 간의 통신을 작성하고 모니터링할 수 있습니다. 분산 큐 관리에 대한 자세한 정보는 [분산 큐 관리 소개](#)를 참조하십시오.

관련 개념

[WebSphere MQ 소개](#)

[13 페이지의 『IBM WebSphere MQ 아키텍처 설계』](#)

포인트-투-포인트 및 Pub/Sub 메시징 스타일에 대해 IBM WebSphere MQ에서 지원하는 서로 다른 아키텍처에 대해 확인하십시오.

관련 태스크

[큐 관리자 작성 및 관리](#)

네트워크 및 네트워크 계획

WebSphere MQ는 큐 관리자 및 채널을 사용하여 네트워크에서 애플리케이션 간에 데이터를 송신하고 수신합니다. 네트워크 계획은 네트워크에서 이 시스템에 연결하기 위한 프레임워크를 작성하는 요구사항 정의를 포함합니다.

채널은 통신해야 하는 다른 시스템과 사용자 시스템 간에 작성할 수 있습니다. 멀티홉 채널을 작성하여 직접 연결이 없는 시스템에 연결할 수 있습니다. 시나리오에 설명된 메시지 채널 연결은 [16 페이지의 그림 1](#)에 네트워크 다이어그램으로 표시됩니다.

채널 및 전송 큐 이름

전송 큐에 어느 이름이나 지정할 수 있습니다. 하지만 해당하는 경우 혼란을 피하기 위해 목적지 큐 관리자 이름 또는 큐 관리자 알리어스 이름과 동일한 이름을 제공할 수 있습니다. 이는 사용하는 라우트와 전송 큐를 연관시켜서 중간(멀티호핑된) 큐 관리자를 통해 작성된 병렬 라우트의 명확한 개요를 제공합니다.

이는 채널 이름에는 명확하지 않습니다. 예를 들어, QM2에 대한 [16 페이지의 그림 1](#)에 있는 채널 이름은 들어오는 채널과 나가는 채널에 따라 달라야 합니다. 모든 채널 이름은 전송 큐 이름을 여전히 포함할 수 있지만 고유한 이름이 되도록 규정해야 합니다.

예를 들어, QM2에는 QM1에서 들어오는 QM3 채널이 하나 있고 QM3으로 나가는 QM3 채널이 하나 있습니다. 이름을 고유하게 하려면 첫 번째 이름을 'QM3_from_QM1'으로 지정하고 두 번째 이름을 'QM3_from_QM2'로 지정할 수 있습니다. 이러한 방식으로 채널 이름은 이름의 첫 번째 부분에 전송 큐 이름을 표시합니다. 방향 및 인접 큐 관리자 이름은 이름의 두 번째 부분에 표시됩니다.

[16 페이지의 그림 1](#)에 대해 제안되는 채널 이름 목록은 [16 페이지의 표 1](#)과 같습니다.

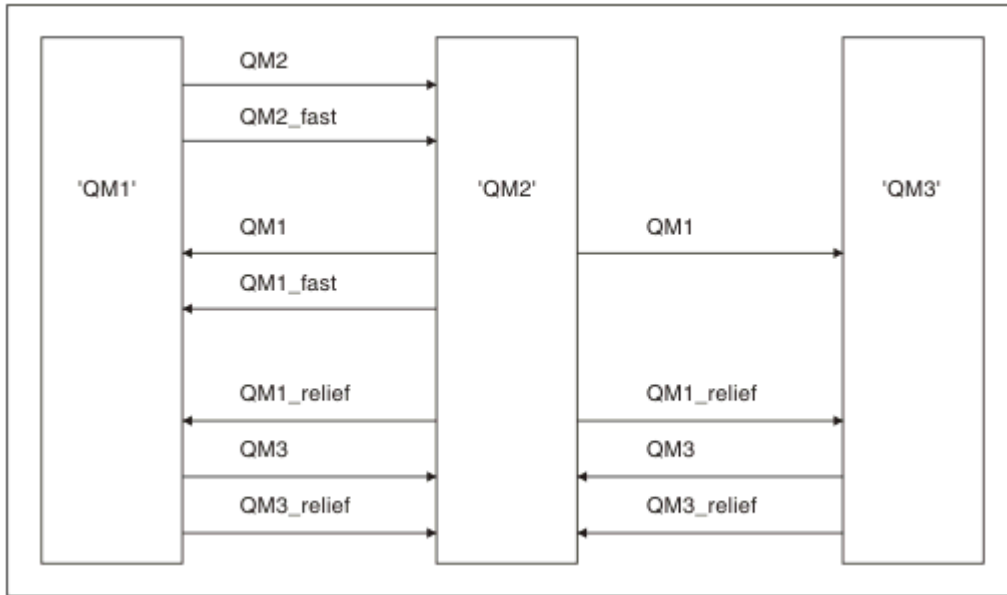


그림 1. 모든 채널이 표시된 네트워크 다이어그램

라우트 이름	채널을 호스트하는 큐 관리자	전송 큐 이름	제안된 채널 이름
QM1	QM1 & QM2	QM1(QM2에서)	QM1.from.QM2
QM1	QM2 & QM3	QM1(QM3에서)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_fast(QM2에서)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief(QM2에서)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief(QM3에서)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2(QM1에서)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_fast(QM1에서)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3(QM1에서)	QM3.from.QM1
QM3	QM2 & QM3	QM3(QM2에서)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief(QM1에서)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief(QM2에서)	QM3_relief.from.QM2

참고:

1. On WebSphere MQ for z/OS, queue manager names are limited to four characters.
2. 네트워크의 모든 채널 이름을 고유하게 지정하십시오. 16 페이지의 표 1에 표시된 대로, 채널 이름에 소스 및 대상 큐 관리자 이름을 포함하는 것은 좋은 방법입니다.

네트워크 계획표

네트워크 작성 시에는 팀의 다른 멤버가 계획을 구현하는 네트워크 계획표의 상위 레벨 기능이 하나 더 있다고 가정합니다.

광범위하게 사용되는 애플리케이션의 경우, 17 페이지의 그림 2에 표시된 대로 로컬 액세스 사이트 간의 광대역 링크를 사용하여 메시지 트래픽의 집중을 위한 로컬 액세스 사이트의 관점에서 생각하는 것이 보다 경제적입니다.

이 예에는 두 개의 기본 시스템 및 여러 위성 시스템이 있습니다. 실제 구성은 비즈니스 고려사항에 따라 다릅니다. 두 개의 집선기 큐 관리자는 편의상 중앙에 위치해 있습니다. 각 QM 집선기에는 로컬 큐 관리자로 향하는 메시지 채널이 있습니다.

- QM 집선기 1에는 세 개의 로컬 큐 관리자 QM1, QM2, QM3 각각에 대한 메시지 채널이 있습니다. 이 큐 관리자를 사용하는 애플리케이션은 QM 집선기를 통해 서로 통신할 수 있습니다.
- QM 집선기 2에는 세 개의 로컬 큐 관리자 QM4, QM5, QM6 각각에 대한 메시지 채널이 있습니다. 이 큐 관리자를 사용하는 애플리케이션은 QM 집선기를 통해 서로 통신할 수 있습니다.
- QM 집선기에는 서로 간에 메시지 채널이 있어서 큐 관리자의 애플리케이션이 다른 큐 관리자의 다른 애플리케이션과 메시지를 교환할 수 있습니다.

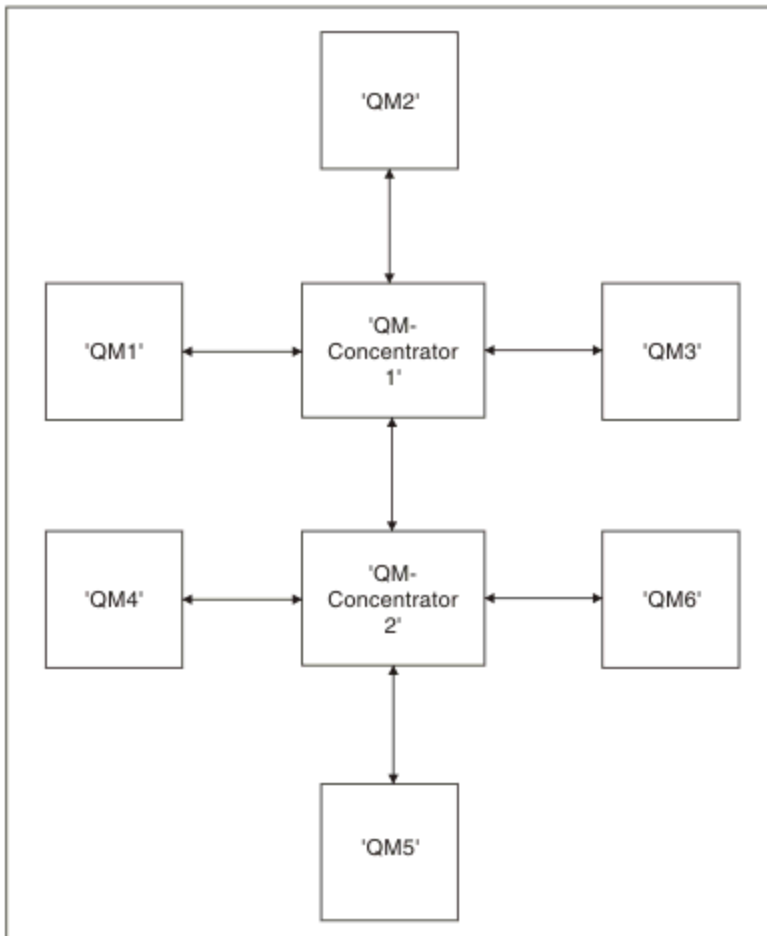


그림 2. QM 집선기가 표시된 네트워크 다이어그램

클러스터링

이 주제는 IBM WebSphere MQ 클러스터 계획 및 관리를 위한 지침을 제공합니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

다음 정보는 사용자가 IBM WebSphere MQ 클러스터에 대한 기본 지식을 가지고 있다고 가정한 것입니다. 이 정보는 "모든 경우에 사용할 수 있는(one size fits all)" 솔루션으로 제공되는 것이 아니라 대신 공통 문제점에 대한 공통 접근 방식을 공유하기 위한 것입니다.

클러스터는 시스템을 설정하는 데 필요한 초기 구성과 필요한 지속적 관리를 단순화하는 방식으로 큐 관리자를 상호 연결하기 위한 메커니즘을 제공합니다. 구성의 규모가 클수록 이점이 커집니다.

올바로 동작하는지 확인하고 특히 크거나 복잡한 클러스터형 시스템의 경우 시스템에 필요한 가용성 및 응답성 레벨을 확인하려면 클러스터형 시스템 계획 시 주의해야 합니다.

성공적인 클러스터 설정은 양질의 애플리케이션 관리 및 네트워크 설계와 같은 IBM WebSphere MQ 기본사항에 대한 완벽한 이해와 우수한 계획에 의존합니다. [상호통신의 개념 및 클러스터 작동 방법](#)에 있는 정보를 숙지했는지 확인하십시오.

클러스터의 개념 및 클러스터가 사용되는 이유

클러스터링은 다음과 같은 두 가지 주요 이점을 제공합니다.

- 클러스터는 채널, 전송 큐 및 구성할 리모트 큐에 대한 많은 오브젝트 정의가 필요한 IBM WebSphere MQ 네트워크 관리를 단순화합니다. 이 상황은 특히 많은 큐 관리자를 상호 연결해야 하는, 잠재적으로 계속 변화하는 대규모 네트워크에서 적용됩니다. 이 아키텍처는 특히 구성 및 활동적인 유지보수가 어렵습니다.
- 클러스터는 큐 및 클러스터의 큐 관리자에서 메시지 트래픽의 워크로드를 분산하는 데 사용할 수 있습니다. 이러한 분산을 통해 단일 큐의 메시지 워크로드를 여러 큐 관리자에 있는 해당 큐의 동등한 인스턴스에 분산할 수 있습니다. 이러한 워크로드 분산을 사용하여 시스템에서 특히 활성 메시지 플로우의 확장 성능을 향상시키고 시스템 장애에 대해 더 많은 회복 기능을 달성할 수 있습니다. 이러한 환경에서 분산된 큐의 각 인스턴스는 메시지를 처리하는 응용 애플리케이션을 보유합니다.

관련 정보

[클러스터링: 우수 사례](#)

포인트-투-포인트 메시징

IBM WebSphere MQ의 가장 단순한 메시징 형식은 포인트-투-포인트 메시징입니다.

포인트-투-포인트 메시징에서 송신 애플리케이션이 수신 애플리케이션에 대한 정보를 알아야 수신 애플리케이션으로 메시지를 송신할 수 있습니다. 예를 들어, 송신 애플리케이션은 정보를 송신할 큐 이름을 알아야 하며 큐 관리자 이름도 지정할 수 있습니다.

IBM WebSphere MQ에서 사용할 수 있는 대체 메시징 스타일은 발행/구독 메시징입니다. 발행/구독 메시징을 사용하면 해당 정보의 이용자로부터 정보의 제공자를 분리시킬 수 있습니다. 정보가 송수신되기 위해 송신 애플리케이션과 수신 애플리케이션이 서로에 대해 어느 것도 알 필요가 없습니다. 발행/구독 메시징에 대한 자세한 정보는 [WebSphere MQ 발행/구독 메시징 소개](#)를 참조하십시오.

관련 정보

[애플리케이션 개발](#)

[WebSphere MQ 메시지](#)

IBM WebSphere MQ 발행/구독 메시징 소개

발행/구독 메시징을 사용하면 해당 정보의 이용자로부터 정보의 제공자를 분리시킬 수 있습니다. 정보가 송수신되기 위해 송신 애플리케이션과 수신 애플리케이션이 서로에 대해 어느 것도 알 필요가 없습니다.

포인트-투-포인트 IBM WebSphere MQ 애플리케이션은 메시지를 다른 애플리케이션으로 송신하기 전에 해당 애플리케이션에 대한 일부 정보를 알아야 합니다. 예를 들어 정보를 송신할 큐의 이름을 알아야 하며 큐 관리자 이름을 지정할 수도 있습니다.

IBM WebSphere MQ 발행/구독에서는 애플리케이션이 대상 애플리케이션을 알지 않아도 됩니다. 송신 애플리케이션이 수행해야 하는 모든 작업은 원하는 정보를 포함하는 IBM WebSphere MQ 메시지를 넣고, 정보의 주제에 대해 설명하는 토픽에 지정한 후 IBM WebSphere MQ가 해당 정보의 분배를 핸들링하도록 하는 것입니다. 유사하게 대상 애플리케이션은 수신한 정보의 소스에 대해 알 필요가 없습니다.

19 페이지의 [그림 3](#)에서는 가장 간단한 발행/구독 시스템을 보여줍니다. 하나의 발행자, 하나의 큐 관리자 및 하나의 구독자가 있습니다. 구독이 구독자로부터 큐 관리자로 송신되고, 발행이 발행자로부터 큐 관리자로 송신된 후 발행이 큐 관리자에 의해 구독자에게 전달됩니다.

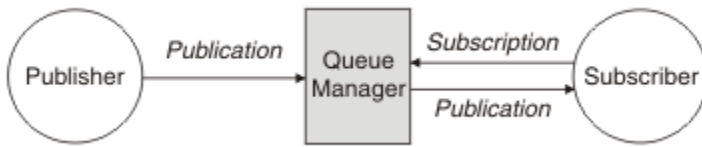


그림 3. 간단한 발행/구독 구성

일반적인 발행/구독 시스템에는 둘 이상의 발행자 및 둘 이상의 구독자가 있고 종종 둘 이상의 큐 관리자도 포함됩니다. 애플리케이션은 발행자 및 구독자 모두가 될 수 있습니다.

발행/구독 컴포넌트의 개요

발행/구독은 구독자가 발행자로부터 메시지 양식으로 정보를 수신할 수 있는 메커니즘입니다. 발행자와 구독자 간 상호작용은 표준 WebSphere MQ 기능을 사용하여 큐 관리자에 의해 제어됩니다.

일반적인 발행/구독 시스템에는 둘 이상의 발행자 및 둘 이상의 구독자가 있고 종종 둘 이상의 큐 관리자도 포함됩니다. 애플리케이션은 발행자 및 구독자 모두가 될 수 있습니다.

정보 제공자는 발행자라고 합니다. 발행자는 해당 정보에 관심이 있는 애플리케이션에 대해 알지 않고도 주제에 대한 정보를 제공합니다. 발행자는 이 메시지의 토픽을 발행하고 정의하려는 발행이라고 하는 메시지 양식으로 이 정보를 생성합니다.

정보의 이용자는 구독자라고 합니다. 구독자는 구독자가 관심이 있는 토픽을 설명하는 구독을 작성합니다. 따라서 구독은 구독자에게 전달되는 발행을 판별합니다. 구독자는 여러 구독을 작성하고 많은 다양한 발행자로부터 정보를 수신할 수 있습니다.

발행된 정보는 WebSphere MQ 메시지로 송신되며 정보의 주제는 해당 토픽으로 식별됩니다. 발행자는 정보를 발행할 때 토픽을 지정하고 구독자는 발행을 수신하려는 토픽을 지정합니다. 구독자는 구독하는 토픽에 대한 정보만 받습니다.

토픽을 통해 포인트-투-포인트 메시징에 필요한 경우 각 메시지에서 특정 목적지를 포함해야 하는 필요를 없애 정보의 제공자와 이용자를 발행/구독 메시징에서 분리할 수 있습니다.

발행자와 구독자 사이의 상호작용은 큐 관리자에 의해 모두 제어됩니다. 큐 관리자는 발행자로부터 메시지를 수신하고 토픽 범위에 따라 구독자로부터 구독을 수신합니다. 큐 관리자의 작업은 메시지 토픽에서 관심사를 등록한 구독자에게 발행된 메시지를 라우팅하는 것입니다.

표준 WebSphere MQ 기능을 사용하여 메시지를 분배할 수 있으므로, 사용자 애플리케이션이 기존 WebSphere MQ 애플리케이션에서 사용 가능한 모든 기능을 사용할 수 있습니다. 즉, 지속 메시지를 사용하여 한 번만 보장된 전달을 받을 수 있으며, 발행자가 커밋한 경우에만 구독자에게 메시지를 전달하도록 메시지를 트랜잭션 작업 단위로 포함할 수 있음을 의미합니다.

단일 큐 관리자 발행/구독 구성 예제

20 페이지의 그림 4에서는 기본적인 단일 큐 관리자 발행/구독 구성을 보여줍니다. 예제에서는 뉴스 서비스에 대한 구성을 보여줍니다. 여기에서 발행자로부터 여러 토픽에 대한 정보를 사용할 수 있습니다.

- 발행자 1은 Sport 토픽을 사용하여 스포츠 결과에 대한 정보를 발행함
- 발행자 2는 Stock 토픽을 사용하여 주가에 대한 정보를 발행함
- 발행자 3은 Films 토픽을 사용하여 영화 비평에 대한 정보를 발행하고 TV 토픽을 사용하여 TV 프로그램에 대한 정보를 발행함

3명의 구독자는 서로 다른 토픽을 관심사로 등록했으므로 큐 관리자는 관심이 있는 정보를 해당 구독자에게 송신합니다.

- 구독자 1은 스포츠 결과와 주가를 수신함
- 구독자 2는 영화 비평을 수신함
- 구독자 3은 스포츠 결과를 수신함

TV 프로그램을 관심사로 등록한 구독자는 없으므로 이는 배포되지 않습니다.

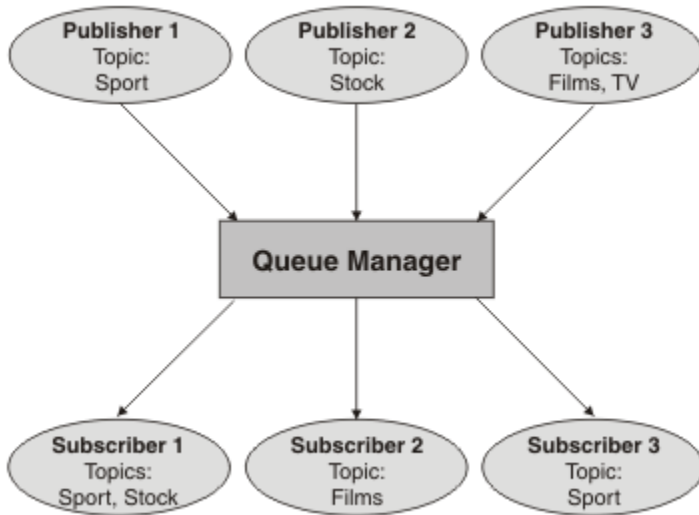


그림 4. 단일 큐 관리자 발행/구독 예제

발행자 및 발행물

WebSphere MQ 발행/구독에서 발행자는 발행물이라는 표준 WebSphere MQ 메시지 형식으로 지정된 토픽에 대한 정보를 큐 관리자가 사용 가능하게 하는 애플리케이션입니다. 발행자는 둘 이상의 토픽에 대한 정보를 발행할 수 있습니다.

발행자는 MQPUT 동사를 사용하여 이전에 열린 토픽에 메시지를 넣으며, 이 메시지는 발행물입니다. 그리고 로컬 큐 관리자는 발행물의 토픽을 구독하는 구독자에게 해당 발행물을 라우팅합니다. 발행된 메시지를 둘 이상의 구독자가 이용할 수 있습니다.

해당 구독을 보유하는 모든 로컬 구독자에게 발행물을 배포함은 물론, 큐 관리자는 이에 연결된 기타 큐 관리자에게 직접적으로 또는 토픽에 대한 구독자가 있는 큐 관리자의 네트워크를 통해 발행물을 배포할 수도 있습니다.

WebSphere MQ 발행/구독 네트워크에서 발행 애플리케이션이 구독자가 될 수도 있습니다.

동기점 하의 발행물

발행자는 작업 단위로 구독자에게 전달된 모든 메시지를 포함하도록 동기점에서 MQPUT 또는 MQPUT1 호출을 실행할 수 있습니다. MQPMO_RETAIN 옵션 또는 토픽 전달 옵션 NPMSGDLV 또는 PMSGDLV(ALL 또는 ALLDUR 값의)가 지정되면, 큐 관리자는 발행자 MQPUT 또는 MQPUT1 호출 범위 내의 동기점에서 내부 MQPUT 또는 MQPUT1 호출을 사용합니다.

상태 및 이벤트 정보

발행은 주식의 현재 가격과 같은 상태 발행이나, 해당 주식의 거래와 같은 이벤트 발행으로 분류할 수 있습니다.

상태 발행

상태 발행은 축구 경기의 현재 점수 또는 주가와 같이 어떤 항목의 현재 상태에 대한 정보를 포함합니다. 주가가 변동되거나 축구 경기의 점수가 변경되는 것과 같이 무슨 일이 발생하면 이전 상태 정보가 새 정보로 대체되므로 더 이상 필요하지 않게 됩니다.

구독자는 시작할 때 상태 정보의 최신 버전을 수신하려고 하며, 상태가 변경되면 새 정보가 전송되기를 원합니다.

발행에 상태 정보가 포함되는 경우, 흔히 보유된 발행으로 발행됩니다. 일반적으로 새 구독자는 즉시 현재 상태 정보를 원하며, 구독자는 정보가 다시 발행되는 이벤트를 기다리기를 원하지 않습니다. 구독자가 MQSO_PUBLICATIONS_ON_REQUEST 또는 MQSO_NEW_PUBLICATIONS_ONLY 옵션을 사용하지 않는 한, 구독할 때 구독자는 자동으로 토픽의 보유된 발행을 수신합니다.

이벤트 발행

이벤트 발행은 일부 주식의 거래나 특정 골의 득점과 같이 발생하는 개별 이벤트에 대한 정보를 포함합니다. 각 이벤트는 다른 이벤트와 서로 독립적입니다.

구독자는 이벤트가 발생하면 해당 이벤트에 대한 정보를 수신하려고 합니다.

보유된 발행물

기본적으로 발행은 관련된 모든 구독자에게 전송된 후에 제거됩니다. 그러나 발행자는 발행 사본을 보유하도록 지정할 수 있습니다. 그러면 토픽에 관심사를 등록한 향후 구독자에게 이를 전송할 수 있습니다.

이벤트 정보의 경우 관련된 모든 구독자에게 전송된 후에 발행을 삭제하는 것이 적절하지만, 상태 정보에서는 항상 적절하지 않을 수도 있습니다. 메시지를 보유하면 새 구독자가 초기 상태 정보를 수신하기 전에 정보 발행을 다시 기다리지 않아도 됩니다. 예를 들어 주가에 대해 구독하는 구독자는 주가가 변경되고 이후에 다시 발행되기를 기다리지 않고도 현재 가격을 직접 수신할 수 있습니다.

큐 관리자는 각 토픽에 대해 하나의 발행만 보유할 수 있으므로 새 보유된 발행이 큐 관리자에 도달하면 토픽의 기존 보유된 발행은 삭제됩니다. 그러나 기존 발행의 삭제가 새 보유된 발행의 도착과 동시에 수행되지 않을 수도 있습니다. 따라서 토픽에서 보유된 발행을 송신하는 발행자가 하나만 존재할 수 있습니다.

구독자는 MQSO_NEW_PUBLICATIONS_ONLY 구독 옵션을 사용하여 보유된 발행을 수신하지 않도록 지정할 수 있습니다. 기존 구독자는 보유된 발행의 중복 사본을 전송해줄 것을 요청할 수 있습니다.

상태 정보라도 발행을 보유하지 않으려는 경우가 있습니다.

- 토픽에 대한 모든 구독이 해당 토픽에 대한 발행을 수행하기 전에 이루어진 경우 새 구독을 예상하지 않았거나 이를 허용하지 않으면 처음 발행될 때 전체 구독자 세트에 전달되므로 발행을 보유하지 않아도 됩니다.
- 발행이 자주(가령 매초마다) 나타나면 새 구독자 또는 장애에서 복구한 구독자는 초기 구독 이후 거의 즉시 현재 상태를 수신하므로 이러한 발행을 보유하지 않아도 됩니다.
- 발행이 큰 경우 각 토픽에 대한 보유된 발행을 저장하는 데 상당한 스토리지 공간이 필요하게 될 수도 있습니다. 다중 큐 관리자 환경에서 보유된 발행은 일치하는 구독이 있는 네트워크의 모든 관리자에서 저장합니다.

보유된 발행의 사용 여부를 결정할 때 구독 애플리케이션이 실패에서 복구하는 방법을 고려하십시오. 발행자가 보유된 발행을 사용하지 않을 경우, 구독자 애플리케이션은 현재 상태를 로컬로 저장해야 할 수도 있습니다.

발행을 보유하려면 MQPMO_RETAIN 메시지 넣기 옵션을 사용하십시오. 이 옵션이 사용되고 발행물을 보유할 수 없는 경우에는 메시지가 발행되지 않고 호출은 MQRC_PUT_NOT_RETAINED가 발생하여 실패합니다.

메시지가 보유된 발행이면 MQIsRetained 메시지 특성으로 표시됩니다. 메시지의 지속성은 원래 발행될 때 속성 그대로입니다.

동기점 하의 발행물

IBM WebSphere MQ 발행/구독에서 동기점은 발행자가 사용하거나 큐 관리자가 내부적으로 사용할 수 있습니다.

발행자는 MQPMO_SYNCPOINT 옵션을 사용하여 MQPUT/MQPUT1 호출을 발행할 때 동기점을 사용합니다. 구독자에 전달된 모든 메시지는 작업 단위에서 커밋되지 않은 최대 메시지 수로 계산됩니다. MAXUMSGS 큐 관리자 속성이 이 한계를 지정합니다. 한계에 도달하면 발행자가 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_REACHED 이유 코드를 수신합니다.

발행자가 MQPMO_RETAIN 옵션 또는 값이 ALL이나 ALLDUR인 토픽 전달 옵션 NPMSGDLV/PMSGDLV와 함께 MQPMO_NO_SYNCPOINT를 사용하여 MQPUT/MQPUT1 호출을 발행하는 경우 큐 관리자는 메시지가 요청된 대로 전달될 수 있도록 내부 동기점을 사용합니다. 한계가 발행자 MQPUT/MQPUT1 호출의 범위에 도달하면 발행자가 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_REACHED 이유 코드를 수신할 수 있습니다.

구독자 및 구독

WebSphere MQ 발행/구독에서, 구독자는 발행/구독 네트워크의 큐 관리자에서 특정 토픽에 대한 정보를 요청하는 애플리케이션입니다. 구독자는 둘 이상의 발행자로부터 동일하거나 다른 토픽에 대한 메시지를 수신할 수 있습니다.

구독은 애플리케이션 또는 MQSC 명령을 사용하여 수동으로 작성할 수 있습니다. 이러한 구독은 로컬 큐 관리자로 발행되고, 구독자가 수신하려는 발행에 대한 정보를 포함합니다.

- 구독자가 관심이 있는 토픽. 이는 와일드카드를 사용하면 여러 토픽으로 해석될 수 있습니다.

- 발행된 메시지에 적용할 선택적 선택 문자열.
- 선택한 발행물을 배치해야 하는 큐(구독자 큐라고도 함)에 대한 핸들 및 선택적 CorrelId입니다.

로컬 큐 관리자는 구독 정보를 저장하고 발행을 수신하면 구독의 토픽 및 선택 문자열과 일치하는 구독이 있는지 여부를 판별하도록 정보를 스캔합니다. 일치하는 각 구독에 대해 큐 관리자는 구독자의 구독자 큐로 발행을 지정합니다. 큐 관리자가 구독에 대해 저장하는 정보는 DIS SUB 및 DIS SBSTATUS 명령을 사용하여 볼 수 있습니다.

다음과 같은 이벤트 중 하나가 발생하는 경우에만 구독이 삭제됩니다.

- 구독자는 MQCLOSE 호출을 사용하여 구독을 해제합니다(구독이 비지속으로 작성된 경우).
- 구독이 만료됩니다.
- 구독은 시스템 관리자가 DELETE SUB 명령을 사용하여 삭제합니다.
- 구독자 애플리케이션이 종료됩니다(구독이 비지속으로 작성된 경우).
- 큐 관리자가 중지되거나 재시작됩니다(구독이 비지속으로 작성된 경우).

메시지를 가져올 때 MQGET 호출에서 적절한 옵션을 사용하십시오. 애플리케이션이 한 구독의 메시지만 처리하는 경우에는 최소한 get-by-correlid를 C 샘플 프로그램 amqssbxa.c 및 비관리 MQ 구독자에 시연되어 있는 바와 같이 사용해야 합니다. 사용할 **CorrelId** 는 MQSD의 MQSUB에서 리턴됩니다.**SubCorrelId** 필드.

관리 큐 및 발행/구독

구독을 작성하면 관리 큐잉을 사용하도록 선택할 수 있습니다. 관리 대상 큐잉을 사용하는 경우, 구독을 작성할 때 구독 큐가 자동으로 작성됩니다. 관리 큐는 구독의 지속성에 따라 자동으로 정리됩니다. 관리 큐의 사용은 발행을 수신하는 큐를 작성하지 않아도 되며, 지속 불가능 구독 연결이 닫히면 이용하지 않은 발행은 구독자 큐에서 자동으로 제거됨을 의미합니다.

애플리케이션이 특정 큐를 구독자 큐, 즉 수신하는 발행물의 대상으로 사용하지 않아도 되는 경우, MQSO_MANAGED 구독 옵션을 사용하여 관리 대상 구독을 이용할 수 있습니다. 관리 구독을 작성하면 큐 관리자는 발행이 수신되는 큐 관리자가 작성한 구독자 큐의 구독자에게 오브젝트 핸들을 리턴합니다. 큐 오브젝트 핸들을 리턴하여 큐를 찾아보거나 가져오거나 조회할 수 있습니다(임시 동적 큐에 대한 액세스 권한을 명시적으로 부여받지 않는 한, 관리 큐의 속성을 설정하거나 여기에 배치할 수 없음).

구독의 지속성은 구독 애플리케이션에서 큐 관리자로의 연결이 끊어진 경우 관리 큐가 남아 있는지 여부를 판별합니다.

관리 구독은 애플리케이션 연결이 종료될 때 (그렇지 않은 경우) 이용되지 않은 메시지가 구독 큐에 계속 남아 있어서 큐 관리자에서 무한으로 공간을 차지하므로 지속 불가능 구독과 함께 사용할 때 특히 유용합니다. 관리 구독을 사용하는 경우 관리 큐는 임시 동적 큐이며, 다음과 같은 이유로 연결이 끊어질 때 이용되지 않은 메시지와 함께 삭제됩니다.

- MQCO_REMOVE_SUB를 포함하는 MQCLOSE가 사용되고 관리 Hobj가 닫힙니다.
- 지속 불가능 구독(MQSO_NON_DURABLE)을 사용하는 애플리케이션과의 연결이 끊어집니다.
- 구독이 만료되었고 관리 Hobj가 닫혔으므로 구독이 제거됩니다.

관리 구독을 지속 가능 구독과 함께 사용할 수도 있지만 연결을 다시 열 때 검색할 수 있도록 이용되지 않은 메시지를 구독자 큐에 남겨둘 수 있습니다. 이러한 이유로 지속 가능 구독에 대한 관리 큐는 영구적 동적 큐 양식을 취하고 구독 애플리케이션과 큐 관리자의 연결이 끊어진 경우에도 그대로 남아 있습니다.

영구 동적 관리 대상 큐를 사용하려는 경우, 연결이 끊긴 후에도 큐가 아직 있지만 계속 남아 있지는 않도록 구독에 만기를 설정할 수 있습니다.

관리 큐를 삭제하면 오류 메시지가 수신됩니다.

작성된 관리 큐의 이름은 각각 고유하도록 끝에 숫자(시간 소인)가 추가됩니다.

구독 지속성

구독은 지속 가능 또는 지속 불가능으로 구성될 수 있습니다. 구독 지속성은 구독 애플리케이션과 큐 관리자의 연결이 끊어질 때 구독에서 나타나는 상황을 판별합니다.

지속 가능 구독

지속 가능 구독의 경우 구독 애플리케이션과 큐 관리자의 연결이 단힌 경우에도 계속 존재합니다. 구독이 지속적 인 경우 구독 애플리케이션의 연결이 끊기면, 구독이 그대로 남게 되고, 구독 작성 시 리턴된 SubName으로 다시 구독을 요청하여 재연결할 때 구독 애플리케이션에서 이 구독을 사용할 수 있습니다.

지속적으로 구독하는 경우, 구독 이름(SubName)이 필요합니다. 구독을 식별하는 데 사용할 수 있도록 구독 이름 은 큐 관리자에서 고유해야 합니다. 구독 식별은 구독에 대한 핸들을 의도적으로 닫거나(MQCO_KEEP_SUB 옵션 을 사용하여) 큐 관리자에서 연결을 끊은 경우 재개할 구독을 지정할 때 필요합니다. MQSO_RESUME 옵션과 함 께 MQSUB 호출을 사용하여 기존 구독을 계속할 수 있습니다. SUBTYPE ALL 또는 ADMIN과 함께 DISPLAY SBSTATUS 명령을 사용하는 경우에도 구독 이름이 표시됩니다.

애플리케이션에 지속 가능 구독이 더 이상 필요하지 않으면 MQCO_REMOVE_SUB 옵션과 함께 MQCLOSE 함수 를 사용하여 제거할 수 있거나 수동으로 MQSC 명령 DELETE SUB를 사용하여 삭제할 수 있습니다.

토픽에 대한 지속적 구독을 작성할 수 있는지의 여부는 DURSUB 토픽 속성을 사용하여 제어할 수 있습니다.

MQSO_RESUME 옵션을 사용하여 MQSUB 호출에서 리턴할 때 구독 만료는 남은 만료 시간이 아닌, 원래 구독의 만료로 설정됩니다.

큐 관리자는 계속해서 발행을 송신하여 구독자 애플리케이션이 연결되지 않아도 지속 가능 구독을 만족시킬 수 있습니다. 이로 인해 구독자 큐에 메시지가 누적될 수 있습니다. 이 문제점을 피하는 가장 쉬운 방법은 해당되는 경우에만 지속 불가능 구독을 사용하는 것입니다. 그러나 지속 가능 구독을 사용해야 하는 경우 구독자가 보유된 발행 옵션을 사용하여 구독하면 메시지 누적을 방지할 수 있습니다. 그러면 구독자는 MQSUBRQ 호출을 사용하 여 발행을 수신하는 시기를 제어할 수 있습니다.

지속 불가능 구독

지속 불가능 구독은 구독 애플리케이션과 큐 관리자의 연결이 열려 있는 경우에만 존재합니다. 이 구독은 고의적 으로 또는 연결 유실에 의해 구독 애플리케이션과 큐 관리자의 연결이 끊길 때 제거됩니다. 연결이 닫히면 구독에 대한 정보가 큐 관리자에서 제거되고 DISPLAY SBSTATUS 명령을 사용하여 구독을 표시하는 경우 더 이상 표시 되지 않습니다. 더 이상 구독자 큐에 메시지를 넣지 않습니다.

지속 불가능 구독에 대한 구독자 큐에서 이용되지 않은 발행의 처리 방법은 다음과 같이 판별됩니다.

- 구독 애플리케이션이 관리 목적지를 사용하는 경우 이용되지 않은 발행물이 자동으로 제거됩니다.
- 구독 애플리케이션이 구독 시 고유한 구독자 큐로 핸들을 제공하는 경우 이용되지 않은 메시지는 자동으로 제 거되지 않습니다. 적절한 경우 큐를 지우는 것은 애플리케이션의 몫입니다. 큐가 둘 이상의 구독자 또는 다른 포인트-투-포인트 애플리케이션에서 공유되는 경우 큐를 완전히 지우는 것은 적절하지 않을 수도 있습니다.

비지속 구독에 필요하지 않지만 구독 이름이 제공되는 경우 큐 관리자에서 사용됩니다. 구독을 식별하는 데 사용 할 수 있도록 구독 이름은 큐 관리자에서 고유해야 합니다.

선택 문자열

선택 문자열은 구독과 일치하는지 여부를 판별하기 위해 발행에 적용되는 표현식입니다. 선택 문자열은 와일드 카드 문자를 포함할 수 있습니다.

구독하는 경우 토픽 지정 외에도 선택 문자열을 지정하여 메시지 특성에 따라 발행을 선택할 수 있습니다.

토픽

토픽은 발행/구독 메시지에서 발행된 정보의 제목입니다.

포인트-투-포인트 시스템의 메시지는 특정 목적지 주소로 송신됩니다. 주제 기반 발행/구독 시스템의 메시지는 메시지의 콘텐츠에 대해 설명하는 주제에 따른 구독자로 송신됩니다. 콘텐츠 기반 시스템에서, 메시지는 메시지 의 콘텐츠를 기반으로 한 구독자로 송신됩니다.

IBM WebSphere MQ 발행/구독 시스템은 주제 기반 발행/구독 시스템입니다. 발행자는 메시지를 작성하고 발행 주제에 맞는 토픽 문자열을 사용하여 해당 메시지를 발행합니다. 구독자는 발행을 수신하기 위해 발행 토픽을 선택하도록 토픽 문자열과 일치하는 패턴의 구독을 작성합니다. 큐 관리자는 발행 토픽과 일치하는 구독을 가진 구 독자에게 발행을 전달하며 이 큐 관리자에게 발행을 수신할 수 있는 권한이 부여됩니다. 24 페이지의 『토픽 문 자열』 글에서는 발행 주제를 식별하는 토픽 문자열의 구분에 대해 설명합니다. 또한 구독자는 수신할 토픽을 선택하도록 토픽 문자열을 작성합니다. 구독자가 작성하는 토픽 문자열에는 발행에 있는 토픽 문자열의 패턴 일치

에 대한 두 가지 대체 와일드카드 설계 중 하나가 있습니다. 패턴 일치는 [25 페이지의 『와일드카드 설계』](#)에 설명되어 있습니다.

주제 기반 발행/구독에서는, 발행자 또는 관리자가 주제를 토픽으로 분류해야 합니다. 일반적으로 주제는 '/' 문자로 토픽 문자열에서 하위 토픽을 작성하여 계층적으로 토픽 트리로 구성됩니다. 토픽 트리 예는 [30 페이지의 『토픽 트리』](#)의 내용을 참조하십시오. 토픽은 토픽 트리의 노드입니다. 토픽은 추가 하위 토픽이 없는 리프 노드 또는 하위 토픽이 있는 중간 노드입니다.

주제를 계층적 토픽 트리로 구성하면서, 토픽을 관리 토픽 오브젝트와 연관시킬 수 있습니다. 토픽을 관리 토픽 오브젝트와 연관시켜 토픽을 클러스터에 분배할지의 여부와 같은 속성을 토픽에 지정합니다. 관리 토픽 오브젝트의 TOPICSTR 속성을 사용하여 토픽의 이름을 지정하면 연관이 이루어집니다. 명시적으로 관리 토픽 오브젝트를 토픽에 연관시키지 않으면, 토픽이 관리 토픽 오브젝트와 연관시킨 토픽 트리의 가장 가까운 상위의 속성을 상속합니다. 상위 주제를 전혀 정의하지 않은 경우 SYSTEM.BASE.TOPIC을 참조하십시오. 관리 토픽 오브젝트는 [33 페이지의 『관리 토픽 오브젝트』](#)에 설명되어 있습니다.

참고: SYSTEM.BASE.TOPIC에서 SYSTEM.BASE.TOPIC을 참조하십시오. 예를 들어, 토픽 공간인 미국의 주 USA/Alabama, USA/Alaska 등에서는 USA가 루트 토픽입니다. 루트 토픽은 주로 잘못된 구독과 발행이 일치하는 것을 피하기 위해 겹치지 않는 개별 토픽 공간을 작성하는 데 사용됩니다. 또한 이 루트 토픽을 사용하면 전체 토픽 공간에 영향을 미치도록 루트 토픽의 속성을 변경할 수 있습니다. 예를 들어, **CLUSTER** 속성에 이름을 설정할 수 있습니다.

발행자 또는 구독자로 토픽을 참조하는 경우, 토픽 문자열을 제공하거나 토픽 오브젝트를 참조할 수 있습니다. 또한 제공하는 토픽 문자열에서 토픽 오브젝트의 하위 토픽을 정의하는 경우에는 둘 다 수행할 수 있습니다. 큐 관리자는 토픽 오브젝트에서 이름 지정된 토픽 문자열 접두부에 토픽 문자열을 추가하고 두 토픽 문자열 사이에 '/'를 삽입하여 토픽을 식별합니다(예: *topic string/object string*). [29 페이지의 『토픽 문자열 결합』](#)에서는 이에 대해 자세히 설명합니다. 이렇게 생성된 토픽 문자열은 토픽을 식별하고 이 토픽을 관리 토픽 오브젝트와 연관시키는 데 사용됩니다. 관리 토픽 오브젝트는 마스터 토픽에 해당하는 토픽 오브젝트와 다를 수도 있습니다.

컨텐츠 기반 발행/구독에서는, 모든 메시지의 콘텐츠를 검색하는 선택 문자열을 제공하여 수신할 메시지 내용을 정의합니다. WebSphere MQ에서는 메시지의 전체 콘텐츠와 다른 메시지 특성을 스캔하는 메시지 선택자를 사용하여 컨텐츠 기반 발행/구독의 중간 형식을 제공합니다. 선택자를 참조하십시오. 메시지 선택자는 전형적으로 토픽을 구독한 다음 숫자 특성에 대한 선택 사항을 규정하는 데 사용됩니다. 선택자를 사용하면 특정 범위의 값에만(문자 또는 토픽 기반 와일드카드를 사용하여 수행할 수 없는 내용) 관심이 있음을 지정할 수 있습니다. 메시지의 전체 콘텐츠를 기반으로 하여 필터링해야 하는 경우, WebSphere Message Broker를 사용해야 합니다.

토픽 문자열

토픽 문자열을 사용하여 토픽으로 발행하는 레이블 정보입니다. 문자 또는 토픽 기반 와일드카드 토픽 문자열을 사용하여 토픽 그룹을 구독합니다.

토픽

토픽 문자열은 발행/구독 메시지의 토픽을 식별하는 문자열입니다. 토픽 문자열을 구성할 때 원하는 모든 문자를 사용할 수 있습니다.



3자는 버전 7 발행/구독에서 특별한 의미를 지닙니다. 이는 토픽 문자열에서 허용되지만, 주의해서 사용해야 합니다. 특수 문자의 사용은 [25 페이지의 『토픽 기반 와일드카드 설계』](#)에서 설명됩니다.

슬래시(/)

토픽 레벨 구분 기호입니다. '/' 문자를 사용하여 토픽 트리로 토픽을 구성합니다.

가능한 경우 빈 토픽 레벨('//')은 피하십시오. 이는 토픽 문자열이 없는 토픽 계층의 노드에 대응합니다. 토픽 문자열에서 선두 또는 후미 문자 '/'는 선두 또는 후미의 빈 노드에 대응하므로 이러한 사용도 피해야 합니다.

해시 부호(#)

'/'와 함께 사용하여 구독에서 다중 레벨 와일드카드를 구성합니다. 발행된 토픽의 이름을 지정하는 데 사용되는 토픽 문자열에서 '/' 가까이에서 '#'를 사용할 때 주의하십시오. [25 페이지의 『토픽 문자열 예제』](#) '#'의 사용 가능한 사용을 표시합니다.

'.../#/...', '#/...', '.../#' 문자열에는 구독 토픽 문자열에서 특별한 의미를 지닙니다. 문자열은 토픽 계층에서 하나 이상의 레벨에 있는 모든 토픽과 일치합니다. 따라서 이러한 순서 중 하나로 토픽을 작성한 경우 토픽 계층의 다중 레벨에서 모든 토픽을 구독하지 않는 한, 이를 구독할 수 없습니다.

더하기 부호(+)

'/'와 함께 사용하여 구독에서 단일 레벨 와일드카드를 구성합니다. 발행된 토픽의 이름을 지정하는 데 사용되는 토픽 문자열에서 '/' 가까이에서 '+'를 사용할 때 주의하십시오.

'.../+/...', '+/...', '.../+' 문자열에는 구독 토픽 문자열에서 특별한 의미를 지닙니다. 문자열은 토픽 계층에서 한 레벨에 있는 모든 토픽과 일치합니다. 따라서 이러한 순서 중 하나로 토픽을 작성한 경우 토픽 계층의 한 레벨에서 모든 토픽을 구독하지 않는 한, 이를 구독할 수 없습니다.

토픽 문자열 예제

```
IBM/Business Area#/Results
IBM/Diversity/%African American
```

와일드카드 설계

여러 토픽을 구독하는 데 사용하는 2개의 와일드카드 설계가 있습니다. 설계 선택은 구독 옵션입니다.

MQSO_WILDCARD_TOPIC

토픽 기반 와일드카드 설계를 사용하여 구독할 토픽을 선택합니다.

이는 와일드카드 스키마를 명확하게 선택하지 않은 경우 기본값입니다.

MQSO_WILDCARD_CHAR

문자 기반 와일드카드 설계를 사용하여 구독할 토픽을 선택합니다.

DEFINE SUB 명령에서 **wschema** 매개변수를 지정하여 설계를 설정합니다. 자세한 정보는 [DEFINE SUB](#) 를 참조하십시오.

참고: WebSphere MQ 버전 7.0이 문자 기반 와일드카드 설계를 사용하기 전에 작성된 구독입니다.

예:

```
IBM/+/Results
#/Results
IBM/Software/Results
IBM/*ware/Results
```

토픽 기반 와일드카드 설계

토픽 기반 와일드카드를 사용하면 구독자가 한 번에 둘 이상의 토픽을 구독할 수 있습니다.

토픽 기반 와일드카드는 WebSphere MQ 발행/구독에 있는 토픽 시스템의 강력한 기능입니다. 다중 레벨 와일드카드와 단일 레벨 와일드카드는 구독에 사용될 수 있으나, 메시지 발행자가 토픽에서 사용할 수는 없습니다.

토픽 기반 와일드카드 설계에서는 토픽 레벨로 그룹화된 발행을 선택할 수 있습니다. 해당 토픽 레벨에서 구독의 문자열이 발행의 문자열과 일치하는지 여부와 관련하여 토픽 계층의 각 레벨을 선택할 수 있습니다. 예를 들어 구독 IBM/+/Results는 모든 토픽을 선택합니다.

```
IBM/Software/Results
IBM/Services/Results
IBM/Hardware/Results
```

두 종류의 와일드카드가 있습니다.

다중 레벨 와일드카드

- 다중 레벨 와일드카드는 구독에서 사용됩니다. 발행에서 사용되면 리터럴로 처리됩니다.

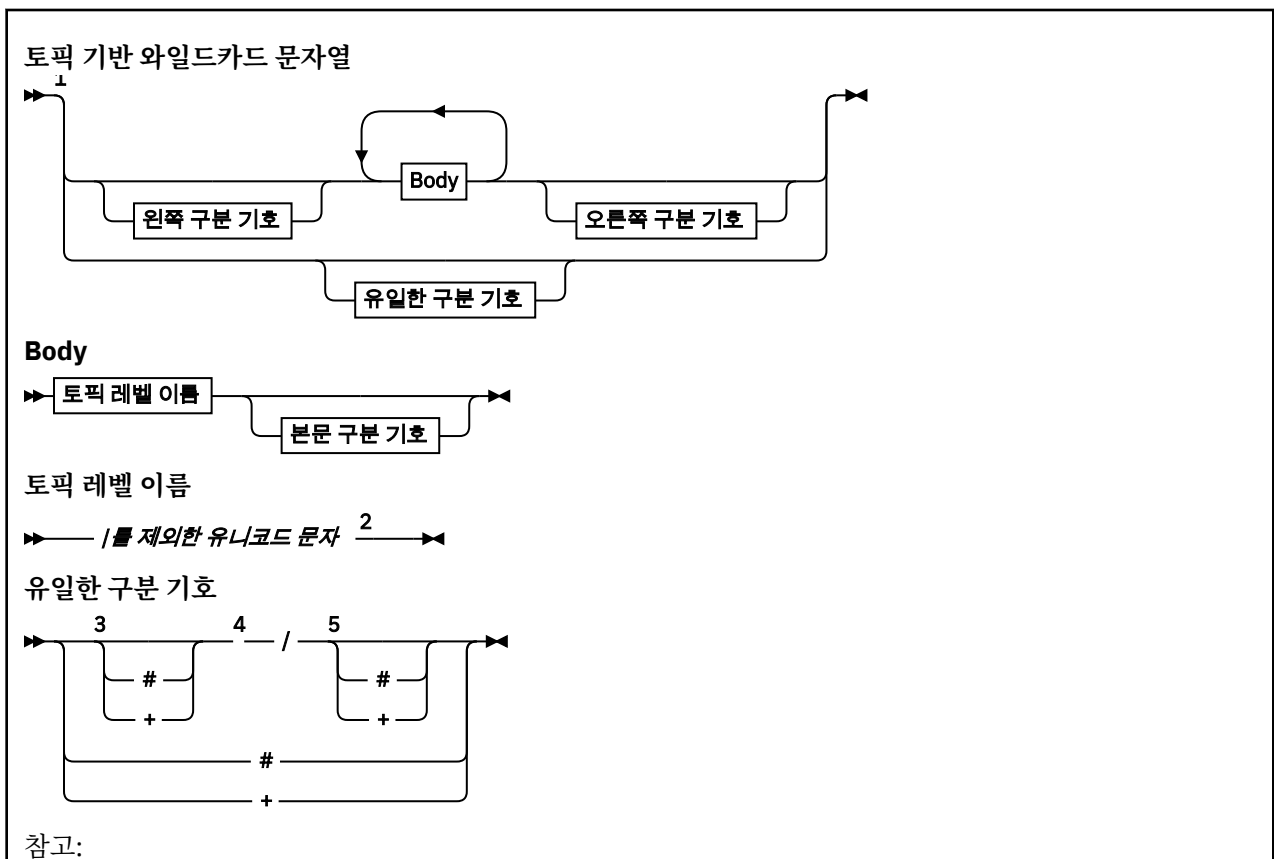
- 다중 레벨 와일드카드 문자 '#' 는 토픽 내 많은 레벨과 일치시키는 데 사용됩니다. 예를 들어 토픽 트리 예제를 사용하면 'USA/Alaska/#' 를 구독할 때 토픽 'USA/Alaska' 및 'USA/Alaska/Juneau' 에서 메시지를 수신합니다.
- 다중 레벨 와일드카드는 0개 이상의 레벨을 표시할 수 있습니다. 따라서 'USA/#' 는 단일 'USA' 와도 일치할 수 있습니다. 여기서 '#' 기호는 0 레벨을 나타냅니다. 토픽 레벨 구분 기호는 분리할 레벨이 없기 때문에 이 컨텍스트에서는 의미가 없습니다.
- 다중 레벨 와일드카드는 자체에 또는 토픽 레벨 분리 문자 옆에 지정된 경우에만 유효합니다. 따라서 '#' 및 'USA/#' 은 '#' 문자가 와일드카드 처리되는 유효한 토픽입니다. 'USA#' 도 유효한 토픽 문자열이지만 '#' 문자는 와일드카드로 간주되지 않으며, 특별한 의미를 지니지 않습니다. 자세한 정보는 27 페이지의 『토픽 기반 와일드카드가 와일드카드가 아닌 경우』의 내용을 참조하십시오.

단일 레벨 와일드카드

- 단일 와일드카드는 구독에서 사용됩니다. 발행에서 사용되면 리터럴로 처리됩니다.
- 단일 레벨 와일드카드 문자 '+' 는 하나의 유일한 토픽 레벨과 일치합니다. 예를 들어 'USA/+' 는 'USA/Alabama/Auburn' 이 아닌 'USA/Alabama' 와 일치합니다. 단일 레벨 와일드카드는 단일 레벨과만 일치하므로 'USA/+' 는 'USA' 와 일치하지 않습니다.
- 단일 레벨 와일드카드는 토픽 트리의 레벨에서 그리고 다중 레벨 와일드카드와 연결하여 사용될 수 있습니다. 단일 레벨 와일드카드는 자체에 지정된 경우를 제외하고 토픽 레벨 분리 문자 옆에 지정해야 합니다. 따라서 '+' 및 'USA/+' 은 '+' 문자가 와일드카드 처리되는 유효한 토픽입니다. 'USA+' 도 유효한 토픽 문자열이지만 '+' 문자는 와일드카드로 간주되지 않으며, 특별한 의미를 지니지 않습니다. 자세한 정보는 27 페이지의 『토픽 기반 와일드카드가 와일드카드가 아닌 경우』의 내용을 참조하십시오.

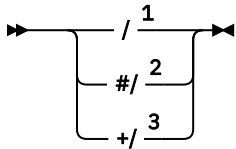
토픽 기반 와일드카드 설계의 구문에는 이스케이프 문자가 없습니다. '#' 및 '+' 가 와일드카드 처리되는지 여부는 컨텍스트에 따라 달라집니다. 자세한 정보는 27 페이지의 『토픽 기반 와일드카드가 와일드카드가 아닌 경우』의 내용을 참조하십시오.

참고: 토픽 문자열의 시작과 끝은 특별한 방식으로 처리됩니다. 문자열의 끝을 표시하기 위해 '\$' 를 사용하는 경우, '\$#/...' 는 다중 레벨 와일드카드 및 '\$#/..' 입니다. 루트의 빈 노드이고 다음에 다중 레벨 와일드카드가 나옵니다.

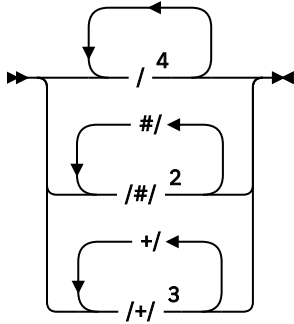


- 1 널 또는 길이가 0인 토픽 문자열은 유효하지 않음
- 2 문자 기반 및 주제 기반 와일드카드 설계 간의 호환성을 위해 레벨 이름 문자열에서 *, ?, % 을 사용하지 않는 것이 좋습니다.
- 3 이 경우는 왼쪽 구분 기호 패턴과 동일합니다.
- 4 와일드카드가 없는 /는 단일 빈 토픽과 일치합니다.
- 5 이 경우는 오른쪽 구분 기호 패턴과 동일합니다.
- 6 모든 토픽과 일치합니다.
- 7 1개 레벨만 존재하는 모든 토픽과 일치합니다.

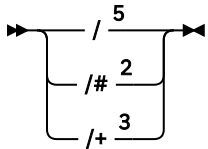
왼쪽 구분 기호



본문 구분 기호



오른쪽 구분 기호



참고:

- 1 토픽 문자열이 빈 토픽으로 시작함
- 2 0개 이상의 레벨과 일치합니다. 여러 개의 다중 레벨 일치 문자열은 하나의 다중 레벨 일치 문자열과 효과는 동일합니다.
- 3 정확히 하나의 레벨과 일치합니다.
- 4 //는 빈 토픽(토픽 문자열이 없는 토픽 오브젝트)입니다.
- 5 토픽 문자열이 빈 토픽으로 끝남

토픽 기반 와일드카드가 와일드카드가 아닌 경우

와일드카드 문자 '+' 및 '#'은 토픽 레벨에서 해당 문자를 포함하여 다른 문자와 함께 혼합된 경우 특별한 의미는 없습니다.

즉 토픽 레벨에서 다른 문자와 함께 '+' 또는 '#'을 포함하는 토픽을 발행할 수 있음을 의미합니다.

예를 들어 다음 두 개의 토픽을 고려하십시오.

1. level0/level1/+/level4/#
2. level0/level1/#+/level4/level#

토픽 문자열 결합

메시지를 구독할 수 있도록 구독을 작성하거나 토픽을 열 경우, 토픽 문자열이 두 개의 별도 하위 토픽 문자열 또는 "subtopics"를 결합하여 형식화될 수 있습니다. 한 하위 토픽은 애플리케이션 또는 관리 명령에 의해 토픽 문자열로 제공되며, 다른 하위 토픽은 토픽 오브젝트와 연관된 토픽 문자열입니다. 하위 토픽을 자체에서 토픽 문자열로 사용하거나 새 토픽 이름을 형성하도록 둘을 결합할 수 있습니다.

예를 들어, MQSC 명령 **DEFINE SUB**를 사용하여 구독을 정의할 경우 이 명령은 **TOPICSTR**(토픽 문자열) 또는 **TOPICOBJ**(토픽 오브젝트) 중 하나 또는 둘 다를 속성으로 사용할 수 있습니다. **TOPICOBJ**만 제공되면 해당 토픽 오브젝트와 연관된 토픽 문자열이 토픽 문자열로 사용됩니다. **TOPICSTR**만 제공되면 이것이 토픽 문자열로 사용됩니다. 모두가 제공되면, 병합하여 **TOPICOBJ/TOPICSTR**의 양식으로 단일 토픽 문자열을 형성합니다. 여기서 **TOPICOBJ** 구성 토픽 문자열이 먼저이고 문자열의 두 부분은 항상 "/" 문자로 구분됩니다.

마찬가지로 MQI 프로그램에서 전체 토픽 이름은 MQOPEN에 의해 작성됩니다. 이는 다음에 나열된 순서대로 발행/구독 MQI 호출에서 사용되는 2개의 필드로 구성됩니다.

1. **ObjectName** 필드에 이름 지정된, 토픽 오브젝트의 **TOPICSTR** 속성.
2. 애플리케이션에서 제공하는 하위 토픽을 정의하는 **ObjectString** 매개변수.

결과로 생성되는 토픽 문자열은 **ResObjectString** 매개변수에서 리턴됩니다.

이러한 필드는 각 필드의 첫 번째 문자가 공백 또는 널 문자가 아니면 존재한다고 간주되며 필드 길이는 0보다 큼니다. 필드 중 하나만 존재하면 이는 토픽 이름으로 그대로 사용됩니다. 필드 모두에 값이 없는 경우 전체 토픽 이름이 유효하지 않으면 호출은 이유 코드 MQRC_UNKNOWN_OBJECT_NAME 또는 MQRC_TOPIC_STRING_ERROR로 실패합니다.

두 필드가 모두 있으면 "/" 문자는 결과로 생성되어 결합된 토픽 이름의 두 요소 사이에 삽입됩니다.

29 페이지의 표 2에서는 토픽 문자열 연결의 예제를 보여줍니다.

표 2. 토픽 문자열 연결 예제			
토픽 오브젝트의 TOPICSTR	애플리케이션 또는 DEFINE SUB 명령에서 제공되는 토픽 문자열	전체 토픽 이름	주석
Football/Scores	' '	Football/Scores	토픽 오브젝트의 TOPICSTR 은 단독으로 사용됩니다.
' '	Football/Scores	Football/Scores	ObjectString/TOPICSTR 은 단독으로 사용됩니다.
Football	Scores	Football/Scores	"/" 문자가 연결 지점에 추가되었습니다.
Football	/Scores	Football//Scores	두 문자열 사이에 '빈 노트'가 생성됩니다. 이는 "Football/Scores"와 다릅니다.
/Football	Scores	/Football/Scores	토픽이 '빈 노트'로 시작합니다. 이는 "Football/Scores"와 다릅니다.

"/" 문자는 30 페이지의 『토픽 트리』에서 전체 토픽 이름에 구조를 제공하는 특수 문자로 간주됩니다. "/" 문자는 토픽 트리의 구조가 영향을 받기 때문에 다른 이유로 사용해서는 안 됩니다. 토픽 "/Football"은 토픽 "Football"과 동일하지 않습니다.

참고: 구독을 작성할 때 토픽 오브젝트를 사용하면, 토픽 오브젝트 토픽 문자열의 값이 정의 시에 구독에서 수정됩니다. 토픽 오브젝트의 후속 변경사항은 구독이 정의되는 토픽 문자열에 영향을 미치지 않습니다.

토픽 문자열의 와일드카드 문자

다음 와일드카드 문자는 특수 문자입니다.

- 더하기 부호(+)
- 숫자 부호(#)
- 별표(*)
- 물음표(?)

와일드카드 문자는 구독에서 사용될 경우에만 특별한 의미를 가집니다. 이러한 문자가 다른 곳에서 사용될 경우 적합하지 않다고 간주되지 않지만, 사용 방법을 이해하고 있어야 하며 토픽 오브젝트 발행 또는 정의 시에 토픽 문자열에 이러한 문자를 사용하지 않는 것이 좋습니다.

한 토픽 레벨 내의 다른 문자(해당 문자 포함)와 함께 # 또는 +가 결합된 토픽 문자열에서 발행할 경우, 토픽 문자열은 와일드카드 설계와 함께 구독될 수 있습니다.

두 / 문자 사이의 유일한 문자로 # 또는 +를 사용하여 토픽 문자열에서 발행할 경우, 토픽 문자열은 와일드카드 설계 MQSO_WILDCARD_TOPIC을 사용하여 애플리케이션에 의해 명시적으로 구독될 수 없습니다. 이 상황으로 인해 애플리케이션은 예상보다 많은 발행을 가져옵니다.

정의된 토픽 오브젝트의 토픽 문자열에서 와일드카드 문자를 사용해서는 안 됩니다. 이렇게 할 경우, 오브젝트가 발행자에 의해 사용될 때는 문자가 리터럴 문자로 처리되고 구독에서 사용될 경우 와일드카드 문자로 처리됩니다. 이는 혼동을 야기할 수 있습니다.

코드 스니펫 예제

예제 프로그램 [예제 2: 가변 토픽에 대한 발행자](#)에서 추출한 이 코드 스니펫은 가변 토픽 문자열에 토픽 오브젝트를 결합합니다.

```
MQOD      td = {MQOD_DEFAULT}; /* Object Descriptor          */
td.ObjectType = MQOT_TOPIC; /* Object is a topic      */
td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
strcpy(td.ObjectName, MQ_TOPIC_NAME_LENGTH);
td.ObjectString.VSPtr = topicString;
td.ObjectString.VSLength = (MQLONG)strlen(topicString);
td.ResObjectString.VSPtr = resTopicStr;
td.ResObjectString.VSBufSize = sizeof(resTopicStr)-1;
MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
```

토픽 트리

정의한 각 토픽은 요소 또는 토픽 트리의 노드입니다. 토픽 트리는 비워 두거나 MQSC 또는 PCF 명령을 사용하여 이미 정의된 토픽으로 시작되거나 해당 토픽을 포함할 수 있습니다. 토픽 작성 명령을 사용하거나 발행 또는 구독에 토픽을 처음으로 지정하면 새 토픽을 정의할 수 있습니다.

아무 문자열이나 사용해도 토픽의 토픽 문자열을 정의할 수 있으나 계층 구조 트리에 적합한 토픽 문자열을 선택해야 합니다. 토픽 문자열 및 토픽 트리를 신중하게 디자인하면 다음과 같은 조작에 도움이 될 수 있습니다.

- 다중 토픽을 구독합니다.
- 보안 정책을 설정합니다.

토픽 트리를 평면적 선형 구조로 구성할 수도 있지만, 토픽 트리를 하나 이상의 루트 토픽이 있는 계층 구조로 빌드하는 것이 더 좋습니다. 보안 계획 및 토픽에 대한 자세한 정보는 [발행/구독 보안](#)을 참조하십시오.

[31 페이지의 그림 5](#)에서는 루트 토픽이 하나인 토픽 트리 예제를 보여줍니다.

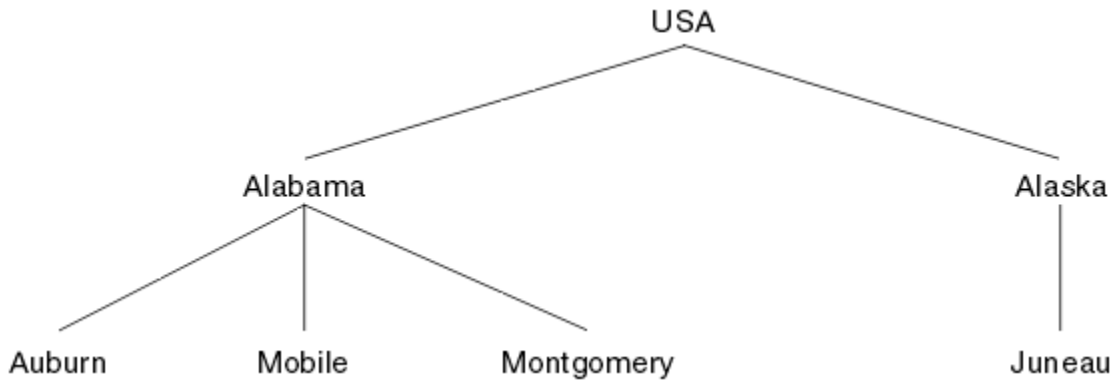


그림 5. 토픽 트리 예제

그림의 각 문자열은 토픽 트리의 노드를 나타냅니다. 전체 토픽 문자열은 토픽 트리에 있는 하나 이상의 레벨에서 노드를 집계하여 작성됩니다. 레벨은 "/" 문자로 구분됩니다. 지정된 전체 토픽 문자열의 형식은 "root/level2/level3"입니다.

31 페이지의 그림 5에 표시된 토픽 트리의 올바른 토픽은 다음과 같습니다.

"USA"
 "USA/Alabama"
 "USA/Alaska"
 "USA/Alabama/Auburn"
 "USA/Alabama/Mobile"
 "USA/Alabama/Montgomery"
 "USA/Alaska/Juneau"

토픽 문자열 및 토픽 트리를 디자인할 때, 큐 관리자는 토픽 문자열의 의미를 도출하도록 시도하거나 토픽 문자열 자체를 해석하지 않습니다. 단순히 선택한 메시지를 해당 토픽의 구독자에게 보내는 데 토픽 문자열을 사용합니다.

토픽 트리의 구성 및 콘텐츠에는 다음 원칙이 적용됩니다.

- 토픽 트리의 레벨 수에 제한이 없습니다.
- 토픽 트리의 레벨 이름 길이에 제한이 없습니다.
- "루트" 노드 수에 제한이 없습니다. 즉 토픽 트리 수에 제한이 없습니다.

토픽 트리에서 불필요한 토픽 수 감소

발행/구독 시스템의 성능은 토픽 트리의 원하지 않는 토픽 수를 줄여서 개선됩니다. 원하지 않는 토픽은 무엇이고 이들을 제거하는 방법은 무엇입니까?

성능에 부정적인 영향을 주지 않고 많은 수의 토픽을 작성할 수 있습니다. 그러나, 발행/구독 사용의 몇 가지 방법은 토픽 트리를 지속적으로 팽창시킵니다. 유난히 많은 수의 토픽이 한 번 작성되고 다시는 사용되지 않습니다. 커지는 토픽 수는 성능 문제점이 될 수 있습니다.

크고 커져가는 원하지 않는 토픽 수로 이어지는 설계를 피할 수 있는 방법은 무엇입니까? 큐 관리자가 토픽 트리에서 원하지 않는 토픽을 제거하도록 도와주기 위해 무엇을 할 수 있습니까?

큐 관리자는 30분 동안 사용되지 않은 토픽을 원하지 않는 토픽으로 인식합니다. 큐 관리자는 토픽 트리에서 원하지 않는 토픽을 제거합니다. 30분 지속 기간은 큐 관리자 속성 **TREELIFE**를 변경하여 바꿀 수 있습니다. 토픽이 큐 관리자에 사용되지 않는 것으로 나타나는지 확인하여 큐 관리자가 원하지 않는 토픽을 제거하도록 도울 수 있습니다. 32 페이지의 『[사용되지 않는 토픽의 개념](#)』 절에는 사용하지 않은 토픽의 개념에 대한 설명이 있습니다.

애플리케이션을 설계하고 특히 장기 실행하는 애플리케이션을 설계하는 프로그래머는 자원 사용법, 즉 프로그램에 필요한 자원의 양, 한정되지 않은 수요가 있는지 여부 및 자원 누출이 있는지 여부 등을 고려합니다. 토픽은 발행/구독 프로그램이 사용하는 자원입니다. 프로그램이 사용하는 다른 자원과 마찬가지로 토픽의 사용을 면밀히 조사하십시오.

사용되지 않는 토픽의 개념

사용되지 않는 토픽의 개념을 정의하기 전에, 무엇이 정확하게 토픽으로 계수됩니까?

USA/Alabama/Auburn과 같은 토픽 문자열을 토픽으로 변환하면 토픽이 토픽 트리에 추가됩니다. 추가 토픽 노드 및 해당 토픽이 필요한 경우 트리에 작성됩니다. 토픽 문자열 USA/Alabama/Auburn이 다음 세 토픽을 갖는 트리로 변환됩니다.

- USA
- USA/Alabama
- USA/Alabama/Auburn

토픽 트리의 모든 토픽을 표시하려면 **runmqsc** 명령 `DISPLAY TPSTATUS('#') TYPE(TOPIC)` 을 사용하십시오.

토픽 트리의 미사용 토픽은 다음 특성을 갖습니다.

토픽 오브젝트와 연관되지 않음

관리 토픽 오브젝트는 토픽과 연관시키는 토픽 문자열을 갖습니다. 토픽 오브젝트 Alabama를 정의하면 연관되는 USA/Alabama 토픽이 없는 경우 토픽이 토픽 문자열에서 작성됩니다. 토픽이 존재하는 경우 토픽 오브젝트 및 토픽이 토픽 문자열을 사용하여 함께 연관됩니다.

보유된 발행을 갖지 않음

발행자가 MQPMO_RETAIN 옵션을 사용하여 토픽에 메시지를 넣으면 보유된 발행물이 있는 토픽이 생성됩니다.

Use the **runmqsc** command `DISPLAY TPSTATUS('USA/Alabama') RETAINED` to check if USA/Alabama has a retained publication. 응답은 YES 또는 NO입니다.

runmqsc 명령 `CLEAR TOPICSTR('USA/Alabama') CLTRTYPE(RETAINED)` 를 사용하여 USA/Alabama 에서 보유된 발행물을 제거하십시오.

하위 토픽이 없음

USA/Alabama/Auburn은 하위 토픽이 없는 토픽입니다. USA/Alabama/Auburn은 USA/Alabama의 직접 하위 토픽입니다.

runmqsc 명령 `DISPLAY TPSTATUS('USA/Alabama/+')` 를 사용하여 USA/Alabama 의 직접 하위를 표시하십시오.

노드에 대한 활성 발행자가 없음

노드에 대한 활성 발행자는 출력을 위해 열린 토픽을 갖는 애플리케이션입니다.

예를 들어, 애플리케이션은 열려 있는 옵션 MQOO_OUTPUT을(를) 사용하여 **Alabama**(이)라는 토픽 오브젝트를 엽니다.

활성 공개자를 USA/Alabama 및 모든 하위에 표시하려면 **runmqsc** 명령 `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACTCONN`을 사용하십시오.

노드에 대한 활성 구독자가 없음

활성 구독자는 지속 가능한 등록이거나, MQSUB가 있는 토픽에 대한 등록을 등록한 애플리케이션이거나 이를 닫지 않을 수 있습니다.

USA/Alabama에 대한 활성 등록을 표시하려면 **runmqsc** 명령 `DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACTCONN`을 사용하십시오.

USA/Alabama 및 모든 하위에 대한 활성 등록을 표시하려면 **runmqsc** 명령 `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACTCONN`을 사용하십시오.

토픽 트리에서 토픽의 수 줄이기

즉, 토픽 트리에서 토픽의 수를 줄이는 여러 가지 방법이 있습니다.

TREELIFE 수정

미사용 토픽은 기본적으로 30분의 수명을 갖습니다. 미사용 토픽의 수명을 더 작게 만들 수 있습니다.

예를 들어, `runmqsc` 명령 (ALTER QMGR TREELIFE(900)) 은 사용하지 않는 주제의 수명을 30분에서 15분으로 단축합니다.

예외적으로 큐 관리자 재시작

큐 관리자가 재시작될 때, 토픽 트리는 토픽 오브젝트, 보유된 발행을 갖는 노드 및 지속 가능 구독으로부터 다시 초기화됩니다. 발행자 및 구독자 프로그램의 조작에 의해 작성된 토픽은 제거됩니다.

정기적으로 `runmqsc` 명령 DISPLAY TPSTATUS('#') TYPE(TOPIC) 을 사용하여 모든 주제를 나열하고 숫자가 증가하고 있는지 확인하십시오.

최후의 수단으로서, 원하지 않는 토픽의 성장이 과거의 성능 문제점의 원인이었던 경우 큐 관리자를 재시작하십시오.

관리 토픽 오브젝트

관리 토픽 오브젝트를 사용하여 기본이 아닌 특정 속성을 토픽에 지정할 수 있습니다.

33 페이지의 그림 6에서는 서로 다른 스포츠를 포함하는 별도의 토픽으로 구분된 Sport의 상위 레벨 토픽을 토픽 트리로 시각화하는 방법을 보여줍니다.



그림 6. 토픽 트리의 시각화

33 페이지의 그림 7에서는 각 스포츠에 대한 여러 유형의 정보를 구분하기 위해 추가로 토픽 트리를 구분하는 방법을 보여줍니다.



그림 7. 확장된 토픽 트리

표시된 토픽 트리를 작성하려는 경우 관리 토픽 오브젝트를 정의하지 않아도 됩니다. 이 트리의 각 노드는 발행 또는 구독 조작에서 작성된 토픽 문자열에 의해 정의됩니다. 트리에서 각 토픽은 해당 상위에서 속성을 상속합니다. 기본적으로 모든 속성은 ASPARENT로 설정되므로 속성은 상위 토픽 오브젝트에서 상속됩니다. 이 예제에서 모든 토픽에는 Sport 토픽과 동일한 속성이 있습니다. Sport 토픽에는 관리 토픽 오브젝트가 없으며 SYSTEM.BASE.TOPIC에서 해당 속성을 상속합니다.

토픽 트리의 루트 노드(즉, SYSTEM.BASE.TOPIC)에서 mqm이 아닌 사용자에게는 권한을 부여하지 않는 것이 좋습니다. 권한은 상속되지만 제한할 수 없기 때문입니다. 따라서 이 레벨에서 권한을 제공하면 전체 트리에 권한을 제공하게 됩니다. 따라서 계층의 더 낮은 토픽 레벨에서 권한을 제공해야 합니다.

관리 토픽 오브젝트는 토픽 트리에 있는 특정 노드에 대한 특정 속성을 정의하는 데 사용할 수 있습니다. 다음 예제에서 관리 토픽 오브젝트는 축구 토픽의 지속 가능한 등록 특성 DURSUB를 NO값으로 설정하도록 정의됩니다.

```
DEFINE TOPIC(FOOTBALL.EUROPEAN)
TOPICSTR('Sport/Soccer')
DURSUB(NO)
DESCR('Administrative topic object to disallow durable subscriptions')
```

이제 토픽 트리는 다음과 같이 시각화될 수 있습니다.

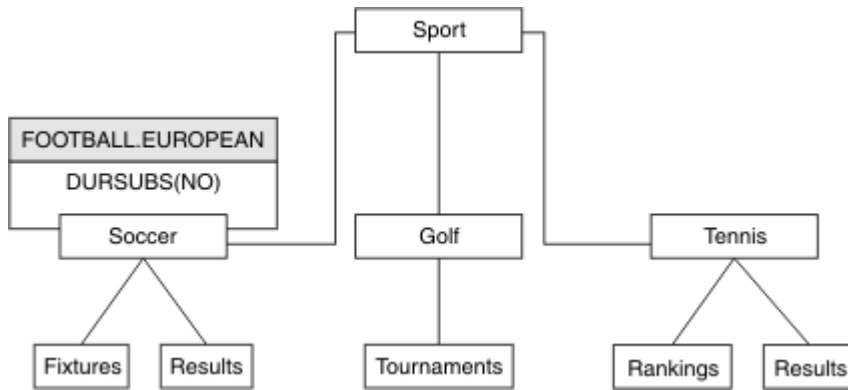


그림 8. Sport/Soccer 토픽과 연관된 관리 토픽 오브젝트의 시각화

트리에서 Soccer 아래 토픽을 구독하는 애플리케이션은 계속해서 관리 토픽 오브젝트를 추가하기 전에 사용하던 토픽 문자열을 사용할 수 있습니다. 하지만 애플리케이션은 문자열 /Sport/Soccer 대신 오브젝트 이름 FOOTBALL.EUROPEAN을 사용하여 구독하기 위해 기록할 수 있습니다. 예를 들어, /Sport/Soccer/Results를 구독하려면, 애플리케이션이 MQSD.ObjectName을 FOOTBALL.EUROPEAN으로 MQSD.ObjectString을 Results로 지정할 수 있습니다.

이 기능을 사용하면 애플리케이션 개발자로부터 토픽 트리의 일부를 숨길 수 있습니다. 토픽 트리의 특정 노드에서 관리 토픽 오브젝트를 정의하면 애플리케이션 개발자가 노드의 하위로 고유한 토픽을 정의할 수 있습니다. 개발자는 상위 토픽에 대해 알아야 하지만 상위 트리의 다른 노드에 대해서는 알지 않아도 됩니다.

속성 상속

토픽 트리에 관리 토픽 오브젝트가 많으면 기본적으로 각 관리 토픽 오브젝트는 가장 가까운 상위 관리 토픽에서 해당 속성을 상속합니다. 이전 예제는 34 페이지의 그림 9에서 확장되었습니다.

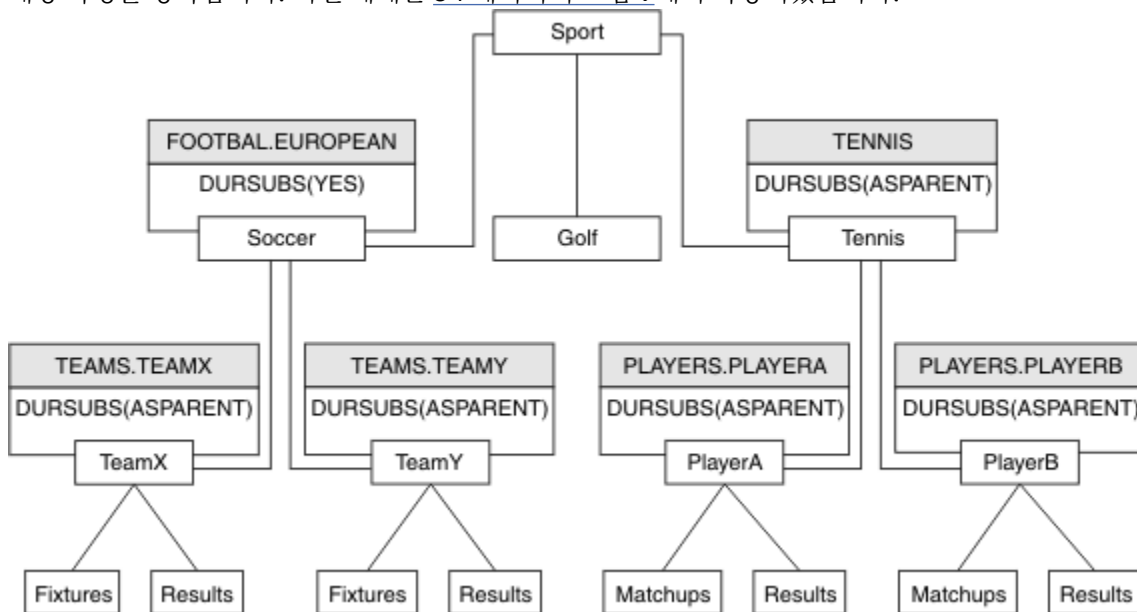


그림 9. 여러 관리 토픽 오브젝트를 포함하는 토픽 트리

예를 들어 상속을 사용하여 /Sport/Soccer의 모든 하위 토픽에 구독이 비지속인 특성을 제공합니다. FOOTBALL.EUROPEAN의 DURSUB 속성을 NO로 변경하십시오.

이 속성은 다음과 같은 명령을 사용하여 설정할 수 있습니다.

```
ALTER TOPIC(FOOTBALL.EUROPEAN) DURSUB(NO)
```

Sport/Soccer 의 하위 주제의 모든 관리 토픽 오브젝트에는 DURSUB 특성이 기본값 ASPARENT로 설정되어 있습니다. After changing the DURSUB property value of FOOTBALL . EUROPEAN to NO, the child topics of Sport/Soccer inherit the DURSUB property value NO . Sport/Tennis의 모든 하위 토픽은 SYSTEM.BASE.TOPIC 오브젝트에서 DURSUB의 값을 상속합니다. SYSTEM.BASE.TOPIC의 값은 YES입니다.

이제 Sport/Soccer/TeamX/Results 토픽에서 지속 가능 구독을 작성하려고 하면 실패합니다. 그러나 Sport/Tennis/PlayerB/Results의 지속 가능 구독 작성에는 성공합니다.

WILDCARD 특성으로 와일드카드 사용 제어

MQSC **Topic** WILDCARD 특성 또는 동등한 PCF 토픽 WildcardOperation 특성을 사용하여 와일드카드 토픽 문자열 이름을 사용하는 구독자 애플리케이션으로 발행물을 전달하도록 제어하십시오. WILDCARD 특성은 다음과 같은 두 가지 가능한 값을 보유할 수 있습니다.

WILDCARD

이 토픽에 관한 와일드카드 구독의 동작입니다.

PASSTHRU

이 토픽 오브젝트의 토픽 문자열보다 덜 특정한 와일드카드 토픽에 대한 구독이 이 토픽 및 이 토픽보다 더욱 특정한 토픽 문자열에 대한 발행물을 수신합니다.

BLOCK

이 토픽 오브젝트의 토픽 문자열보다 덜 특정한 와일드카드 토픽에 대한 구독이 이 토픽 또는 이 토픽보다 더욱 특정한 토픽 문자열에 대한 발행물을 수신하지 않습니다.

이 속성의 값은 구독이 정의될 때 사용됩니다. 이 속성을 대체할 경우 기존 구독에 포함된 토픽 세트는 수정의 영향을 받지 않습니다. 이 시나리오는 토픽 오브젝트를 작성 또는 삭제할 때 토픽로지가 변경된 경우에도 적용됩니다. WILDCARD 속성을 수정한 후에 작성된 구독과 일치하는 토픽 세트가 수정된 토픽로지를 사용하여 작성됩니다. 일치하는 토픽 세트를 강제로 기존 구독에 대해 재평가하려는 경우 큐 관리자를 재시작해야 합니다.

예제, 39 페이지의 『예제: Sport 발행/구독 클러스터 작성』에서는 35 페이지의 그림 10에 표시된 토픽 트리 구조를 작성하는 단계를 수행할 수 있습니다.

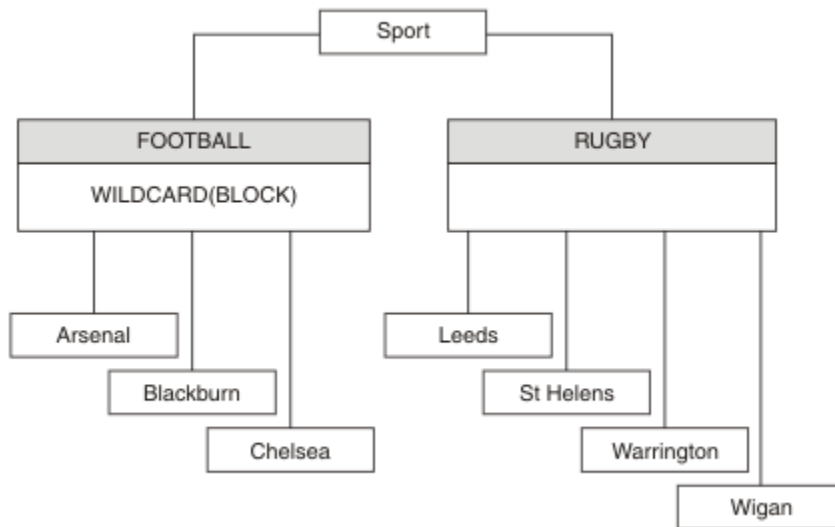


그림 10. WILDCARD 특성, BLOCK을 사용하는 토픽 트리

와일드카드 토픽 문자열 #을 사용하는 구독자는 Sport 토픽 및 Sport/Rugby 하위 트리에 대한 모든 발행을 수신합니다. 구독자는 Sport/Football 하위 트리에 대한 발행을 수신하지 않습니다. Sport/Football 토픽의 WILDCARD 특성 값은 BLOCK입니다.

PASSTHRU는 기본 설정입니다. WILDCARD 특성 값 PASSTHRU 을 Sport 트리의 노드로 설정할 수 있습니다. 노드에 WILDCARD 특성 값 BLOCK이 없으면 PASSTHRU를 설정해도 Sports 트리의 노드에서 구독자가 관찰하는 작동은 대체되지 않습니다.

예제에서 구독을 작성하여 와일드카드 설정이 전달되는 발행에 미치는 영향을 확인하십시오(40 페이지의 그림 14 참조). 일부 발행을 작성하려면 41 페이지의 그림 17에서 발행 명령을 실행하십시오.

pub QMA

그림 11. QMA에 발행

결과는 36 페이지의 표 3에 표시됩니다. WILDCARD 특성 값 BLOCK을 설정할 때 와일드카드를 포함하는 구독이 와일드카드 범위 내 토픽에 대한 발행을 수신하지 않도록 하는 방법에 주의하십시오.

구독	토픽 문자열	수신된 발행	Notes®
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Sports/Football에서 WILDCARD(BLOCK)으로 차단된 Football 하위 트리에 대한 모든 발행
SARSENAL	Sports/#/Arsenal	-	Sports/Football의 WILDCARD(BLOCK)은 Arsenal에서 와일드카드 서브스크립션을 방지합니다.
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby의 기본 WILDCARD는 Leeds에서 와일드카드 구독을 금지하지 않습니다.

참고:

구독에 WILDCARD 특성 값 BLOCK이 있는 토픽 오브젝트와 일치하는 와일드카드가 있다고 가정합니다. 또한 구독이 일치하는 와일드카드 오른쪽에 토픽 문자열을 포함하면 구독은 발행을 수신하지 않습니다. 차단되지 않은 발행 세트는 차단된 와일드카드 상위에 해당하는 토픽에 대한 발행입니다. BLOCK 특성 값이 설정된 토픽의 하위에 해당하는 토픽에 대한 발행은 와일드카드로 차단됩니다. 따라서 와일드카드 오른쪽에 토픽을 포함하는 구독 토픽 문자열은 일치하는 발행을 수신하지 않습니다.

WILDCARD 특성 값을 BLOCK으로 설정해도 와일드카드를 포함하는 토픽 문자열을 사용하여 구독할 수 없음을 의미하지는 않습니다. 이러한 구독은 정상적입니다. 구독이 WILDCARD 특성 값 BLOCK이 설정된 토픽 오브젝트를 포함하는 토픽과 일치하는 명시적 토픽을 보유합니다. 이는 WILDCARD 특성 값 BLOCK이 설정된 토픽의 상위 또는 하위에 해당하는 토픽에 대한 와일드카드를 사용합니다. 35 페이지의 그림 10의 예제에서 Sports/Football/#과 같은 구독은 발행을 수신할 수 있습니다.

와일드카드 및 클러스터 토픽

클러스터 토픽 정의는 클러스터의 모든 큐 관리자로 전파됩니다. 클러스터의 한 큐 관리자에서 클러스터 토픽에 대한 구독으로 인해 큐 관리자가 프록시 구독을 작성할 수 있습니다. 프록시 구독은 클러스터의 다른 모든 큐 관리자에서 작성됩니다. 클러스터 토픽과 결합해 와일드카드를 포함하는 토픽 문자열을 사용하는 구독은 작동을 예상하기 어려울 수 있습니다. 작동은 다음 예제에서 설명됩니다.

In the cluster set up for the example, 39 페이지의 『예제: Sport 발행/구독 클러스터 작성』, QMB has the same set of subscriptions as QMA, yet QMB received no publications after the publisher published to QMA, see 36 페이지의 그림 11. Sports/Football 및 Sports/Rugby 주제는 클러스터 토픽이지만 fullsubs.tst에 정의된 등록은 클러스터 토픽을 참조하지 않습니다. QMB에서 QMA로 프록시 구독이 전파되지 않습니다. 프록시 구독이 없으면 QMA에 대한 발행물은 QMB로 전달되지 않습니다.

Sports/Rugby의 경우, 일부 구독(예: Sports/#/Leeds)은 클러스터 토픽을 참조하는 것처럼 보입니다. Sports/#/Leeds 구독은 실제로 토픽 오브젝트 SYSTEM.BASE.TOPIC으로 해석됩니다.

Sports/#/Leeds와 같이 구독에서 참조하는 토픽 오브젝트를 해석하는 규칙은 다음과 같습니다. 토픽 문자열을 첫 번째 와일드카드까지 자릅니다. 토픽 문자열을 왼쪽부터 스캔하여 연관된 관리 토픽 오브젝트가 있는 첫 번째 토픽을 찾습니다. 이 토픽 오브젝트는 클러스터 이름을 지정하거나 로컬 토픽 오브젝트를 정의할 수 있습니다.

Sports/#/Leeds예제에서 잘림 후의 토픽 문자열은 토픽 오브젝트가 없는 Sports이고, Sports/#/Leeds는 로컬 토픽 오브젝트인 SYSTEM.BASE.TOPIC에서 상속합니다.

클러스터 토픽을 구독하여 와일드카드 전파 방식을 변경하는 방법을 확인하려면 배치 스크립트, [upsubs.bat](#)를 실행하십시오. 스크립트는 구독 큐를 지우고 [fullsubs.tst](#)에서 클러스터 토픽 구독을 추가합니다. [puba.bat](#)를 다시 실행하여 발행 배치를 작성하십시오(36 페이지의 [그림 11](#) 참조).

37 페이지의 표 4에서는 발행이 발행된, 동일한 큐 관리자에 2개의 새 구독을 추가하는 결과를 보여줍니다. 결과는 예상한 대로입니다. 새 구독은 각각 하나의 발행을 수신하고 다른 구독에서 수신하는 발행 수는 변경되지 않습니다. 다른 클러스터 큐 관리자에서 예상치 못한 결과가 발생합니다(37 페이지의 표 5 참조).

표 4. QMA에서 수신된 발행			
구독	토픽 문자열	수신된 발행	참고
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Sports/Football에서 WILDCARD(BLOCK)으로 차단된 Football 하위 트리에 대한 모든 발행
SARSENAL	Sports/#/Arsenal	-	Sports/Football의 WILDCARD(BLOCK)은 Arsenal에서 와일드카드 서브스크립션을 방지합니다.
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby의 기본 WILDCARD는 Leeds에서 와일드카드 구독을 금지하지 않습니다.
FARSENAL	Sports/Football/ Arsenal	Sports/Football/ Arsenal	Arsenal은 구독이 와일드카드를 포함하지 않으므로 발행을 수신합니다.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds는 모든 이벤트에서 발행을 수신합니다.

37 페이지의 표 5에서는 QMB에서 2개의 새 구독을 추가하고 QMA에서 발행하는 결과를 보여줍니다. QMB가 이러한 2개의 새 구독 없이 발행을 수신하지 않는다는 점을 상기하십시오. 예상대로 두 개의 새 구독은 발행물을 수신합니다. Sports/FootBall 및 Sports/Rugby는 둘 다 클러스터 토픽이기 때문입니다. QMB forwarded proxy subscriptions for Sports/Football/Arsenal and Sports/Rugby/Leeds to QMA, which then sent the publications to QMB.

예상치 않은 결과는 이전에 발행물을 수신하지 않았던 두 개의 구독 Sports/# 및 Sports/#/Leeds가 이제 발행을 수신하는 것입니다. 이는 다른 구독의 QMB로 전달된 Sports/Football/Arsenal 및 Sports/Rugby/Leeds 발행물이 QMB에 첨부된 구독자에서 사용 가능하기 때문입니다. 결과적으로 로컬 토픽 Sports/# 및 Sports/#/Leeds에 대한 구독은 Sports/Rugby/Leeds 발행을 수신합니다. Sports/#/Arsenal은 계속해서 발행을 수신하지 않습니다. Sports/Football에서 해당 WILDCARD 특성 값이 BLOCK으로 설정되었기 때문입니다.

표 5. QMB에서 수신된 발행			
구독	토픽 문자열	수신된 발행	참고
SPORTS	Sports/#	Sports/Rugby/Leeds	Sports/Football에서 WILDCARD(BLOCK)에 의해 차단된 Football 하위 트리에 대한 모든 발행
SARSENAL	Sports/#/Arsenal	-	Sports/Football의 WILDCARD(BLOCK)은 Arsenal에서 와일드카드 서브스크립션을 방지합니다.

표 5. QMB에서 수신된 발행 (계속)

구독	토픽 문자열	수신된 발행	참고
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby의 기본 WILDCARD는 Leeds의 와일드카드 구독을 방해하지 않음
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal은 구독이 와일드카드를 포함하지 않으므로 발행을 수신합니다.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds는 모든 이벤트에서 발행을 수신합니다.

대부분의 애플리케이션에서는 한 구독이 다른 구독의 작동에 영향을 주는 것은 바람직하지 않습니다. 값이 BLOCK인 WILDCARD 특성의 한 가지 중요한 사용법은 균일하게 작동하는 와일드카드를 포함하는 동일한 토픽 문자열에 대한 구독을 작성하는 것입니다. 발행자와 동일한 큐 관리자 또는 다른 큐 관리자의 구독인지에 상관없이 구독 결과는 동일합니다.

와일드카드 및 스트림

WebSphere MQ 버전 6 스트림은 WebSphere MQ 버전 7에 의해 토픽으로 맵핑됩니다. 41 페이지의 『스트림 및 토픽』의 내용을 참조하십시오. 버전 7의 **strmqbrk**로 수행되는 기본 맵핑에서 스트림 Sports의 모든 토픽이 토픽 Sports로 맵핑됩니다. 스트림 Business의 모든 주제는 Business주제에 맵핑됩니다.

Sports 스트림의 *에 대한 WebSphere MQ 버전 6의 구독은 Sports 트리의 모든 발행물을 수신하며 Business 트리의 발행물은 수신하지 않습니다. 버전 7의 동일한 구독은 Sports 트리의 모든 발행물 및 Business 트리의 모든 발행물을 수신합니다. 이 작동을 차단하려면 스트림이 버전 7로 마이그레이션될 때 **strmqbrk**이 WILDCARD 특성을 설정합니다. 이 명령은 스트림에서 마이그레이션된 최상위 레벨 토픽 각각에 대해 이 특성을 BLOCK 값으로 설정합니다. Sports 및 Business의 WILDCARD 특성은 Sports 및 Business라는 버전 6스트림에서 변환하여 BLOCK 값으로 설정됩니다.

발행/구독 API에 기록되는 새 응용프로그램의 경우, *에 대한 등록은 발행을 수신하지 않습니다. 모든 Sports 발행물을 수신하려면 Sports/* 또는 Sports/#을 구독해야 하며 Business 발행물의 경우에도 이와 유사합니다.

발행/구독 브로커가 WebSphere MQ 버전 7로 이주될 때 기존 큐된 발행/구독 애플리케이션의 동작은 변경되지 않습니다. **Publish, Register Publisher** 또는 **Subscriber** 명령의 **StreamName** 특성은 스트림이 마이그레이션된 토픽의 이름으로 맵핑됩니다.

와일드카드 및 구독 지점

WebSphere Message Broker 구독 지점이 WebSphere MQ 버전 7에 의해 토픽으로 맵핑됩니다. 43 페이지의 『구독 지점 및 토픽』의 내용을 참조하십시오. 버전 7의 **migmqbrk**로 수행되는 기본 맵핑에서, 구독 지점 Sports의 모든 토픽이 Sports 토픽으로 맵핑됩니다. All the topics in the subscription point Business would be mapped to the topic Business.

Sports 구독 지점의 *에 대한 WebSphere Message Broker 버전 6의 구독은 Sports 트리의 모든 발행물을 수신하며 Business 트리의 발행물은 수신하지 않습니다. 버전 7에서 동일한 구독은 Sports 트리의 모든 발행물과 Business 트리의 모든 발행물을 수신합니다. 이 동작을 차단하려면 구독 지점이 버전 7로 마이그레이션될 때 **migmqbrk**가 WILDCARD 특성을 설정합니다. 이 명령은 구독 지점에서 마이그레이션된 최상위 레벨 토픽 각각에 대해 이 특성을 BLOCK 값으로 설정합니다. Sports 및 Business의 WILDCARD 특성은 Sports 및 Business라는 WebSphere Message Broker 구독 지점에서 변환하여 BLOCK 값으로 설정됩니다.

발행/구독 API에 기록된 새 응용프로그램의 경우, 이주의 영향은 *에 대한 등록이 발행을 수신하지 않는다는 것입니다. 모든 Sports 발행물을 수신하려면 Sports/* 또는 Sports/#을 구독해야 하며 Business 발행물의 경우에도 이와 유사합니다.

발행/구독 브로커가 WebSphere MQ 버전 7로 이주될 때 기존 큐된 발행/구독 애플리케이션의 동작은 변경되지 않습니다. **Publish, Register Publisher** 또는 **Subscriber** 명령의 **SubPoint** 특성은 등록이 마이그레이션된 토픽의 이름으로 매핑됩니다.

예제: Sport 발행/구독 클러스터 작성

다음 단계에서는 두 개의 전체 저장소 (CL1A 및 CL1B) 와 두 개의 부분 저장소 (QMA 및 QMB) 가 있는 클러스터 (CL1) 를 작성합니다. 전체 저장소는 클러스터 정의만 보유하는 데 사용됩니다. QMA는 클러스터 토픽 호스트를 지정합니다. 지속 가능 구독은 QMA 및 QMB 모두에서 정의됩니다.

참고: Windows에 대한 예가 코드화됩니다. 다른 플랫폼에서 예제를 구성하고 테스트하도록 `qmgrs.bat` 작성 및 `pub.bat` 작성을 다시 코딩해야 합니다.

1. 스크립트 파일을 작성하십시오.
 - a. `topics.tst` 작성
 - b. `wildsubs.tst` 작성
 - c. `fullsubs.tst` 작성
 - d. `qmgrs.bat` 작성
 - e. `pub.bat` 작성
2. `qmgrs.bat` 작성을 실행하여 구성을 작성하십시오.

```
qmgrs
```

35 페이지의 그림 10에서 토픽을 작성하십시오. 그림 5의 스크립트는 클러스터 토픽 Sports/Football 및 Sports/Rugby를 작성합니다.

참고: REPLACE 옵션은 주제의 TOPICSTR 특성을 바꾸지 않습니다. TOPICSTR은(는) 다른 토픽 트리를 테스트하는 예제에서 사용되는 특성입니다. 토픽을 변경하려면 먼저 토픽을 삭제하십시오.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports')      TOPICSTR('Sports')
DEFINE TOPIC ('Football')   TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal')    TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn')  TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea')    TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby')      TOPICSTR('Sports/Rugby')      CLUSTER(CL1)
DEFINE TOPIC ('Leeds')      TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan')      TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

그림 12. 토픽 삭제 및 작성: `topics.tst`

참고: REPLACE가 토픽 문자열을 바꾸지 않으므로 토픽을 삭제합니다.

와일드카드를 사용하여 구독을 작성하십시오. 35 페이지의 그림 10에서 토픽 오브젝트를 포함하는 토픽에 대응하는 와일드카드. 각 구독에 대한 큐를 작성하십시오. 스크립트를 실행하거나 다시 실행하면 큐가 지워지고 구독이 삭제됩니다.

참고: REPLACE 옵션은 구독의 TOPICOBJ 또는 TOPICSTR 특성을 바꾸지 않습니다. TOPICOBJ 또는 TOPICSTR 는 다른 등록을 테스트하는 예제에서 사용되는 특성입니다. 변경하려면 먼저 구독을 삭제하십시오.

```

DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)

```

그림 13. 와일드카드 구독 작성: *wildsubs.tst*

클러스터 토픽 오브젝트를 참조하는 구독을 작성하십시오.

참고:

구분 기호, /는 TOPICOBJ에서 참조하는 토픽 문자열과 TOPICSTR에서 정의하는 토픽 문자열 사이에 자동으로 삽입됩니다.

DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) 정의는 동일한 등록을 작성합니다. TOPICOBJ는 이미 정의한 토픽 문자열을 참조하는 빠른 방법으로 사용됩니다. 작성되면 구독은 더 이상 토픽 오브젝트를 참조하지 않습니다.

```

DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)

```

그림 14. 구독 삭제 및 삭제: *fullsubs.tst*

2개의 저장소를 포함하는 클러스터를 작성하십시오. 발행 및 구독을 위해 2개의 부분 저장소를 작성하십시오. 모두 삭제하고 다시 시작하도록 스크립트를 다시 실행하십시오. 또한 스크립트는 토픽 계층 및 초기 와일드카드 구독을 작성합니다.

참고:

다른 플랫폼에 비슷한 스크립트를 작성하거나 모두 명령을 입력하십시오. 스크립트를 사용하면 모두 삭제하고 동일한 구성으로 다시 빠르게 시작할 수 있습니다.


```

@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof

```

그림 15. 큐 관리자 작성: *qmgrs.bat*

클러스터 토픽에 구독을 추가하여 구성을 업데이트하십시오.

```

@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst

```

그림 16. 구독 업데이트: *upsubs.bat*

큐 관리자에서 매개변수로 *pub.bat*를 실행하여 발행 토픽 문자열을 포함하는 메시지를 발행하십시오. *Pub.bat*는 샘플 프로그램 *amqspub*를 사용합니다.

```

@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1

```

그림 17. 발행: *pub.bat*

스트림 및 토픽

큐에 있는 발행/구독에는 통합된 발행/구독 모델에 존재하지 않는 발행 스트림의 개념이 내포되어 있습니다. 큐에 있는 발행/구독에서는 스트림이 여러 가지 토픽에 대한 정보 플로우를 분리하는 방법을 제공합니다. IBM WebSphere MQ Version 6.0에서 스트림은 큐로 구현되며 이 큐는 스트림을 지원하는 각 브로커에 정의되어 있습니다. 각 큐의 이름은 동일합니다(스트림의 이름). IBM WebSphere MQ Version 7.0 이후부터 스트림은 관리 상 다른 토픽 ID에 매핑될 수 있는 최상위 레벨 토픽으로 구현됩니다.

기본 스트림 SYSTEM.BROKER.DEFAULT.STREAM은 네트워크의 모든 브로커 및 큐 관리자에 대해 자동으로 설정되며 기본 스트림을 사용하기 위해 추가 구성이 필요하지 않습니다. 디폴트 스트림을 이름이 지정되지 않은 디폴트 토픽 공간으로 생각해보십시오. 기본 스트림에 발행된 토픽은 사용 가능한 큐된 발행/구독에서 연결된 모든 Version 6.0 브로커 및 Version 7.0의 모든 큐 관리자를 즉시 사용할 수 있습니다. 이름 지정된 스트림은 별도의 이름 지정된 토픽 공간입니다. 이름 지정된 스트림을 사용되는 각 브로커에 정의해야 합니다.

토픽을 정의하는 경우, 토픽은 Version 6.0 발행/구독 브로커 및 IBM WebSphere MQ의 나중 버전에서 실행하는 발행자와 구독자를 특별한 구성 없이 사용할 수 있습니다.

발행자 및 구독자가 다른 큐 관리자에 있으면 브로커를 동일한 브로커 계층에 연결한 후에 발행 및 구독에서 이들 사이의 전달을 위해 추가 구성이 필요하지 않습니다. 동일한 상호 운용성이 역으로도 작용합니다.

이름 지정된 스트림

큐된 발행/구독 프로그래밍 모델에 대해 작업하는 솔루션 디자이너는 모든 스포츠 서적을 Sport라는 이름 지정된 스트림에 배치하도록 결정할 수 있습니다. Version 6.0에서는 종종 스트림이 모델 큐 SYSTEM.BROKER.MODEL.STREAM을 사용하여 다른 브로커에 자동으로 복제됩니다. 그러나 큐에 있는 발행/구독이 활성화된 Version 7.0 이상에서 실행되는 큐 관리자에서 스트림을 사용 가능하게 하려면 스트림을 수동으로 추가해야 합니다.

큐 관리자를 Version 6.0에서 마이그레이션하는 경우, **strmqbrk** 명령을 실행하면 Version 6.0 이름 지정된 스트림을 토픽으로 마이그레이션합니다. 스트림 Sport 이 (가) Sport주제에 맵핑됩니다. 이는 z/OS에 적용할 수 없습니다.

스트림 Sport의 Soccer/Results를 구독하는 큐에 있는 발행/구독 애플리케이션은 그대로 작동합니다. MQSUB를 사용하여 Sport 토픽을 구독하고 토픽 문자열 Soccer/Results 을 (를) 제공하는 통합 발행/구독 애플리케이션도 동일한 발행을 수신합니다.

Soccer/Result 주제가 **strmqbrk**에 의해 작성되는 경우, 토픽 문자열 Sport이 있는 Sport토픽의 하위로 정의됩니다. Soccer/Results 에 대한 등록은 Sport/Soccer/Results에 대한 등록으로 실현되므로 Sport 스트림에 대한 발행물은 Business 와 같은 다른 스트림에 대한 발행물에 대한 토픽 영역의 다른 위치에 맵핑됩니다.

strmqbrk를 사용하여 수행하는 자동 마이그레이션이 적합하지 않은 경우가 있으며 이 경우 스트림을 수동으로 추가해야 합니다. 스트림 추가 태스크는 [스트림 추가](#) 주제에서 설명됩니다. 다음은 스트림을 수동으로 추가해야 하는 세 가지 이유입니다.

1. 버전 6 큐 관리자에서 발행/구독 애플리케이션을 계속 유지보수하며, 해당 애플리케이션은 새로 작성된 발행/구독 애플리케이션(나중 버전에서 실행)과 상호 운영됩니다.
2. 통합된 발행/구독 MQI 인터페이스로 애플리케이션을 마이그레이션하는 대신, 이후 버전 큐 관리자에서 실행되는 큐에 있는 발행/구독 애플리케이션을 계속 개발합니다.
3. 스트림을 토픽에 디폴트 맵핑하면 토픽 공간에 "충돌"이 발생하고 스트림의 발행에 다른 곳의 발행과 동일한 토픽 문자열이 생깁니다.

권한

기본적으로 토픽 트리의 루트에는 여러 토픽 오브젝트 (SYSTEM.BASE.TOPIC, SYSTEM.BROKER.DEFAULT.STREAM및 SYSTEM.BROKER.DEFAULT.SUBPOINT) 가 있습니다. 권한 (예: 공개 또는 등록의 경우) 은 SYSTEM.BASE.TOPIC의 권한에 의해 결정됩니다.

SYSTEM.BROKER.DEFAULT.STREAM 또는 SYSTEM.BROKER.DEFAULT.SUBPOINT 의 권한은 무시됩니다. SYSTEM.BROKER.DEFAULT.STREAM 또는 SYSTEM.BROKER.DEFAULT.SUBPOINT 중 하나가 삭제되고 비어 있지 않은 토픽 문자열로 다시 작성되는 경우, 해당 오브젝트에 대해 정의된 권한이 일반 토픽 오브젝트와 동일한 방식으로 사용됩니다.

스트림과 토픽 간의 맵핑

큐에 있는 발행/구독 스트림은 Version 7.0 이후부터 큐를 작성하고 스트림과 동일한 이름을 제공하여 모방됩니다. 때로 큐를 스트림 큐라 부릅니다. 스트림 큐가 큐에 있는 발행/구독 애플리케이션에 이와 같이 표시되기 때문입니다. SYSTEM.QPUBSUB.QUEUE.NAMELIST라는 특수 이름 목록에 큐를 추가해서 발행/구독 엔진에 큐가 식별됩니다. 추가적인 특수 큐를 이름 목록에 추가해서 필요한 만큼 스트림을 추가할 수 있습니다. 마지막으로 토픽에 발행 및 구독할 수 있도록 스트림과 동일한 이름의 토픽 및 스트림 이름과 동일한 토픽 문자열을 추가할 필요가 있습니다.

그러나 예외적인 상황에서 스트림에 대응하는 토픽에 토픽을 정의할 때 선택한 토픽 문자열을 제공할 수 있습니다. 토픽 문자열의 목적은 토픽 공간에서 토픽에 고유한 이름을 제공하는 것입니다. 일반적으로 스트림 이름은 이 목적을 완벽하게 수행합니다. 때때로 스트림 이름과 기존 토픽 이름이 충돌합니다. 이 문제점을 해결하기 위해 스트림에 연관된 토픽에 대해 다른 토픽 문자열을 선택할 수 있습니다. 토픽 문자열을 선택하고 고유한지 확인하십시오.

토픽 정의에 정의된 토픽 문자열에는 MQOPEN 또는 MQSUB MQI 호출을 사용하여 발행자 및 구독자가 제공한 토픽 문자열에 정상적인 방식으로 접두부가 붙습니다. 토픽 오브젝트를 사용하여 토픽을 참조하는 애플리케이션은 접두부 토픽 문자열의 선택에 의해 영향을 받지 않습니다. 따라서 토픽 공간에서 발행을 고유하게 만드는 임의의 토픽 문자열을 선택할 수 있습니다.

다른 스트림을 다른 토픽으로 다시 매핑하는 작업은 한 토픽 세트를 다른 토픽 세트와 완전히 분리하기 위해 고유하도록 토픽 문자열에 사용되는 접두부에 따라 다릅니다. 매핑이 작동하도록 엄격하게 지켜지는 보편적인 토픽 이름 지정 규칙을 정의해야 합니다. Version 7.0에서 토픽 문자열이 충돌하면 스트림을 사용하여 토픽 공간을 분리할 수도 있습니다. Version 7.0 이후부터 접두부 메커니즘을 사용하여 토픽 문자열을 토픽 공간의 다른 위치로 다시 매핑할 수 있습니다.

참고: 스트림을 삭제할 때에는 먼저 스트림의 모든 구독을 삭제하십시오. 이 조치는 브로커 계층의 기타 브로커에서 구독이 생성되는 경우 가장 중요합니다.

예

In 43 페이지의 그림 18, topic 'Sport' has the topic string 'xyz' resulting in publications that originate from stream 'Sport' being prefixed with the string 'xyz' in the version 7 queue manager topic space. 버전 7에서는 토픽 'Sport'로 발행 또는 구독하면 토픽 문자열에 'xyz' 접두부가 붙습니다. 발행물이 버전 6 구독자로 플로우되는 경우, 접두부 'xyz'이 발행물에서 제거되며 'Sport' 스트림에 배치됩니다. 반대로, 발행물이 버전 6에서 버전 7로, 'Sport' 스트림에서 'Sport' 토픽으로 플로우되는 경우 접두부 'xyz'가 토픽 문자열에 추가됩니다.

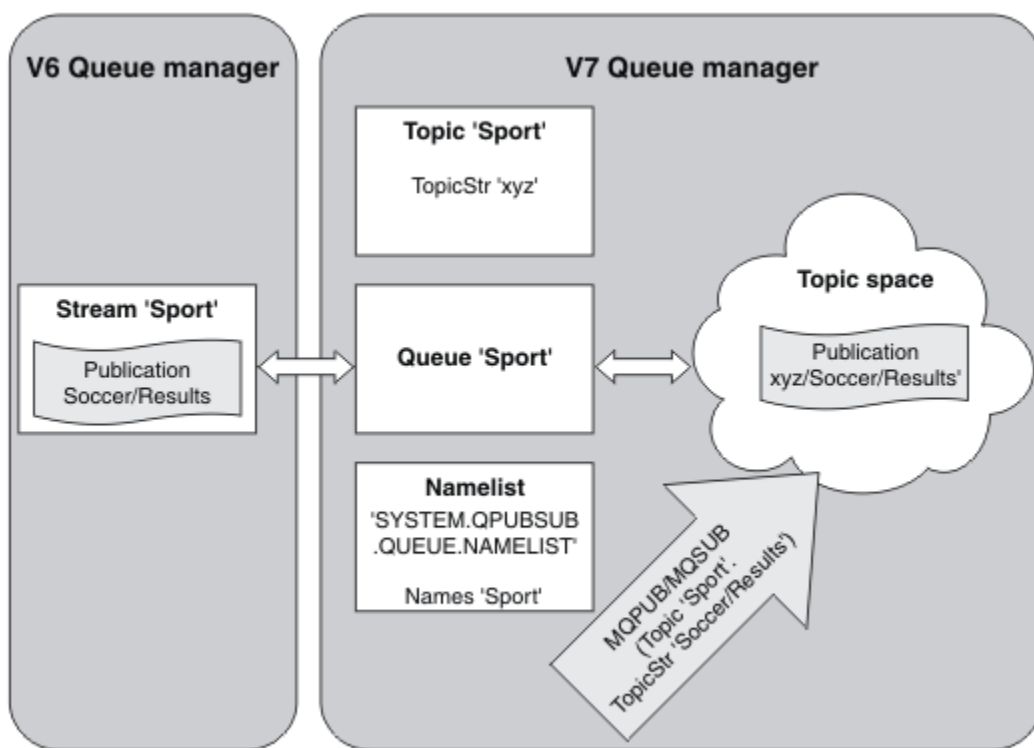


그림 18. 버전 7 토픽과 함께 있는 버전 6 스트림

구독 지점 및 토픽

WebSphere MQ Event Broker 및 Message Broker의 발행 노드의 특정 세트에서 발행물을 요청하는 데 사용되는 구독 지점입니다. 이름 지정된 구독 지점은 토픽 및 토픽 오브젝트에 의해 에뮬레이트됩니다.

WebSphere MQ Event Broker V6.0에서 WebSphere MQ V7.0.1로의 마이그레이션 프로시저인 **migmbbrk**는 이름 지정된 구독 지점을 토픽 및 토픽 오브젝트로 변환합니다. 구독 지점은 보유된 발행물 또는 등록된 구독자를 갖는 경우 자동으로 마이그레이션됩니다. **migmbbrk**는 이름 지정된 구독 지점으로부터 토픽 오브젝트를 작성합니다. 구독 지점의 이름이 토픽 오브젝트의 이름 및 토픽 문자열 자체가 됩니다. 토픽 오브젝트는 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST에 추가됩니다.

동일한 이름을 갖는 토픽 오브젝트가 존재하는 경우 **migmbbrk**는 다음 중 하나를 수행합니다.

1. 토픽 오브젝트가 다른 토픽 문자열을 갖는 경우 또는 구독 지점 이름이 오브젝트 이름보다 긴 경우 **migmbbrk**는 생성된 이름을 갖는 토픽 오브젝트를 작성합니다.
2. 토픽 오브젝트가 동일한 토픽 문자열을 갖는 경우 **migmbbrk**는 기존 오브젝트를 이름 목록에 추가합니다.

수동으로 구독 지점을 추가하려면 [구독 지점 추가](#)를 참조하십시오.

WebSphere MQ Event Broker의 구독 지점

발행 노드는 WebSphere MQ Event 및 Message Broker 메시지 플로우에서 메시지를 필터링하고 구독자에게 메시지를 전송하는 데 사용됩니다. 발행자는 일반적으로 발행 노드에 구독 지점을 설정하지 않습니다. 구독자는 특정 토픽 세트에 관심을 등록하고 일반적으로 구독 지점을 지정하지 않습니다.

구독 지점은 구독에 메시지를 전달하는 발행 노드를 선택하는 방법입니다. 구독자는 구독 지점의 이름으로 토픽 세트에 대한 관심을 규정합니다.

등록 지점 이름을 설정하려면 공개 노드의 **Subscription point** 특성에 이름을 지정하십시오.

구독 지점 특성은 토픽에 대한 발행물이 동일한 토픽에 대한 구독자에게 전달되는지 여부를 제어합니다. 이름 지정된 구독 지점을 갖는 발행 노드에서의 발행물은 동일한 구독 지점에 대한 구독자에게만 전달됩니다. 이름 지정된 구독 지점이 없는 발행 노드에서의 발행물(기본값)은 구독 지점을 이름 지정하지 않은 구독자에게만 전달됩니다.

Nodes with a named subscription point send Publish command messages in MQRFH2 format, with the **SubPoint** property set. 이름 지정된 등록 위치에 대한 등록은 MQRFH2 Register subscriber 명령 메시지에서 **SubPoint** 특성을 설정해야 합니다.

WebSphere MQ의 구독 지점

WebSphere MQ는 구독 지점을 WebSphere MQ 토픽 트리의 다른 토픽 공간에 맵핑합니다. 구독 지점이 없는 명령 메시지의 토픽은 WebSphere MQ 토픽 트리의 루트에 변경되지 않은 채로 맵핑되고 `SYSTEM.BASE.TOPIC`에서 특성을 상속합니다.

구독 지점을 포함하는 명령 메시지는 `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST`에서 토픽 오브젝트 목록을 사용하여 처리됩니다. 명령 메시지의 구독 지점 이름은 목록에 있는 각 토픽 오브젝트에 대한 토픽 문자열과 비교하여 일치됩니다. 일치 항목을 찾으면 구독 지점 이름이 토픽 문자열에 토픽 노드로 추가됩니다. 토픽은 `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST`에 있는 연관된 토픽 오브젝트에서 해당 속성을 상속합니다.

구독 지점 사용 효과는 각 구독 지점에 대해 별도의 토픽 공간을 작성하는 것입니다. 토픽 공간은 구독 지점과 이름이 같은 토픽에 기반을 둡니다. 각 토픽 공간의 토픽은 구독 지점과 이름이 같은 토픽 오브젝트에서 해당 속성을 상속합니다.

일치하는 토픽 오브젝트에 설정되지 않은 속성은 일반적인 방식으로 `SYSTEM.BASE.TOPIC`에서 상속됩니다.

MQRFH2 메시지 헤더를 사용하여 기존 큐된 발행/구독 애플리케이션은 Publish 또는 Register subscriber 명령 메시지에서 **SubPoint** 특성을 설정하여 작업을 계속합니다. 구독 지점은 명령 메시지에서 토픽 문자열과 결합되며 결과로 생성되는 토픽은 다른 것과 마찬가지로 처리됩니다.

새 WebSphere MQ V7 애플리케이션은 구독 지점의 영향을 받지 않습니다. 일치하는 토픽 오브젝트 중 하나로부터 상속하는 토픽을 사용하는 경우 일치하는 구독 지점을 사용하여 큐된 애플리케이션과 상호작용합니다.

예

집합에 있는 기존 WebSphere MQ Event Broker 발행/구독 애플리케이션은 구독 지점을 사용하여 공유 가격을 여러 통화로 발행합니다. IBM 주식의 달러 현물 가격이 구독 지점 USD와 토픽 NYSE/IBM/SPOT을 사용하여 발행됩니다. 파운드화 가격은 동일한 토픽 및 구독 지점 GBP를 사용하여 발행됩니다.

WebSphere MQ의 마이그레이션 프로시저가 대응하는 토픽 문자열 'GBP' 및 'USD'를 갖는 두 토픽 오브젝트 GBP 및 USD를 작성합니다.

구독 지점 USD를 사용하는 WebSphere MQ에서 실행하기 위해 마이그레이션된 토픽 NYSE/IBM/SPOT의 기존 발행자가 토픽 USD/NYSE/IBM/SPOT에 발행물을 작성합니다. 마찬가지로 구독 지점 USD를 사용하는 NYSE/IBM/SPOT에 대한 기존 구독자는 USD/NYSE/IBM/SPOT에 대한 구독을 작성합니다.

MQSUB를 호출하여 버전 7 발행/구독 프로그램에서 달러 현물 가격을 구독하십시오. 'C' 코드 단편에서 보여준 대로 USD 토픽 오브젝트 및 'NYSE/IBM/SPOT' 토픽 문자열을 사용하여 구독을 작성하십시오.

```
strncpy(sd.ObjectName, "USD", MQ_TOPIC_NAME_LENGTH);
sd.ObjectString.VSPtr = "NYSE/IBM/SPOT";
sd.ObjectString.VSLength = MQVS_NULL_TERMINATED;
MQSUB(Hconn, &sd, &Hobj, &Hsub, &CompCode, &Reason);
```

집합의 WebSphere MQ Event Broker가 항상 구독 지점 USD 및 GBP를 사용했는지 여부를 고려하십시오. 그 경우 클러스터 토픽 호스트에서 USD 및 GBP 토픽 오브젝트를 한 번만 클러스터 토픽으로 작성하십시오. 클러스터의 모든 큐 관리자에서 클러스터 토픽으로 SYSTEM.BASE.TOPIC를 변경하기 위해 마이그레이션 프로시저의 [./com.ibm.mq.mig.doc/q007670_dita#q007670_clusterstep](http://com.ibm.mq.mig.doc/q007670_dita#q007670_clusterstep) 단계를 수행할 필요가 없습니다. 대신, 다음 단계를 수행하십시오.

1. 클러스터 토픽 오브젝트에서 USD 및 GBP 토픽 오브젝트의 CLUSTER 속성을 설정하십시오.
2. 클러스터의 다른 큐 관리자에서 USD 및 GBP 토픽 오브젝트의 모든 사본을 삭제하십시오.
3. USD 및 GBP가 클러스터의 모든 큐 관리자에 있는 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST에 정의되었는지 확인하십시오.

분산 발행/구독

이 절에는 큐 관리자 간에 발행/구독 메시징 수행 방법과 큐 관리자, 클러스터 및 계층 연결에 사용할 수 있는 두 개의 서로 다른 큐 관리자 토폴로지에 대한 정보가 포함되어 있습니다.

큐 관리자는 WebSphere MQ 발행/구독 시스템에 있는 다른 큐 관리자와 통신할 수 있으므로, 해당 구독자는 하나의 큐 관리자를 구독하고 초기에 다른 큐 관리자에 발행된 메시지를 수신할 수 있습니다. [45 페이지의 그림 19](#)에 설명되어 있습니다.

[45 페이지의 그림 19](#)에서는 두 큐 관리자와의 발행/구독 시스템을 표시합니다.

- 발행자 4는 큐 관리자 2를 사용하여 일기 예보 정보(날씨 토픽 사용) 및 주요 도로의 트래픽 상황에 대한 정보(트래픽 토픽 사용)를 발행합니다.
- 구독자 4도 이 큐 관리자를 사용하며, 트래픽 토픽을 사용하여 트래픽 상황에 대한 정보를 구독합니다.
- 구독자 3은 발행자와 다른 큐 관리자를 사용하긴 하지만 역시 날씨 상황에 대한 정보를 구독합니다. 큐 관리자가 서로 링크되어 있기 때문에 가능합니다.

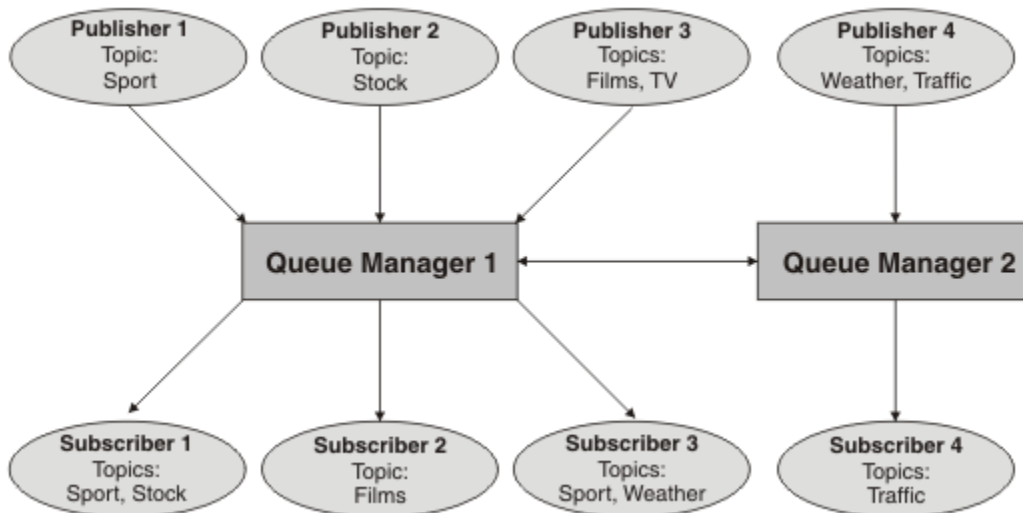


그림 19. 큐 관리자가 2개인 발행/구독 예제

분배된 발행/구독 작동 방식

WebSphere MQ 발행/구독에서는 프록시 구독을 사용하여 구독자가 리모트 큐 관리자로 발행된 메시지를 수신할 수 있게 합니다.

분배된 발행/구독에서는 큐 관리자 및 해당 큐 관리자에 연결되는 애플리케이션의 네트워크를 연결하기 위해 분산 큐잉과 동일한 컴포넌트를 사용합니다. 큐 관리자 간 연결을 설정하여 관련된 컴포넌트와 큐 관리자 간 메시징에 대해 자세히 알아보려면 상호통신 문서를 참조하십시오.

구독자는 분배된 발행/구독 시스템에서 표준 구독 조작이 아닌 조작은 수행하지 않아도 됩니다. 큐 관리자에서 구독이 이루어지는 경우, 큐 관리자가 구독을 연결된 큐 관리자로 전파한 프로세스를 관리합니다. 프록시 구독은 네트워크에서 모든 큐 관리자로 플로우됩니다. 이러한 구독은 발행물이 원래 구독이 작성된 큐 관리자로 다시 라우트되도록 작성됩니다. 46 페이지의 그림 20의 내용을 참조하십시오.

발행물은 해당 토픽에 대한 구독이 리모트 큐 관리자에 있는 경우에만 리모트 큐 관리자로 전파됩니다.

큐 관리자는 로컬 애플리케이션이나 리모트 큐 관리자로부터, 작성된 모든 구독을 통합합니다. 또한 구독이 없는 경우 인접 항목이 있는 구독의 토픽에 대한 프록시 구독을 작성합니다. 47 페이지의 그림 21의 내용을 참조하십시오.

애플리케이션이 정보를 발행하면, 수신 큐 관리자가 이를 리모트 큐 관리자에 올바른 구독이 있는 애플리케이션으로 전달합니다. 하나 이상의 중간 큐 관리자를 통해 전달할 수 있습니다. 47 페이지의 그림 22의 내용을 참조하십시오.

구독자 1은 Asia 큐 관리자(1)에서 특정 토픽에 대한 구독 등록합니다. 이 토픽에 대한 구독은 네트워크(2,3,4)의 다른 모든 큐 관리자로 전달됩니다.

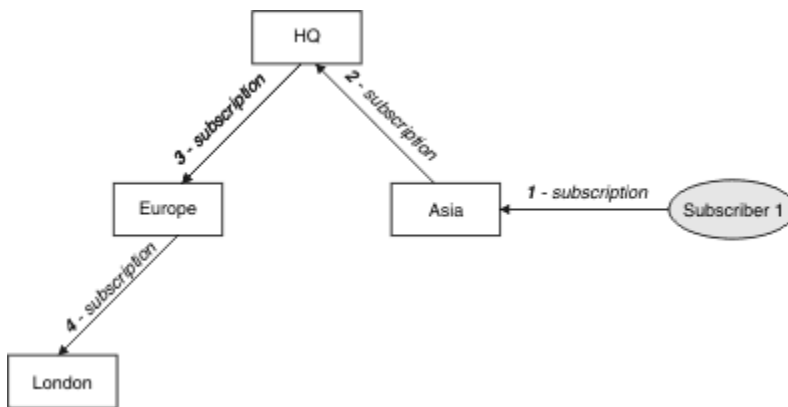


그림 20. 큐 관리자 네트워크를 통한 구독 전파

구독자 2는 HQ 큐 관리자 (5)에서 46 페이지의 그림 20의 경우와 동일한 토픽에 대한 구독을 등록합니다. 이 토픽에 대한 구독은 구독이 네트워크 (6)에 있는지 인식하도록 Asia 큐 관리자로 전달됩니다. 이 토픽에 대한 구독이 이미 등록되었으므로 구독이 Europe 큐 관리자로 전달되지 않습니다. 46 페이지의 그림 20에서 3단계 참조하십시오.

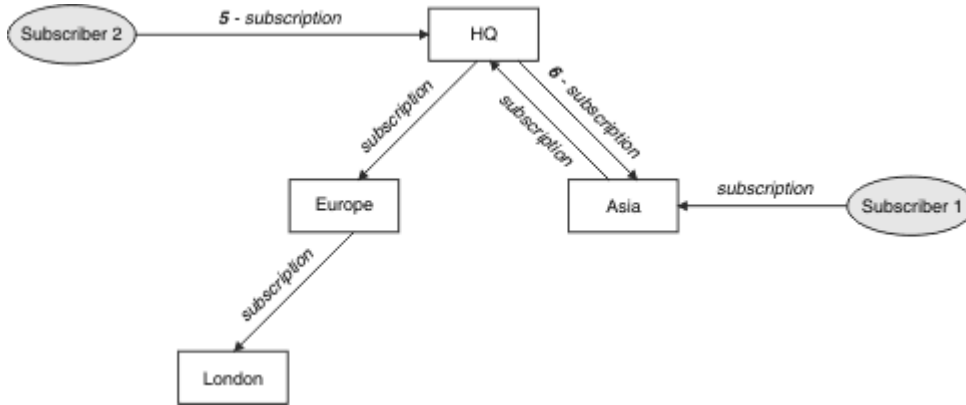


그림 21. 다중 구독

발행자는 47 페이지의 그림 21의 경우와 동일한 토픽에 대한 발행물을 Europe 큐 관리자 (7)로 송신합니다. HQ에서 Europe으로 향하는 이 토픽에 대한 구독이 있으므로, 발행물이 HQ 큐 관리자 (8)로 전달됩니다. 하지만 London에서 Europe으로 향하는 구독이 없으므로(Europe에서 London으로 향하는 구독만 있음), 발행물이 London 큐 관리자로 전달되지 않습니다. HQ 큐 관리자가 직접 발행물을 구독자 2 및 Asia 큐 관리자 (9)로 송신합니다. 발행물은 Asia (10)에서 구독자 1로 전달됩니다.

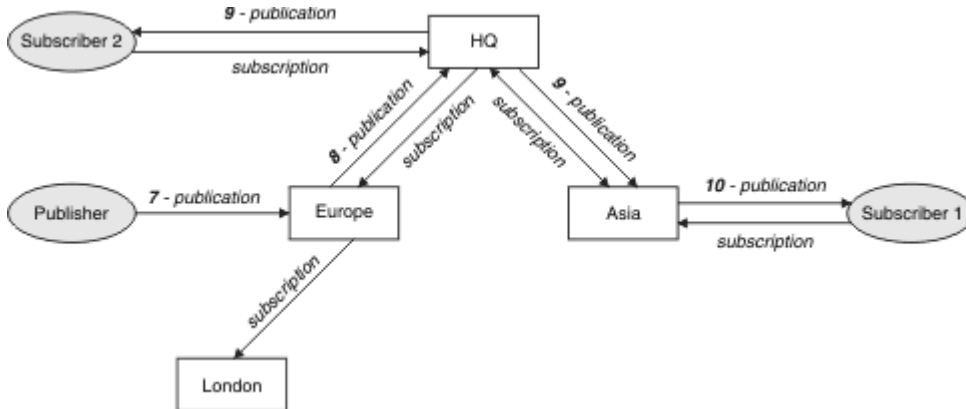


그림 22. 큐 관리자 네트워크를 통한 발행물 전파

큐 관리자는 다른 큐 관리자에 발행물이나 구독을 보낼 때 자체 사용자 ID를 메시지에 설정합니다. 발행/구독 계층을 사용하고 수신 채널이 메시지에서 사용자 ID의 권한으로 메시지를 넣도록 설정하는 경우, 송신 큐 관리자의 사용자 ID 권한을 부여해야 합니다. 97 페이지의 『큐 관리자 계층으로 기본 사용자 ID 사용』의 내용을 참조하십시오. 발행/구독 클러스터를 사용하는 경우, 권한이 클러스터에서 핸들링됩니다.

발행/구독 큐 관리자의 상호 연결된 네이처로 인해 프록시 구독이 네트워크의 모든 노드 주위에 전파되는 데 시간이 걸립니다. 원격 발행물은 바로 구독하지 않아도 됩니다. 48 페이지의 『라우팅 메커니즘에 대한 세부사항』에 설명된 대로 값이 FORCE인 Topic 속성 PROXYSUB를 사용하여 구독 지연을 없앨 수 있습니다.

프록시 구독을 직접 연결된 각 큐 관리자에 대한 적절한 전송 큐에 넣으면 구독 조작이 완료됩니다. 구독 조작은 나머지 토픽로 프록시 구독이 전파되기를 기다리지 않습니다.

프록시 구독은 이 구독을 작성한 큐 관리자 이름과 연관되어 있습니다. 계층의 큐 관리자에 동일한 이름이 있는 경우, 발행물이 도달할 수 없습니다. 이러한 문제점을 피하려면 큐 관리자가 WebSphere MQ 네트워크에 직접 또는 간접적으로 연결되는 경우 포인트-투-포인트 메시징과 같이 큐 관리자에 고유 이름을 제공하십시오.

분배된 발행/구독 네트워크 내에서는 발행물 및 구독 범위를 사용하여 발행물 및 구독 플로우를 제어하고 제한(해당하는 경우)할 수 있습니다.

프록시 구독 집계 및 발행물 집계

분배된 발행/구독 발행물 수 및 프로кси 구독 수를 집계하여 발행/구독 큐 관리자 사이에 전달되는 메시지의 양을 최소화합니다.

프록시 구독은 하나의 큐 관리자에 발행된 토픽에 대해 다른 큐 관리자가 작성하는 구독입니다. 명시적으로 프로кси 구독을 작성하지 않으면 큐 관리자가 사용자를 대신하여 작성합니다. 46 페이지의 『분배된 발행/구독 작동 방식』의 내용을 참조하십시오.

큐 관리자를 발행/구독 계층에 연결하거나 발행/구독 클러스터에 연결할 수 있습니다. 프로кси 구독은 연결된 큐 관리자 사이에서 전달됩니다. 프로кси 구독을 사용하면 한 큐 관리자에 연결된 발행자가 작성한 토픽에 대한 발행물을 다른 큐 관리자에 연결된 해당 토픽의 구독자가 받게 됩니다. 56 페이지의 『발행/구독 토폴로지』의 내용을 참조하십시오.

프록시 구독은 구독에서 구독하는 각 개별 토픽 문자열에 대해 큐 관리자 사이에서 플로우됩니다.

Topic 속성 **PUBSCOPE**와 **SUBSCOPE**를 사용하여 연결된 큐 관리자 간의 프로кси 구독 및 발행물의 플로우를 제한할 수 있습니다. 또한 **Topic** 속성 **WILDCARD**를 **BLOCK**으로 설정하여 와일드카드를 포함한 프로кси 구독의 플로우를 제한할 수도 있습니다. 50 페이지의 『와일드카드 규칙』의 내용을 참조하십시오.

프록시 구독은 구독의 생성에 비동기적으로 큐 관리자 사이에서 플로우됩니다. 구독할 토픽 상위 또는 토픽에서 **Topic** 속성 **PROXYSUB**를 **FORCE**로 설정하여 프로кси 구독을 연결된 모든 큐 관리자로 전파하기를 기다리는 시간을 줄일 수 있습니다. 48 페이지의 『라우팅 메커니즘에 대한 세부사항』의 내용을 참조하십시오.

프록시 구독 집계

프록시 구독은 중복 제거 시스템을 사용하여 집계됩니다. 해석된 특정 토픽 문자열의 경우 프로кси 구독은 수신된 프로кси 구독 또는 첫 번째 로컬 구독에서 전송됩니다. 동일한 토픽 문자열에 대한 후속 구독은 이 기존 프로кси 구독을 사용합니다.

마지막 로컬 구독 후에 프로кси 구독이 취소되거나 수신된 프로кси 구독이 취소됩니다.

개별 토픽 문자열을 수천 번 구독하는 발행/구독 토폴로지에서는 또는 해당 구독의 존재 여부가 급격하게 변하는 경우, 프로кси 구독 전파에 대한 오버헤드를 고려해야 합니다. 개별 프로кси 구독은 **FORCE**로 설정되는 토픽 속성 **PROXYSUB**의 사용을 통해 통합될 수 있습니다. 라우팅 메커니즘 및 클러스터 토픽 성능에 대한 자세한 정보는 48 페이지의 『라우팅 메커니즘에 대한 세부사항』의 내용을 참조하십시오.

발행 집계

큐 관리자에 동일한 토픽 문자열에 대한 구독이 둘 이상인 경우 해당 토픽 문자열과 일치하는 각 발행의 사본 하나만 발행/구독 토폴로지의 다른 큐 관리자에서 전송됩니다. 메시지가 도착하면 로컬 큐 관리자는 일치하는 각 구독에 메시지의 사본을 전달합니다.

프로кси 구독이 와일드카드를 포함하는 경우 단일 발행의 토픽 문자열과 둘 이상의 구독이 일치할 수 있습니다. 메시지가 하나의 연결된 큐 관리자에서 작성한 둘 이상의 프로кси 구독과 일치하는 큐 관리자에서 발행되는 경우 발행의 사본 하나만 리모트 큐 관리자로 전달되어 다중 프로кси 구독을 만족시킵니다.

라우팅 메커니즘에 대한 세부사항

모든 곳에 발행은 개별 프로кси 구독 전달에 대한 대체 라우팅 메커니즘입니다. 개별 프로кси 구독 전달은 토픽 문자열에서 일치하는 구독이 있는 발행물만 원격 메시징 서버로 송신됨을 의미합니다. 모든 곳에 발행 또는 브로드캐스트는 메시징 서버에 발행된 모든 발행물을 분산 발행/구독 네트워크의 다른 모든 메시징 서버에 전달하여 작동합니다. 그런 다음, 수신하는 메시징 서버는 지역 구독에 해당하는 발행물을 전달합니다.

각 메커니즘은 장점을 가지고 있지만 제한사항도 가지고 있습니다.

개별 프록시 구독 전달

이 메커니즘에서는 큐 관리자의 구독과 일치하는 발행물만 송신되므로 최소량의 큐 관리자 사이 발행 트래픽이 발생합니다.

그러나,

- 각 개별 토픽 문자열을 구독하면 결과적으로 발행/구독 토플로지의 다른 모든 큐 관리자로 프록시 구독이 송신됩니다. 작성 또는 삭제할 수천 개의 구독(예: 큐 관리자 재시작 이후의 모든 비지속 가능 구독)이 있는 경우 또는 구독 세트가 각각 다른 토픽 문자열로 빠르게 변경되는 경우, 이 메시징 오버헤드가 중요할 수 있습니다.
- 프록시 구독은 비동기 메시징을 사용하여 다른 큐 관리자로 플로우됩니다, 따라서 구독 작성과 다른 큐 관리자에 의한 프록시 구독 작성, 전달 및 처리 사이에서 지연됩니다. 해당 간격으로 이 큐 관리자에서 발행되는 메시지는 원격 구독으로 전달되지 않습니다.

모든 위치에서 발행

다음 메커니즘 사용:

- 빠른 구독 작성, 삭제 또는 변경으로 인해 네트워크 로드 및 처리가 증가하지 않는 시스템에서는 토픽 문자열당 프록시 구독 오버헤드가 없습니다.
- 발행물이 항상 모든 큐 관리자로 플로우되므로 구독 작성과 큐 관리자로의 발행물 플로우 사이에는 지연이 없습니다. 따라서 새로 작성된 원격 구독에 발행물이 전달되지 않는 장은 없습니다.

그러나,

- 모든 발행물이 발행/구독 토플로지의 모든 큐 관리자로 송신되면, 잠재적으로 발행물이 각 큐 관리자의 구독과 일치하지 않는 초과 네트워크 트래픽이 발생합니다.

클러스터 또는 계층에 있는 큐 관리자의 상당한 부분에서 발행물이 구독될 것이라고 예상되는 경우 또는 구독 변경의 빈도 때문에 프록시 구독 오버헤드가 너무 큰 경우에는 모든 곳에 발행 메커니즘을 사용할 수 있습니다. 이 작업 메소드는 일치하는 구독이 포함된 큐 관리자가 아니라 모든 큐 관리자에 발행물이 송신될 때 메시징 트래픽이 증가한 경우보다는 이러한 경우에 더욱 효과적일 수 있습니다.

상위 레벨 토픽 오브젝트에 대해 **PROXYSUB** 속성을 FORCE로 설정하여 IBM WebSphere MQ 분산 발행/구독 토플로지에서 모든 위치 메커니즘을 사용할 수 있습니다.

개별 프록시 구독을 사용 안함으로 설정하는 방법에 대한 자세한 정보는 [69 페이지의 『개별 프록시 구독 사용 안함』](#)의 내용을 참조하십시오.

강제 실행된 프록시 구독이 토플로지 전체에 전파될 때 새 구독은 지체하지 않고 즉시 연결되어 있는 다른 큐 관리자로부터 발행물을 수신합니다.

이러한 시스템을 구성할 때는 주의해야 합니다. **PROXYSUB**가 FORCE로 설정된 토픽 아래에 있는 토픽 오브젝트는 **PROXYSUB**가 FORCE로 설정된 노드와 다른 클러스터 또는 계층 스트림에 없어야 합니다. 마찬가지로 하위 토픽 오브젝트에서 해당 **WILDCARD** 속성을 BLOCK으로 설정할 수 없습니다. 두 경우 모두 발행된 메시지가 한 큐 관리자에서 다른 큐 관리자로 플로우되지 않습니다.

PROXYSUB를 FORCE로 설정한 경우에도 구독하는 각 개별 토픽 문자열에 대한 프록시 구독이 계속 전파됩니다. 구독의 수와 빈도가 시스템에 중요한 오버헤드를 야기할 만큼 충분히 높은 경우, 큐 관리자의 모든 토픽에 대해 해당 구독을 사용할 수 없도록 설정할 수 있습니다. 개별 프록시 구독을 사용 안함으로 설정하는 방법에 대한 자세한 정보는 [69 페이지의 『개별 프록시 구독 사용 안함』](#)의 내용을 참조하십시오.

멀티캐스트 및 구독 지연 시간

구독 지연 시간 및 **PROXYSUB(FORCE)** 옵션을 사용하면 프록시 구독을 유지보수할 수 있습니다.

예를 들어, 잠재적인 문제점으로 인해 모든 구독자의 연결이 끊긴 후 **QM_B**에서 **QM_A**로의 프록시 구독이 이루어지지 않을 수 있습니다. 큐 관리자에 대한 유니캐스트 연결이 종료되는 경우에도 멀티캐스트 트래픽이 계속되어야 하면 이러한 상황을 원치 않을 수 있습니다. WebSphere MQ에 대한 멀티캐스트는 새 구독자가 연결되는 경우 마지막 구독자가 종료할 때 구독이 이루어지지 않도록 각 프록시 구독에 몇 분의 지연 시간을 추가하여 단기간 동안 프록시 구독을 유지보수합니다.

또한 토픽에서 **PROXYSUB(FORCE)** 옵션을 사용하여 미해결 프록시 구독이 항상 미해결 상태인지 확인할 수 있습니다. 구독이 활성 상태인 동안 하나 이상의 구독자에게 큐에 플로우되는 메시지가 필요한지 확인해야 합니다.

PROXYSUB(FORCE)가 설정되면, 첫 번째 로컬 구독 또는 수신한 프록시 구독보다 먼저 프록시 구독을 송신할 수 있고 마지막 로컬 구독 또는 수신한 프록시 구독을 취소한 경우에도 이 프록시 구독은 취소되지 않습니다.

구독이 계속 작성되지 않으면 피어 투 피어 통신을 사용하여 메시지 전송이 계속되도록 보장할 수 있습니다. 자세한 정보는 [멀티캐스트에 대한 고가용성을 참조하십시오](#).

와일드카드 규칙

프록시 구독의 와일드카드는 토픽 와일드카드를 사용하도록 변환됩니다.

와일드카드의 구독을 수신하는 경우, 이는 문자이며 WebSphere MQ 버전 6.0에서 사용됩니다. 또한 토픽이 될 수도 있으며 WebSphere Message Broker 버전 6.0 및 WebSphere MQ 버전 7.0에서 사용됩니다.

- 문자 와일드카드는 *를 사용하여 임의의 문자(/ 포함)를 나타냅니다.
- 토픽 와일드카드는 #을 사용하여 / 문자 사이의 토픽 공간 부분을 나타냅니다.

WebSphere MQ 버전 7.0에서는, 모든 프록시 구독이 토픽 와일드카드를 사용하도록 변환됩니다. 문자 와일드카드가 있는 경우, 가장 가까운 / 뒤의 이 와일드카드를 # 문자로 바꿉니다. 예를 들어 /aaa/bbb/c*d 는 /aaa/bbb/#으로 변환됩니다. 이러한 변환으로 인해 리모트 큐 관리자는 명시적으로 구독하는 것보다 약간 많은 발행을 송신합니다. 추가 발행은 로컬 구독자로 발행을 전달할 때 로컬 큐 관리자로 필터링됩니다.

WILDCARD 특성으로 와일드카드 사용 제어

MQSC **Topic WILDCARD** 특성 또는 동등한 PCF 토픽 WildcardOperation 특성을 사용하여 와일드카드 토픽 문자열 이름을 사용하는 구독자 애플리케이션으로 발행물을 전달하도록 제어하십시오. WILDCARD 특성은 다음과 같은 두 가지 가능한 값을 보유할 수 있습니다.

WILDCARD

이 토픽에 관한 와일드카드 구독의 동작입니다.

PASSTHRU

이 토픽 오브젝트의 토픽 문자열보다 덜 특정한 와일드카드 토픽에 대한 구독이 이 토픽 및 이 토픽보다 더욱 특정한 토픽 문자열에 대한 발행물을 수신합니다.

BLOCK

이 토픽 오브젝트의 토픽 문자열보다 덜 특정한 와일드카드 토픽에 대한 구독이 이 토픽 또는 이 토픽보다 더욱 특정한 토픽 문자열에 대한 발행물을 수신하지 않습니다.

이 속성의 값은 구독이 정의될 때 사용됩니다. 이 속성을 대체할 경우 기존 구독에 포함된 토픽 세트는 수정의 영향을 받지 않습니다. 이 시나리오는 토픽 오브젝트를 작성 또는 삭제할 때 토픽로지가 변경된 경우에도 적용됩니다. WILDCARD 속성을 수정한 후에 작성된 구독과 일치하는 토픽 세트가 수정된 토픽로지를 사용하여 작성됩니다. 일치하는 토픽 세트를 강제로 기존 구독에 대해 재평가하려는 경우 큐 관리자를 재시작해야 합니다.

예제, 39 페이지의 『예제: Sport 발행/구독 클러스터 작성』에서는 35 페이지의 그림 10에 표시된 토픽 트리 구조를 작성하는 단계를 수행할 수 있습니다.

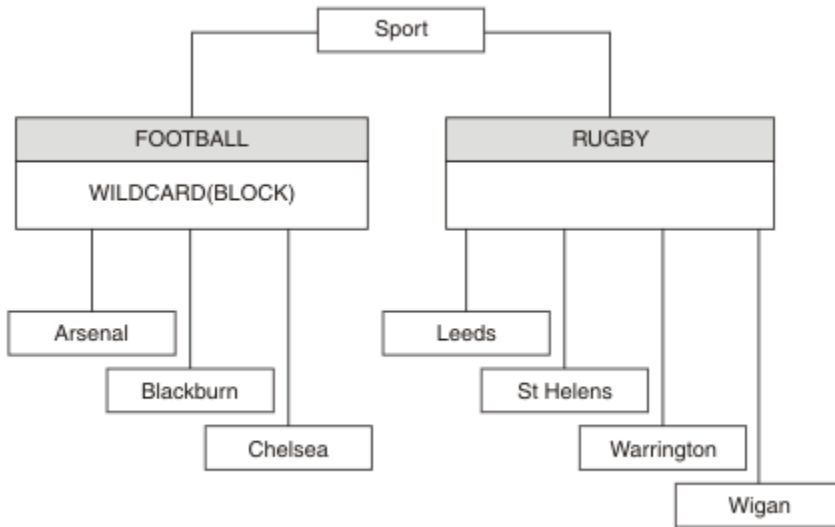


그림 23. WILDCARD 특성, BLOCK을 사용하는 토픽 트리

와일드카드 토픽 문자열 #을 사용하는 구독자는 Sport 토픽 및 Sport/Rugby 하위 트리에 대한 모든 발행을 수신합니다. 구독자는 Sport/Football 하위 트리에 대한 발행을 수신하지 않습니다. Sport/Football 토픽의 WILDCARD 특성 값은 BLOCK입니다.

PASSTHRU는 기본 설정입니다. WILDCARD 특성 값 PASSTHRU 을 Sport 트리의 노드로 설정할 수 있습니다. 노드에 WILDCARD 특성 값 BLOCK이 없으면 PASSTHRU를 설정해도 Sports 트리의 노드에서 구독자가 관찰하는 작동은 대체되지 않습니다.

예제에서 구독을 작성하여 와일드카드 설정이 전달되는 발행에 미치는 영향을 확인하십시오(40 페이지의 그림 14 참조). 일부 발행을 작성하려면 41 페이지의 그림 17에서 발행 명령을 실행하십시오.

```
pub QMA
```

그림 24. QMA에 발행

결과는 36 페이지의 표 3에 표시됩니다. WILDCARD 특성 값 BLOCK을 설정할 때 와일드카드를 포함하는 구독이 와일드카드 범위 내 토픽에 대한 발행을 수신하지 않도록 하는 방법에 주의하십시오.

표 6. QMA에서 수신된 발행			
구독	토픽 문자열	수신된 발행	참고
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Sports/Football에서 WILDCARD(BLOCK)으로 차단된 Football 하위 트리에 대한 모든 발행
SARSENAL	Sports/#/Arsenal	-	Sports/Football의 WILDCARD(BLOCK)은 Arsenal에서 와일드카드 서브스크립션을 방지합니다.
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby의 기본 WILDCARD는 Leeds에서 와일드카드 구독을 금지하지 않습니다.

참고:

구독에 WILDCARD 특성 값 BLOCK가 있는 토픽 오브젝트와 일치하는 와일드카드가 있다고 가정합니다. 또한 구독이 일치하는 와일드카드 오른쪽에 토픽 문자열을 포함하면 구독은 발행을 수신하지 않습니다. 차단되지 않은

발행 세트는 차단된 와일드카드 상위에 해당하는 토픽에 대한 발행입니다. BLOCK 특성 값이 설정된 토픽의 하위에 해당하는 토픽에 대한 발행은 와일드카드로 차단됩니다. 따라서 와일드카드 오른쪽에 토픽을 포함하는 구독 토픽 문자열은 일치하는 발행을 수신하지 않습니다.

WILDCARD 특성 값을 BLOCK으로 설정해도 와일드카드를 포함하는 토픽 문자열을 사용하여 구독할 수 없음을 의미하지는 않습니다. 이러한 구독은 정상적입니다. 구독이 WILDCARD 특성 값 BLOCK이 설정된 토픽 오브젝트를 포함하는 토픽과 일치하는 명시적 토픽을 보유합니다. 이는 WILDCARD 특성 값 BLOCK이 설정된 토픽의 상위 또는 하위에 해당하는 토픽에 대한 와일드카드를 사용합니다. 35 페이지의 그림 10의 예제에서 Sports/Football/#과 같은 구독은 발행을 수신할 수 있습니다.

와일드카드 및 클러스터 토픽

클러스터 토픽 정의는 클러스터의 모든 큐 관리자로 전파됩니다. 클러스터의 한 큐 관리자에서 클러스터 토픽에 대한 구독으로 인해 큐 관리자가 프록시 구독을 작성할 수 있습니다. 프록시 구독은 클러스터의 다른 모든 큐 관리자에서 작성됩니다. 클러스터 토픽과 결합해 와일드카드를 포함하는 토픽 문자열을 사용하는 구독은 작동을 예상하기 어려울 수 있습니다. 작동은 다음 예제에서 설명됩니다.

In the cluster set up for the example, 39 페이지의 『예제: Sport 발행/구독 클러스터 작성』, QMB has the same set of subscriptions as QMA, yet QMB received no publications after the publisher published to QMA, see 36 페이지의 그림 11. Sports/Football 및 Sports/Rugby 주제는 클러스터 토픽이지만 fullsubs.tst에 정의된 등록은 클러스터 토픽을 참조하지 않습니다. QMB에서 QMA로 프록시 구독이 전파되지 않습니다. 프록시 구독이 없으면 QMA에 대한 발행물은 QMB로 전달되지 않습니다.

Sports/Rugby의 경우, 일부 구독(예: Sports/#/Leeds)은 클러스터 토픽을 참조하는 것처럼 보입니다. Sports/#/Leeds 구독은 실제로 토픽 오브젝트 SYSTEM.BASE.TOPIC으로 해석됩니다.

Sports/#/Leeds와 같이 구독에서 참조하는 토픽 오브젝트를 해석하는 규칙은 다음과 같습니다. 토픽 문자열을 첫 번째 와일드카드까지 자릅니다. 토픽 문자열을 왼쪽부터 스캔하여 연관된 관리 토픽 오브젝트가 있는 첫 번째 토픽을 찾습니다. 이 토픽 오브젝트는 클러스터 이름을 지정하거나 로컬 토픽 오브젝트를 정의할 수 있습니다. Sports/#/Leeds예제에서 잘림 후의 토픽 문자열은 토픽 오브젝트가 없는 Sports이고, Sports/#/Leeds는 로컬 토픽 오브젝트인 SYSTEM.BASE.TOPIC에서 상속합니다.

클러스터 토픽을 구독하여 와일드카드 전파 방식을 변경하는 방법을 확인하려면 배치 스크립트, [upsubs.bat](#)를 실행하십시오. 스크립트는 구독 큐를 지우고 fullsubs.tst에서 클러스터 토픽 구독을 추가합니다. [puba.bat](#)를 다시 실행하여 발행 배치를 작성하십시오(36 페이지의 그림 11 참조).

37 페이지의 표 4에서는 발행이 발행된, 동일한 큐 관리자에 2개의 새 구독을 추가하는 결과를 보여줍니다. 결과는 예상한 대로입니다. 새 구독은 각각 하나의 발행을 수신하고 다른 구독에서 수신하는 발행 수는 변경되지 않습니다. 다른 클러스터 큐 관리자에서 예상치 못한 결과가 발생합니다(37 페이지의 표 5 참조).

구독	토픽 문자열	수신된 발행	참고
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Sports/Football에서 WILDCARD(BLOCK)으로 차단된 Football 하위 트리에 대한 모든 발행
SARSENAL	Sports/#/Arsenal	-	Sports/Football의 WILDCARD(BLOCK)은 Arsenal에서 와일드카드 서브스크립션을 방지합니다.
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby의 기본 WILDCARD는 Leeds에서 와일드카드 구독을 금지하지 않습니다.
FARSENAL	Sports/Football/ Arsenal	Sports/Football/ Arsenal	Arsenal은 구독이 와일드카드를 포함하지 않으므로 발행을 수신합니다.

표 7. QMA에서 수신된 발행 (계속)			
구독	토픽 문자열	수신된 발행	참고
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds는 모든 이벤트에서 발행을 수신합니다.

37 페이지의 표 5에서는 QMB에서 2개의 새 구독을 추가하고 QMA에서 발행하는 결과를 보여줍니다. QMB가 이러한 2개의 새 구독 없이 발행을 수신하지 않는다는 점을 상기하십시오. 예상대로 두 개의 새 구독은 발행물을 수신합니다. Sports/FootBall 및 Sports/Rugby 는 둘 다 클러스터 토픽이기 때문입니다. QMB forwarded proxy subscriptions for Sports/Football/Arsenal and Sports/Rugby/Leeds to QMA, which then sent the publications to QMB.

예상치 않은 결과는 이전에 발행물을 수신하지 않았던 두 개의 구독 Sports/# 및 Sports/#/Leeds 가 이제 발행을 수신하는 것입니다. 이는 다른 구독의 QMB로 전달된 Sports/Football/Arsenal 및 Sports/Rugby/Leeds 발행물이 QMB에 첨부된 구독자에서 사용 가능하기 때문입니다. 결과적으로 로컬 토픽 Sports/# 및 Sports/#/Leeds에 대한 구독은 Sports/Rugby/Leeds 발행을 수신합니다. Sports/#/Arsenal은 계속해서 발행을 수신하지 않습니다. Sports/Football에서 해당 WILDCARD 특성 값이 BLOCK으로 설정되었기 때문입니다.

표 8. QMB에서 수신된 발행			
구독	토픽 문자열	수신된 발행	참고
SPORTS	Sports/#	Sports/Rugby/Leeds	Sports/Football에서 WILDCARD (BLOCK)에 의해 차단된 Football 하위 트리에 대한 모든 발행
SARSENAL	Sports/#/Arsenal	-	Sports/Football의 WILDCARD (BLOCK)은 Arsenal에서 와일드카드 서브스크립션을 방지합니다.
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby의 기본 WILDCARD는 Leeds의 와일드카드 구독을 방해하지 않음
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal은 구독이 와일드카드를 포함하지 않으므로 발행을 수신합니다.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds는 모든 이벤트에서 발행을 수신합니다.

대부분의 애플리케이션에서는 한 구독이 다른 구독의 작동에 영향을 주는 것은 바람직하지 않습니다. 값이 BLOCK인 WILDCARD 특성의 한 가지 중요한 사용법은 균일하게 작동하는 와일드카드를 포함하는 동일한 토픽 문자열에 대한 구독을 작성하는 것입니다. 발행자와 동일한 큐 관리자 또는 다른 큐 관리자의 구독인지에 상관없이 구독 결과는 동일합니다.

와일드카드 및 스트림

WebSphere MQ 버전 6 스트림은 WebSphere MQ 버전 7에 의해 토픽으로 맵핑됩니다. 41 페이지의 『스트림 및 토픽』의 내용을 참조하십시오. 버전 7의 **strmqbrk**로 수행되는 기본 맵핑에서 스트림 Sports의 모든 토픽이 토픽 Sports로 맵핑됩니다. 스트림 Business의 모든 주제는 Business주제에 맵핑됩니다.

Sports 스트림의 *에 대한 WebSphere MQ 버전 6의 구독은 Sports 트리의 모든 발행물을 수신하며 Business 트리의 발행물은 수신하지 않습니다. 버전 7의 동일한 구독은 Sports 트리의 모든 발행물 및 Business 트리의 모든 발행물을 수신합니다. 이 작동을 차단하려면 스트림이 버전 7로 마이그레이션될 때 **strmqbrk**이 WILDCARD 특성을 설정합니다. 이 명령은 스트림에서 마이그레이션된 최상위 레벨 토픽 각각에

대해 이 특성을 BLOCK 값으로 설정합니다. Sports 및 Business의 WILDCARD 특성은 Sports 및 Business라는 버전 6 스트림에서 변환하여 BLOCK 값으로 설정됩니다.

발행/구독 API에 기록되는 새 응용프로그램의 경우, *에 대한 등록은 발행을 수신하지 않습니다. 모든 Sports 발행물을 수신하려면 Sports/* 또는 Sports/#을 구독해야 하며 Business 발행물의 경우에도 이와 유사합니다.

발행/구독 브로커가 WebSphere MQ 버전 7로 이주될 때 기존 큐된 발행/구독 애플리케이션의 동작은 변경되지 않습니다. **Publish, Register Publisher** 또는 **Subscriber** 명령의 **StreamName** 특성은 스트림이 마이그레이션된 토픽의 이름으로 매핑됩니다.

와일드카드 및 구독 지점

WebSphere Message Broker 구독 지점이 WebSphere MQ 버전 7에 의해 토픽으로 매핑됩니다. [43 페이지의 『구독 지점 및 토픽』](#)의 내용을 참조하십시오. 버전 7의 **migmqbrk**로 수행되는 기본 매핑에서, 구독 지점 Sports의 모든 토픽이 Sports 토픽으로 매핑됩니다. All the topics in the subscription point Business would be mapped to the topic Business.

Sports 구독 지점의 *에 대한 WebSphere Message Broker 버전 6의 구독은 Sports 트리의 모든 발행물을 수신하며 Business 트리의 발행물은 수신하지 않습니다. 버전 7에서 동일한 구독은 Sports 트리의 모든 발행물과 Business 트리의 모든 발행물을 수신합니다. 이 동작을 차단하려면 구독 지점이 버전 7로 마이그레이션될 때 **migmqbrk**가 WILDCARD 특성을 설정합니다. 이 명령은 구독 지점에서 마이그레이션된 최상위 레벨 토픽 각각에 대해 이 특성을 BLOCK 값으로 설정합니다. Sports 및 Business의 WILDCARD 특성은 Sports 및 Business라는 WebSphere Message Broker 구독 지점에서 변환하여 BLOCK 값으로 설정됩니다.

발행/구독 API에 기록된 새 응용프로그램의 경우, 이주의 영향은 *에 대한 등록이 발행을 수신하지 않는다는 것입니다. 모든 Sports 발행물을 수신하려면 Sports/* 또는 Sports/#을 구독해야 하며 Business 발행물의 경우에도 이와 유사합니다.

발행/구독 브로커가 WebSphere MQ 버전 7로 이주될 때 기존 큐된 발행/구독 애플리케이션의 동작은 변경되지 않습니다. **Publish, Register Publisher** 또는 **Subscriber** 명령의 **SubPoint** 특성은 등록이 마이그레이션된 토픽의 이름으로 매핑됩니다.

예제: Sport 발행/구독 클러스터 작성

다음 단계에서는 두 개의 전체 저장소 (CL1A 및 CL1B)와 두 개의 부분 저장소 (QMA 및 QMB)가 있는 클러스터 (CL1)를 작성합니다. 전체 저장소는 클러스터 정의만 보유하는 데 사용됩니다. QMA는 클러스터 토픽 호스트를 지정합니다. 지속 가능 구독은 QMA 및 QMB 모두에서 정의됩니다.

참고: Windows에 대한 예가 코드화됩니다. 다른 플랫폼에서 예제를 구성하고 테스트하도록 [qmgrs.bat](#) 작성 및 [pub.bat](#) 작성을 다시 코딩해야 합니다.

1. 스크립트 파일을 작성하십시오.
 - a. [topics.tst](#) 작성
 - b. [wildsubs.tst](#) 작성
 - c. [fullsubs.tst](#) 작성
 - d. [qmgrs.bat](#) 작성
 - e. [pub.bat](#) 작성
2. [qmgrs.bat](#) 작성을 실행하여 구성을 작성하십시오.

```
qmgrs
```

35 페이지의 [그림 10](#)에서 토픽을 작성하십시오. 그림 5의 스크립트는 클러스터 토픽 Sports/Football 및 Sports/Rugby를 작성합니다.

참고: REPLACE 옵션은 주제의 TOPICSTR 특성을 바꾸지 않습니다. TOPICSTR은(는) 다른 토픽 트리를 테스트하는 예제에서 사용되는 특성입니다. 토픽을 변경하려면 먼저 토픽을 삭제하십시오.

```

DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports')      TOPICSTR('Sports')
DEFINE TOPIC ('Football')   TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal')    TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn')  TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea')    TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby')      TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds')      TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan')      TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')

```

그림 25. 토픽 삭제 및 작성: *topics.tst*

참고: REPLACE가 토픽 문자열을 바꾸지 않으므로 토픽을 삭제합니다.

와일드카드를 사용하여 구독을 작성하십시오. 35 페이지의 그림 10에서 토픽 오브젝트를 포함하는 토픽에 대응하는 와일드카드. 각 구독에 대한 큐를 작성하십시오. 스크립트를 실행하거나 다시 실행하면 큐가 지워지고 구독이 삭제됩니다.

참고: REPLACE 옵션은 구독의 TOPICOBJ 또는 TOPICSTR 특성을 바꾸지 않습니다. TOPICOBJ 또는 TOPICSTR 는 다른 등록을 테스트하는 예제에서 사용되는 특성입니다. 변경하려면 먼저 구독을 삭제하십시오.

```

DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)

```

그림 26. 와일드카드 구독 작성: *wildsubs.tst*

클러스터 토픽 오브젝트를 참조하는 구독을 작성하십시오.

참고:

구분 기호, /는 TOPICOBJ에서 참조하는 토픽 문자열과 TOPICSTR에서 정의하는 토픽 문자열 사이에 자동으로 삽입됩니다.

DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) 정의는 동일한 등록을 작성합니다. TOPICOBJ는 이미 정의한 토픽 문자열을 참조하는 빠른 방법으로 사용됩니다. 작성되면 구독은 더 이상 토픽 오브젝트를 참조하지 않습니다.

```

DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)

```

그림 27. 구독 삭제 및 삭제: *fullsubs.tst*

2개의 저장소를 포함하는 클러스터를 작성하십시오. 발행 및 구독을 위해 2개의 부분 저장소를 작성하십시오. 모두 삭제하고 다시 시작하도록 스크립트를 다시 실행하십시오. 또한 스크립트는 토픽 계층 및 초기 와일드카드 구독을 작성합니다.

참고:

다른 플랫폼에 비슷한 스크립트를 작성하거나 모두 명령을 입력하십시오. 스크립트를 사용하면 모두 삭제하고 동일한 구성으로 다시 빠르게 시작할 수 있습니다.

```
@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof
```

그림 28. 큐 관리자 작성: *qmgrs.bat*

클러스터 토픽에 구독을 추가하여 구성을 업데이트하십시오.

```
@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst
```

그림 29. 구독 업데이트: *upsubs.bat*

큐 관리자에서 매개변수로 *pub.bat*를 실행하여 발행 토픽 문자열을 포함하는 메시지를 발행하십시오. *Pub.bat*는 샘플 프로그램 *amqspub*를 사용합니다.

```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

그림 30. 발행: *pub.bat*

발행/구독 토플로지

발행/구독 토플로지는 큐 관리자와 발행/구독 애플리케이션을 지원하는 큐 관리자 간의 연결로 구성됩니다.

발행/구독 애플리케이션은 서로 연결된 큐 관리자의 네트워크로 구성할 수 있습니다. 큐 관리자는 모두 동일한 물리적 시스템에 있거나 몇몇 물리적 시스템에 분산되어 있을 수 있습니다. 큐 관리자를 서로 연결하면 네트워크에 있는 큐 관리자를 통해 애플리케이션이 발행물을 수신할 수 있습니다.

이 애플리케이션의 이점은 다음과 같습니다.

- 클라이언트 애플리케이션이 멀리 있는 큐 관리자가 아닌 가까이 있는 큐 관리자와 통신할 수 있으며 그로 인해 응답 시간이 더 빨라질 수 있습니다.
- 둘 이상의 큐 관리자를 사용하면 보다 많은 구독자를 지원할 수 있습니다.

서로 다른 두 가지 방법(클러스터와 계층)으로 발행/구독 메시지를 수행하는 큐 관리자를 배열할 수 있습니다. 단순 클러스터 및 단순 계층의 예제는 [57 페이지의 그림 32](#) 및 [57 페이지의 그림 31](#)을 참조하십시오. 이러한 두 토폴로지에 대한 자세한 정보를 보고 사용자에게 가장 적합한 내용을 찾으려면, 제품 문서의 이 절에 있는 정보를 참조하십시오.

계층에서 클러스터를 조인하면 두 토폴로지를 결합하여 사용할 수 있습니다.

Cluster

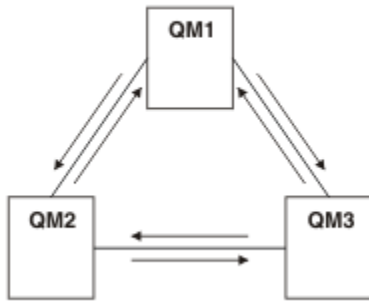


그림 31. 간단한 발행/구독 클러스터

Hierarchy

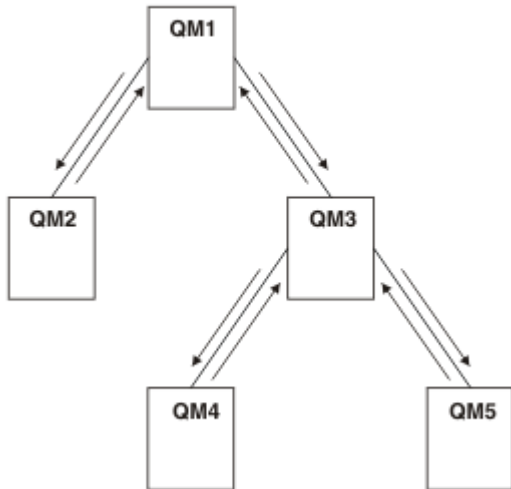


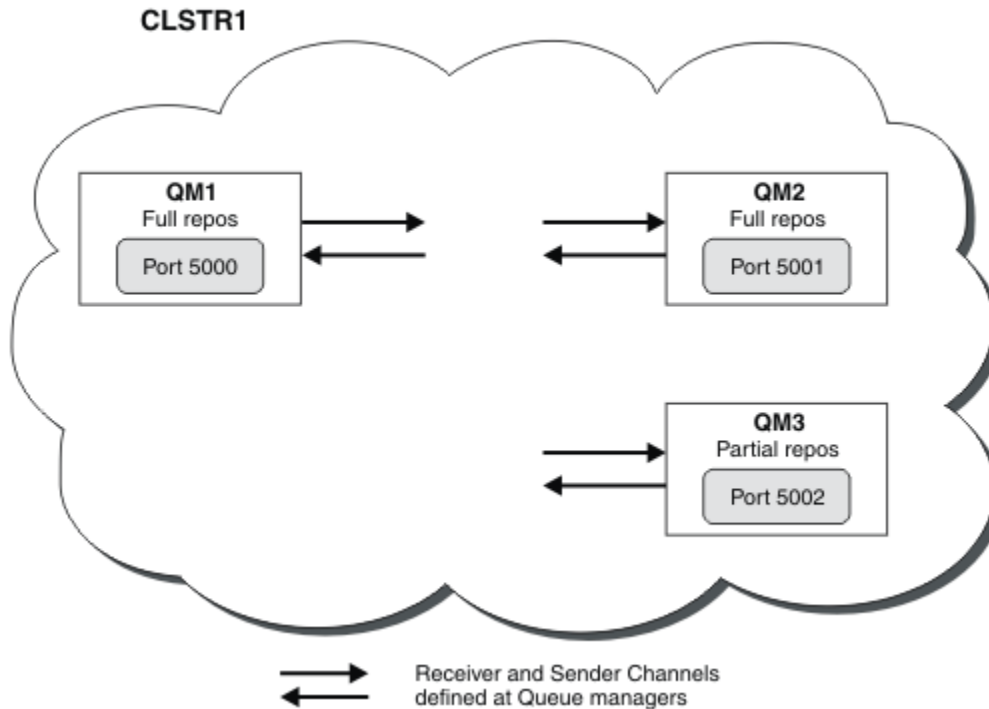
그림 32. 간단한 발행/구독 계층

발행/구독 클러스터 설정: 시나리오 1

두 개의 큐 관리자를 전체 저장소로 클러스터에 추가하고 이들 간의 채널을 정의합니다.

이 태스크 정보

다음 다이어그램에는 세 개의 큐 관리자 (QM1, QM2 및 QM3)가 있습니다.



QM1 및 QM2 는 클러스터의 전체 저장소이고 QM3 는 부분 저장소입니다.

시나리오 1은 QM1 및 QM2를 클러스터 DEMO에 전체 저장소로 추가합니다.

시나리오 2는 QM3를 클러스터 DEMO에 부분 저장소로 추가합니다.

이 태스크에는 최소 한 개의 명령 창이 필요합니다.

프로시저

1. QM1 및 QM2를 DEMO 클러스터의 전체 저장소로 설정하십시오.

```
alter QMGR REPOS(DEMO)
```

2. QM1의 리스너를 정의하고 시작하십시오.

```
define listener(QM1_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5000)
start listener(QM1_LS)
```

3. QM2의 리스너를 정의하고 시작하십시오.

```
define listener(QM2_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5001)
start listener(QM2_LS)
```

4. QM1의 수신자 채널을 정의하십시오.

```
DEFINE CHANNEL(DEMO.QM1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
CLUSTER(DEMO) DESCR('TCP Cluster-receiver channel for queue manager QM1')
```

5. QM1에서 QM2로 송신자 채널을 정의하십시오.

```
DEFINE CHANNEL(DEMO.QM2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')
CLUSTER(DEMO) DESCR('TCP Cluster-sender channel from QM1 to queue manager QM2')
```

6. QM2의 수신자 채널을 정의하십시오.


```
DEFINE CHANNEL(DEMO.QM2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')
CLUSTER(DEMO) DESCR('TCP Cluster-receiver channel for queue manager QM2')
```

7. QM2 에서 QM1로 송신자 채널을 정의하십시오.

```
DEFINE CHANNEL(DEMO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
CLUSTER(DEMO) DESCR('TCP Cluster-sender channel from qm2 to qm1')
```

8. QM1에서 클러스터 토픽 scores 을 정의하십시오.

```
define topic(scores) TOPICSTR(/football) CLUSTER(DEMO)
```

9. 다음 명령으로 설정을 확인하십시오.

```
display topic(scores) type(all) clusinfo
display clusqmgr(*)
display chstatus(*)
```

10. 두 개의 명령 창을 사용하여 설정을 테스트하십시오.

a. 첫 번째 명령 창에 다음 명령을 입력하십시오.

```
/opt/mqm/samp/bin/amqspub /FOOTBALL/scores QM1
```

b. 두 번째 명령 창에 다음 명령을 입력하십시오.

```
/opt/mqm/samp/bin/amqssub /FOOTBALL/scores QM2
```

관련 태스크

[WebSphere MQ 클러스터 관리](#)

[새 클러스터 설정](#)

발행/구독 클러스터 설정: 시나리오 2

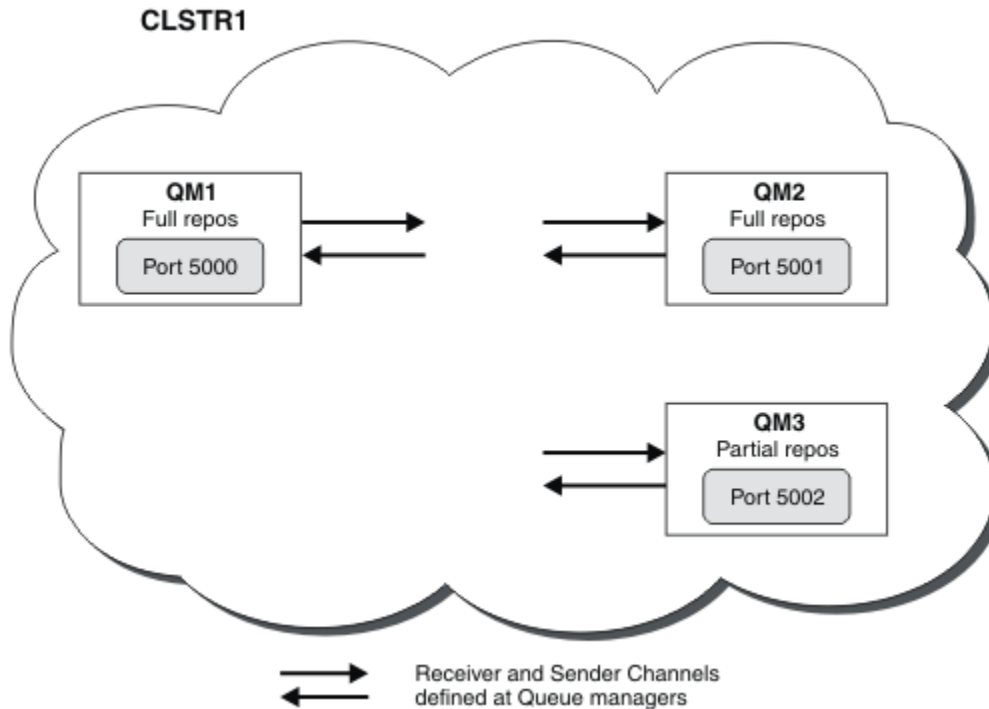
세 번째 큐 관리자를 클러스터에 부분 저장소로 추가합니다.

시작하기 전에

이 태스크를 완료하기 전에 [57 페이지의 『발행/구독 클러스터 설정: 시나리오 1』](#)의 태스크를 완료했어야 합니다.

이 태스크 정보

다음 다이어그램에는 QM1, QM2, QM3의 세 가지 큐 관리자가 있습니다.



QM1 및 QM2는 클러스터의 전체 저장소이고 QM3는 부분 저장소입니다.

시나리오 1은 QM1 및 QM2를 클러스터 DEMO에 전체 저장소로 추가합니다.

시나리오 2는 QM3를 클러스터 DEMO에 부분 저장소로 추가합니다.

이 태스크에는 최소 1개의 명령 창이 필요합니다.

프로시저

1. QM3의 리스너를 정의하고 시작하십시오.

```
define listener(QM3_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5002)
start listener(QM3_LS)
```

2. QM3의 수신자 채널을 정의하십시오.

```
DEFINE CHANNEL (DEMO.QM3) CHLTYPE (CLUSRCVR) TRPTYPE (TCP) CONNAME ('$HOSTNAME(5002)') CLUSTER
(DEMO) DESCR ('TCP Cluster-receiver channel for queue manager QM3')
```

3. QM3에서 QM1으로의 송신자 채널을 정의하십시오.

```
DEFINE CHANNEL (DEMO.QM1) CHLTYPE (CLUSSDR) TRPTYPE (TCP) CONNAME ('$HOSTNAME(5000)')
CLUSTER (DEMO) DESCR ('TCP Cluster-sender channel from qm3 to qm1')
```

4. 다음 명령으로 설정을 확인하십시오.

```
display topic(scores) type(all) clusinfo
display clusqmgr(*)
display chstatus(*)
```

5. 2개의 명령 창을 사용하여 설정을 테스트하십시오.

- a. 첫 번째 명령 창에 다음 명령을 입력하십시오.

```
/opt/mqm/samp/bin/amqspub /FOOTBALL/scores QM2
```

b. 두 번째 명령 창에 다음 명령을 입력하십시오.

```
/opt/mqm/samp/bin/amqssub /FOOTBALL/scores QM3
```

발행/구독 클러스터

발행/구독 클러스터는 발행물이 자동으로 발행 애플리케이션에서 클러스터의 임의 큐 관리자에 있는 구독으로 이동하는 상호 연결된 큐 관리자의 표준 IBM WebSphere MQ 클러스터입니다.

메시징 발행/구독에 사용되는 클러스터는 표준 IBM WebSphere MQ 클러스터와 다르지 않습니다. 마찬가지로 발행/구독 클러스터 내 큐 관리자는 실제로 별도의 컴퓨터에 존재할 수 있으며, 큐 관리자의 각 쌍은 필요한 경우 클러스터 채널에 의해 자동으로 함께 연결됩니다. IBM WebSphere MQ 클러스터를 계획하고 구성하는 방법에 대한 정보는 [클러스터 작업 방법](#)을 참조하십시오.

클러스터에서 큐 관리자에 의해 구성된 토픽에 대해 **CLUSTER** 속성을 설정하면 클러스터된 토픽 오브젝트가 정의될 때 발행/구독 클러스터가 작성됩니다. 토픽 정의가 클러스터의 모든 구성원으로 전파됩니다. 클러스터에 있는 임의의 큐 관리자에서 토픽과 토픽 트리에서 그 아래에 있는 어떤 토픽 문자열에 대해서도 발행 및 구독할 수 있습니다. 발행물은 클러스터의 다른 큐 관리자에 연결된 구독자로 자동 전파됩니다.

클러스터된 토픽 오브젝트에 해당하지 않는 토픽 문자열로 작업하면 클러스터되지 않은 발행/구독 활동도 발행/구독 클러스터에서 발생할 수 있습니다. 이 배열은 전체 계층에 걸쳐 모든 구독이 전파되는 발행/구독 계층과는 다릅니다. 두 경우 모두, 구독 및 발행 범위를 사용하여 더욱 세밀하게 제어할 수 있습니다.

발행/구독 토픽로지의 클러스터를 사용하면 다음과 같은 이점이 있습니다.

- 동일한 클러스터의 특정 큐 관리자에 대한 구독에 예정된 메시지는 해당 큐 관리자에게 직접 전송되므로 중간 큐 관리자를 통과할 필요가 없습니다. 이렇게 되면 성능이 향상되고 계층 구조 토픽로지와 비교해서 상호 큐 관리자 발행/구독 트래픽이 최적화됩니다.
- 모든 큐 관리자가 서로 직접 연결되어 있기 때문에, 이 토픽로지에는 단일 실패 지점이 없습니다. 한 큐 관리자가 사용 불가능한 경우 클러스터의 다른 큐 관리자에 대한 구독이 사용 가능한 큐 관리자에서 여전히 메시지를 수신할 수 있습니다.
- 여러 개의 분리된 클러스터가 있는 시스템에서(예: 클러스터가 지리적으로 분산되어 있는 시스템) 클러스터를 클러스터 계층으로 연결할 수 있습니다. 각 클러스터에서 단일 큐 관리자를 조인하는 방법으로 이 연결이 작성되며, 네트워크를 통해 발행 및 구독을 플로우할 수 있습니다(89 페이지의 『다중 클러스터의 토픽 공간 결합』 참조). 또한 하나의 클러스터에서 또 다른 클러스터로 플로우되는 발행물을 제어할 수 있습니다. 91 페이지의 『다중 클러스터의 토픽 공간 결합 및 분리』의 내용을 참조하십시오.
- 구독 애플리케이션은 가장 가까운 큐 관리자에 연결되어 성능을 향상시킬 수 있습니다. 큐 관리자는 클러스터 내의 모든 큐 관리자에서 클라이언트의 구독 등록과 일치하는 모든 메시지를 수신합니다.

또한 이 큐 관리자에서 요청된 다른 서비스에 대한 클라이언트 애플리케이션의 성능도 향상됩니다. 클라이언트 애플리케이션은 발행/구독 메시징 및 포인트-투-포인트 메시징을 모두 사용할 수 있습니다.

- 클러스터에 더 많은 큐 관리자를 추가하여 워크로드를 공유하면 각각의 큐 관리자에 대한 클라이언트 및 구독의 수를 줄일 수 있습니다. 새 큐 관리자의 클라이언트로 발행물이 자동으로 분배됩니다. 일부 사용 패턴의 경우, 이 프로세스를 통해 발행/구독 클러스터 토픽로지의 확장성을 높일 수 있습니다.

발행/구독에서 클러스터를 사용할 때 고려할 사항:

- 발행/구독 클러스터의 모든 큐 관리자는 클러스터에 있는 다른 모든 큐 관리자를 자동으로 인식하게 됩니다. 이 프로세스는 어떤 큐 관리자에 관심이 있는 큐 관리자만 알려져 있는 포인트-투-포인트 클러스터에는 다릅니다.
- 클러스터된 토픽에 하나 이상의 구독을 호스트하는 발행/구독 클러스터의 큐 관리자는 클러스터에 있는 다른 모든 큐 관리자에 대한 클러스터 송신자 채널을 자동으로 작성합니다. 또한, 큐 관리자는 수신 큐 관리자가 임의의 클러스터된 토픽에 대해 메시지를 발행하지 않을 때도 그 각각의 구독에 관한 정보를 전송합니다.
- 클러스터된 토픽 하의 토픽 문자열에 대한 큐 관리자의 첫 번째 구독으로 인해 클러스터의 다른 모든 큐 관리자에게 메시지가 송신됩니다. 마찬가지로, 삭제할 토픽 문자열에 대한 마지막 구독도 메시지를 발생시킵니다. 클러스터된 토픽에서 개별 토픽 문자열이 더 많이 사용될수록, 큐 관리자 간 통신이 더 많이 이루어집니다.



경고:

이 토픽에서 앞서 나열한 이유 때문에, 대형 IBM WebSphere MQ 클러스터(즉, 다수의 큐 관리자를 포함한 클러스터)에 클러스터된 토픽을 도입하면 클러스터의 각 큐 관리자에 즉시 추가적인 로드 발생할 수

있고, 어떤 경우에는 성능 감소로 이어질 수 있습니다. 자세한 정보는 67 페이지의 『클러스터 토픽 성능』의 내용을 참조하십시오.

큐 관리자의 클러스터, 특히 기존 클러스터에 발행/구독을 도입할 때 이런 성능 감소를 수용할 수 있도록 세심하게 계획해야 합니다.

클러스터가 발행/구독의 성능 저하를 수용할 수 없는 것으로 알려진 경우에는 **PSCLUS** 매개변수를 사용하여 큐 관리자에서 클러스터된 발행/구독 기능을 사용하지 않을 수 있습니다. **PSCLUS** 매개변수는 기본적으로 실수 또는 클러스터된 토픽을 잘못 정의하여 행/구독 클러스터를 작성할 때 발생할 수 있는 심각한 문제점을 중지시킵니다. 이 기능의 비활성화에 대한 자세한 정보는 66 페이지의 『클러스터에서 클러스터된 발행/구독 금지』의 내용을 참조하십시오.

발행/구독 클러스터링: 우수 사례

이 주제에서는 IBM WebSphere MQ 발행/구독 클러스터 계획 및 관리에 대한 안내를 제공합니다. 이 정보는 테스트와 고객 피드백을 바탕으로 합니다.

다음 정보는 사용자가 IBM WebSphere MQ 클러스터, 발행/구독에 대한 기본 이해가 있고 45 페이지의 『분산 발행/구독』의 주제에 대해 숙지하고 있다고 가정합니다. 이 정보는 "모든 경우에 사용할 수 있는(one size fits all)" 솔루션으로 제공되는 것이 아니라 대신 공통 문제점에 대한 공통 접근 방식을 공유하기 위한 것입니다.

발행/구독 클러스터

클러스터를 사용하면, 필요한 경우 클러스터의 큐 관리자 사이에서 보편적인("any-to-any") 직접 연결성이 제공 됩니다. 클러스터가 포인트-투-포인트 메시징으로 사용된 경우, 연결되는 애플리케이션이 이를 사용하도록 요청 할 때 클러스터 및 클러스터된 큐의 다른 큐 관리자와 같은 다른 클러스터 자원에 관한 정보만은 클러스터의 각 큐 관리자가 알아야 합니다. 필요한 것만 알려주는 방식으로 동작합니다.

일반 **CLUSDR** 및 **CLUSRCVR** 채널 정의와 함께 발행/구독 클러스터는 큐 관리자의 클러스터입니다. 그러나 발행/구독 클러스터에는 토픽 오브젝트가 클러스터 이름을 식별한 클러스터에서 하나 이상의 큐 관리자에 정의된 하나 이상의 토픽 오브젝트가 포함됩니다.

클러스터에 정의된 토픽 오브젝트를 사용하면, 클러스터에 있는 한 큐 관리자에 연결된 애플리케이션이 해당 토픽 또는 해당 토픽 아래에 있는 토픽 트리의 노드를 구독하고 클러스터에 있는 다른 큐 관리자로부터 해당 토픽에 대한 발행물을 수신할 수 있습니다. 구독이 존재하는 큐 관리자를 식별하는 클러스터에 있는 다른 모든 큐 관리자에서 프록시 구독을 작성하면 이 프로세스를 완료할 수 있습니다. 따라서 문제가 되는 토픽에 대해 자신의 큐 관리자에서 발행이 이루어지면, 클러스터의 다른 적절한 멤버로 발행물을 전달하고 그곳에서 개별 애플리케이션 구독으로 전달해야 하는 것으로 파악합니다.

이 전달을 수행하려면, 토픽이 클러스터에 추가되자마자 클러스터의 모든 큐 관리자가 클러스터의 기타 모든 큐 관리자의 ID를 알아야 합니다. 이러한 지식은 클러스터의 전체 저장소 큐 관리자를 통해 전파됩니다. 어떤 큐 관리자에서 발행된 메시지는 같은 토픽에 대한 구독을 호스팅하는 것으로 알려진 클러스터의 다른 큐 관리자로부터 발송됩니다. 이 프로세스를 완료하려면 애플리케이션이 클러스터링된 토픽에 대한 구독을 작성할 때, 해당 큐 관리자가 클러스터-송신자 채널을 통해 프록시 구독을 전파하는 방식으로 클러스터에 있는 다른 모든 큐 관리자와 직접 통신할 수 있어야 합니다.

이 프로세스는 포인트-투-포인트 전달을 위해 클러스터를 사용할 경우 필요한 제한된 알아야 하는 정보 및 연결 성과 크게 다릅니다. 그러므로 발행/구독 클러스터의 요구사항은 포인트-투-포인트 클러스터의 요구사항과 다릅니다(클러스터된 토픽이 없는 클러스터).

클러스터된 토픽을 사용하면 큐 관리자 간의 발행/구독 도메인을 간단하게 확장할 수 있지만, 메커니즘과 숨겨진 의미를 파악하지 못하고 발행/구독에 사용되는 클러스터에 관해 이런 점을 고려하는 경우 문제점으로 이어질 수 있습니다. 다음 우수 사례는 이 이해 및 전파에서 도움이 되도록 설계됩니다.

요약하자면, 클러스터된 발행/구독의 성능 관련 의미는 대형 클러스터에 무익할 수 있고, 기존 클러스터에서 발행/구독을 사용하기 전에 이를 주의 깊게 고려하고 이해할 필요가 있습니다. 예를 들어, 클러스터된 토픽 오브젝트도 간단하게 작성할 수 있습니다. 발행/구독 활동 전용의 새로운 소형 클러스터부터 시작하여 클러스터의 규모를 성장시키는 것이 더 나을지 모릅니다.

발행/구독 토픽로지 계획

이전에 설명한 대로, 클러스터에서 발행/구독을 사용하는 경우 용량 및 성능에 대해 고려해야 할 사항이 있습니다. 그러므로 큐 관리자에서 발행/구독의 필요성을 신중하게 고려하고 필요한 큐 관리자의 수만으로 발행/구독을

제한하는 것이 좋습니다. 토픽 세트에 대해 발행하고 구독할 필요가 있는 큐 관리자의 최소 세트가 식별된 후, 이런 큐 관리자는 자신만 포함되어 있고 다른 큐 관리자는 없는 클러스터의 멤버일 수 있습니다.

이는 포인트-투-포인트 메시징을 위해 이미 잘 동작하고 있는 설정된 클러스터에서 특히 참입니다. 이러한 이유로 기존의 대형 클러스터를 발행/구독 클러스터로 변환시키는 경우, 초기에 현재 클러스터를 사용하는 대신 애플리케이션이 시도할 수 있는 발행/구독 작업용으로 별도의 클러스터를 작성하는 것이 좋습니다. 하나 이상의 포인트-투-포인트 클러스터에 이미 있는 기존의 큐 관리자를 계속 사용할 수 있고, 이런 큐 관리자의 서브세트를 새 발행/구독 클러스터의 멤버로 만들어야 합니다. 그러나 이 새 클러스터에는 기존 클러스터 전체 저장소로부터의 추가 로드를 분리하기 위해 전체 저장소로서 구성된 별도의 큐 관리자가 있어야 합니다.

크기 또는 현재의 로드로 인해 어떤 클러스터가 발행/구독에 사용되지 않도록 설정하는 경우, 클러스터에서 임의의 큐 관리자에 클러스터된 토픽을 간단히 작성함으로써 이 클러스터가 예상치 못하게 발행/구독 클러스터가 되지 않도록 하는 것이 좋은 방법입니다. 이 설계를 달성하려면 **PSCLUS** 큐 관리자 특성을 사용하십시오. 자세한 내용은 클러스터에서 클러스터된 발행/구독 금지를 참조하십시오.

클러스터에 추가될 토픽을 주의 깊게 선택하는 것도 중요합니다. 토픽 트리가 높을수록 이 토픽은 보다 광범위하게 됩니다. 이런 이유로, 보여진 동작을 고려하지 않고 클러스터로 토픽 루트 노드를 넣는 것은 권장되지 않습니다. 토픽 문자열에 상위 레벨 규정자를 사용하여(예를 들어, /global 또는 /cluster) 가능하면 글로벌 토픽을 명확하게 하십시오.

시스템 크기 조정 방법

모델이 포인트-투-포인트 메시징과 다르기 때문에 발행/구독 클러스터는 여러 채널을 필요로 합니다. 각 큐 관리자가 해당 클러스터의 기타 모든 큐 관리자에게 알려야 합니다. 포인트-투-포인트 모델은 선택(opt in) 모델이지만 발행/구독 클러스터에는 구독 팬아웃을 포함한 무차별 네이처가 있습니다. 그러므로 전체 저장소 큐 관리자 및 발행/구독 클러스터의 로컬 구독을 호스팅하는 큐 관리자에는 동시에 클러스터의 모든 멤버로 채널을 설정하는 기능이 있어야 합니다.

발행/구독 클러스터의 모든 큐 관리자가 이런 기능을 충족할 수 있도록 하는 것이 최선이겠지만, 구독을 결코 호스트하지 않는 것으로 알려진 큐 관리자는 다른 모든 큐 관리자와 채널을 설정할 필요가 없고, 따라서 이 레벨의 기능이 필요하지 않은 것으로 인정됩니다.

하지만, 그와 같은 큐 관리자에서 작성된 부수적인 구독이나 해당 큐 관리자를 클러스터에 있는 다른 큐 관리자와 수동으로 재동기화하려고 하면 모든 채널이 동시에 시작되므로 주의해야 합니다. 자세한 정보는 [63 페이지의 『프록시 구독의 재동기화』](#)의 내용을 참조하십시오.

클러스터된 발행/구독을 사용하여 전달될 하나의 큐 관리자에 발행된 메시지를 다른 큐 관리자의 구독으로 전달할 수 있습니다. 하지만 포인트-투-포인트 메시징에서와 같이 큐 관리자 사이에서 메시지 전송 비용이 성능에 크게 해가 될 수 있습니다. 그러므로 메시지가 발행 중인 동일 큐 관리자에 토픽에 대한 구독을 작성할 수 있는 곳에서 시도가 이루어져야 합니다.

다른 고려사항은 프록시 구독을 전파하는 시스템에서의 성능에 대한 영향입니다. 일반적으로, (구성된 토픽 오버젝트뿐 아니라) 특정 클러스터 토픽 문자열에 대해 첫 번째 구독이 작성될 때, 큐 관리자는 클러스터에 있는 다른 모든 큐 관리자로 프록시 구독 메시지를 발송합니다. 발행/구독 솔루션이 구독되는 다수의 고유 토픽 문자열로 구성되거나 토픽에 대한 구독과 구독 취소가 자주 이루어지는 경우, 클러스터에 있는 모든 큐 관리자 사이에서 상당한 양의 프록시 구독 트래픽이 생성되어 시스템의 전체적 성능에 나쁜 영향을 미칠 수 있습니다. 프록시 구독 오버헤드를 줄이는 자세한 방법은 [67 페이지의 『클러스터 토픽 성능』](#)의 내용을 참조하십시오.

프록시 구독의 재동기화

일반 상황에서, 큐 관리자는 시스템의 프록시 구독이 클러스터의 각 큐 관리자의 구독에 반영되는지를 자동으로 확인합니다.

그러나 `REFRESH QMGR TYPE(PROXYSUB)` 명령을 사용하여 클러스터에서 전파되는 프록시 구독과 큐 관리자의 로컬 구독을 수동으로 다시 동기화할 수 있습니다.

참고: 동기화하면 일시적으로 명령이 실행되는 큐 관리자에서 생성되는 갑작스러운 추가 프록시 구독 로드가 클러스터에서 발생합니다. 이러한 이유로 IBM WebSphere MQ 서비스, IBM WebSphere MQ 문서 또는 오류 로그가 이 지시하지 않으면 이를 사용하지 마십시오.

재동기화가 필요한 경우의 예제는 큐 관리자가 프록시 구독을 올바로 전파할 수 없는 경우입니다. 이는 채널이 중지되고 전송을 위해 모든 메시지를 큐잉할 수 없거나 운영자 오류로 인해

SYSTEM.CLUSTER.TRANSMIT.QUEUE 큐에서 메시지가 잘못 삭제되었기 때문입니다. 이런 경우, 먼저 원래 문제점을 수정한 후(예를 들어 채널을 다시 시작하여) 큐 관리자에서 **REFRESH QMGR TYPE(PROXYSUB)** 명령을 실행하십시오. 프록시 구독이 제대로 수행되지 않아 놓친 발행물은 영향을 받은 구독에 대해 복구되지 않습니다. 이 문제점을 고려해야 합니다.

재동기화하려면 큐 관리자가 클러스터에 있는 다른 모든 큐 관리자에 대해 채널을 시작해야 합니다. 그러므로 새로 고치는 큐 관리자는 클러스터의 다른 모든 큐 관리자와 통신에 대처할 충분한 능력이 있어야 합니다.

클러스터 토픽

클러스터 토픽은 **cluster** 속성이 정의된 관리 토픽입니다. 클러스터 토픽에 대한 정보는 클러스터의 모든 구성원에게 푸시되고 로컬 토픽에 결합되어 각 큐 관리자에서 다른 토픽 공간을 작성할 수 있습니다.

큐 관리자에 대한 클러스터 토픽을 정의하면 클러스터 토픽 정의가 전체 저장소 큐 관리자로 송신됩니다. 그러면 전체 저장소에서 클러스터 토픽 정의가 클러스터 내의 모든 큐 관리자로 전파되므로 클러스터의 모든 큐 관리자에 있는 발행자와 구독자가 동일한 클러스터 토픽을 사용할 수 있습니다. 클러스터 토픽이 작성된 큐 관리자를 클러스터 토픽 호스트라고 합니다. 클러스터의 어떤 큐 관리자든 클러스터 토픽을 사용할 수 있지만, 전체 저장소를 통해 클러스터의 모든 멤버에게 수정사항이 전파되는 지점에서(호스트) 토픽이 정의되는 큐 관리자에서 클러스터 토픽에 대한 수정 작업을 수행해야 합니다.

각 큐 관리자의 단일 토픽 이름 공간은 인식되는 로컬 및 클러스터 토픽 정의에서 구성됩니다. 애플리케이션이 클러스터된 토픽으로 해석되는 토픽을 구독하면, IBM WebSphere MQ가 프록시 구독을 작성하고 이 구독을 구독이 작성된 큐 관리자에서 클러스터의 다른 모든 구성원에게 바로 송신됩니다. 클러스터된 토픽 자체와는 달리, 프록시 구독은 전체 저장소 큐 관리자를 통해 흐르지 않습니다.

토픽에 발행된 메시지는 발행자가 연결된 큐 관리자에게 알려진 모든 구독에 송신됩니다. 이 구독이 프록시 구독인 경우, 발행된 메시지의 사본이 프록시 구독을 생성한 큐 관리자로 송신됩니다. 그러면 수신 큐 관리자가 메시지 사본을 개개의 로컬 구독으로 보냅니다. 이 프로세스를 통해 클러스터된 토픽에 대한 구독자는 클러스터의 큐 관리자에 연결된 발행자로부터 발행물을 수신하고 클러스터를 통해 최소 개수의 발행 메시지를 전파할 수 있습니다.

클러스터된 토픽과 로컬 토픽 오브젝트가 있으면 로컬 토픽이 우선합니다. 자세한 정보는 65 페이지의 『다중 클러스터 토픽 정의』의 내용을 참조하십시오.

클러스터 토픽을 표시하는 데 사용할 명령에 대한 자세한 정보는 다음 관련 링크를 참조하십시오.

와일드카드 구독

프록시 구독은 클러스터된 토픽 오브젝트(또는 그 아래)로 해석되는 토픽 문자열에 대한 로컬 구독이 이루어질 때 작성됩니다. 와일드카드 구독이 클러스터 토픽보다 높은 계층에서 이루어지면 일치하는 클러스터 토픽에 대한 프록시 구독이 클러스터 주변에 송신되지 않아서, 클라이언트의 기타 멤버로부터 발행물을 수신하지 않습니다. 하지만 로컬 큐 관리자로부터 발행물을 수신합니다.

하지만 또 다른 애플리케이션이 클러스터 토픽 또는 그 아래로 해석되는 토픽 문자열을 구독하는 경우에는 프록시 구독이 생성되고 발행물이 이 큐 관리자에게 전파됩니다. 원래의 더 높은 와일드카드 구독이 이 발행물의 적합한 수신자로 간주되어 사본을 수신합니다.

이 작동은 같은 토픽에서 로컬로 발행된 메시지와는 다릅니다. 이 작동이 필요하지 않은 경우 클러스터된 토픽에 대해 **WILDCARD (BLOCK)**를 설정하면 원래 와일드카드가 정당한 구독으로 간주되지 않아 클러스터 토픽이나 하위 토픽에 대해 (로컬이나 클러스터의 다른 곳에서) 발행물을 수신하지 못합니다.

관련 개념

[관리 토픽에 대한 작업](#)

[구독에 대한 작업](#)

관련 참조

[DISPLAY TOPIC](#)

[DISPLAY TPSTATUS](#)

[DISPLAY SUB](#)

클러스터 토픽 속성

발행/구독 클러스터를 디자인하고 관리하는 데 필요한 클러스터 토픽 속성에 대해 설명합니다.

토픽 오브젝트에는 다중 큐 관리자 발행/구독 토폴로지에 적용되는 여러 속성이 있습니다. IBM WebSphere MQ 클러스터를 사용하여 해당 토폴로지를 작성할 경우 이러한 속성은 다음과 같이 작동합니다.

PROXYSUB

- **PROXYSUB**는 프록시 구독이 작성되는 시기를 제어하는 속성입니다. FIRSTUSE의 기본값에서 이 속성을 변경하려는 이유에 대한 세부사항은 48 페이지의 『라우팅 메커니즘에 대한 세부사항』의 내용을 참조하십시오.
- 클러스터된 토픽의 다른 속성과 마찬가지로, **PROXYSUB** 속성은 토픽이 정의된 큐 관리자뿐 아니라 클러스터의 모든 큐 관리자로도 잔파됩니다. 따라서 클러스터의 모든 큐 관리자가 바로 다른 모든 큐 관리자에 대해 와일드카드 프록시 구독을 작성하게 됩니다. 이 프로세스의 결과로 모든 큐 관리자가 다른 모든 큐 관리자에 대해 클러스터 송신자 채널을 작성하고, 발행된 모든 메시지가 모든 큐 관리자에게 발송됩니다.

PUBSCOPE 및 SUBSCOPE

PUBSCOPE 및 **SUBSCOPE**는 이 큐 관리자가 토폴로지(발행/구독 클러스터 또는 계층)에 있는 큐 관리자에게 발행물을 전파하거나 해당 로컬 큐 관리자로서 범위를 제한하는지 여부를 판별합니다.

MQPMO_SCOPE_QMGR/MQSO_SCOPE_QMGR을 사용하여 프로그래밍 방식으로 동일한 작업을 수행할 수 있습니다.

- **PUBSCOPE** 클러스터 토픽 오브젝트가 **PUBSCOPE (QMGR)**를 사용하여 정의된 경우, 정의는 클러스터와 공유되지만 해당 토픽을 기준으로 하는 발행물의 범위가 로컬 전용이므로 발행물은 클러스터의 다른 큐 관리자로서 송신되지 않습니다.
- **SUBSCOPE** 클러스터 토픽 오브젝트가 **SUBSCOPE (QMGR)**를 사용하여 정의된 경우, 정의는 클러스터와 공유되지만 해당 토픽을 기준으로 하는 발행물의 범위가 로컬 전용이므로 발행물은 클러스터의 다른 큐 관리자로서 송신되지 않습니다.

두 속성은 일반적으로 큐 관리자가 특정 토픽에 대해 클러스터의 기타 멤버와 상호작용하지 못하게 하는 용도로 함께 사용됩니다. 큐 관리자가 클러스터의 기타 멤버로부터 이 토픽에 대한 발행물을 발행하지도, 수신하지도 않습니다. 토픽 오브젝트가 하위 토픽에 대해 정의된 경우에는 이 상황에서 발행 또는 구독이 차단되지 않습니다.

토픽의 로컬 정의에서 **SUBSCOPE**를 QMGR로 설정하면 클러스터의 다른 큐 관리자가 **SUBSCOPE (ALL)**와 함께 토픽의 클러스터된 버전을 사용하는 경우 해당 프록시 구독을 큐 관리자로서 전파하지 않습니다. 하지만 로컬 정의에 **PUBSCOPE**도 QMGR로 설정되면 이 큐 관리자로부터 프록시 구독이 송신되지 않습니다.

다중 클러스터 토픽 정의

로컬 토픽 정의는 같은 이름의 원격 정의된 클러스터 토픽 정의를 대체합니다. 클러스터에 있는 여러 가지 큐 관리자에서 같은 클러스터 토픽에 대한 여러 가지 정의를 작성하는 것도 가능합니다. 하지만, 이 두 시나리오 모두 주의가 필요하며 이 토픽에서 그 이유를 설명합니다.

클러스터된 큐와 마찬가지로, 클러스터에 같은 클러스터 토픽 오브젝트의 정의가 여러 가지이면 각각에 대해 정의된 서로 다른 특성을 가질 가능성이 있습니다. 클러스터에 있는 각 큐 관리자에게 보이는 토픽 정의의 버전이 무엇인지 판별하기 쉽지 않으므로, 예상되는 작동을 판별하기 어렵습니다.

단일 토픽 문자열에 대해 둘 이상의 클러스터 토픽 정의가 서로 다른 속성을 갖거나 둘 이상의 클러스터에 존재하는 경우, 메시지 (AMQ5465 & AMQ5466)가 오류 로그에 기록되고 가장 최근에 수신된 클러스터 토픽 정의가 사용됩니다.

클러스터 토픽 호스트 큐 관리자는 토픽 정의를 삭제하면 안 되고, 클러스터의 모든 멤버가 클러스터된 토픽을 계속 알 수 있도록 클러스터에 그대로 남습니다. 클러스터 토픽 정의는 전체 저장소 큐 관리자와 부분 클러스터 저장소에 있는 다른 모든 큐 관리자에 의해 캐시되기 때문에, 이 호스트 큐 관리자를 반드시 계속 사용할 수 있어야 하는 것은 아닙니다. 이러한 캐싱을 사용하면 호스트 큐 관리자를 사용할 수 없는 동안 최소 60일 이상 사용할 수 있습니다. 이 주제에 대해서는 70 페이지의 『발행/구독 클러스터 큐 관리자의 주요 역할』을 참조하십시오.

로컬로 클러스터 토픽 정의 대체

클러스터의 특정 큐 관리자에 대해 클러스터된 토픽의 작동을 대체해야 할 수도 있습니다. 클러스터 토픽 오브젝트를 같은 토픽 문자열로 대체하고 로컬로 연결된 구독자에게만 이를 발행하는 데 사용하기 위한 로컬 토픽 오브젝트를 정의하여 대체할 수 있습니다.

큐 관리자에 클러스터된 토픽을 대체하기 위해 토픽의 로컬 정의가 작성되어도, 큐 관리자는 클러스터된 토픽 정의를 사용하여 클러스터의 다른 멤버로부터 프록시 구독을 계속 수신합니다. 기본적으로, 로컬에서 발행되는 메

시지는 프록시 구독을 지원하기 위해 계속 리모트 큐 관리자로 발송됩니다. 이런 배열이 필요하지 않은 경우에는 로컬 토픽 오브젝트에 대해 **PUBSCOPE** (QMGR)를 지정하여 이 큐 관리자에 연결된 발행자 애플리케이션이 로컬 구독자에게만 발행하도록 하십시오.

클러스터 토픽 정의 수정

클러스터 토픽 정의를 대체해야 하는 경우, 해당 정의를 이 정의가 클러스터 토픽 호스트에 정의된 동일한 큐 관리자에서 수정하십시오. 클러스터의 다른 큐 관리자에서 동일한 클러스터 토픽 정의를 작성하지 마십시오. 토픽을 다시 정의하면 동일한 클러스터 토픽에 대해 두 클러스터 토픽 호스트가 사용됩니다.

여러 번 클러스터 토픽을 정의하면 잠재적으로 충돌하는 정의가 작성되고 여러 큐 관리자가 여러 경우에 서로 다른 정의를 사용할 수도 있습니다.

클러스터 토픽 정의를 클러스터의 다른 큐 관리자로 이동

예를 들어, 클러스터에서 큐 관리자를 커밋 해제할 때 클러스터에 있는 한 큐 관리자에서 다른 큐 관리자로 클러스터 토픽 정의를 이동해야 할 수도 있습니다. 발행물의 플로우를 방해하지 않고 클러스터에 있는 다른 큐 관리자로 클러스터 토픽 정의를 이동시키려면 다음 단계를 수행해야 합니다. The example moves a definition from QM1 to QM2.

1. QM1의 정의와 같은 속성을 가진 QM2에 대해 클러스터 토픽 정의의 복제를 작성하십시오.
2. 전체 저장소 큐 관리자가 클러스터 전체로 새 정의를 전파할 때까지 기다리십시오. (**DISPLAY CLUSTER**) 명령을 사용하여 각 클러스터 멤버에 클러스터 토픽을 표시하고 QM2에서 시작하는 정의를 확인하여 전파를 판별할 수 있습니다.
3. QM1에서 클러스터 토픽 정의를 삭제하십시오.

QM1에서 원래 정의가 삭제된 후, 필요하다면 특성에 충돌을 일으키지 않고 QM2에 대한 정의를 수정할 수 있습니다.

실패한 큐 관리자에서 클러스터 토픽 정의 바꾸기

이전 시나리오에서는 일정한 기간 동안 QM1을 사용할 수 없으면 QM1에서 정의를 삭제하는 것이 가능하지 않을 수도 있습니다. 이 시나리오에서는 두 가지 정의가 모두 있는 상태에서 실행하도록 허용됩니다.

그런 다음 클러스터된 토픽 정의를 수정해야 할 필요성이 생기면, QM2 정의가 QM1 정의보다 새로운 정의이므로 우선한다는 점을 알고 있는 상태에서 QM2에서 버전을 수정할 수 있습니다. 하지만, 이 기간 중에 충돌하는 클러스터 토픽 정의가 있기 때문에 큐 관리자의 오류 로그에 오류가 기록됩니다. QM1을 다시 시작할 수 있을 때 QM1에서 중복 클러스터 토픽 정의를 제거하여 최대한 빨리 오류를 해결하십시오.

또는 QM1이 클러스터로 결코 되돌아가지 않을 경우(예: 치명적인 하드웨어 장애가 발생한 후 예상치 못하게 커밋 해제된 경우), **RESET CLUSTER** 명령을 사용하여 큐 관리자를 강제로 내보낼 수 있습니다. **RESET CLUSTER**는 대상 큐 관리자에서 호스팅하는 모든 토픽 오브젝트를 자동으로 삭제합니다.

클러스터에서 클러스터된 발행/구독 금지

큐 관리자의 클러스터, 특히 기존 클러스터에 발행/구독을 도입할 때 성능 감소를 수용할 수 있도록 세심하게 계획해야 합니다.

대형 IBM WebSphere MQ 클러스터(다수의 큐 관리자가 포함된 클러스터)에 클러스터된 토픽을 도입하면 클러스터의 각 큐 관리자에 즉시 추가적인 로드가 부과될 수 있고 일부 경우에는 성능이 저하될 수 있습니다. 따라서 발행/구독 도입을 신중하게 계획해야 합니다. 자세한 정보는 [67 페이지의 『클러스터 토픽 성능』](#)의 내용을 참조하십시오.

클러스터가 발행/구독의 오버헤드를 수용할 수 없는 것으로 알려진 경우에는 큐 관리자 속성 **PSCLUS**를 **DISABLED**로 설정하여 큐 관리자에서 클러스터된 발행/구독 기능을 사용하지 않을 수 있습니다.

PSCLUS를 **DISABLED**로 설정하면 다음과 같이 큐 관리자 기능의 세 가지 부분이 수정됩니다.

- 이 큐 관리자의 관리자는 더 이상 Topic 오브젝트를 클러스터된 것으로 정의할 수 없습니다.

- 다른 큐 관리자에서 수신되는 토픽 정의 또는 프로시 구독이 거부됩니다(관리자에게 잘못된 구성을 알려주기 위해 경고 메시지가 로그로 기록됨).
- 전체 저장소가 토픽 정의를 수신할 때 더 이상 자동으로 다른 모든 부분 저장소와 모든 큐 관리자에 대한 정보를 공유하지 않습니다.

PSCLUS는 클러스터의 각 개별 큐 관리자 매개변수이지만 클러스터의 큐 관리자 서브세트에서 발행/구독을 선택적으로 사용 불가능하게 하는 용도로 사용되지 않습니다. 다른 것과는 별개로, 이 방법을 사용하면 오류 메시지가 자주 프로시 구독으로 보이게 되고 토픽 정의가 늘 보이고 거부됩니다. 이 옵션을 사용할 때는 클러스터에 있는 모든 큐 관리자를 사용 안함으로 설정하는 것이 이상적입니다. 큐 관리자가 하나 이상의 발행/구독 클러스터에 참가하고 하나 이상의 일반 클러스터에도 참가하는 경우, 해당 큐 관리자에서 **PSCLUS**를 **ENABLED**로 설정해야 합니다. 전체 저장소에서 사용 안함으로 설정하는 데 관한 다음 정보를 참조하십시오.

중요한 점은, 클러스터의 모든 전체 저장소 큐 관리자에서 **PSCLUS**를 **DISABLED**로 설정하면 잘못 구성된 부분 저장소에 클러스터된 토픽 정의가 클러스터의 다른 큐 관리자에 영향을 주지 못한다는 것입니다. 그런 상황에서는 전체 저장소 큐 관리자의 오류 로그에 불일치가 보고됩니다.

기존의 포인트-투-포인트 클러스터를 발행/구독 클러스터와 겹칠 때, 각각 별개의 전체 저장소 세트를 사용하는 것이 중요합니다. 이 배열에서는 토픽 정의와 '모든 큐 관리자' 정보가 발행/구독 클러스터에서만 플로우될 수 있습니다.

이 매개변수를 사용할 때 일관성 없는 구성을 피하는 데 도움이 되는 몇 가지 중요한 사항이 있습니다. **ENABLED**에서 **DISABLED**로 수정할 경우 이 큐 관리자가 멤버인 클러스터에는 클러스터 토픽 오브젝트가 존재할 수 없습니다. 이 기능을 사용 불가능하게 하기 전에 (원격으로 정의된 것이라도) 그런 토픽을 전부 삭제해야 합니다.

PSCLUS에 대한 자세한 정보는 **ALTER QMGR (PSCLUS)**를 참조하십시오.

클러스터 토픽 성능

클러스터 토픽의 성능 특성은 클러스터 큐의 성능 특성과 다르므로 특별한 고려가 필요한데, 별로 고려하지 않고 사용하면 대형 클러스터 또는 밸런싱되지 않은 클러스터에서 성능 문제점의 원인이 될 수 있습니다.

성능에서 발행/구독의 영향 줄이기

클러스터의 큐 관리자에 두 개의 워크로드 소스(애플리케이션 프로그램의 메시지 수동 핸들링과 클러스터를 관리하기 위해 필요한 메시지 및 채널 핸들링)가 있습니다. 일반적인 포인트-투-포인트 클러스터에서 클러스터 시스템 워크로드는 주로 필요에 따라 클러스터의 멤버가 명시적으로 요청하는 정보로 제한됩니다(70 페이지의 『발행/구독 클러스터의 성능 특성』에서 비교한 내용 참조). 따라서 매우 큰 클러스터(예: 수천 개의 큐 관리자가 포함된 클러스터) 이외의 클러스터에서는 큐 관리자 성능을 고려할 때 클러스터 관리에 따라 성능에 미치는 효과를 크게 줄일 수 있습니다.

발행/구독 클러스터에서는 모든 클러스터 큐 관리자가 발행/구독 메시징에 능동적으로 참가하는지 상관없이, 클러스터된 토픽 및 프로시 구독과 같은 정보가 클러스터의 모든 멤버에게 푸시됩니다. 이 프로세스는 시스템에 상당한 추가 로드를 생성합니다. 그러므로 타이밍 및 크기 모두에서 큐 관리자 성능에 관한 클러스터 관리 효과를 고려해야 합니다.

클러스터의 성능에 발행/구독 클러스터 관리가 미치는 영향을 줄이려면 다음 두 가지 제안사항을 고려하십시오.

1. 하루의 오프 피크 시간에 클러스터, 토픽 및 구독 업데이트를 수행하십시오.
2. 클러스터가 이미 있다는 이유만으로 기존의 대형 클러스터에 발행/구독 토픽을 추가하려고 생각하고 있다면, 발행/구독에 관여하는 훨씬 작은 큐 관리자 서브세트를 정의하고 그것을 "겹치는" 클러스터로 만들 수 있을지 고려하십시오. 이 클러스터가 클러스터 토픽이 정의되는 클러스터입니다. 일부 큐 관리자가 이제 두 클러스터에 있지만, 발행/구독의 전체 효과는 줄어듭니다.
 - a. 발행/구독 클러스터의 크기가 더 작습니다.
 - b. 클러스터 관리 트래픽이 발행/구독 클러스터에 없는 큐 관리자에 미치는 영향이 훨씬 줄어들었습니다.

작성자 및 이용자 밸런싱

비동기 메시징 성능의 중요한 개념은 밸런싱입니다. 메시지 이용자가 메시지 작성자와 밸런스를 이루지 않는 경우, 이용되지 않는 메시지의 백로그가 쌓이고 복수 애플리케이션의 성능에 심각하게 영향을 줄 수 있다는 위험이 있습니다.

포인트-투-포인트 메시징 토폴로지에서 메시지 이용자와 메시지 작성자 사이의 관계가 쉽게 이해됩니다. 채널 별, 큐별 예상 메시지 생산량 및 이용량을 얻을 수 있습니다. 밸런스 부족이 있는 경우 병목현상을 쉽게 식별한 후 구제합니다.

발행자 및 구독자가 발행/구독 토폴로지에서 밸런스를 이루는지 여부를 밝히는 것은 더 어렵습니다. 클러스터된 토픽으로 표시되는 각 subscription에서 시작하고 토픽에 대한 publisher가 있는 큐 관리자로 다시 돌아가십시오. 각 큐 관리자에서 각 구독자로 흐르는 발행물의 수를 계산하십시오.

(프록시 구독을 바탕으로) 클러스터에 있는 리모트 큐 관리자에 대한 구독과 일치하는 각각의 발행물은 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 넣게 됩니다. 다중 리모트 큐 관리자는 해당 발행에 대한 프록시 구독이 있는 경우, 메시지의 여러 사본이 다른 클러스터 송신자 채널을 대상으로 하는 전송 큐에 넣어집니다.

이 발행물은 리모트 큐 관리자에서 SYSTEM.INTER.QMGR.PUBS 큐를 대상으로 합니다. 각각의 큐 관리자는 해당 큐에 도착하는 메시지를 처리하여 해당 큐 관리자에 대한 올바른 구독으로 배달합니다.

이 때문에 병목현상이 발생할 수 있는 다음 지점의 로드를 모니터하십시오.

- 개별 구독 큐 자신:

- 이런 병목 상태는 구독 애플리케이션이 발행물이 발행되는 속도만큼 빠르게 발행물을 이용하지 못하고 있음을 의미합니다.

- SYSTEM.INTER.QMGR.PUBS 큐:

- 큐 관리자가 로컬 구독에 발행물을 분배할 수 있는 속도보다 빠르게 하나 이상의 리모트 큐 관리자로부터 발행물을 수신하고 있습니다.

- 발행 큐 관리자, 구독 큐 관리자, 발행 큐 관리자의 클러스터 전송 큐(기본적으로 SYSTEM.CLUSTER.TRANSMIT.QUEUE) 사이의 클러스터 채널:

- 하나 이상의 클러스터 채널이 실행되고 있지 않거나, 채널이 리모트 큐 관리자로 메시지를 배달할 수 있는 것보다 빠른 속도로 로컬 큐 관리자로 메시지가 발행되고 있습니다.

- 발행 애플리케이션이 큐된 발행/구독 인터페이스를 사용하고 있는 경우, SYSTEM.BROKER.DEFAULT.STREAM 큐 및 SYSTEM.QPUBSUB.QUEUE.NAMELIST에 나열된 기타 스트림 큐 및 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST에 나열된 SYSTEM.BROKER.DEFAULT.SUBPOINT 큐 및 기타 서브포인트 큐도 고려해야 합니다.

- 로컬 큐 관리자가 메시지를 처리하는 것보다 빠른 속도로 로컬 발행 애플리케이션이 메시지를 넣고 있습니다.

구독 성능 고려사항

이전에 설명된 대로, 클러스터된 토픽을 해석하는 토픽 문자열에 대한 큐 관리자에서 구독이 작성되면, 해당 큐 관리자는 클러스터의 기타 모든 큐 관리자에 토픽의 위치에 프록시 구독이 있는지 확인해야 합니다. 이 결과를 얻으려면, 큐 관리자가 작성되며 프록시 구독 메시지를 클러스터의 기타 모든 큐 관리자에게 송신합니다.

기본 구성을 사용하여 클러스터된 토픽에 대한 구독을 작성할 때 새 프록시 구독이 전송되지 않는 유일한 경우는 정확히 같은 토픽 문자열에 대한 구독이 로컬 큐 관리자에 이미 있는 경우입니다. 이 상황에서, 도착되는 발행물이 토픽 문자열에 대한 원래 구독만이 아닌 일치되는 모든 구독에 전달되기 때문에 추가 프록시 구독이 필요하지 않습니다.

기본 구성에 대한 대안은 69 페이지의 『개별 프록시 구독 사용 안함』의 내용을 참조하십시오.

구독 선택자는 고려되지 않으므로, 같은 토픽 문자열에 대해 서로 다른 선택자를 사용하여 두 개의 구독이 계속 프록시 구독을 공유합니다. 또한 이 상황은 발행문이 구독 선택자와 일치하지 않더라도 토픽 문자열과 일치하는 발행물이 구독자 큐 관리자에게 전파됨을 의미할 수 있습니다.

토픽 문자열에 대한 마지막 구독이 큐 관리자에서 삭제되면 프록시 구독 메시지에 해당되는 메시지가 작성되어 모든 큐 관리자에게 송신됩니다. 이 프로세스는 리모트 큐 관리자에서 프록시 구독을 제거합니다.

이런 이유로, 클러스터의 크기 및 다른 토픽 문자열에 대한 구독 빈도가 클러스터 자체에 중요한 로드를 행사할 수 있으며 클러스터 및 토픽 계획을 발행/구독 애플리케이션에서 사용할 경우를 고려해야 합니다.

67 페이지의 『작성자와 이용자 밸런싱』 절에 나열된 큐 모니터 외에 프록시 구독 트래픽으로부터 시스템에 로드를 고려할 경우, 다음 큐를 모니터하십시오.

- 구독자 큐 관리자의 SYSTEM.INTER.QMGR.FANREQ 큐.
- 클러스터에 있는 다른 모든 큐 관리자의 SYSTEM.INTER.QMGR.CONTROL 큐.

이 큐에서의 중요 메시지 백로그는 구독 변경의 등급이 해당 시스템에 대해 너무 크거나 큐 관리자가 클러스터에서 올바르게 기능한다는 것을 의미합니다. 발행/구독 지원이 사용 불가능하거나 (ALTER QMGR에서 **PSMODE** 참조) 추가 조사를 필요로 하는 문제가 발생하기 때문에, 확인되는 지점에서 큐 관리자 오류가 로그됩니다.

프록시 구독 트래픽 감소

프록시 구독 오버헤드가 높으면 이를 줄이기 위한 단계를 수행해야 합니다. 이것은 일반 토픽 통합 또는 큐 관리자 사이의 발행물에 대한 브로드캐스트 모델로의 변경을 통해 가능합니다.

시스템의 자원에 대한 전체 로드를 줄이는 방식으로 토픽 문자열을 통합할 수 있을지 알아보기 위해 토픽 문자열 사용을 평가하는 것이 일반적인 발행/구독 권장사항입니다. 많은 특징적인 임시 토픽 문자열의 사용은 발행자 또는 구독이 첨부되는 시스템의 각 큐 관리자에서 일정 수준의 관리 오버헤드를 가져옵니다. 토픽 문자열의 수와 일시적 특성을 줄이면 이런 문자열에 대한 발행자와 구독이 시스템에 미치는 영향을 줄입니다.

프록시 구독 트래픽을 줄이는 한 가지 방법은 같은 큐 관리자에서 같은 토픽 문자열에 대한 구독을 찾는 것입니다. 이 큐 관리자는 이 방법을 통해 같은 토픽 문자열에 대한 고유의 구독 세트에 대해 각각 다수의 큐 관리자가 프록시 구독을 발송하는 것이 아니라, 다른 큐 관리자에게 단일 프록시 구독을 발송할 수 있습니다. 이렇게 하면 클러스터 전반적으로 발행물 라우팅도 최적화됩니다.

개별 프록시 구독 사용 안함

클러스터 전체에 걸쳐 구독되고 있는 고유의 토픽 문자열 세트가 크고 계속 변하는 상황에서는 구독 전파 모델에서 발행 브로드캐스트 모델로 변경하는 것이 더 나을 수 있습니다. 발행 브로드캐스트 모델에서는 해당 큐 관리자에서의 구독 존재 여부에 상관없이 클러스터의 모든 큐 관리자로 클러스터 토픽에 대한 모든 발행물이 자동으로 발송됩니다.

그러면 수신 큐 관리자가 로컬 구독이 존재하는 경우 그 로컬 구독으로 메시지를 배달하거나 메시지를 제거할 수 있습니다. 이 모델에서는 구독의 존재 여부를 바탕으로 개별 프록시 구독을 작성하고 삭제할 필요가 없습니다. 이 모드에서 실행 시, 모든 발행물이 모든 큐 관리자로 발송되므로 발행된 메시지 자원 로드가 아마 증가할 것입니다. 따라서 클러스터의 큐 관리자가 이런 추가 로드를 핸들링할 수 있는 용량을 가지고 있어야 합니다.

다음 구성 단계를 수행하여 브로드캐스트 모델을 사용하십시오.

1. 구독을 호스트하는 모든 큐 관리자는 클러스터된 토픽에 대한 로컬 구독과 일치하는 프록시 구독을 발송하지 않도록 구성되어야 합니다. 이 구성에서는 클러스터 토픽을 정의하거나 클러스터에서 구독을 작성하기 전에 각각의 큐 관리자 `qm.ini` 파일에서 다음 성능 조정 매개변수를 설정해야 합니다.

```
TuningParameters:
  pscProxySubFlags=1
```

2. 성능 조정 매개변수를 설정한 후, 모든 큐 관리자를 재시작해야 합니다.
3. 큐 관리자가 재시작된 후 클러스터된 토픽을 정의할 수 있습니다. 각 클러스터 토픽은 **PROXYSUB**를 **FORCE**로 설정해야 합니다.

작동 반전

69 페이지의 『개별 프록시 구독 사용 안함』에서 이전에 설명한 조작 모드를 반전하려면 다음 단계를 수행하십시오.

1. 모든 큐 관리자에 대해 `qm.ini` 파일에서 성능 조정 매개변수를 제거합니다.
2. 모든 큐 관리자를 재시작하십시오.
3. 구독을 호스트하는 모든 큐 관리자에서 **REFRESH QMGR TYPE(PROXYSUB)** 명령을 실행하십시오.
4. 클러스터된 토픽에 대해 **PROXYSUB**를 **FIRSTUSE**로 설정하십시오.



경고: 이 작동 사용 및 반전에서 모두, 문서화된 순서에 따라 모든 단계가 완료되지 않은 경우 구독에 대한 발행물이 올바르게 플로우되지 않을 수 있습니다.

참고: PROXYSUB(FORCE) 설정의 의미

이 토픽에서 앞서 설명한 바와 같이, **PROXYSUB(FORCE)** 토픽 속성은 프록시 구독 트래픽을 줄일 수 있지만 주의해서 사용해야 합니다. **PROXYSUB(FORCE)** 속성은 토픽이 정의된 큐 관리자뿐 아니라, 클러스터의 모든 큐 관리자로도 전파됩니다. 따라서 클러스터의 모든 큐 관리자가 바로 다른 모든 큐 관리자에 대해 와일드카드 프록시 구독을 작성하게 됩니다. 이 프로세스의 결과, 모든 큐 관리자가 다른 모든 큐 관리자에 대해 클러스터 송신자 채널을 작성하고 발행된 모든 메시지가 모든 큐 관리자로 발송됩니다.

크거나 사용 중인 클러스터에서 이 특성을 설정하면 시스템 자원에 추가로 로드가 걸릴 수 있습니다.

발행/구독 클러스터의 성능 특성

큐 관리자, 토픽 또는 구독을 클러스터에 추가하는 것처럼, 발행/구독 클러스터의 속성 변경이 클러스터에서 실행 중인 애플리케이션의 성능에 어떤 영향을 미칠지 고려하는 것이 중요합니다.

두 개의 관리 태스크에 대해 포인트-투-포인트 클러스터를 발행/구독 클러스터와 비교하십시오.

첫 번째, 포인트-투-포인트 클러스터:

1. 새 클러스터 큐가 정의되어 있을 때 목적지 정보가 전체 저장소 큐 관리자로 푸시되고, 처음에 클러스터 큐를 참조할 때(예: 애플리케이션이 클러스터 큐를 열려고 할 때) 다른 클러스터 멤버로만 전송됩니다. 이 정보는 큐가 액세스될 때마다 정보를 원격으로 검색할 필요성을 제거하기 위해 큐 관리자에 의해 로컬로 캐시됩니다.
2. 클러스터에 큐 관리자 추가가 다른 큐 관리자의 로드에도 직접적으로 영향을 주지 않습니다. 새 큐 관리자에 관한 정보가 전체 저장소로 푸시되지만, 클러스터의 다른 큐 관리자로부터 새 큐 관리자로의 채널은 트래픽이 새 큐 관리자로 또는 새 큐 관리자로부터 이동하기 시작할 때만 작성되고 시작됩니다.

즉, 포인트-투-포인트 클러스터에 있는 큐 관리자의 로드는 애플리케이션 프로그램에서 핸들링하는 메시지 트래픽과 관련되어 있으며 클러스터의 크기와는 직접 관련되지 않습니다.

그 다음, Pub/Sub 클러스터에 대한 내용입니다.

1. 새 클러스터 토픽이 정의되어 있을 때 정보가 전체 저장소 큐 관리자로 푸시되고, 거기서 바로 클러스터의 모든 멤버에게 즉시 푸시되므로, 채널이 아직 시작되지 않은 경우 전체 저장소에서 클러스터의 각 멤버로 채널이 시작됩니다.
2. 새 토픽 문자열의 클러스터 토픽에 구독이 작성될 때, 해당 정보가 즉시 해당 큐 관리자에서 클러스터의 다른 모든 멤버로 직접 푸시되어 아직 시작되지 않은 경우 해당 큐 관리자에서 클러스터의 각 멤버로 채널이 시작되게 합니다.
3. 새 큐 관리자가 기존 클러스터를 조인할 때, 클러스터된 모든 토픽에 대한 정보가 전체 저장소 큐 관리자에서 새 큐 관리자로 푸시됩니다. 그러면 새 큐 관리자가 클러스터의 모든 멤버가 있는 클러스터에서 모든 구독에 관한 정보를 클러스터 토픽으로 동기화하므로, 새 큐 관리자에서 클러스터의 각 멤버에 대해 채널이 작성되고 시작됩니다.

요약하자면, 클러스터에 있는 임의의 큐 관리자에서 클러스터 관리 로드는 각 큐 관리자에서 해당 클러스터 토픽을 로컬에서 사용하는 것과 상관없이 클러스터 내부에 있는 큐 관리자, 클러스터된 토픽 및 프록시 구독의 수와 함께 증가합니다.

발행/구독 클러스터 큐 관리자의 주요 역할

포인트-투-포인트 클러스터와 유사하게, 발행/구독 클러스터에는 큐 관리자에 대해 전체 저장소 큐 관리자와 클러스터 토픽 호스트로서의 두 가지 주요 역할이 있습니다.

전체 저장소

전체 저장소 큐 관리자에는 오브젝트 정의를 클러스터의 다른 구성원으로 푸시하는 역할이 있습니다. 발행/구독 클러스터의 경우에는 클러스터된 토픽 오브젝트 정의를 클러스터의 다른 구성원으로 푸시합니다.

클러스터 토픽 호스트

클러스터 토픽 호스트는 클러스터된 토픽 오브젝트가 정의된 큐 관리자입니다. 발행/구독 클러스터에 있는 모든 큐 관리자에서 클러스터된 토픽 오브젝트를 정의할 수 있습니다. 클러스터 토픽 오브젝트는 전체 저장소 큐 관리자로 푸시되며, 클러스터의 임의의 큐 관리자를 실행하는 발행자 및 구독자에서 사용하도록 캐시되는 클러스터의 기타 모든 큐 관리자에게 푸시됩니다.

사용 가능성 및 관리

클러스터에서 클러스터 토픽 정의의 가용성을 극대화하려면 클러스터에서 두 개의 전체 저장소를 정의해야 합니다.

큐된 메시징 클러스터에 관해서는, 많은 컴퓨터 사이에서 가용성이 높은 컴퓨터가 두 대만 있는 발행/구독 클러스터에서 가용성이 높은 컴퓨터를 전체 저장소로 정의하는 것이 좋은 방법입니다.

큐된 클러스터에서는, 클러스터의 여러 큐 관리자에 있는 동일한 클러스터 큐를 정의하여 클러스터 큐의 사용 가능성 및 처리량을 늘릴 수 있습니다. 그러면 메시지가 여러 큐 관리자 간에 워크로드 밸런스가 조정됩니다. 그에 반해, 발행/구독 클러스터에서 클러스터된 토픽을 클러스터의 모든 큐 관리자에서 사용 가능하지만, 발행/구독 트래픽의 워크로드 밸런스 조정이 수행되지 않습니다. 그 대신, 서로 다른 여러 큐 관리자에 걸쳐 별개의 구독 및 발행자를 분배하여 발행/구독 로드를 분산해야 합니다. 클러스터 토픽을 정의한 큐 관리자가 사용 불가능한 경우, 다른 큐 관리자가 토픽에 대한 발행/구독 요청을 계속 처리합니다.

그러나 클러스터 토픽 오브젝트를 정의한 큐 관리자를 다시 사용할 수 없는 경우, 결국 다른 큐 관리자의 캐시된 토픽 오브젝트가 삭제되고 토픽이 사용 불가능해집니다. 이 프로세스는 토픽 정의가 사용 불가능하게 된 시점으로부터 최소한 60일 이상 지난 후에(토픽 정의가 마지막으로 새로 고쳐진 시점에 따라 다름) 발생합니다.

클러스터 토픽 오브젝트를 정의한 큐 관리자를 복구하는 데 60일의 기간이 주어지므로, 클러스터 토픽 호스트의 가용성을 높이기 위해 특별한 방법을 동원할 필요는 거의 없습니다. 기술적 문제점에 대응하는 데 60일이라는 기간이면 충분합니다. 60일을 초과한다면, 관리상 오류가 원인일 가능성이 매우 높습니다. 클러스터 토픽 호스트를 사용할 수 없는 경우 사용 가능성을 높이기 위해 클러스터의 모든 구성원이 캐시된 클러스터 토픽 오브젝트를 새로 고치지 않은 오류 로그 메시지를 시간마다 기록합니다. 클러스터 토픽 오브젝트를 정의한 큐 관리자가 실행 중인지 확인하여 이 메시지에 응답합니다.

다른 큐 관리자에서 동일한 클러스터 토픽 오브젝트를 정의하는 사례를 채택할 수 있습니다. 정의할 때마다 추가 클러스터 토픽 오브젝트가 다른 클러스터 토픽 호스트를 포함한, 클러스터의 다른 큐 관리자로 푸시됩니다. 60일 이상 클러스터 토픽 호스트를 사용할 수 없게 되는 경우, 해당 버전의 클러스터 토픽 오브젝트만 다른 호스트에서 제거됩니다. 클러스터 토픽 오브젝트의 다른 버전은 남아 있습니다. 클러스터에서 특정 토픽에 대한 모든 정의가 동일해야 하며, 그렇지 않으면 큐 관리자에서 어떤 토픽 정의를 사용하고 있는지 확인하기 어렵습니다. 호스트의 최근 사본은 항상 사용되는 클러스터 토픽 오브젝트입니다.

증가된 관리의 복잡도에 대응하기 위해 여러 클러스터 토픽 정의에 대한 추가적인 보호 조치에 비중을 두십시오. 복잡도가 증가하면 작업에 대한 실수가 늘어나게 됩니다.

클러스터된 큐 호스트와 달리, 클러스터된 토픽 정의에 대한 호스트 큐 관리자는 추가 애플리케이션 메시지 트래픽을 소개하지 않습니다. 이 트래픽은 구독이 작성되고 메시지가 발행된 큐 관리자로 제한됩니다. 아무 것도 수행하지 않는 큐 관리자에서 클러스터된 토픽을 호스트할 수 있습니다. 즉, 필수적인 것은 아니지만 이런 큐 관리자가 더 높은 레벨의 가용성으로 제공되고 클러스터된 토픽에 대해 더욱 긴밀한 관리 제어 능력을 가질 수 있으므로 클러스터에 대한 전체 저장소 큐 관리자에 이런 토픽을 호스트하는 것이 합당할 때가 많습니다. 이렇게 하면 정의나 심지어는 큐 관리자까지도 틀리게 수정하거나 삭제할 가능성이 줄어듭니다.

겹치는 클러스터 지원 및 발행/구독

IBM WebSphere MQ 클러스터를 사용하면 단일 큐 관리자가 둘 이상의 클러스터의 구성원이 될 수 있습니다. 이 지정은 클러스터 겹치기라고도 합니다. 하나의 큐 관리자에 클러스터가 겹칠 때 발행/구독 클러스터에 클러스터된 토픽은 서로 다르게 작동합니다. 겹치는 클러스터가 있는 클러스터된 발행/구독을 사용할 때 이 작동을 분명히 이해해야 합니다.

큐와 달리, 둘 이상의 클러스터와 토픽 정의를 연관시킬 수 없습니다. 따라서 클러스터에서 작성된 프록시 구독의 범위는 클러스터된 토픽이 정의된 단일 클러스터로 제한됩니다. 하지만, 각각의 큐 관리자에는 모든 로컬 토픽과 이들이 멤버인 클러스터에서 알려진 클러스터된 토픽을 포함하는 단일 토픽 트리기가 있습니다. 이 때문에, 발행/구독 작동이 이해하기 어려울 수 있는 시스템을 설계할 가능성이 있습니다.

다중 발행/구독 클러스터 통합

포인트-투-포인트 메시지의 경우, 단일 큐 관리자를 둘 이상의 클러스터의 멤버로 만드는 이유는 두 클러스터 사이에서 클러스터 게이트웨이를 작성하기 위한 것입니다. 이 주제에 대한 자세한 정보는 클러스터 겹침을 참조하십시오. 이 클러스터 게이트웨이를 사용하면 한 클러스터에서 생성되는 포인트-투-포인트 메시지를 다른 클러스터의 조회로 라우팅할 수 있습니다. 발행/구독 클러스터는 기존 큐 관리자 클러스터에서 겹치는 기능을 상속했습니다. 하지만, 이 메커니즘을 사용하여 한 클러스터에서 다른 클러스터로 발행물과 구독을 라우팅할 수는 없습니다.

그 대신, 한 클러스터에 있는 큐 관리자를 다른 클러스터에 있는 큐 관리자로 발행물과 구독을 전달하려면 발행/구독 계층을 사용하여 큐 관리자를 함께 연결해야 합니다. 기타 클러스터의 다른 큐 관리자와 함께 하나의 클러스터의 하나의 큐 관리자 간에 상위-하위 계층 구조 관계를 명시적으로 작성하여 이 지정을 수행할 수 있습니다. 이 관계는 클러스터 간에 모든 프로시 구독의 플로우를 사용하여 발행물을 일치시킵니다. 이 관계에 대해서는 74 페이지의 『발행/구독 계층』을 참조하십시오.

클러스터 사이에서 플로우할 발행물과 구독을 제한하는 한 가지 방법은 어느 클러스터에도 없는 게이트웨이 큐 관리자를 사용하는 것입니다(91 페이지의 『다중 클러스터의 토픽 공간 결합 및 분리』 참조).

겹치는 클러스터, 단일 토픽 트리

각각의 큐 관리자에는 로컬 토픽과 알려진 모든 클러스터 토픽을 포함하는 단일 토픽 트리가 있습니다. 구독/발행 모두를 사용한 두 클러스터 겹치기에 대해 각 클러스터의 큐 관리자가 동일한 이름으로 클러스터된 토픽을 정의하거나 동일한 토픽 문자열이 있는 이름이 다른 클러스터된 토픽을 정의하는 것이 가능하다는 점을 추가로 고려해야 합니다. 두 클러스터의 멤버인 큐 관리자에서는 다중 클러스터 토픽 정의가 알려지는 경우 각 클러스터에 대해 한 번씩 충돌이 발생할 수 있습니다. 큐 관리자가 문제점을 보고하지만 큐 관리자는 가장 최근에 클러스터된 토픽 정의만을 사용하여 계속 조작합니다. 그러므로 작동은 비결정적이 되어 기대할 수 없습니다.

이 때문에, 클러스터된 발행/구독을 사용하는 겹치는 클러스터는 토픽 정의 네임스페이스의 범위가 모든 클러스터에 이르는 것으로 간주하고, 그에 따라 적절히 토픽 오브젝트의 이름을 지정하고 토픽 문자열을 구성해야 합니다. 그 다음, 겹치는 부분의 큐 관리자를 사용하여 두 개의 클러스터에서 주기적으로 발행 및 구독할 수 있습니다.

72 페이지의 그림 33에서, T_B 및 T_C 는 겹치지 않는 토픽 정의입니다. 클러스터 겹침에서 QM3에 연결된 발행자는 각 클러스터에서 두 토픽 모두에 대해 발행할 수 있습니다. 겹침에서 QM3에 연결된 구독자는 두 클러스터 모두에 있는 토픽에 대해 구독할 수 있습니다.

72 페이지의 그림 33에 대한 대체 방법으로 프로시 구독을 고려합니다. 큐 관리자 QM3에 연결된 애플리케이션이 토픽 오브젝트 T_B (CLUSTER 1에만 존재함)로 해석되는 토픽을 구독하면 프로시 구독이 큐 관리자 QM3에서 큐 관리자 QM1 및 QM2만 송신됩니다. 큐 관리자 QM3에 연결된 애플리케이션은 (CLUSTER 2에만 있는) 토픽 오브젝트 T_C 로 해석되는 토픽에 대해 구독합니다. 이 구독으로 인해 큐 관리자 QM3에서 큐 관리자 QM4 및 QM5로만 프로시 구독이 발송됩니다.

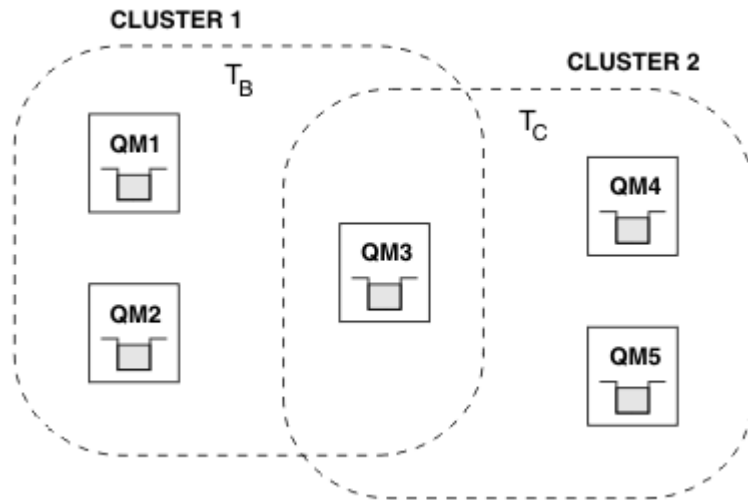


그림 33. 겹치는 클러스터: 각각 서로 다른 토픽을 구독하는 두 개의 클러스터

겹침에 없는 큐 관리자에 대한 발행자와 구독자는 자신의 클러스터에만 있는 토픽에 대해 발행하고 구독할 수 있습니다. 예를 들어, QM2에서 토픽 문자열에 대한 구독자는 토픽의 클러스터링과는 상관없이 QM5에서 발행된 같은 토픽 문자열에 대해 발행된 메시지를 수신하지 않습니다. 이 배열을 달성하려면 발행/구독 계층이 필요합니다.

겹치는 클러스터, 와일드카드로 표시된 구독

이 토픽의 이전 섹션에서 설명한 이유 때문에, 다중 클러스터의 멤버인 큐 관리자에서 토픽을 구독하기 위해 와일드카드를 사용할 때 주의해야 합니다.

이전 예에서는 두 개의 토픽 오브젝트가 다음과 같이 구성된 것으로 가정합니다.

- T_B: 토픽 이름 'Football', 클러스터 'CLUSTER1'. 토픽 문자열 '/Sport/Football'
- T_C: 토픽 이름 'Tennis', 클러스터 'CLUSTER2'. 토픽 문자열 '/Sport/Tennis'

이 시나리오에서는 토픽 이름 또는 토픽 문자열에 겹침이 없이 두 개의 클러스터된 토픽이 명확히 구분됩니다.

QM3에 연결된 애플리케이션은 '/Sport/Football'에 대한 구독과 '/Sport/Tennis'에 대한 구독을 작성할 수 있습니다. 그러면 두 클러스터로부터 모든 발행물을 수신하게 됩니다. 하지만, 33 페이지의 『관리 토픽 오브젝트』에 설명된 것처럼 '/Sport/Football'과 '/Sport/Tennis' 모두에 대한 발행물을 수신할 의도로 '/Sport/#'에 대해 구독한 경우, 어느 클러스터에서도 이 모델이 클러스터된 토픽으로 인식되지 않으므로 프록시 구독이 작성되지 않습니다. 그러면 어느 한쪽의 클러스터에 있는 다른 큐 관리자에서 제공되는 발행물을 놓치게 됩니다.

이미 설명한 대로 CLUSTER 1 및 CLUSTER 2 둘 다에서 '/Sport/#'에 대해 클러스터된 토픽을 작성하는 것은 유효하지 않습니다. 이러한 클러스터된 토픽이 충돌하고 이를 표시하기 위해 오류 로그에 정보 메시지가 기록되기 때문입니다. 그러나 클러스터 중 하나에 이러한 주제를 작성하는 것은 '허용'입니다 (CLUSTER 1). Now a subscription to '/Sport/#' in QM3 would result in proxy subscriptions being sent to the queue managers in CLUSTER 1 only, so still, publications to '/Sport/Tennis' from QM4 or QM5 would still fail to be received.

이 시나리오에서 유일한 해결책은 계속 두 개의 구독을 별도로 작성하는 것입니다.

발행/구독 클러스터에 대한 REFRESH CLUSTER 고려사항

REFRESH CLUSTER 명령을 발행하면 큐 관리자가 클러스터 토픽 및 연관된 프록시 구독을 포함하여, 로컬에 보유된 클러스터에 대한 정보를 일시적으로 제거합니다.

REFRESH CLUSTER 명령 발행에서 큐 관리자가 클러스터된 발행/구독에 대한 필요한 전체 정보를 다시 수집한 시점까지 걸리는 시간은 전체 저장소 큐 관리자의 응답성, 가용성, 클러스터의 크기에 따라 다릅니다.

새로 고치기 처리 중에는 발행/구독 클러스터의 발행/구독 트래픽 방해가 발생합니다. 대형 클러스터의 경우, **REFRESH CLUSTER** 명령을 사용하면 진행 중인 클러스터를 방해할 수 있으며, 클러스터 오브젝트가 모든 관심 있는 큐 관리자에게 자동으로 상태 업데이트를 보내는 경우에는 다시 27일 간격으로 클러스터를 중단할 수 있습니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오. 이러한 이유로 **REFRESH CLUSTER** 명령은 IBM 지원 센터의 지시가 있을 때에만 발행/구독 클러스터에 사용해야 합니다.

클러스터 방해는 외부에서 보면 다음 증상으로 표시될 수 있습니다.

- 이 큐 관리자의 클러스터 토픽에 대한 구독이 클러스터의 기타 큐 관리자에 연결된 발행자로부터 발행물을 수신하지 않습니다.
- 이 큐 관리자의 클러스터 토픽에 발행된 메시지가 기타 큐 관리자의 구독에 전파되지 않습니다.
- 이 기간 중 작성된 이 큐 관리자의 클러스터 토픽에 대한 구독이 클러스터의 기타 멤버에 프록시 구독을 일관되게 보내지 않습니다.
- 이 기간 중 삭제된 이 큐 관리자의 클러스터 토픽에 대한 구독이 클러스터의 기타 멤버에서 프록시 구독을 일관되게 제거하지 않습니다.
- 메시지 전달 시 10초 이상 일시정지됩니다.
- MQPUT 실패(예: MQRC_PUBLICATION_FAILURE).
- 발행물이 데드-레터 큐에 배치되었습니다(이유: MQRC_UNKNOWN_REMOTE_Q_MGR).

위와 같은 이유로 **REFRESH CLUSTER** 명령을 발행하기 전에 발행/구독 애플리케이션을 일시정지해야 합니다.

REFRESH CLUSTER에 대한 사용 노트 및 클러스터링: REFRESH CLUSTER 우수 사례 사용도 참조하십시오.

REFRESH CLUSTER 명령이 발행/구독 클러스터의 큐 관리자에서 실행된 후, 모든 클러스터 큐 관리자 및 클러스터 토픽이 성공적으로 새로 고쳐질 때까지 기다린 후 63 페이지의 『프록시 구독의 재동기화』에 설명된 대로 프록시 구독을 다시 동기화하십시오. 이 배열에서는 클러스터에 있는 다른 모든 큐 관리자에 대해 이 큐 관리자에서 클러스터 송신자 채널을 시작해야 합니다. 모든 프록시 구독이 제대로 재동기화되면 발행/구독 애플리케이션을 다시 시작하십시오.

REFRESH CLUSTER 명령을 완료하는 데 많은 시간이 소요되는 경우, SYSTEM.CLUSTER.COMMAND.QUEUE의 CURDEPTH를 보고 모니터링하십시오.

관련 개념

REFRESH CLUSTER를 실행할 때 표시되는 애플리케이션 문제

[클러스터링: REFRESH CLUSTER 사용 우수 사례](#)

관련 참조

[MQSC 명령 참조: REFRESH CLUSTER](#)

발행/구독 계층

계층에서 큐 관리자를 그룹화할 수 있으며, 계층에는 직접 연결된 하나 이상의 큐 관리자가 포함되어 있습니다. 큐 관리자는 연결 시간 상위 및 하위 관계를 사용하여 서로 연결되어 있습니다. 처음에 두 개의 큐 관리자가 서로 연결되면 하위 큐 관리자가 상위 큐 관리자에 연결됩니다.

상위 및 하위 큐 관리자가 계층에서 연결되어 있는 경우, 계층에서 큐 관리자의 연결이 끊어질 때까지 두 개의 큐 관리자 사이에 기능적 차이는 없습니다.

참고: IBM WebSphere MQ 계층 연결에서는 큐 관리자 속성 PSMODE가 ENABLED로 설정되어야 합니다.

Hierarchy

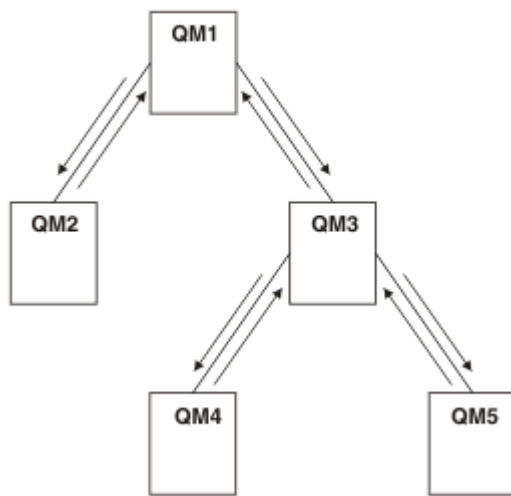


그림 34. 간단한 발행/구독 계층

큐 관리자를 브로커 계층에 연결

로컬 큐 관리자를 상위 큐 관리자에 연결하여 브로커 계층을 수정할 수 있습니다.

시작하기 전에

1. 큐된 발행/구독 모드를 사용하도록 설정합니다. [큐된 발행/구독 시작](#)을 참조하십시오.
2. 이 변경사항은 IBM WebSphere MQ 연결을 사용하여 상위 큐 관리자에 전파됩니다. 연결 설정 방법은 두 가지입니다.
 - 큐 관리자를 IBM WebSphere MQ 클러스터에 연결하십시오. [클러스터에 큐 관리자 추가](#)를 참조하십시오.
 - 이름이 상위 큐 관리자와 동일한 전송 큐 또는 큐 관리자 알리어스를 사용하여 포인트-투-포인트 채널 연결을 설정합니다. 지점간 채널 연결을 설정하는 방법에 대한 자세한 정보는 [WebSphere MQ 분산 메시징 기술](#)을 참조하십시오.

이 태스크 정보

ALTER QMGR PARENT (PARENT_NAME) runmqsc 명령을 사용하여 하위 항목과 상위 항목을 연결합니다.

분산 발행/구독은 큐 관리자 클러스터와 클러스터된 토픽 정의를 사용하여 구현됩니다. IBM WebSphere MQ Version 6.0 및 WebSphere Message Broker Version 6.1 및 WebSphere Event Broker Version 6.1 이전 버전과의 상호 운영성을 위해, 큐된 발행/구독 모드가 사용 가능한 한 Version 7.1 이상의 큐 관리자를 브로커 계층에 연결할 수도 있습니다.

프로시저

ALTER QMGR PARENT(PARENT)

예

첫 번째 예는 QM1의 하위로 QM2를 연결한 후 해당 연결에 대해 QM2를 조회하는 방법을 보여줍니다.

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
  1 : alter qmgr parent(QM1)
AMQ8005: WebSphere MQ queue manager changed.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
      QMNAME(QM2)                TYPE(LOCAL)
      STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
      QMNAME(QM1)                TYPE(PARENT)
      STATUS(ACTIVE)
```

다음 예는 해당 연결에 대해 QM1을 조회한 결과를 보여줍니다.

```
C:\Documents and Settings\Admin>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
      QMNAME(QM1)                TYPE(LOCAL)
      STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
      QMNAME(QM2)                TYPE(CHILD)
      STATUS(ACTIVE)
```

다음에 수행할 작업

연결된 큐 관리자의 발행자 및 구독자에 사용 가능한 하나의 브로커 또는 큐 관리자에 대해 토픽을 정의할 수 있습니다. 자세한 정보는 [관리 토픽 정의](#)를 참조하십시오.

관련 개념

[스트림 및 토픽](#)

[eSphere MQ 발행/구독 메시징 소개](#)

관련 참조

[DISPLAY PUBSUB](#)

브로커 계층에서 큐 관리자의 연결 끊기

브로커 계층의 상위 큐 관리자에서 하위 큐 관리자의 연결을 끊습니다.

이 태스크 정보

ALTER QMGR 명령을 사용하여 브로커 계층에서 큐 관리자의 연결을 끊을 수 있습니다. 언제든지 원하는 순서로 큐 관리자 연결을 끊을 수 있습니다.

상위를 업데이트하기 위한 해당 요청은 큐 관리자 사이의 연결이 실행 중일 때 송신됩니다.

프로시저

ALTER QMGR PARENT('')

예

```
C:\Documents and Settings\Admin>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
  1 : alter qmgr parent('')
```

```
AMQ8005: WebSphere MQ queue manager changed.
  2 : display pubsub type(child)
AMQ8147: WebSphere MQ object not found.
display pubsub type(parent)
  3 : display pubsub type(parent)
AMQ8147: WebSphere MQ object not found.
```

다음에 수행할 작업

더 이상 필요하지 않은 스트림, 큐, 수동으로 정의된 채널을 삭제할 수 있습니다.

발행/구독 계층의 예: 시나리오 1

큐 관리자 이름 알리어스를 가진 포인트-투-포인트 채널을 사용하여 발행/구독 계층 토폴로지를 설정합니다.

이 태스크 정보

이런 시나리오에서는 큐 관리자 간에 연결을 설정하기 위해 여러 가지 방법으로 발행/구독 계층을 설정합니다. 이런 시나리오에서는 모두 QM1이라는 상위 큐 관리자와 QM2 및 QM3이라는 2개의 하위 큐 관리자를 사용합니다.

시나리오 1은 프로세스를 더 쉽게 따라할 수 있도록 더 작은 섹션으로 분할합니다.

시나리오 1 파트 1: 큐 관리자 작성

프로시저

1. 다음 명령을 사용하여 QM1, QM2 및 QM3이라고 하는 세 개의 큐 관리자를 작성하고 시작하십시오.

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3
```

2. 세 개의 큐 관리자 모두에서 다음 명령을 사용하여 큐 관리자 발행/구독 모드를 사용하도록 설정하십시오.

```
ALTER QMGR PSMODE(ENABLED)
```

시나리오 1 파트 2: 포인트-투-포인트 채널 연결

이 태스크 정보

상위 큐 관리자와 동일한 이름과 함께 큐 관리자 알리어스를 사용하여 큐 관리자 사이에 포인트-투-포인트 채널 연결을 설정합니다.

프로시저

1. QM1에 대해 QM2의 전송 큐 및 큐 관리자 알리어스를 정의하십시오. QM1에 대해 송신자 채널을 정의하고 QM2에 대해 QM1에서 작성된 송신자 채널의 수신자 채널을 정의하십시오.

```
DEFINE QLOCAL(QM1.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE(QM1) RNAME('') RQMNAME(QM1) XMITQ(QM1.XMITQ)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1.XMITQ)
TRPTYPE(TCP)

DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(RCVR) TRPTYPE(TCP)
```

2. QM1에 대해 QM3의 전송 큐 및 큐 관리자 알리어스를 정의하십시오. QM1에 대해 송신자 채널을 정의하고 QM3에 대해 QM1에서 작성된 송신자 채널의 수신자 채널을 정의하십시오.

```
DEFINE QLOCAL(QM1.XMITQ) USAGE(XMITQ)
```



```

DEFINE QREMOTE (QM1) RNAME(' ') RQMNAME(QM1) XMITQ(QM1.XMITQ)

DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1.XMITQ)
TRPTYPE(TCP)

DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(RCVR) TRPTYPE(TCP)

```

3. QM2 및 QM3에 대해 QM1의 전송 큐 및 큐 관리자 알리어스를 정의하십시오. QM2 및 QM3에 대해 송신자 채널을 정의하고, QM1에 대해 QM2 및 QM3에서 작성된 송신자 채널의 수신자 채널을 정의하십시오.

```

DEFINE QLOCAL(QM2.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE (QM2) RNAME(' ') RQMNAME(QM2) XMITQ(QM2.XMITQ)

DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(SDR) CONNAME('localhost(7777)') XMITQ(QM2.XMITQ)
TRPTYPE(TCP)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)

DEFINE QLOCAL(QM3.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM3.XMITQ)

DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(SDR) CONNAME('localhost(8888)') XMITQ(QM3.XMITQ)
TRPTYPE(TCP)

DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)

```

4. 큐 관리자에서 적절한 리스너를 시작하십시오.

```

runmqclsr -m QM1 -t TCP -p 9999 &
runmqclsr -m QM2 -t TCP -p 7777 &
runmqclsr -m QM3 -t TCP -p 8888 &

```

5. 다음 채널을 시작하십시오.

a. QM1:

```

START CHANNEL('QM1.TO.QM2')
START CHANNEL('QM1.TO.QM3')

```

b. QM2:

```

START CHANNEL('QM2.TO.QM1')

```

c. QM3:

```

START CHANNEL('QM3.TO.QM1')

```

6. 모든 채널이 시작되었는지 확인하십시오.

```

DISPLAY CHSTATUS('QM1.TO.QM2')
DISPLAY CHSTATUS('QM1.TO.QM3')
DISPLAY CHSTATUS('QM2.TO.QM1')
DISPLAY CHSTATUS('QM3.TO.QM1')

```

시나리오 1 파트 3: 큐 관리자를 연결하고 토픽을 정의합니다.

이 태스크 정보

상위 큐 관리자 QM1에 하위 큐 관리자 QM2 및 QM3을 연결합니다.

프로시저

1. QM2 및 QM3에서 상위 큐 관리자를 QM1로 설정하십시오.

```
ALTER QMGR PARENT (QM1)
```

2. 하위 큐 관리자가 상위 큐 관리자에 연결되어 있는지 확인하기 위해 모든 큐 관리자에 대해 다음 명령을 실행하십시오.

```
DISPLAY PUBSUB TYPE(ALL)
```

3. 토픽 오브젝트를 정의하십시오.

```
define topic(FOOTBALL) TOPICSTR('Sport/Soccer')
```

시나리오 1 파트 4: 토픽 발행 및 구독

이 태스크 정보

토픽을 발행하고 구독하기 위해 `amqspub.exe` 및 `amqssub.exe` 애플리케이션을 사용합니다.

프로시저

1. 첫 번째 명령 창에서 이 명령을 실행하십시오.

```
amqspub Sport/Soccer QM2
```

2. 두 번째 명령 창에서 이 명령을 실행하십시오.

```
amqssub Sport/Soccer QM1
```

3. 세 번째 명령 창에서 이 명령을 실행하십시오.

```
amqssub Sport/Soccer QM3
```

결과

두 번째와 세 번째 명령 창의 `amqssub.exe` 애플리케이션은 첫 번째 명령 창에서 발행된 메시지를 수신합니다.

관련 태스크

78 페이지의 『[발행/구독 계층의 예: 시나리오 2](#)』

원격 큐 관리자와 같은 전송 큐 이름을 가진 포인트-투-포인트 채널을 사용하여 발행/구독 계층 토폴로지를 설정합니다.

81 페이지의 『[발행/구독 계층의 예: 시나리오 3](#)』

클러스터 채널을 사용하여 계층 토폴로지에 큐 관리자를 추가합니다.

발행/구독 계층의 예: 시나리오 2

원격 큐 관리자와 같은 전송 큐 이름을 가진 포인트-투-포인트 채널을 사용하여 발행/구독 계층 토폴로지를 설정합니다.

이 태스크 정보

이런 시나리오에서는 큐 관리자 간에 연결을 설정하기 위해 여러 가지 방법으로 발행/구독 계층을 설정합니다. 이런 시나리오에서는 모두 QM1이라는 상위 큐 관리자와 QM2 및 QM3이라는 2개의 하위 큐 관리자를 사용합니다.

시나리오 2는 프로세스를 더 쉽게 따라할 수 있도록 더 작은 섹션으로 분할합니다. 이 시나리오는 76 페이지의 『[발행/구독 계층의 예: 시나리오 1](#)』에서 시나리오 1 파트 1, 시나리오 1 파트 3, 시나리오 1 파트 4를 재사용합니다.

시나리오 2 파트 1: 큐 관리자 작성 및 `PSMODE` 설정

프로시저

1. 다음 명령을 사용하여 QM1, QM2 및 QM3이라고 하는 세 개의 큐 관리자를 작성하고 시작하십시오.

```

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3

```

2. 세 개의 큐 관리자 모두에서 다음 명령을 사용하여 큐 관리자 발행/구독 모드를 사용하도록 설정하십시오.

```
ALTER QMGR PSMODE(ENABLED)
```

시나리오 2 파트 2: 포인트-투-포인트 채널 연결

이 태스크 정보

상위 큐 관리자와 동일한 이름으로 전송 큐를 사용하여 큐 관리자 사이의 포인트-투-포인트 채널 연결을 설정합니다.

프로시저

1. QM1에 대해 QM2의 전송 큐를 정의하십시오. QM1에 대해 송신자 채널을 정의하고 QM1에서 작성된 QM2에 대한 송신자 채널의 수신자 채널을 정의하십시오.

```

DEFINE QLOCAL(QM1) USAGE(XMITQ)
DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1) TRPTYPE(TCP)
DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(RCVR) TRPTYPE(TCP)

```

2. QM1에 대해 QM3의 전송 큐를 정의하십시오. QM1에 대해 송신자 채널을 정의하고 QM3에 대해 QM1에서 작성된 송신자 채널의 수신자 채널을 정의하십시오.

```

DEFINE QLOCAL(QM1) USAGE(XMITQ)
DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1) TRPTYPE(TCP)
DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(RCVR) TRPTYPE(TCP)

```

3. QM2 및 QM3에 대해 QM1에서 전송 큐를 정의하십시오. QM2 및 QM3에 대해 송신자 채널을 정의하고, QM1에 대해 QM2 및 QM3에서 작성된 송신자 채널의 수신자 채널을 정의하십시오.

```

DEFINE QLOCAL(QM2) USAGE(XMITQ)
DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(SDR) CONNAME('localhost(7777)') XMITQ(QM2) TRPTYPE(TCP)
DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE QLOCAL(QM3) USAGE(XMITQ)
DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(SDR) CONNAME('localhost(8888)') XMITQ(QM3) TRPTYPE(TCP)
DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)

```

4. 큐 관리자에서 적절한 리스너를 시작하십시오.

```

runmqclsr -m QM1 -t TCP -p 9999 &
runmqclsr -m QM2 -t TCP -p 7777 &
runmqclsr -m QM3 -t TCP -p 8888 &

```

5. 다음 채널을 시작하십시오.

a. QM1:

```

START CHANNEL('QM1.TO.QM2')
START CHANNEL('QM1.TO.QM3')

```

b. QM2:

```
START CHANNEL('QM2.TO.QM1')
```

c. QM3:

```
START CHANNEL('QM3.TO.QM1')
```

6. 모든 채널이 시작되었는지 확인하십시오.

```
DISPLAY CHSTATUS('QM1.TO.QM2')
DISPLAY CHSTATUS('QM1.TO.QM3')
DISPLAY CHSTATUS('QM2.TO.QM1')
DISPLAY CHSTATUS('QM3.TO.QM1')
```

시나리오 2 파트 3: 큐 관리자를 연결하고 토픽을 정의합니다.

이 태스크 정보

상위 큐 관리자 QM1에 하위 큐 관리자 QM2 및 QM3을 연결합니다.

프로시저

1. QM2 및 QM3에서 상위 큐 관리자를 QM1로 설정하십시오.

```
ALTER QMGR PARENT (QM1)
```

2. 하위 큐 관리자가 상위 큐 관리자에 연결되어 있는지 확인하기 위해 모든 큐 관리자에 대해 다음 명령을 실행하십시오.

```
DISPLAY PUBSUB TYPE(ALL)
```

3. 토픽 오브젝트를 정의하십시오.

```
define topic(FOOTBALL) TOPICSTR('Sport/Soccer')
```

시나리오 2 파트 4: 토픽을 발행하고 구독합니다.

이 태스크 정보

토픽을 발행하고 구독하기 위해 `amqspub.exe` 및 `amqssub.exe` 애플리케이션을 사용합니다.

프로시저

1. 첫 번째 명령 창에서 이 명령을 실행하십시오.

```
amqspub Sport/Soccer QM2
```

2. 두 번째 명령 창에서 이 명령을 실행하십시오.

```
amqssub Sport/Soccer QM1
```

3. 세 번째 명령 창에서 이 명령을 실행하십시오.

```
amqssub Sport/Soccer QM3
```

결과

두 번째와 세 번째 명령 창의 `amqssub.exe` 애플리케이션은 첫 번째 명령 창에서 발행된 메시지를 수신합니다.

관련 태스크

76 페이지의 『발행/구독 계층의 예: 시나리오 1』

큐 관리자 이름 알리어스를 가진 포인트-투-포인트 채널을 사용하여 발행/구독 계층 토폴로지를 설정합니다.

81 페이지의 『발행/구독 계층의 예: 시나리오 3』

클러스터 채널을 사용하여 계층 토폴로지에 큐 관리자를 추가합니다.

발행/구독 계층의 예: 시나리오 3

클러스터 채널을 사용하여 계층 토폴로지에 큐 관리자를 추가합니다.

이 태스크 정보

이런 시나리오에서는 큐 관리자 간에 연결을 설정하기 위해 여러 가지 방법으로 발행/구독 계층을 설정합니다. 이런 시나리오에서는 모두 QM1이라는 상위 큐 관리자와 QM2 및 QM3이라는 2개의 하위 큐 관리자를 사용합니다.

시나리오 3은 프로세스를 더 쉽게 따라할 수 있도록 더 작은 섹션으로 분할합니다. 이 시나리오는 76 페이지의 『발행/구독 계층의 예: 시나리오 1』에서 시나리오 1 파트 1, 시나리오 1 파트 3, 시나리오 1 파트 4를 재사용합니다.

이 시나리오는 QM1 및 QM2가 전체 저장소이고 QM3이 부분 저장소인 DEMO 클러스터를 작성합니다. QM1 큐 관리자는 QM2 및 QM3 큐 관리자의 상위 큐 관리자입니다.

시나리오 2 파트 1: 큐 관리자 작성 및 PSMODE 설정

프로시저

1. 다음 명령을 사용하여 QM1, QM2 및 QM3이라고 하는 세 개의 큐 관리자를 작성하고 시작하십시오.

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3
```

2. 세 개의 큐 관리자 모두에서 다음 명령을 사용하여 큐 관리자 발행/구독 모드를 사용하도록 설정하십시오.

```
ALTER QMGR PSMODE(ENABLED)
```

시나리오 2 파트 2: 포인트-투-포인트 채널 연결

이 태스크 정보

큐 관리자와 클러스터 사이의 포인트-투-포인트 채널 연결을 설정합니다.

프로시저

1. QM1 및 QM2에서 **REPOS** 매개변수를 DEMO 클러스터의 이름으로 설정하십시오.

```
ALTER QMGR REPOS(DEMO)
```

2. 큐 관리자에서 적절한 리스너를 시작하십시오.

```
runmqclsr -m QM1 -t TCP -p 9999 &
runmqclsr -m QM2 -t TCP -p 7777 &
runmqclsr -m QM3 -t TCP -p 8888 &
```

3. 각 큐 관리자에 클러스터 수신자 채널을 정의하십시오.

a. QM1:

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(9999)')
CLUSTER(DEMO)
```

b. QM2:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(7777)')
CLUSTER(DEMO)
```

c. QM3:

```
DEFINE CHANNEL(TO.QM3) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(8888)')
CLUSTER(DEMO)
```

4. 클러스터의 각 큐 관리자에서 전체 저장소에 대해 클러스터 송신자 채널을 정의하십시오.

a. QM1:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(7777)')
CLUSTER(DEMO)
```

b. QM2:

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(9999)')
CLUSTER(DEMO)
```

c. QM3에는 QM1 또는 QM2의 전체 저장소에 대한 클러스터 송신자 채널이 있을 수 있습니다. 이 예에서는 QM1에 대해 채널을 정의합니다.

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(9999)')
CLUSTER(DEMO)
```

시나리오 2 파트 3: 큐 관리자를 연결하고 토픽을 정의합니다.

이 태스크 정보

상위 큐 관리자 QM1에 하위 큐 관리자 QM2 및 QM3을 연결합니다.

프로시저

1. QM2 및 QM3에서 상위 큐 관리자를 QM1로 설정하십시오.

```
ALTER QMGR PARENT (QM1)
```

2. 하위 큐 관리자가 상위 큐 관리자에 연결되어 있는지 확인하기 위해 모든 큐 관리자에 대해 다음 명령을 실행하십시오.

```
DISPLAY PUBSUB TYPE(ALL)
```

3. 토픽 오브젝트를 정의하십시오.

```
define topic(FOOTBALL) TOPICSTR('Sport/Soccer')
```

시나리오 2 파트 4: 토픽을 발행하고 구독합니다.

이 태스크 정보

토픽을 발행하고 구독하기 위해 amqspub.exe 및 amqssub.exe 애플리케이션을 사용합니다.

프로시저

1. 첫 번째 명령 창에서 이 명령을 실행하십시오.


```
amqspub Sport/Soccer QM2
```

2. 두 번째 명령 창에서 이 명령을 실행하십시오.

```
amqssub Sport/Soccer QM1
```

3. 세 번째 명령 창에서 이 명령을 실행하십시오.

```
amqssub Sport/Soccer QM3
```

결과

두 번째와 세 번째 명령 창의 `amqssub.exe` 애플리케이션은 첫 번째 명령 창에서 발행된 메시지를 수신합니다.

관련 태스크

76 페이지의 『발행/구독 계층의 예: 시나리오 1』

큐 관리자 이름 알리어스를 가진 포인트-투-포인트 채널을 사용하여 발행/구독 계층 토폴로지를 설정합니다.

78 페이지의 『발행/구독 계층의 예: 시나리오 2』

원격 큐 관리자와 같은 전송 큐 이름을 가진 포인트-투-포인트 채널을 사용하여 발행/구독 계층 토폴로지를 설정합니다.

발행 및 구독 플로우 제어

분산 발행/구독 토폴로지에 함께 연결되는 큐 관리자는 공용 연합 토픽 공간을 공유합니다. 발행 및 구독이 로컬 인지 글로벌인지를 선택하여 토폴로지 내에서 발행 및 구독의 플로우를 제어할 수 있습니다.

로컬 발행과 구독은 발행자 또는 구독자가 연결된 큐 관리자를 넘어 전파되지 않습니다.

클러스터 또는 계층에서 큐 관리자를 모두 연결하여 작성된 토픽 공간의 범위를 제어할 수 있습니다. 발행/구독 클러스터에서 토픽 오브젝트는 '클러스터'되어야 합니다. 그렇지 않으면 모든 요소가 로컬로 유지되고 발행 또는 구독이 적용되지 않습니다.

구독이 여러 발행의 토픽 문자열과 일치하면 구독이 여러 토픽 오브젝트로 해석될 수 있습니다. 이를 토픽 겹침이라고 합니다. 특정 일치 내용에 대해 발행과 연관된 토픽 오브젝트에서는 토픽 속성을 제공하고 구독자가 발행물을 수신하는지의 여부를 판별합니다.

발행 범위

발행물 범위는 큐 관리자가 발행물을 리모트 큐 관리자에게 전달하는지의 여부를 제어합니다. **PUBSCOPE** 토픽 속성을 사용하여 발행 범위를 관리합니다.

발행이 리모트 큐 관리자에 전달되지 않으면 로컬 구독자만 발행을 수신합니다.

PUBSCOPE 토픽 속성을 사용하여 특정 토픽에 대한 발행물 범위를 결정할 수 있습니다. 속성을 다음 값 중 하나로 설정할 수 있습니다.

큐 관리자

발행은 로컬 구독자에게만 전달됩니다. 이러한 발행을 로컬 발행이라고 합니다. 로컬 발행은 리모트 큐 관리자로 전달되지 않으므로 리모트 큐 관리자에 연결된 구독자는 수신하지 않습니다.

모두

발행물은 로컬 구독자 및 리모트 큐 관리자에게 연결된 구독자에게 전달됩니다. 이러한 발행을 글로벌 발행이라고 합니다.

ASPARENT

상위의 **PUBSCOPE** 설정을 사용합니다.

또한 발행자는 `MQPMO_SCOPE_QMGR` 입력 메시지 옵션을 사용하여 발행 종류(로컬 또는 글로벌)를 지정할 수 있습니다. 이 옵션을 사용하면 **PUBSCOPE** 토픽 속성을 사용하여 설정된 동작을 대체합니다.

구독 범위

구독 범위는 한 큐 관리자의 구독이 발행/구독 클러스터 또는 계층에 있는 다른 큐 관리자에서 발행된 발행을 수신하는지, 아니면 로컬 발행자의 발행을 수신하는지 여부를 제어합니다.

큐 관리자로 구독 범위를 제한하면 프록시 구독이 발행/구독 토폴로지에 있는 다른 큐 관리자로 전달되는 작업이 중지됩니다. 이로 인해 큐 관리자 내 발행/구독 메시징 트래픽을 줄입니다.

SUBSCOPE 토픽 속성을 사용하여 특정 토픽에 대해 작성된 구독 범위를 결정할 수 있습니다. 속성을 다음 값 중 하나로 설정할 수 있습니다.

큐 관리자

구독은 로컬 발행만 수신하며 프록시 구독은 리모트 큐 관리자로 전파되지 않습니다.

모두

프록시 구독이 리모트 큐 관리자로 전파되며 구독자가 로컬 및 리모트 발행물을 받습니다.

ASPARENT

상위에 대한 **SUBSCOPE** 설정을 사용합니다.

개별 등록자는 등록을 작성할 때 MQSO_SCOPE_QMGR 등록 옵션을 지정하여 ALL의 **SUBSCOPE** 설정을 대체할 수 있습니다. 구독은 토픽의 **SUBSCOPE** 설정인 ALL을 대체할 수 있습니다.

참고: 개별 구독자는 토픽의 **SUBSCOPE**만을 제한할 수 있습니다. 개별 구독에서 **SUBSCOPE**를 ALL로 설정하면, 구독은 일치하는 토픽의 **SUBSCOPE** 설정을 사용합니다.

발행 및 구독 범위 결합

WebSphere MQ 버전 7 이상에서, 발행 및 구독 범위가 독립적으로 작용하여 큐 관리자 간의 발행물 플로우를 결정합니다.

발행물은 발행/구독 토폴로지에서 연결된 모든 큐 관리자 또는 로컬 큐 관리자로만 이동할 수 있습니다. 이는 프록시 구독의 경우에도 유사합니다. 구독과 일치하는 발행물은 이 두 플로우의 조합으로 결정됩니다.

발행물과 구독물의 범위는 둘 다 QMGR 또는 ALL에 해당될 수 있습니다. 발행자와 구독자가 둘 다 동일한 큐 관리자에 연결된 경우에는 범위 설정이 구독자가 해당 발행자로부터 수신하는 발행물에 영향을 주지 않습니다.

발행자와 구독자가 다른 큐 관리자에 연결되어 있는 경우 원격 발행을 수신하려면 두 설정 모두 ALL이어야 합니다.

발행자가 여러 큐 관리자에 연결된 것으로 가정하십시오. 구독자가 다른 발행자로부터 발행물을 수신하도록 하려면 구독 범위를 ALL로 설정하십시오. 그런 다음 각 발행자마다 발행물 범위를 발행자의 로컬 구독자로 제한할지 여부를 결정할 수 있습니다.

구독자가 여러 큐 관리자에 연결된 것으로 가정하십시오. 발행자의 발행물을 모든 구독자로 보내려면 발행 범위를 ALL로 설정하십시오. 구독자가 동일한 큐 관리자에 연결된 발행자의 발행물만 수신하도록 하려면 구독 범위를 QMGR로 설정하십시오.

버전 6 이전에서는, 발행 및 구독 범위를 사용하여 플로우되는 발행물을 관리하지 않았습니다. 또한 발행물 범위가 구독 범위와 일치해야 합니다.

예: 축구 경기 결과 서비스

축구 리그에 소속된 팀인 것으로 가정하십시오. 각 팀마다 발행/구독 클러스터의 다른 팀 모두에 큐 관리자가 연결되어 있습니다.

팀은 Football/result/Home team name/Away team name(이)라는 주제를 사용하여 홈에서 플레이한 모든 게임의 결과를 공개합니다. 이탤릭체 문자열은 가변적인 토픽 이름이고 발행물은 경기 결과입니다.

또한 각 클럽은 Football/myteam/Home team name/Away team name 토픽 문자열을 사용하여 클럽에 대한 결과를 다시 공개합니다.

두 토픽 모두 전체 클러스터에 발행됩니다.

다음 구독은 모든 팀의 팬이 세 가지 흥미로운 방법으로 결과를 구독할 수 있도록 리그에서 설정되었습니다.

SUBSCOPE (QMGR)로 클러스터 토픽을 설정할 수 있습니다. 토픽 정의는 클러스터의 각 멤버로 전파되지만 구독 범위는 로컬 큐 관리자뿐입니다. 따라서 각 큐 관리자의 구독자는 동일한 구독에서 다른 발행물을 수신합니다.

모든 결과 수신

```
DEFINE TOPIC(A) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(ALL)
```

모든 홈 결과 수신

```
DEFINE TOPIC(B) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(QMGR)
```

구독의 범위는 QMGR이므로 홈 그라운드에서 발행된 결과만 일치 여부를 확인합니다.

내 팀 결과 모두 수신

```
DEFINE TOPIC(C) TOPICSTR('Football/myteam/') CLUSTER SUBSCOPE(QMGR)
```

구독의 범위는 QMGR이므로 로컬로 재발행된 로컬 팀 결과의 일치 여부만을 확인합니다.

토픽 공간

토픽 공간은 구독할 수 있는 토픽 세트입니다. 분배된 발행/구독 토픽로지의 큐 관리자에 연결된 구독자에게 연결된 큐 관리자에 정의된 토픽이 잠재적으로 포함된 토픽 공간이 있습니다.

토픽은 사용자가 토픽 오브젝트나 지속적 구독을 정의할 때 또는 애플리케이션이 발행물 또는 구독을 동적으로 작성할 때 처음으로 관리 면에서 작성됩니다.

토픽은 프록시 구독 및 관리 클러스터 토픽 오브젝트를 작성하는 방식 모두를 통해 다른 큐 관리자로 전파됩니다. 프록시 구독의 경우 발행자가 연결된 큐 관리자에서 구독의 큐 관리자로 발행이 전달됩니다. 프록시 구독은 여러 큐 관리자에서 정의된 토픽이 공용 토픽 공간에 결합되는 메커니즘입니다.

프록시 구독은 큐 관리자 계층에서 상위-하위 관계로 함께 연결되는 모든 큐 관리자 사이에 전파됩니다. 결과적으로 한 큐 관리자에서 계층의 다른 큐 관리자에 정의된 토픽을 구독할 수 있습니다. 큐 관리자 사이에 연결된 경로가 있는 한, 큐 관리자의 연결 방식은 중요하지 않습니다.

또한 클러스터의 모든 구성원에 클러스터 토픽에 대한 프록시 구독이 전파됩니다. 클러스터 토픽은 **CLUSTER** 속성이 있거나 상위로부터 속성을 상속하는 토픽 오브젝트에 연결된 토픽입니다. 클러스터 토픽이 아닌 토픽은 로컬 토픽이라고 하며 클러스터에 복제되지 않습니다. 클러스터에 대해서는 구독에서 로컬 토픽으로 프록시 구독이 전파되지 않습니다.

즉, 프록시 구독은 두 가지 상황에서 구독자에 대해 작성됩니다.

1. 큐 관리자가 계층의 멤버이고 프록시 구독이 큐 관리자의 상위와 하위로 전달됩니다.
2. 큐 관리자가 클러스터의 멤버이고 구독 토픽 문자열이 클러스터 토픽 오브젝트와 연관된 토픽으로 분석됩니다. 프록시 구독은 클러스터의 모든 구성원으로 전달됩니다. 자세한 정보는 93 페이지의 『[접치는 토픽](#)』의 내용을 참조하십시오.

큐 관리자가 클러스터 및 계층의 멤버이면 구독자에게 중복 발행을 전달하지 않고도 두 메커니즘에서 프록시 구독이 전파됩니다.

클러스터 토픽 오브젝트 작성의 영향은 두 가지입니다. 클러스터 토픽에 대한 구독이 이루어지면 토픽에 대한 프록시 구독은 클러스터의 다른 구성원으로 송신됩니다. 또한 토픽 오브젝트의 사본을 클러스터의 다른 구성원으로 송신합니다. 클러스터 토픽 오브젝트를 전달하면 토픽 관리가 단순화됩니다. 일반적으로 클러스터 토픽 오브젝트는 클러스터의 단일 큐 관리자(클러스터 토픽 호스트라고 함)에 정의되어 있습니다.

3개 발행/구독 토픽로지의 토픽 공간은 다음 목록에서 설명됩니다.

- 86 페이지의 『[케이스 1. 발행/구독 클러스터](#)』.
- 87 페이지의 『[케이스 2. 버전 7의 발행/구독 계층](#)』.
- 87 페이지의 『[케이스 3. 버전 6의 발행/구독 계층 구조 및 스트림](#)』.

개별 토픽에서, 다음 태스크는 토픽 공간을 결합하는 방법에 대해 설명합니다.

- 87 페이지의 『[발행/구독 클러스터에 단일 토픽 공간 작성](#)』.
- 88 페이지의 『[기존의 버전 6 토픽 공간에 버전 7 큐 관리자 추가](#)』.
- 89 페이지의 『[다중 클러스터의 토픽 공간 결합](#)』.
- 91 페이지의 『[다중 클러스터의 토픽 공간 결합 및 분리](#)』.
- 92 페이지의 『[다중 클러스터의 토픽 공간에 발행 및 구독](#)』.

케이스 1. 발행/구독 클러스터

예제에서는 큐 관리자가 발행/구독 계층에 연결되지 않았다고 가정합니다.

큐 관리자가 발행/구독 클러스터의 멤버이면 해당 토픽 공간은 로컬 토픽 및 클러스터 토픽에서 구성됩니다. 로컬 토픽은 **CLUSTER** 속성 없이 토픽 오브젝트에 연관됩니다. 큐 관리자에 로컬 토픽 오브젝트 정의가 있으면 해당 토픽 공간은 로컬로 정의된 고유한 토픽 오브젝트도 포함하는 클러스터의 다른 큐 관리자와는 다릅니다.

구독한 토픽을 클러스터 토픽 오브젝트로 해석하지 않는 한, 발행/구독 클러스터에서 다른 큐 관리자에 정의된 토픽을 구독할 수 없습니다.

클러스터에 정의된 클러스터 토픽의 정의가 충돌하면 최근 정의를 우선하는 방향으로 해결됩니다. 클러스터 토픽을 여러 번 정의하는 경우, 여러 큐 관리자의 클러스터 토픽 정의가 다를 수 있습니다.

토픽 오브젝트의 로컬 정의는 클러스터 토픽에 대한 정의인지, 로컬 토픽에 대한 정의인지에 상관없이 클러스터의 다른 위치에서 정의된 동일한 토픽 오브젝트보다 우선합니다. 다른 곳에서 정의된 오브젝트가 보다 최근 항목이어도 로컬로 정의된 토픽이 사용됩니다.

PUBSCOPE 및 **SUBSCOPE** 옵션을 QMGR로 설정하여 클러스터의 여러 큐 관리자로 플로우되는 클러스터 토픽에 대한 발행물 또는 구독이 이루어지지 않게 합니다.

클러스터 토픽 호스트에서 토픽 문자열 USA/Alabama 을 사용하여 클러스터 토픽 오브젝트 Alabama 를 정의한다고 가정하십시오. 결과는 다음과 같습니다.

1. 클러스터 토픽 호스트의 토픽 공간에 클러스터 토픽 오브젝트 Alabama 및 토픽 USA/Alabama가 포함됩니다.
2. 클러스터 토픽 오브젝트 Alabama는 각 큐 관리자의 토픽 공간과 결합되는 클러스터의 모든 큐 관리자에 복제됩니다. 클러스터의 각 큐 관리자에서 발생하는 상황은 토픽 오브젝트 Alabama가 큐 관리자에 있는지의 여부에 따라 다릅니다.
 - Alabama가 새 토픽 오브젝트인 경우, 큐 관리자가 클러스터 토픽 오브젝트 Alabama 및 토픽 USA/Alabama를 토픽 공간에 추가합니다.
 - Alabama가 로컬 정의인 경우, 클러스터 토픽 오브젝트 Alabama가 추가됩니다. 로컬 정의를 삭제하지 않으면 원격으로 정의된 클러스터 토픽 오브젝트가 무시됩니다. 큐 관리자는 두 개의 정의 모두 보유합니다.
 - Alabama가 정의된 이전 클러스터 토픽 오브젝트인 경우 새 클러스터 토픽 오브젝트로 바뀝니다.
3. 클러스터에 있는 애플리케이션 또는 관리자는 Alabama 토픽 오브젝트를 참조하여 USA/Alabama에 대한 구독을 작성할 수 있습니다.
4. 클러스터에 있는 애플리케이션은 직접 토픽 문자열 USA/Alabama를 사용하여 토픽 오브젝트 Alabama의 속성을 상속하는 구독을 작성할 수 있습니다. Alabama 토픽 오브젝트는 USA/Alabama로 시작하는 토픽 문자열에서 구성된 구독에 의해 상속됩니다.

다른 큐 관리자 중 하나에 Alabama 토픽 오브젝트에 대한 또 다른 정의가 있는 경우, 이는 클러스터 토픽 호스트의 정의보다 우선합니다. 로컬 오브젝트에 클러스터 속성이 있을 수도 또는 없을 수도 있습니다. 클러스터 속성은 동일한 클러스터 또는 다른 클러스터를 참조합니다. 이렇게 다중 정의가 나타나지 않도록 하십시오. 다중 정의가 생기면 작동에 차이가 생깁니다.

5. 토픽 오브젝트 Alabama에 **PUBSCOPE** 속성 ALL이 있는 경우, Alabama에 대한 구독이 클러스터의 다른 모든 큐 관리자로 송신됩니다.

Alabama **PUBSCOPE** 속성을 QMGR로 설정하여 발행자에서 클러스터의 여러 큐 관리자에 연결된 구독자로 플로우되는 발행물이 이루어지지 않도록 하십시오.

Alabama 토픽 오브젝트가 클러스터의 모든 큐 관리자로 복제되면 **PUBSCOPE** 및 **SUBSCOPE** 속성이 클러스터의 모든 큐 관리자에 적용됩니다.

클러스터 토픽 오브젝트는 클러스터의 모든 위치에서 동일한 토픽 문자열에 연관됩니다. 토픽 오브젝트가 연관된 토픽 문자열은 수정할 수 없습니다. 다른 토픽 문자열과 동일한 토픽 오브젝트를 연관하려면 토픽 오브젝트를 삭제하고 새 토픽 문자열로 다시 작성해야 합니다. 토픽이 클러스터된 경우 효과는 클러스터의 다른 멤버에 저장된 토픽 오브젝트의 사본을 삭제하고 클러스터의 모든 위치에서 새 토픽 오브젝트의 사본을 작성하는 것입니다. 토픽 오브젝트의 사본은 모두 동일한 토픽 문자열을 참조합니다.

하지만, 다른 토픽 문자열을 사용하여 클러스터의 다른 큐 관리자에서 토픽 오브젝트의 중복 정의를 작성할 수 있습니다. 큐 관리자에서 클러스터 토픽 호스트를 관리하여 중복을 피하십시오. 이 내용에 대한 자세한 정보는 65 페이지의 『다중 클러스터 토픽 정의』를 참조하십시오. 서로 다른 토픽 문자열을 가진 동일한 토픽 오브젝트에 대한 다중 정의로 인해 토픽 참조 방법 및 참조 위치에 따라 다른 결과가 나타납니다.

케이스 2. 버전 7의 발행/구독 계층

이 예제에서는 큐 관리자가 발행/구독 클러스터의 멤버가 아니라고 가정합니다.

버전 7에서 큐 관리자가 발행/구독 계층의 구성원인 경우 해당 토픽 공간이 로컬로 정의되어 있고 연결된 큐 관리자에 있는 모든 토픽으로 구성됩니다. 계층에서 모든 큐 관리자의 토픽 공간은 동일합니다. 로컬 토픽과 클러스터 토픽으로 토픽을 나누지 않았습니다.

PUBSCOPE 및 **SUBSCOPE** 옵션을 QMGR로 설정하여 토픽의 발행을 발행자로부터 계층의 다른 큐 관리자에 연결된 구독자로 전달되지 않도록 하십시오.

큐 관리자 QMA에서 토픽 문자열 USA/Alabama를 포함하는 토픽 오브젝트 Alabama를 정의한다고 가정하십시오. 결과는 다음과 같습니다.

1. 이제 QMA의 토픽 공간은 토픽 오브젝트 Alabama 및 토픽 문자열 USA/Alabama를 포함합니다.
2. 애플리케이션 또는 관리자가 토픽 오브젝트 이름 Alabama를 사용하여 QMA에 구독을 작성할 수 있습니다.
3. 애플리케이션은 계층의 큐 관리자에서 USA/Alabama를 포함하여 토픽에 대한 구독을 작성할 수 있습니다. QMA가 로컬로 정의되지 않은 경우 토픽 USA/Alabama가 토픽 오브젝트 SYSTEM.BASE.TOPIC으로 해석됩니다.

케이스 3. 버전 6의 발행/구독 계층 구조 및 스트림

버전 7 이전의 경우 토픽 공간은 여러 스트림으로 구분되었으며, 여기에는 모든 큐 관리자에 존재하는 기본 스트림이 포함되었습니다. 발행은 다른 스트림 사이에서 전달할 수 없습니다. 이름 지정된 스트림을 사용하면 다른 큐 관리자의 토픽 공간은 달라질 수 있습니다. 토픽은 기본 스트림의 토픽과 이름 지정된 다른 스트림의 토픽으로 구분됩니다.

참고: 이름 지정된 각 스트림은 별도의 토픽 공간을 구성합니다. 연결된 토픽로지를 구성하려면 이름 지정된 각 스트림이 연결된 큐 관리자에 존재해야 합니다. 스트림 X는 QMB가 아닌, QMA 및 QMC에 정의되어 있다고 가정합니다. QMA가 QMB의 상위이고 QMB는 QMC의 상위인 경우 스트림 X의 토픽은 QMA 및 QMC 사이에서 전달될 수 없습니다.

PUBSCOPE 및 **SUBSCOPE** 옵션을 모두 QMGR 또는 ALL로 설정하려면 토픽에 대한 발행자 및 구독자가 로컬 사용에 대한 발행만 교환하거나 글로벌 사용에 대한 발행만 교환해야 합니다.

버전 7부터 발행/구독 API를 사용하여 스트림을 사용할 수 없습니다. 버전 7 큐 관리자에서 큐에 있는 발행/구독을 사용하는 경우 스트림은 스트림 효과를 시뮬레이션할 수 있는 다른 토픽 오브젝트로 맵핑됩니다. 스트림은 스트림에 있는 모든 토픽의 루트 토픽인 토픽 오브젝트를 작성하여 시뮬레이션됩니다. 큐 관리자는 스트림 및 각 트리의 대응하는 루트 토픽 사이에서 발행과 구독을 맵핑합니다.

토픽 공간 결합

발행/구독 클러스터 또는 계층에서 큐 관리자의 토픽 공간을 다른 큐 관리자와 결합합니다. 발행/구독 클러스터와, 계층이 있는 발행/구독 클러스터를 결합합니다.

CLUSTER, PUBSCOPE 및 **SUBSCOPE** 속성, 발행/구독 클러스터 및 발행/구독 계층의 빌딩 블록을 사용하여 여러 발행/구독 토픽 공간을 작성할 수 있습니다.

다음 시나리오는 단일 큐 관리자에서 발행/구독 클러스터로의 확장 예부터 다양한 발행/구독 토픽로지를 보여줍니다.

발행/구독 클러스터에 단일 토픽 공간 작성

발행/구독 시스템이 여러 큐 관리자에서 실행될 수 있도록 확장합니다. 발행/구독 클러스터를 사용하여 각 발행자 및 구독자에게 동일한 단일 토픽 공간을 제공하십시오.

시작하기 전에

단일 버전 7 큐 관리자에서 발행/구독 시스템을 구현했습니다.

SYSTEM.BASE.TOPIC 속성 상속에 의존하지 않고 항상 고유한 루트 토픽으로 토픽 공간을 작성하십시오. 발행/구독 시스템을 클러스터로 확장하는 경우 클러스터 토픽 호스트에서 루트 토픽을 클러스터 토픽으로 정의하면 클러스터 전체에서 모든 토픽을 공유합니다.

이 태스크 정보

이제 시스템이 보다 많은 발행자와 구독자를 지원하고 클러스터 전체에서 모든 토픽을 인식할 수 있도록 시스템을 확장할 수 있습니다.

프로시저

1. 발행/구독 시스템에 사용할 클러스터를 작성하십시오.
전통적인 기존 클러스터가 있는 경우에는 성능상의 이유로 새 발행/구독 시스템에 새 클러스터를 설정하는 것이 좋습니다. 두 클러스터의 클러스터 저장소에 동일한 서버를 사용할 수 있습니다.
2. 클러스터 토픽 호스트가 될 하나의 큐 관리자(예를 들어, 저장소 중 하나)를 선택하십시오.
3. 발행/구독 클러스터 전체에서 인식되는 모든 토픽이 관리 토픽 오브젝트로 해석되어야 합니다.
발행/구독 클러스터의 이름을 지정하는 **CLUSTER** 속성을 설정하십시오.

다음에 수행할 작업

발행자 및 구독자 애플리케이션을 클러스터 내 큐 관리자에 연결하십시오.

CLUSTER 속성을 갖는 관리 토픽 오브젝트를 작성하십시오. 토픽은 또한 클러스터 전체에서 전파됩니다. 발행자 및 구독자 프로그램은 관리 토픽을 사용하므로 클러스터 내 다른 큐 관리자에 연결되어도 해당 작동은 변경되지 않습니다.

SYSTEM.BASE.TOPIC이 모든 큐 관리자에서 클러스터 토픽 역할을 해야 하는 경우에는 모든 큐 관리자에서 수정해야 합니다.

기존의 버전 6 토픽 공간에 버전 7 큐 관리자 추가

동일한 토픽 공간을 공유하여 버전 7 큐 관리자와 상호 운영하기 위해 기존 버전 6 발행/구독 시스템을 확장합니다.

시작하기 전에

사용자에게 기존 버전 6 발행/구독 시스템이 있습니다.

새 서버에서 WebSphere MQ 버전 7을 설치하고 큐 관리자를 구성했습니다.

이 태스크 정보

기존 버전 6 발행/구독 시스템을 확장하여 버전 7 큐 관리자와 함께 사용하려 합니다.

사용자는 큐된 발행/구독 인터페이스를 사용하는 버전 6 발행/구독 시스템의 개발을 안정적으로 수행하기로 했습니다. 그리고 버전 7 MQI를 사용하여 시스템에 대한 확장을 추가하려 합니다. 현재 큐된 발행/구독 애플리케이션을 재작성할 계획은 없습니다.

앞으로 버전 6 큐 관리자를 버전 7로 업그레이드할 계획입니다. 현재는 버전 7 큐 관리자에서 기존 큐된 발행/구독 애플리케이션을 실행합니다.

프로시저

1. 한 세트의 송신자-수신자 채널을 작성하여 양방향으로 버전 6 큐 관리자 중 하나에 버전 7 큐 관리자를 연결하십시오.
2. 대상 큐 관리자의 이름으로 두 개 전송 큐를 작성하십시오. 어떠한 이유로 대상 큐 관리자의 이름을 전송 큐 이름으로 사용할 수 없는 경우 큐 관리자 알리어스를 사용하십시오.
3. 송신자 채널을 트리거하도록 전송 큐를 구성하십시오.
4. 버전 6발행/구독 시스템이 스트림을 사용하는 경우 [스트림 추가](#)에 설명된 대로 버전 7큐 관리자에 스트림을 추가하십시오.

5. 버전 7 큐 관리자 **PSMODE**가 ENABLE로 설정되었는지 확인하십시오.
6. 해당 **PARENT** 속성을 대체하여 버전 6 큐 관리자 중 하나를 참조하십시오.
7. 큐 관리자 간의 상위-하위 관계 상태가 양방향에서 모두 활성화되었는지 확인하십시오.

다음에 수행할 작업

태스크를 완료하면 버전 6 및 버전 7 큐 관리자가 동일한 토픽 공간을 공유합니다. 예를 들어, 다음 태스크를 모두 수행할 수 있습니다.

- 버전 6과 버전 7 큐 관리자 사이에서 발행물과 구독을 교환합니다.
- 버전 7 큐 관리자에서 기존 버전 6 발행/구독 프로그램을 실행합니다.
- 버전 6 또는 버전 7 큐 관리자에서 토픽 공간을 보고 수정합니다.
- 버전 7 발행/구독 애플리케이션을 쓰고 버전 7 큐 관리자에서 실행합니다.
- 버전 7 애플리케이션에서 새 발행물과 구독을 작성하고 버전 6 애플리케이션과 교환합니다.

다중 클러스터의 토픽 공간 결합

여러 클러스터에 분산되는 토픽 공간을 작성합니다. 특정 클러스터의 토픽에는 발행하고 다른 클러스터의 토픽은 구독합니다.

시작하기 전에

기존 발행/구독 클러스터가 있고 몇 가지 클러스터 토픽을 모든 클러스터에 전파하려 합니다.

이 태스크 정보

특정 클러스터에서 다른 클러스터로 발행물을 전파하려면 계층에서 클러스터를 함께 결합해야 합니다([90 페이지의 그림 35](#) 참조). 계층적 연결이 연결된 큐 관리자 간에 구독물과 발행물을 전파하며 클러스터는 클러스터 사이가 아닌 각 클러스터 내에 클러스터 토픽을 전파합니다.

이 두 메커니즘을 결합하여 모든 클러스터 간에 클러스터 토픽을 전파할 수 있습니다. 각 클러스터에서 클러스터 토픽 정의를 반복해야 합니다.

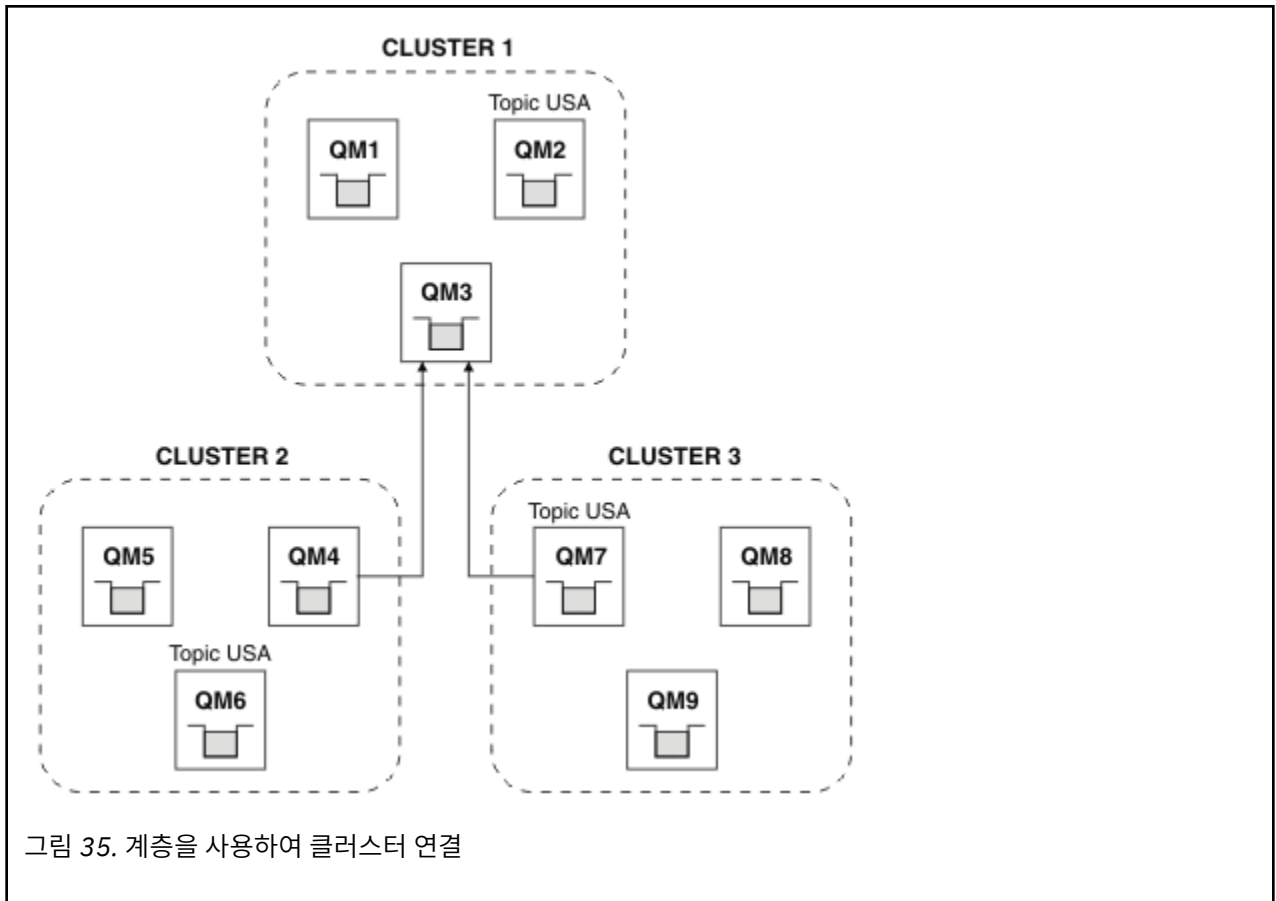


그림 35. 계층을 사용하여 클러스터 연결

다음 단계는 클러스터를 계층에 연결합니다.

프로시저

1. 두 개의 송신자 수신자 채널 세트를 작성하여 두 디렉토리에서 모두 QM3, QM4, QM3, QM7을 연결하십시오. 계층을 연결하기 위해 클러스터가 아닌 전통적인 송신자 수신자 채널과 전송 큐를 사용해야 합니다.
2. 대상 큐 관리자의 이름으로 세 개 전송 큐를 작성하십시오. 어떠한 이유로 대상 큐 관리자의 이름을 전송 큐 이름으로 사용할 수 없는 경우 큐 관리자 알리어스를 사용하십시오.
3. 송신자 채널을 트리거하도록 전송 큐를 구성하십시오.
4. QM3, QM4, QM7의 **PSMODE**가 ENABLE로 설정되었는지 확인하십시오.
5. QM4 및 QM7의 **PARENT** 속성을 QM3으로 대체하십시오.
6. 큐 관리자 간의 상위-하위 관계 상태가 양방향에서 모두 활성화되었는지 확인하십시오.
7. 클러스터 1, 2 및 3의 세 개의 클러스터 토픽 호스트 각각에 **CLUSTER('CLUSTER 1')**, **CLUSTER('CLUSTER 2')** 및 **CLUSTER('CLUSTER 3')** 속성을 사용하여 관리 토픽 USA 을 작성하십시오. 클러스터 토픽 호스트는 계층적으로 연결된 큐 관리자일 필요가 없습니다.

다음에 수행할 작업

이제 90 페이지의 그림 35에서 클러스터 토픽 USA를 발행하고 구독할 수 있습니다. 발행물과 구독물은 세 개 클러스터 모두의 발행자와 구독자로 이동합니다.

다른 클러스터에서 USA를 클러스터 토픽으로 작성하지 않은 것으로 가정하십시오. USA가 QM7에서만 정의된 경우에는 QM7, QM8, QM9, QM3 간에 USA에 대한 발행물과 구독물이 교환됩니다. QM7, QM8, QM9에서 실행되는 발행자와 구독자는 관리 토픽 USA의 속성을 상속합니다. QM3의 발행자와 구독자는 QM3에서 SYSTEM.BASE.TOPIC의 속성을 상속합니다.

다중 클러스터의 토픽 공간 결합 및 분리

일부 토픽 공간을 특정 클러스터로 분리하고 연결된 모든 클러스터에서 액세스할 수 있도록 다른 토픽 공간을 결합합니다.

시작하기 전에

89 페이지의 『다중 클러스터의 토픽 공간 결합』 주제를 확인하십시오. 큐 관리자를 브릿지로 추가하지 않고도 요구를 충분히 충족시킬 수 있습니다.

이 태스크 정보

89 페이지의 『다중 클러스터의 토픽 공간 결합』의 90 페이지의 그림 35에 표시된 토픽로지의 잠재적 개선 사항은 모든 클러스터에서 공유되지 않는 클러스터 토픽을 분리하는 것입니다. 클러스터에 없는 브릿지 큐 관리자를 작성하여 클러스터를 분리하십시오(91 페이지의 그림 36 참조). 브릿지 큐 관리자를 사용하면 특정 클러스터에서 다른 클러스터로 이동할 수 있는 발행물과 구독물을 필터링할 수 있습니다.

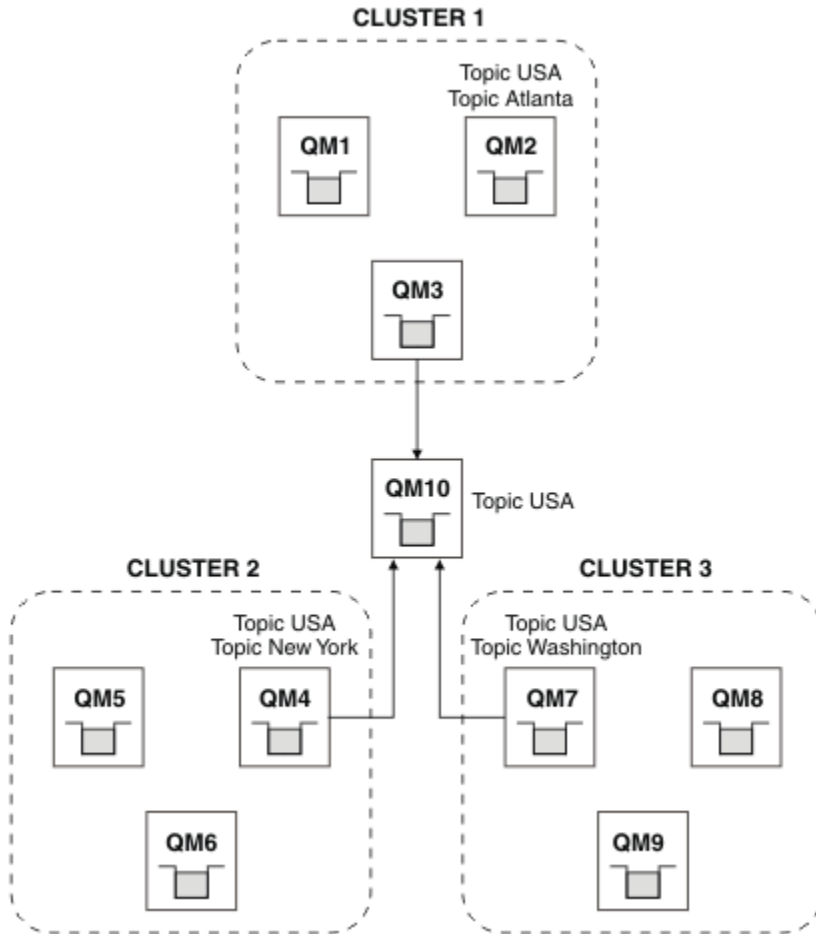


그림 36. 연결된 클러스터

다른 클러스터의 브릿지에서 공유하지 않을 클러스터 토픽을 분리하려면 브릿지를 사용하십시오. 91 페이지의 그림 36에서, USA 는 모든 클러스터에서 공유되는 클러스터 토픽이고 Atlanta, New York 및 Washington 는 각각 하나의 클러스터에서만 공유되는 클러스터 토픽입니다.

다음 프로시저를 사용하여 구성을 모델화하십시오.

프로시저

1. 모든 큐 관리자의 **SUBSCOPE(QMGR)** 및 **PUBSCOPE(QMGR)**를 포함하도록 SYSTEM.BASE.TOPIC 토픽 오브젝트를 모두 수정하십시오.

명시적으로 클러스터 토픽의 루트 토픽에서 **SUBSCOPE(ALL)** 및 **PUBSCOPE(ALL)**를 설정하지 않는 경우 다른 큐 관리자에 토픽(클러스터 토픽의 경우에도)이 전파되지 않습니다.

2. **CLUSTER(clustername)**, **SUBSCOPE(ALL)** 및 **PUBSCOPE(ALL)** 속성을 가진 각 클러스터에서 공유할 세 개의 클러스터 토픽 호스트에서 토픽을 정의하십시오.

일부 클러스터 토픽을 모든 클러스터에서 공유하려면 각 클러스터에서 동일한 토픽을 정의하십시오. 각 클러스터의 클러스터 이름을 클러스터 속성으로 사용하십시오.

3. 모든 클러스터에서 공유하려는 클러스터 토픽에서는, 속성 **SUBSCOPE(ALL)**, 및 **PUBSCOPE(ALL)**를 가진 브릿지 큐 관리자(QM10)에서 토픽을 다시 정의하십시오.

예

91 페이지의 그림 36의 예에서는 USA에서 상속되는 토픽만 세 개 클러스터 간에 모두 전파됩니다.

다음에 수행할 작업

SUBSCOPE(ALL) 및 **PUBSCOPE(ALL)**가 있는 브릿지 큐 관리자에 정의된 토픽 구독이 클러스터 사이에 전파됩니다.

속성 **CLUSTER(clustername)**, **SUBSCOPE(ALL)** 및 **PUBSCOPE(ALL)**가 있는 각 클러스터에 정의된 토픽 구독이 각 클러스터 내에 전파됩니다.

다른 구독은 큐 관리자의 로컬입니다.

다중 클러스터의 토픽 공간에 발행 및 구독

중첩된 클러스터를 사용하여 복수 클러스터에서 토픽을 발행, 구독합니다. 이 기술은 클러스터의 토픽 공간이 중첩되지 않는 경우 사용할 수 있습니다.

시작하기 전에

클러스터 간 교차 지점에서 일부 큐 관리자가 있는 여러 기존 클러스터를 작성하십시오.

이 태스크 정보

다양한 이유로 클러스터를 중첩시키도록 선택했을 수 있습니다.

1. 고가용성 서버 또는 큐 관리자의 수가 제한적입니다. 모든 클러스터 저장소를 배치하기로 결정하고 클러스터 토픽이 저장소를 호스팅합니다.
2. 게이트웨이 큐 관리자를 사용하여 연결되는 기존 큐 관리자 클러스터가 있습니다. 동일한 클러스터 토폴로지에 발행/구독 애플리케이션을 배치하려고 합니다.
3. 여러 가지 자체 포함 발행/구독 애플리케이션이 있습니다. 성능상의 이유로, 발행/구독 클러스터는 작게 또한 기존 클러스터와 별도로 유지하는 것이 좋습니다. 애플리케이션을 다른 클러스터에 배치하기로 결정했습니다. 그러나 모니터링 애플리케이션 사본 중 하나에 대한 라이선스만 있어 모든 발행/구독 애플리케이션을 하나의 큐 관리자에서 모니터링하려고 합니다. 이 큐 관리자는 모든 클러스터의 클러스터 토픽 발행에 대한 액세스 권한이 있어야 합니다.

토픽이 중첩되지 않는 토픽 공간에 정의되었는지 확인함으로써 중첩 발행/구독 클러스터에 토픽을 배치할 수 있습니다(93 페이지의 그림 37 참조). 토픽 공간이 중첩될 때 중첩 클러스터에 배치하면 문제점이 발생합니다.

발행/구독 클러스터가 중첩되므로 중첩된 클러스터에서 큐 관리자를 사용하여 토픽 공간을 발행 및 구독할 수 있습니다.

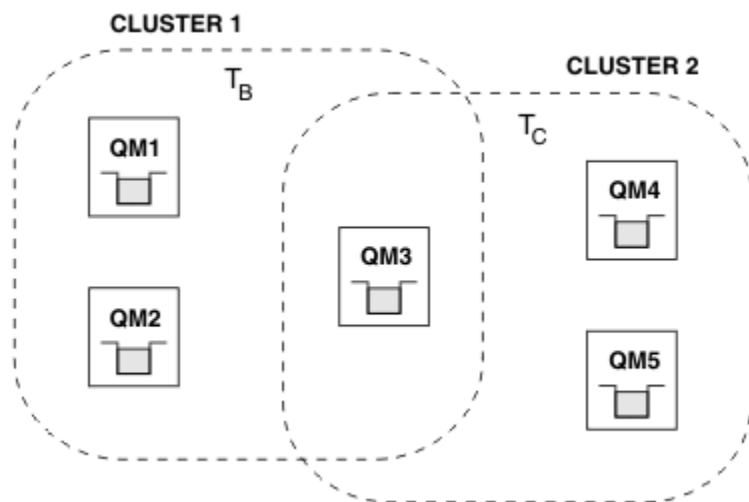


그림 37. 클러스터는 중첩되고 토픽 공간은 중첩되지 않는 경우

프로시저

토픽 공간이 중첩되지 않음을 확인할 수 있는 방법을 작성하십시오.

예를 들어, 각 토픽 공간에 고유한 루트 토픽을 정의하십시오. 루트 토픽을 클러스터 토픽을 만드십시오.

- a) DEFINE TOPIC(B) TOPICSTR('B') CLUSTER('CLUSTER 1') ...
- b) DEFINE TOPIC(C) TOPICSTR('C') CLUSTER('CLUSTER 2') ...

예

93 페이지의 그림 37에서 QM3에 연결된 발행자 및 구독자가 T_B 또는 T_C 를 발행 또는 구독할 수 있습니다.

다음에 수행할 작업

두 클러스터에서 모두 토픽을 사용하는 발행자와 구독자를 중첩된 클러스터의 큐 관리자로 연결하십시오.

특정 클러스터의 토픽만 사용해야 하는 발행자와 구독자를 중첩된 클러스터에 없는 큐 관리자에 연결하십시오.

겹치는 토픽

발행물을 분배된 발행/구독 토픽로지, 발행 및 구독 토픽 문자열에 따라 여러 토픽 오브젝트와 연관시킬 수 있는 경우에 겹치는 토픽이 나타납니다.

토픽을 둘 이상의 토픽 오브젝트로 나타낼 수 있는 경우 토픽 간 겹침을 고려해야 합니다.

클러스터의 로컬 토픽

토픽을 클러스터에 있는 임의의 큐 관리자에서 정의할 수 있습니다. 로컬로 정의한 토픽은 다른 곳에서 정의된 클러스터 토픽보다 우선하며 동일한 토픽 문자열로 표시됩니다.

클러스터의 클러스터 토픽

토픽을 클러스터에 있는 임의의 큐 관리자에서 정의할 수 있습니다. 토픽이 클러스터된 경우 클러스터의 다른 멤버에 복제됩니다. 토픽이 클러스터의 다른 큐 관리자에 있는 클러스터 토픽으로 정의된 경우 오류입니다. 오류 메시지는 기존 클러스터 정의가 있는 큐 관리자의 오류 로그에 기록됩니다.

일반적으로, 클러스터에 있는 하나의 큐 관리자에서만 클러스터 토픽을 정의하여(클러스터 토픽 호스트) 클러스터 토픽에 대한 하나의 정의만 있게 하십시오.

클러스터 토픽을 다시 정의하는 경우, 변경사항이 각 큐 관리자에 도달하기까지 시간이 걸립니다. 결국에는 최근 정의가 비클러스터 토픽 호스트로 복제되는 이전 클러스터 토픽 정의를 대체합니다.

서로 다른 속성이 있는 클러스터의 여러 큐 관리자에서 클러스터 토픽을 정의하는 경우에는, 최근 정의가 이전 로컬 정의를 대체하지 않습니다.

와일드카드 구독이 여러 토픽 문자열로 표시됨

구독에 와일드카드가 있을 때, 토픽 공간에 있는 잠재적으로 서로 다른 토픽이 구독과 일치할 수 있으며 이로 인해 여러 토픽 오브젝트로 해석되는 구독이 작성됩니다.

예를 들어, 클러스터 SPORTS에서 다음 토픽 정의를 고려하십시오.

```
DEFINE TOPIC(A) TOPICSTR('Football/result/#') SUBSCOPE(QMGR) CLUSTER(SPORTS)
DEFINE TOPIC(B) TOPICSTR('Football/#') SUBSCOPE(ALL) CLUSTER(SPORTS)
DEFINE TOPIC(C) TOPICSTR('Football/result/Newport/Cardiff') PUBSCOPE(ALL) SUBSCOPE(ALL)
CLUSTER(SPORTS)
DEFINE TOPIC(D) TOPICSTR('Football/matches/Newport/Cardiff') PUBSCOPE(ALL) SUBSCOPE(QMGR)
CLUSTER(SPORTS)
```

클러스터에 두 개의 큐 관리자 QM1 및 QM2가 있다고 가정합니다. 토픽 C 및 D는 QM1에서 발행됩니다.

이러한 구독이 그룹화되지 않은 경우 QM2의 구독자가 수신하는 내용을 고려하십시오.

- 토픽 A에 대한 구독이 아무 것도 수신하지 않습니다.
 - SUBSCOPE(QMGR) 및 발행물이 다른 큐 관리자에 있습니다.
- 토픽 B에 대한 구독이 두 발행물 모두를 수신합니다.
 - 두 경우 모두에서 SUBSCOPE(ALL) 및 PUBSCOPE(ALL)입니다.
- 토픽 C에 대한 구독이 하나의 발행물을 수신합니다.
 - SUBSCOPE(ALL) 및 PUBSCOPE(ALL), 토픽 C에 대한 발행물과 일치하는 내용입니다.
- 토픽 D에 대한 구독이 아무 것도 수신하지 않습니다.
 - SUBSCOPE(QMGR) 및 발행물이 다른 큐 관리자에 있습니다.

이러한 구독이 그룹화되는 경우 QM2의 구독자가 수신하는 내용을 고려하십시오.

- 구독자가 토픽 C에 대한 하나의 발행물을 수신합니다.
 - SUBSCOPE(QMGR)를 사용하는 토픽 A에 대한 일치하는 구독이 SUBSCOPE(ALL)를 사용하는 토픽 C에 대한 일치하는 구독으로 대체됩니다. 보다 구체적인 구독이 우선하며 발행물이 수신됩니다.
 - 구독이 그룹화되고 C가 보다 구체적이므로 토픽 C에 대한 일치하는 구독이 우선시되어 토픽 B에 대한 일치하는 구독이 거부됩니다. 중복 발행물이 제거됩니다.
- 구독자가 토픽 D에 대한 발행물을 수신하지 않습니다.
 - SUBSCOPE(ALL)를 사용하는 토픽 B에 대한 일치하는 구독이 SUBSCOPE(QMGR)를 사용하는 토픽 D에 대한 일치하는 구독으로 대체됩니다. 보다 구체적인 구독이 우선하며 발행물이 제거됩니다.

루프 감지 작동 방법

분배된 발행/구독 네트워크에서는, 발행물 및 프록시 구독이 루프를 실행할 수 없습니다. 루프를 실행하게 되면 동일한 원본 발행물의 사본을 여러 개 수신하는 구독자가 네트워크에 연결되어 네트워크가 초과 사용됩니다.

48 페이지의 『프록시 구독 집계 및 발행물 집계』에 설명된 프록시 구독 집계 시스템에서는 루프를 형성할 수 있지만 프록시 구독에 대한 무한 루프는 실행할 수 없습니다. 프록시 발행의 존재에 따라 발행 전파가 관별되므로 무한 루프를 입력할 수 있습니다. Websphere MQ V7.0은 발행이 영구적으로 루핑하지 못하도록 하기 위해 다음 기술을 사용합니다.

발행이 발행/구독 토픽로 이동하면 각 큐 관리자는 메시지 헤더에 고유한 지문을 추가합니다. 발행/구독 큐 관리자가 다른 발행/구독 큐 관리자에게 발행물을 수신할 때마다 메시지 헤더에 있는 식별 정보를 확인합니다. 고유한 지문이 이미 있으면 발행은 루프를 모두 순환하므로 큐 관리자는 메시지를 버리고 오류 로그에 항목을 추가합니다.

참고: 루프에서 발행은 루프 내 양방향으로 전파되고 루프 내 각 큐 관리자는 원래 큐 관리자가 루프에 있는 발행을 버리기 전에 발행 모두를 수신합니다. 이로 인해 구독 중인 애플리케이션은 루프가 끊어질 때까지 발행의 중복 사본을 수신합니다.

루프 감지 지문 형식

루프 감지 식별 정보는 RFH2 헤더 또는 플로우에 V7.0 프로토콜의 일부로 삽입됩니다. RFH2 프로그래머는 헤더를 파악하고 식별 정보를 온전하게 전달해야 합니다. WebSphere MessageBroker에서는 식별 정보가 포함되지 않을 RFH1 헤더를 사용합니다.

```
<ibm>
  <Rfp>uuid1</Rfp>
  <Rfp>uuid2</Rfp>
  <Rfp>uuid3</Rfp>
</ibm>
```

<ibm>은 방문한 각각의 큐 관리자에 대한 고유 사용자 ID(uuid)가 포함된 라우팅 지문 목록을 보유한 폴더의 이름입니다.

큐 관리자가 메시지를 발행할 때마다 <Rfp>(라우팅 지문) 태그를 사용하여 해당 uuid를 <ibm> 폴더에 추가합니다. Whenever a publication is received, WebSphere MQ uses the message properties API to iterate through the <Rfp> tags to see if that particular uuid value is present. 큐된 발행/구독 인터페이스를 사용하여 WebSphere MQ의 WebSphere Platform Messaging 컴포넌트가 채널 및 RFH2 구독을 통해 Websphere Message Broker에 연결하는 방식으로 인해 WebSphere MQ도 해당 라우트에 의해 발행물을 수신할 때 식별 정보를 작성합니다.

목적은 예상되는 방식이 아닌 경우 애플리케이션에 RFH2를 전달하는 것이 아닙니다. 지문 정보에 추가했기 때문입니다.

Whenever an RFH2 is converted into message properties, it will also be necessary to convert the <ibm> folder; this removes the fingerprint information from the RFH2 that is passed on or delivered to applications that have used the Websphere MQ V7.0 API.

식별 정보가 있는 메시지를 RFH1 구독자에 전달하거나 Websphere Message Broker V6.0으로 전달할 때마다 식별 정보가 RFH1으로 변환됩니다.

Websphere Message Broker V6.0이 이 메시지를 RFH2 구독자(예: SIB)에 전달할 때, 식별 정보를 다시 RFH2 형식으로 변환해야 합니다.

JMS 인터페이스가 RFH2에서 식별 정보를 추출하지 않아 이 정보를 JMS 애플리케이션에 전달하지 못하므로 이 애플리케이션이 해당 정보를 확인할 수 없습니다.

Rfp 메시지 특성은 propDesc.CopyOptions = MQCOPY_FORWARD and MQCOPY_PUBLISH(으)로 작성됩니다. 이 특성은 동일한 메시지를 수신한 다음 다시 발행하는 애플리케이션에 영향을 미칩니다. 이는 이러한 애플리케이션이 PutMsgOpts.Action = MQACTP_FORWARD(를) 사용하여 라우팅 지문의 체인을 계속할 수 있지만 체인에서 자신의 지문을 제거하기 위해 적절히 코딩되어야 함을 의미합니다. 기본적으로 애플리케이션은 PutMsgOpts.Action = MQACTP_NEW(를) 사용하고 새 체인을 시작합니다.

분배된 발행/구독 토플로지에 보유된 발행물

분배 발행/구독 토플로지에서 보유된 발행물을 사용할 때, 토플로지의 단일 큐 관리자로부터 동일한 토플에 대해 보유된 발행물만 발행하는 것이 좋습니다.

그렇지 않으면 동일한 토플에 대해 보유된 여러 발행물이 여러 큐 관리자에서 활성 상태가 될 수 있으며, 이렇게 되면 예기치 못한 작동이 발생할 수 있습니다. 여러 프록시 구독을 분배하면 보유된 여러 발행물을 수신할 수 있습니다.

큐 관리자 간의 발행/구독 보안

프록시 구독 및 발행물과 같은 발행/구독 내부 메시지는 정상적인 채널 보안 규칙을 사용하여 발행/구독 시스템 큐에 놓입니다. 이 토플에 있는 정보 및 다이어그램은 메시지 전달에 포함되는 다양한 프로세스 및 사용자 ID를 강조표시합니다.

로컬 액세스 제어

발행물 및 구독에 대한 토플의 액세스는 발행/구독 보안에 설명된 로컬 보안 정의 및 규칙으로 관리됩니다. z/OS에서는 액세스 제어를 설정하는 데 로컬 토플 오브젝트가 필요하지 않습니다. 다른 플랫폼에서 액세스 제어에 필요한 로컬 토플도 없습니다. 관리자는 클러스터된 토플 오브젝트가 클러스터에 아직 존재하는지 여부와 관계없이 이에 대한 액세스 제어를 적용하도록 선택할 수 있습니다.

시스템 관리자는 로컬 시스템에서의 액세스 제어를 담당합니다. 이러한 관리자는 액세스 제어 정책을 담당할 클러스터 집합 또는 계층의 다른 멤버의 관리자를 신뢰해야 합니다. 액세스 제어가 별도의 각 시스템에 대해 정의되어 있으므로 상세 레벨 제어가 필요한 경우 번거로운 일이 될 수 있습니다. 액세스 제어를 부여하는 데 필요하지 않거나 액세스 제어가 토픽 트리의 상위 레벨 오브젝트에서 정의될 수 있습니다. 상세 레벨 액세스 제어는 토픽 네임스페이스의 각 세분화를 위해 정의할 수 있습니다.

프록시 구독 작성

조직이 큐 관리자에 해당 큐 관리자를 연결하기 위한 신뢰는 정상 채널 인증 수단으로 확인됩니다. 또한 신뢰되는 조직이 분산 발행/구독을 수행하도록 허용되는 경우 권한 검사가 수행됩니다. 이 검사는 채널이 분산된 발행/구독 큐에 메시지를 넣을 때 수행됩니다. 예를 들어, 메시지를 SYSTEM.INTER.QMGR.CONTROL 큐에 넣는 경우입니다. 큐 권한 검사를 위한 사용자 ID는 수신 채널의 PUTAUT 값에 따라 달라집니다. 예를 들어, 값 및 플랫폼에 따라 채널의 사용자 ID, MCAUSER, 메시지 컨텍스트. 채널 보안에 대한 자세한 정보는 [채널 보안](#)을 참조하십시오.

프록시 구독은 리모트 큐 관리자에서 분배된 발행/구독 에이전트의 사용자 ID로 작성됩니다. 예를 들어, [96 페이지의 그림 38](#)의 QM2입니다. 그러면 사용자 ID가 시스템에 정의되어 있고 따라서 도메인 충돌이 없으므로 이 사용자는 쉽게 로컬 토픽 오브젝트 프로파일에 대한 액세스 권한을 부여받게 됩니다.

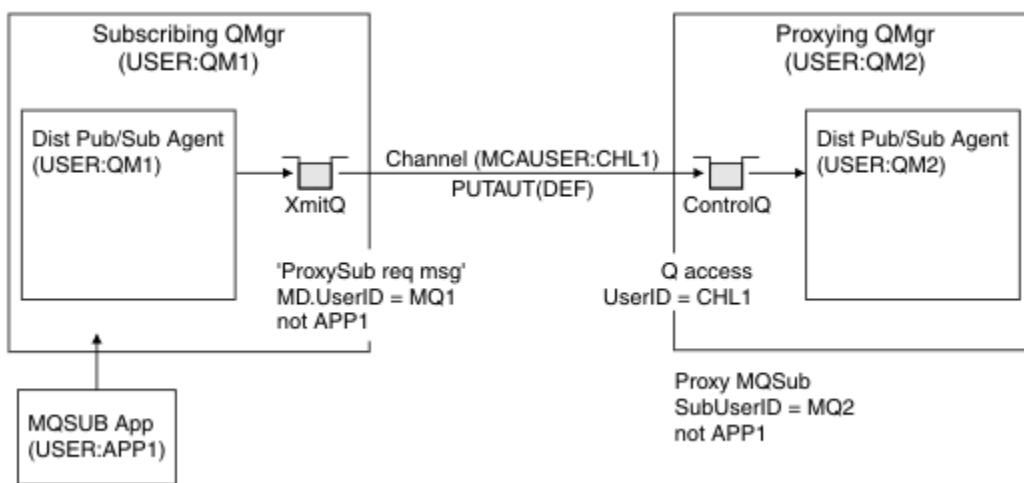


그림 38. 프록시 구독 보안, 구독 작성

원격 발행물 다시 송신

구독 큐 관리자에서 발행물이 작성되면 발행물의 사본이 프록시 구독용으로 작성됩니다. 복사된 발행의 컨텍스트는 구독을 작성한 사용자 ID의 컨텍스트를 포함합니다([97 페이지의 그림 39](#)의 QM2). 프록시 구독은 리모트 큐인 목적지 큐와 함께 작성되므로 발행 메시지는 전송 큐에서 해결됩니다.

다른 큐 관리자 QM1에 조직의 큐 관리자 QM2를 연결하기 위한 조직에 대한 신뢰는 정상적인 채널 인증 수단으로 확인됩니다. 해당되는 신뢰 조직이 분산 발행/구독을 수행하도록 허용되는 경우, 권한 검사는 채널이 발행 메시지를 분산 발행/구독 발행 큐 SYSTEM.INTER.QMGR.PUBS에 넣을 때 수행됩니다. 큐 권한 검사에 대한 사용자 ID는 수신 채널의 PUTAUT 값에 의존합니다(예를 들어, 값과 플랫폼에 따라 채널, MCAUSER, 메시지 컨텍스트 등의 사용자 ID). 채널 보안에 대한 자세한 정보는 [채널 보안](#)을 참조하십시오.

발행 메시지가 구독 큐 관리자에 도달할 때, 토픽에 대한 다른 MQPUT는 해당 큐 관리자의 권한 하에 수행되고, 메시지가 있는 컨텍스트는 로컬 구독자에게 각각 메시지가 제공되는 대로 각 로컬 구독자의 컨텍스트에 의해 대체됩니다.

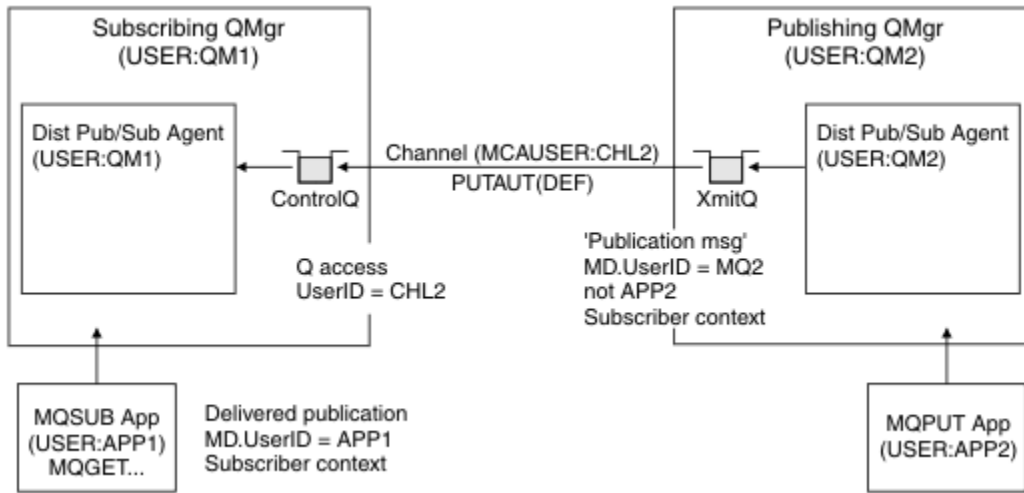


그림 39. 프록시 구독 보안, 발행물 전달

보안과 관련하여 고려할 것이 거의 없는 시스템에서는 분산된 발행/구독이 mqm 그룹의 사용자 ID로 실행되기 쉽고, 채널의 MCAUSER 매개변수는 공백이며(기본값) 메시지는 필요에 따라 다양한 시스템 큐로 전달됩니다. 안전하지 않는 시스템은 분배된 발행/구독을 시연하기 위한 개념 증명을 설정하기가 쉽습니다.

보안이 보다 심각하게 고려되는 시스템에서는 이러한 내부 메시지가 채널을 통해 이동하는 메시지와 동일한 보안 제어의 영향을 받습니다.

채널이 MCAUSER를 검사하도록 지정된 비어 있지 않은 MCAUSER 및 PUTAUT 값으로 설정된 경우 검사 중인 MCAUSER에 SYSTEM.INTER.QMGR.* 큐에 대한 액세스를 부여해야 합니다. 다른 MCAUSER ID에서 채널이 실행 중인 서로 다른 여러 개의 리모트 큐 관리자가 있는 경우 이러한 모든 사용자 ID에 SYSTEM.INTER.QMGR.* 큐에 대한 액세스를 부여해야 합니다. 다른 MCAUSER ID에서 실행 중인 채널은 예를 들어 다중 계층 구조 연결이 단일 큐 관리자에 구성된 경우 발생할 수 있습니다.

메시지의 컨텍스트를 지정하는 PUTAUT 값으로 채널이 설정되는 경우, SYSTEM.INTER.QMGR.* 큐에 대한 액세스는 내부 메시지 안에서 사용자 ID를 기초로 검사됩니다. 이러한 모든 메시지는 내부 메시지 또는 발행 메시지를 송신하는 큐 관리자에서 분배된 발행/구독 에이전트의 사용자 ID를 사용하여 넣었으므로(97 페이지의 그림 39 참조), 이러한 방법으로 분배된 발행/구독 보안을 설정하려는 경우 사용자 ID 세트는 여러 시스템 큐에 액세스를 부여할 수 있는 정도의 크기입니다(리모트 큐 관리자당 하나). 채널 컨텍스트 보안이 항상 가지고 있는 동일한 문제 모두를 계속 가지고 있습니다. 다양한 사용자 ID 도메인의 문제와 메시지의 사용자 ID가 수신 시스템에서 정의되지 않을 수 있다는 문제입니다. 그러나 이는 필요한 경우 완벽히 허용되는 실행 방법입니다.

분산 발행/구독을 위해 모든 내부 큐 관리자 메시징은 정상 채널 보안을 사용하여 실행됩니다.

주제 레벨에서 발행 및 프록시 구독을 제한하는 방법에 대한 정보는 [공개/등록 보안](#) 을 참조하십시오.

큐 관리자 계층으로 기본 사용자 ID 사용

다른 플랫폼에서 실행 중인 큐 관리자의 계층이 있고 기본 사용자 ID를 사용 중인 경우 이러한 기본 사용자 ID는 플랫폼 간에 다를 수 있고 대상 플랫폼에서 인식되지 않을 수 있습니다. 따라서 한 플랫폼에서 실행 중인 큐 관리자는 이유 코드가 MQRC_NOT_AUTHORIZED인 다른 플랫폼의 큐 관리자로부터 수신되는 메시지를 거부합니다.

메시지가 거부되지 않도록 하려면 최소한 다른 플랫폼에서 사용되는 기본 사용자 ID에 다음 권한을 추가해야 합니다.

- SYSTEM.BROKER의 *PUT *GET 권한. 큐
- SYSTEM.BROKER의 *PUB *SUB 권한. 토픽
- SYSTEM.BROKER.CONTROL.QUEUE 큐의 *ADMCR *ADMDLT *ADMCHG 권한.

기본 사용자 ID는 다음과 같습니다.

플랫폼	기본 사용자 ID
Windows	MUSR_MQADMIN 참고: MUSR_MQADMIN은 첫 설치에만 해당하는 기본 사용자 ID입니다. 이후의 설치에서는 IBM WebSphere MQ 준비 마법사에서 MUSR_MQADMINx에 대해 이름 지정된 사용자 계정을 작성합니다. 여기서 x는 현재 없는 사용자 ID를 나타내는 다음에 사용 가능한 수입니다.
UNIX and Linux 시스템	mqm
IBM i	QMQM
z/OS	채널 시작기 주소 공간 사용자 ID

Windows, UNIX, Linux 및 z/OS 플랫폼의 큐 관리자용 IBM i에 있는 큐 관리자에 계층적으로 연결되는 경우 'qmqm' 사용자 ID를 작성하고 액세스를 부여하십시오.

IBM i 및 z/OS 플랫폼의 큐 관리자용 Windows, UNIX 또는 Linux에 있는 큐 관리자에 계층적으로 연결되는 경우 'mqm' 사용자 ID를 작성하고 액세스를 부여하십시오.

Windows, UNIX, Linux 및 IBM 플랫폼의 큐 관리자용 z/OS에 있는 큐 관리자에 계층적으로 연결되는 경우 z/OS 채널 시작기 주소 공간 사용자 ID를 작성하고 액세스를 부여하십시오.

사용자 ID는 대소문자가 구별될 수 있습니다. 발신 큐 관리자(IBM i, Windows, UNIX 또는 Linux 시스템의 경우)는 사용자 ID를 모두 대문자로 만듭니다. 수신 큐 관리자(Windows, UNIX 또는 Linux 시스템의 경우)는 사용자 ID를 모두 소문자로 만듭니다. 따라서 UNIX and Linux 시스템에서 작성된 모든 사용자 ID는 소문자 양식으로 작성되어야 합니다. 메시지 엑시트가 설치된 경우 사용자 ID를 대문자 또는 소문자로 강제 실행하는 작업이 발생하지 않습니다. 메시지 엑시트가 사용자 ID를 처리하는 방법을 이해하려면 경우에는 주의하십시오.

사용자 ID의 변환에 관련된 잠재적인 문제점을 방지하려면 다음을 수행하십시오.

- UNIX, Linux 및 Windows 시스템에서 사용자 ID가 소문자로 지정되어 있는지 확인하십시오.
- IBM i 및 z/OS에서 사용자 ID가 대문자로 지정되어 있는지 확인하십시오.

분산 발행/구독 시스템 큐

네 개의 시스템 큐가 발행/구독 메시징을 위한 큐 관리자에 사용됩니다. 문제점 판별 또는 용량 계획을 위해서만 이러한 큐를 사용해야 합니다.

시스템 큐	목적
SYSTEM.INTER.QMGR.CONTROL	WebSphere MQ의 분배된 발행/구독 제어 큐
SYSTEM.INTER.QMGR.FANREQ	WebSphere MQ의 분배된 발행/구독 내부 프로시 구독 팬아웃 프로세스 입력 큐
SYSTEM.INTER.QMGR.PUBS	WebSphere MQ의 분배된 발행/구독 발행물
SYSTEM.HIERARCHY.STATE	WebSphere MQ의 분배된 발행/구독 계층 관계 상태

발행/구독 시스템 큐의 속성은 [98 페이지의 표 10](#)에 표시되어 있습니다.

속성	기본값
DEFPSIST	예
DEFSOPT	EXC

표 10. 발행/구독 시스템 큐의 속성 (계속)	
속성	기본값
MAXMSGL	AIX, HP-UX, Linux, IBM i, Solaris 및 윈도우 플랫폼: ALTER QMGR 명령의 MAXMSGL 매개변수 값
MAXDEPTH	999999999
SHARE	해당사항 없음
STGCLASS	z/OS에서만 이 속성이 사용됩니다.

발행/구독 시스템 큐 오류

분산 발행/구독 큐 관리자 큐를 사용할 수 없으면 오류가 발생할 수 있습니다.

팬아웃 요청 큐 SYSTEM.INTER.QMGR.FANREQ를 사용할 수 없으면, MQSUB API가 오류 로그에 기록된 오류 메시지 및 이유 코드를 수신하며 이러한 경우 프록시 구독을 직접 연결된 큐 관리자에게 전달해야 합니다.

계층 관계 상태 큐 SYSTEM.HIERARCHY.STATE를 사용할 수 없으면, 오류 메시지가 오류 로그에 기록되며 발행/구독 엔진이 COMPAT 모드 상태에 있게 됩니다.

SYSTEM.INTER.QMGR 큐 중 하나를 사용할 수 없는 경우, 오류 메시지가 오류 로그에 기록되며, 기능은 사용할 수 없지만 발행/구독 메시지가 리모트 큐 관리자에서 큐를 빌드업할 수 있습니다.

상위, 하위 또는 발행/구독 클러스터 큐 관리자로의 전송 큐를 사용할 수 없는 경우, 다음과 같은 일이 발생합니다.

1. MQPUT API는 이유 코드를 수신하고 발행물은 전달되지 않습니다.
2. 수신한 상호 큐 관리자 발행물이 입력 큐로 백아웃된 후 다시 시도되며, 백아웃 임계값에 도달하면 데드-레터 큐에 배치됩니다.
3. 프록시 구독이 팬아웃 요청 큐로 백아웃된 후 다시 시도되며, 백아웃 임계값에 도달하면 데드-레터 큐에 배치됩니다. 이 경우 프록시 구독은 연결된 큐 관리자에서 큐를 빌드업하지 않습니다.
4. 계층 관계 프로토콜 메시지에 실패하고 연결 상태가 PUBSUB 명령에서 ERROR로 표시됩니다.

WebSphere MQ 데드-레터 큐 핸들러로 미배달 메시지 핸들링

데드-레터 큐의 개념과 데드-레터 큐에 메시지 넣는 방법 및 관리 방법

데드-레터 큐(DLQ)(때때로 미전달 메시지 큐로 불림)는 목적지 큐로 전달되지 못한 메시지에 대한 큐를 보유합니다. 네트워크 내의 모든 큐 관리자는 연관된 DLQ가 있어야 합니다.

메시지는 큐 관리자, 메시지 채널 에이전트(MCA) 및 애플리케이션에 의해 DLQ에 넣어질 수 있습니다. DLQ에 대한 모든 메시지는 데드 레터 헤더 구조, MQDLH를 접두부로 사용합니다.

큐 관리자 또는 메시지 채널 에이전트에 의해 DLQ에 넣어진 메시지는 항상 MQDLH를 가집니다. DLQ에 메시지를 넣는 애플리케이션이 MQDLH를 제공해야 합니다. MQDLH 구조의 Reason 필드에는 메시지가 DLQ에 있는 이유를 식별하는 이유 코드가 포함됩니다.

모든 WebSphere MQ 환경에는 DLQ에 있는 메시지를 정기적으로 처리하기 위한 루틴이 필요합니다.

WebSphere MQ는 데드-레터 큐 핸들러(DLQ 핸들러)라는 기본 루틴을 제공하며, runmqdlq 명령을 사용하여 이 루틴을 호출할 수 있습니다.

DLQ에 있는 메시지를 처리하기 위한 지시사항은 사용자 작성 규칙 테이블 방식으로 DLQ 핸들러에 제공됩니다. 즉, DLQ 핸들러는 규칙 테이블의 입력 항목에 대응하여 DLQ에 있는 메시지와 일치합니다. DLQ 메시지가 규칙 테이블에 있는 입력 항목과 일치하면 DLQ 핸들러가 해당 입력 항목과 연관된 조치를 수행합니다.

DLQ 핸들러 호출

runmqdlq 명령을 사용하여 DLQ 핸들러를 호출하십시오. 처리하려는 DLQ 및 두 가지 방법으로 사용하려는 큐 관리자의 이름을 지정할 수 있습니다.

두 가지 방법은 다음과 같습니다.

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

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrulename.rul
```

- 규칙 테이블에서. 예를 들면, 다음과 같습니다.

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

예가 큐 관리자 `ABC1.QUEUE.MANAGER`가 소유하는 `ABC1.DEAD.LETTER.QUEUE`라는 DLQ에 적용됩니다.

표시된 대로 DLQ 또는 큐 관리자를 지정하지 않으면, 설치를 위한 기본 큐 관리자가 해당 큐 관리자에 속하는 DLQ와 함께 사용됩니다.

`runmqdlq` 명령은 `stdin`에서 입력사항을 가져옵니다. 규칙 테이블에서 `stdin`을 경로 재지정하여 규칙 테이블을 `runmqdlq`와 연관시킵니다.

DLQ 핸들러를 실행하려면 DLQ의 메시지가 전달되는 임의의 메시지 큐 및 DLQ 자체 모두에 액세스하기 위한 권한이 부여되어야 합니다. 메시지 컨텍스트에서 사용자 ID의 권한을 가지는 큐에 메시지를 넣기 위한 DLQ 핸들러의 경우, 기타 사용자의 ID를 가정하기 위한 권한이 부여되어야 합니다.

`runmqdlq` 명령에 대한 자세한 정보는 [runmqdlq](#)를 참조하십시오.

샘플 DLQ 핸들러, *amqsdld*

`runmqdlq` 명령을 사용하여 호출된 DLA 핸들러 외에도, WebSphere MQ는 `runmqdlq`가 제공한 것과 유사한 함수가 있는 샘플 DLQ 핸들러 `amqsdld`의 소스를 제공합니다.

요구사항을 충족하는 DLQ 핸들러를 제공하기 위해 `amqsdld`를 사용자 정의할 수 있습니다. 예를 들어, 데드-레터 헤더 없이 메시지를 처리할 수 있는 DLQ 핸들러를 원한다고 결정할 수 있습니다. (기본 DLQ 핸들러 및 샘플 (`amqsdld`) 모두 데드-레터 헤더 `MQDLH`로 시작하는 DLQ의 해당 메시지만 처리합니다. `MQDLH`로 시작하지 않는 메시지는 정기적으로 오류가 있는 것으로 보고되며, 무기한 DLQ에 남게 됩니다.)

`MQ_INSTALLATION_PATH`는 WebSphere MQ가 설치되어 있는 상위 레벨 디렉토리를 표시합니다.

In WebSphere MQ for 윈도우, the source of `amqsdld` is supplied in the directory:

```
MQ_INSTALLATION_PATH\tools\c\samples\dld
```

그리고 컴파일된 버전이 디렉토리에 제공됩니다.

```
MQ_INSTALLATION_PATH\tools\c\samples\bin
```

In WebSphere MQ for UNIX and Linux systems, the source of `amqsdld` is supplied in the directory:

```
MQ_INSTALLATION_PATH/samp/dld
```

그리고 컴파일된 버전이 디렉토리에 제공됩니다.

```
MQ_INSTALLATION_PATH/samp/bin
```

DLQ 핸들러 규칙 테이블

DLQ 핸들러 규칙 테이블은 DLQ 핸들러가 DLQ에 도착한 메시지를 처리하는 방법을 정의합니다.

규칙 테이블에는 두 가지 유형의 입력 항목이 있습니다.

- 표의 첫 번째 항목(선택적)에는 제어 데이터가 포함됩니다.
- 테이블의 기타 모든 입력 항목은 DLQ 핸들러가 따라야 할 규칙입니다. 각 규칙은 메시지가 일치되는 패턴(메시지 특성 세트)과, DLQ의 메시지가 지정된 패턴과 일치할 때 수행되는 조치로 구성됩니다. 규칙 테이블에는 최소 하나의 규칙이 있어야 합니다.

규칙 테이블의 각 입력 항목은 하나 이상의 키워드로 구성됩니다.

제어 데이터

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

참고:

- 수직 행()은 대안을 분리하며 이들 중 하나만 지정할 수 있습니다.
- 모든 키워드는 선택사항입니다.

INPUTQ (QueueName| '___')

처리하려는 DLQ의 이름입니다.

1. runmqdlq 명령에 매개변수로서 제공하는 INPUTQ 값이 규칙 테이블에서 INPUTQ 값을 대체합니다.
2. runmqdlq 명령에 매개변수로서 INPUTQ 값을 지정하지 않지만 규칙 테이블에서 값을 지정하는 경우, 규칙 테이블의 INPUTQ 값이 사용됩니다.
3. DLQ가 지정되지 않거나 규칙 테이블에서 INPUTQ(' ')를 지정하는 경우, runmqdlq 명령에 매개변수로서 제공되는 이름을 가지는 큐 관리자에 속하는 DLQ의 이름이 사용됩니다.
4. runmqdlq 명령에 매개변수로서 INPUTQ 값을 지정하지 않거나 규칙 테이블에서 값으로 INPUTQ 값을 지정하지 않는 경우, 규칙 테이블의 INPUTQ 키워드에서 이름 지정된 큐 관리자에 속하는 DLQ가 사용됩니다.

INPUTQM (QueueManager이름| '___')

INPUTQ 키워드에서 이름이 지정된 DLQ를 소유하는 큐 관리자의 이름입니다.

1. runmqdlq 명령에 매개변수로서 제공하는 INPUTQM 값이 규칙 테이블에서 INPUTQM 값을 대체합니다.
2. runmqdlq 명령에 매개변수로서 INPUTQM 값을 지정하지 않는 경우, 규칙 테이블에서 INPUTQM 값이 사용됩니다.
3. 큐 관리자가 지정되지 않거나 규칙 테이블에 INPUTQM(' ')를 지정하는 경우, 설치를 위한 기본 큐 관리자가 사용됩니다.

RETRYINT (간격| 60)

DLQ 핸들러가 첫 번째 시도에서 처리될 수 없는 DLQ의 메시지를 처리해야 하고 반복되는 시도가 요청되는 간격(초)입니다. 기본적으로, 재시도 간격은 60초입니다.

WAIT (YES|NO|nnn)

DLQ 핸들러가 처리할 수 있는 추가 메시지가 없다는 것을 감지할 때 DLQ에 도착할 추가 메시지를 기다려야 하는지 여부입니다.

YES

DLQ 핸들러가 무기한 대기합니다.

아니오

DLQ 핸들러는 DLQ가 비어 있거나 처리할 수 있는 메시지가 포함되어 있다는 것을 감지할 때 종료됩니다.

nnn

DLQ 핸들러는 큐가 비어 있거나 처리할 수 있는 메시지가 포함되지 않은 것을 감지한 후 종료하기 전에 새 작업이 도착할 때까지 nnn초 동안 대기합니다.

사용 중인 DLQ에 대해서는 WAIT (YES)를 지정하고 활동 레벨이 낮은 DLQ에 대해서는 WAIT (NO) 또는 WAIT (nnn)을 지정하십시오. DLQ 핸들러가 종료하도록 허용되는 경우, 트리거를 사용하여 다시 호출하십시오. 트리거링에 대한 자세한 정보는 [트리거를 사용하여 WebSphere MQ 애플리케이션 시작](#)의 내용을 참조하십시오.

규칙 테이블에 제어 데이터를 포함하는 것에 대한 대체는 runmqdlq 명령에 입력 매개변수로서 해당 큐 관리자 및 DLQ의 이름을 제공하는 것입니다. 규칙 테이블에 값을 지정하고 runmqdlq 명령에 입력으로 값을 지정하는 것이 모두 가능한 경우, runmqdlq 명령에 지정되는 값이 우위에 섭니다.

규칙 테이블에 제어-데이터 입력 항목을 포함하는 경우, 테이블에 첫 번째 입력 항목일 수 있습니다.

규칙(패턴 및 조치)

DLQ의 메시지가 일치되는 패턴 일치 키워드 및 DLQ 핸들러가 일치 메시지를 처리하는 방법을 판별하는 조치 키워드에 대한 설명입니다. 예시 규칙도 제공됩니다.

패턴 일치 키워드

DLQ의 메시지가 일치되는 값을 지정하려면 사용하는 패턴 일치 키워드는 다음과 같습니다. (모든 패턴 일치 키워드는 선택사항입니다):

APPLIDAT (ApplIdentity데이터|_*)

DLQ의 메시지에 대한 메시지 디스크립터(MQMD)에 지정된 *ApplIdentityData* 값입니다.

APPLNAME (PutAppl이름|_*)

DLQ에 있는 메시지의 메시지 디스크립터, MQMD의 *PutAppl* 이름 필드에 지정된 대로 MQPUT 또는 MQPUT1 호출을 실행한 애플리케이션의 이름입니다.

APPLTYPE (PutAppl유형|_*)

DLQ에 메시지의 메시지 디스크립터, MQMD에 지정되는 *PutApplType* 값입니다.

DESTQ (QueueName|_*)

메시지의 목적지로 지정된 메시지 큐의 이름입니다.

DESTQM (QueueManager이름|_*)

메시지가 향하는 메시지 큐의 큐 관리자의 이름입니다.

FEEDBACK (피드백|_*)

MsgType 값이 MQFB_REPORT일 때, *Feedback*은 보고서의 성질을 설명합니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQFB_COA를 사용하여 목적지 큐에서 도착 확인이 필요한 DLQ에 대한 해당 메시지를 식별할 수 있습니다.

FORMAT (형식|_*)

메시지 송신자가 메시지 데이터 형식을 설명하는 데 사용하는 이름입니다.

MSGTYPE (MsgType|_*)

DLQ에 있는 메시지의 메시지 유형입니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQMT_REQUEST를 사용하여 응답이 필요한 DLQ에서 해당 메시지를 식별할 수 있습니다.

PERSIST (지속성|_*)

메시지의 지속성 값입니다. (메시지의 지속성은 큐 관리자 재시작 후에도 지속되는지 여부를 판별합니다.)

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQPER_PERSISTENT를 사용하여 지속적인 DLQ에서 메시지를 식별할 수 있습니다.

REASON (ReasonCode|_*)

메시지가 DLQ에 놓여진 이유에 대해 설명하는 이유 코드입니다.

기호 이름을 사용할 수 있습니다. 예를 들어, 기호 이름 MQRC_Q_FULL을 사용하여 목적지 큐가 가득 찼기 때문에 DLQ에 배치된 메시지를 식별할 수 있습니다.

REPLYQ (QueueName|_*)

DLQ에서 메시지의 메시지 디스크립터(MQMD)에 지정된 응답 대상 큐의 이름입니다.

REPLYQM (QueueManager이름|_*)

DLQ에서 메시지의 메시지 디스크립터(MQMD)에 지정된 것으로 응답 대상 큐의 큐 관리자 이름입니다.

USERID (UserIdentifier|_*)

DLQ에서 메시지의 메시지 디스크립터(MQMD)에 지정된 것으로 DLQ에서 메시지를 생성한 사용자의 사용자 ID입니다.

조치 키워드

일치 메시지가 처리되는 방법에 대해 설명하기 위해 사용되는 조치 키워드는 다음과 같습니다.

ACTION (DISCARD|IGNORE|RETRY|FWD)

이 규칙에 정의되는 패턴과 일치하는 DLQ에서 메시지에 대해 취하는 조치입니다.

DISCARD

DLQ에서 메시지를 삭제하십시오.

IGNORE

DLQ에 메시지를 남기십시오.

RETRY

해당 목적지 큐에 메시지를 넣으려는 첫 번째 시도가 실패하면 다시 시도하십시오. RETRY 키워드는 조치를 구현하기 위한 시도 횟수를 설정합니다. 제어 데이터의 RETRYINT 키워드는 시도 사이의 간격을 제어합니다.

FWD

FWDQ 키워드에 이름이 지정된 큐로 메시지를 전달합니다.

ACTION 키워드를 지정해야 합니다.

FWDQ (QueueName| 앤드 DESTQ |, REPLYQ)

ACTION(FWD)이 요청될 때 메시지가 전달되는 메시지 큐의 이름입니다.

QueueName

메시지 큐의 이름입니다. FWDQ(' ')가 올바르지 않습니다.

&DESTQ

MQDLH 구조의 *DestQName* 필드에서 큐 이름을 가져옵니다.

&REPLYQ

메시지 디스크립터(MQMD)의 *ReplyToQ* 필드에서 큐 이름을 가져옵니다.

FWDQ (S-REPLYQ) 를 지정하는 규칙이 공백 *ReplyToQ* 필드가 있는 메시지와 일치하는 경우 오류 메시지를 방지하려면 메시지 패턴에 REPLYQ (?*) 를 지정하십시오.

FWDQM (QueueManager이름| 앤드 DESTQM | & REPLYQM | ' _')

메시지를 전달할 큐의 큐 관리자입니다.

QueueManagerName

ACTION(FWD)이 요청될 때 메시지가 전달되는 큐의 큐 관리자의 이름입니다.

&DESTQM

MQDLH 구조의 *DestQMGrName* 필드에서 큐 관리자 이름을 가져옵니다.

&REPLYQM

메시지 디스크립터(MQMD)의 *ReplyToQMGr* 필드에서 큐 관리자 이름을 가져옵니다.

..

기본값인 FWDQM(' ')은 로컬 큐 관리자를 식별합니다.

HEADER (YES|NO)

MQDLH가 ACTION(FWD)이 요청된 메시지에 남아 있어야 하는지 여부를 지정합니다. 기본적으로 MQDLH는 메시지에 남아 있습니다. HEADER 키워드는 FWD 이외의 다른 조치에는 유효하지 않습니다.

PUTAUT (DEF|CTX)

DLQ 핸들러가 메시지를 넣을 때 필요한 권한입니다.

DEF

DLQ 핸들러 자체 권한을 가진 메시지를 넣으십시오.

CTX

메시지 컨텍스트에서 사용자 ID의 권한을 가진 메시지를 넣으십시오. PUTAUT(CTX)를 지정하면, 기타 사용자의 ID를 추측할 수 있는 권한이 지정되어야 합니다.

RETRY (RetryCount| 1)

제어 데이터의 RETRYINT 키워드에 지정된 간격에서 조치를 시도하기 위한 횟수(1 - 999,999,999 범위)입니다. DLQ 핸들러가 특정 규칙을 구현하기 위해 시도한 횟수는 DLQ 핸들러의 현재 인스턴스에 따라 다릅니다. 재시작하는 동안에는 계수가 지속되지 않습니다. DLQ 핸들러가 재시작되면, 규칙에 적용되는 시도 횟수는 0으로 재설정됩니다.

예 규칙

다음은 DLQ 핸들러 규칙 테이블의 규칙 예입니다.

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

이 규칙에 따라 DLQ 핸들러는 MQPUT 및 MQPUT1이 금지되어 DLQ에 있는 지속적인 메시지를 목적지 큐에 이동하도록 세 번 시도합니다.

규칙에 사용할 수 있는 모든 키워드에 대해서는 이 절 나머지 부분에서 설명됩니다. 다음에 유의하십시오.

- 키워드가 있는 경우 기본값은 밑줄 그어져 있습니다. 대부분의 키워드의 경우 기본값이 *(별표)이며 어느 값과도 일치합니다.
- 수직 행(|)은 대안을 분리하며 이들 중 하나만 지정할 수 있습니다.
- 모든 키워드 예외 ACTION은 선택사항입니다.

규칙 테이블 규약

DLQ 핸들러 규칙 테이블의 콘텐츠, 구조 및 구문은 이러한 규약을 고수해야 합니다.

규칙 테이블은 다음 규약을 고수해야 합니다.

- 규칙 테이블에는 최소 하나의 규칙이 있어야 합니다.
- 키워드는 임의의 순서로 발생할 수 있습니다.
- 키워드는 규칙에 한 번만 포함될 수 있습니다.
- 키워드는 대소문자를 구분하지 않습니다.
- 키워드 및 매개변수 값은 최소 하나의 공백이나 쉼표로 다른 키워드와 구분해야 합니다.
- 공백은 규칙 처음이나 끝에 넣을 수 있으며, 키워드, 구두점, 값 사이에 공백을 넣을 수 있습니다.
- 각 규칙은 새 행에서 시작해야 합니다.
- Windows 시스템에서 테이블의 마지막 규칙은 캐리지 리턴/개행 문자로 끝나야 합니다. 테이블의 마지막 행이 빈 줄이도록 규칙의 끝에서 Enter 키를 입력하는지 확인하여 이를 얻을 수 있습니다.
- 이식성의 이유로 행의 중요한 길이는 72자를 넘어서는 안됩니다.
- 다음 행의 첫 번째 비공백 문자부터 규칙이 계속되는 것을 표시하려면 행의 마지막 비공백 문자로 더하기 부호(+)를 사용하십시오. 다음 행의 시작부터 규칙이 계속되는 것을 표시하려면 행의 마지막 비공백 문자로 빼기 부호(-)를 사용하십시오. 키워드 및 매개변수 내에서 연속 문자가 발생할 수 있습니다.

예를 들면, 다음과 같습니다.

```
APPLNAME('ABC+  
D')
```

결과적으로 'ABCD'가 되고

```
APPLNAME('ABC-  
D')
```

결과: 'ABC D'.

- 별표(*)로 시작되는 주석 행은 규칙 테이블에서 어디에나 발생할 수 있습니다.
- 빈 줄은 무시됩니다.
- DLQ 핸들러 규칙 테이블의 각 입력 항목은 하나 이상의 키워드 및 연관된 매개변수로 구성됩니다. 매개변수는 다음 구문 규칙을 따라야 합니다.
 - 각 매개변수 값은 최소 하나의 유효 문자를 포함해야 합니다. 따옴표로 묶이는 값을 분리하는 작은따옴표는 중요한 것으로 여겨지지 않습니다. 예를 들어, 올바른 매개변수는 다음과 같습니다.

```
FORMAT('ABC ')      3개의 유효 문자
```

FORMAT (ABC)	3개의 유효 문자
FORMAT (' A ')	1개의 유효 문자
FORMAT (A)	1개의 유효 문자
FORMAT (' ')	1개의 유효 문자

이러한 매개변수는 중요한 문자를 포함하지 않으므로 올바르지 않습니다.

```
FORMAT ( ' ' )
FORMAT ( )
FORMAT ( )
FORMAT
```

- 와일드카드 문자가 지원됩니다. 후미 공백을 제외하고는 단일 문자 대신 물음표(?)를 사용할 수 있습니다. 0 개 이상의 인접 문자 대신 별표(*)를 사용할 수 있습니다. 별표(*) 및 물음표(?)는 매개변수 값에서 **항상** 와일드카드 문자로 해석됩니다.
- 와일드카드 문자는 키워드(ACTION, HEADER, RETRY, FWDQ, FWDQM 및 PUTAUT)의 매개변수에 포함될 수 없습니다.
- 매개변수 값과 DLQ 메시지의 해당 필드에 있는 후미 공백은 와일드카드 일치룰 수행할 때 유의미하지 않습니다. 그러나, 작은따옴표로 묶인 문자열 내의 선두 문자 및 임베드된 공백은 와일드카드 일치에 중요합니다.
- 숫자 매개변수는 물음표(?) 와일드카드 문자를 포함할 수 없습니다. 전체 숫자 매개변수 대신 별표(*)를 사용할 수 있지만, 숫자 매개변수의 일부로서 사용할 수 없습니다. 예를 들어, 다음은 올바른 숫자 매개변수입니다.

MSGTYPE (2)	응답 메시지만 적합합니다.
MSGTYPE (*)	모든 메시지 유형이 적합합니다.
MSGTYPE ('*')	모든 메시지 유형이 적합합니다.

그러나 MSGTYPE ('2*')는 숫자 매개변수의 일부로 별표(*)가 포함되어 있으므로 올바르지 않습니다.

- 숫자 매개변수는 범위 0-999 999 999에 있어야 합니다. 매개변수 값이 이 범위에 있으면 키워드와 관련된 필드에서 현재 그 값이 올바르지 않더라도 허용됩니다. 숫자 매개변수에 기호 이름을 사용할 수 있습니다.
- 문자열 값이 키워드와 관련된 MQDLH 또는 MQMD에 있는 필드보다 짧으면, 필드 길이에 맞게 값이 공백으로 채워집니다. 별표(*)를 제외하고 값이 필드보다 길면 오류가 진단됩니다. 예를 들어, 다음은 모두 8개 문자 필드에서 올바른 문자열 값입니다.

'ABCDEFGH'	8자
'A*C*E*G*I'	별표를 제외한 5문자
'*A*C*E*G*I*K*M*O*'	별표를 제외한 8문자

- 공백, 소문자 또는 마침표(.), 정방향 슬래시(/), 밑줄(_) 및 퍼센트 부호(%) 이외의 특수 문자가 포함된 문자열을 작은따옴표로 묶으십시오. 작은따옴표로 묶이지 않은 소문자는 대문자로 접힙니다. 문자열에 따옴표가 포함되면, 두 개의 작은따옴표를 사용하여 따옴표 시작과 끝을 모두 나타내야 합니다. 문자열의 길이를 계산할 때, 큰따옴표는 각각 하나의 문자로 계수됩니다.

규칙 테이블이 처리되는 방법

DLQ 핸들러는 패턴이 DLQ의 메시지와 일치하는 규칙에 대해 규칙 테이블을 검색합니다.

검색은 테이블의 첫 번째 규칙부터 시작되며 테이블 전체에서 순차적으로 계속됩니다. DLQ 핸들러가 일치하는 패턴을 가진 규칙을 찾을 때, 해당 규칙으로부터 조치를 취합니다. DLQ 핸들러는 해당 규칙에 적용될 때마다 규칙에 대한 재시도 수가 1씩 증가합니다. 첫 번째 시도에 실패한 경우, DLQ 핸들러는 RETRY 키워드에 지정되는

수와 시도 수가 일치할 때까지 다시 시도합니다. 모든 시도가 실패하면, DLQ 핸들러는 테이블에서 그 다음으로 일치하는 규칙을 검색합니다.

이 프로세스는 조치가 성공할 때까지 일치하는 후속 규칙에 대해 반복됩니다. 일치하는 각 규칙이 RETRY 키워드에 지정된 횟수만큼 시도되고 모든 시도가 실패하면, ACTION(IGNORE)이 가정됩니다. 일치하는 규칙을 찾지 못한 경우에도 ACTION(IGNORE)이 가정됩니다.

참고:

1. DLQ에서 MQDLH로 시작하는 메시지에 대해서만 일치하는 규칙 패턴을 찾습니다. MQDLH로 시작하지 않는 메시지는 정기적으로 오류가 있는 것으로 보고되며, 무기한 DLQ에 남게 됩니다.
2. 규칙이 조치로만 구성될 수 있도록, 모든 패턴 키워드는 기본값에 허용될 수 있습니다. 그러나 조치만으로 구성된 규칙은 MQDLH가 있고 테이블의 다른 규칙에 따라 아직 처리되지 않은 큐의 모든 메시지에 적용됩니다.
3. DLQ 핸들러가 시작될 때 규칙 테이블의 유효성이 검증되고 그 때 오류가 플래그됩니다. 언제든지 규칙 테이블로 변경할 수 있지만, DLQ 핸들러가 다시 시작할 때까지 해당 변경사항이 발효되지 않습니다.
4. DLQ 핸들러는 메시지, MQDLH 또는 메시지 디스크립터의 콘텐츠를 대체하지 않습니다. DLQ 핸들러는 메시지 옵션 MQPMO_PASS_ALL_CONTEXT를 사용하여 항상 다른 큐에 메시지를 넣습니다.
5. 규칙 테이블의 연속 구문 오류는 규칙 테이블이 유효성 검증 동안 반복적인 오류의 생성을 제거하도록 설계되기 때문에 인식되지 않을 수도 있습니다.
6. DLQ 핸들러는 MQOO_INPUT_AS_Q_DEF 옵션을 사용하여 DLQ를 엽니다.
7. DLQ 핸들러의 다중 인스턴스는 동일한 규칙 테이블을 사용하여 동일한 큐에 대해 동시에 실행될 수 있습니다. 그러나, 일반적으로 DLQ와 DLQ 핸들러 간에는 1:1 관계가 성립됩니다.

모든 DLQ 메시지가 처리되었는지 확인

DLQ 핸들러는 발견되었지만 제거되지 않은 모든 메시지의 레코드를 DLQ에 보관합니다.

DLQ 핸들러를 필터로 사용하여 DLQ에서 소규모 메시지 서브세트를 추출하는 경우, DLQ 핸들러는 여전히 DLQ에 아직 처리하지 않은 메시지에 대한 기록을 보존해야 합니다. 또한 DLQ 핸들러는 DLQ가 FIFO(First-In-First-Out)로 정의된 경우에도 DLQ에 도착하는 새 메시지를 볼 수 있다고 보장할 수 없습니다. 큐가 비어있지 않으면, DLQ는 모든 메시지를 검사하기 위해 주기적으로 재검색됩니다.

이러한 이유로, DLQ에 가능한 적은 메시지가 포함되는지 확인하십시오. 기타 큐로 전달되거나 삭제될 수 없는(어떤 이유든) 메시지가 큐에 축적되도록 허용되지 않는 경우, DLQ 핸들러의 워크로드가 증가하고 DLQ 자체가 가득 차게 됩니다.

DLQ 핸들러가 DLQ를 비울 수 있도록 특정 조치를 수행할 수 있습니다. 예를 들어, DLQ에 메시지를 남기는 ACTION(IGNORE)을 사용하지 않도록 하십시오. (ACTION(IGNORE)은 테이블의 다른 규칙에 의해 명시적으로 주소 지정되지 않은 메시지에 대해 가정되는 점을 기억하십시오.) 대신, 무시할 메시지에 대해서는 메시지를 다른 큐로 이동하는 조치를 사용하십시오. 예를 들면, 다음과 같습니다.

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

마찬가지로, 테이블의 마지막 규칙은 테이블의 이전 규칙이 설명하지 못한 메시지를 처리해야 합니다. 예를 들어, 테이블의 최종 규칙은 다음과 같을 수 있습니다.

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

이는 테이블의 최종 규칙으로 보내진 메시지를 큐 REALLY.DEAD.QUEUE로 전달합니다. 여기서 메시지는 수동으로 처리될 수 있습니다. 이러한 규칙이 없으면 메시지가 DLQ에 무기한 남아 있을 가능성이 높습니다.

DLQ 핸들러 규칙 테이블 예

단일 제어 데이터 입력 항목 및 여러 규칙을 포함하는 runmqdlq 명령에 대한 예 규칙 테이블입니다.

```
*****
*           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
* 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)

```

다중 설치

UNIX, Linux, and Windows에서는 시스템에 IBM WebSphere MQ의 사본이 둘 이상 있을 수 있습니다.

IBM WebSphere MQ의 각 사본을 설치할 위치를 선택할 수 있지만, 이러한 사본은 별도의 위치에 설치해야 합니다. 최대 128개의 IBM WebSphere MQ 설치가 한 번에 단일 시스템에 존재할 수 있습니다. 하나의 설치는 IBM WebSphere MQ Version 7.0.1, 수정팩 6 이상의 설치가 될 수 있습니다. 이제 다음과 같은 선택사항이 제공됩니다.

- 시스템에서 IBM WebSphere MQ의 단일 설치를 유지보수 및 관리하는 작업의 단순성을 유지합니다.
- 다중 IBM WebSphere MQ 설치를 통해 제공되는 유연성을 이용합니다.

IBM WebSphere MQ의 다중 사본을 설치하기 전에 여러사항을 결정해야 합니다.

- 시스템에서 IBM WebSphere MQ Version 7.0.1 사본 하나를 작성합니까?

IBM WebSphere MQ Version 7.0.1, 수정팩 6 이상이 시스템에 설치된 경우, 여러가지 고려해야 하는 제한사항이 있습니다.

- UNIX and Linux 시스템에서 IBM WebSphere MQ Version 7.0.1은 기본 위치에 설치해야 합니다.
- IBM WebSphere MQ Version 7.0.1이 시스템에 있는 첫 번째 설치여야 합니다. 버전 7.1 이상을 설치한 후에는 IBM WebSphere MQ Version 7.0.1은 설치할 수 없습니다. 버전 7.0.1을 설치 제거하는 경우, WebSphere MQ의 이후 버전을 설치하는 동안 다시 설치할 수 없습니다.
- IBM WebSphere MQ Version 7.0.1은 자동으로 기본 설치입니다. IBM WebSphere MQ Version 7.0.1을 설치하는 동안 기본 설치로서 다른 설치를 선택할 수 없습니다.

- IBM WebSphere MQ의 각 사본을 어디에 설치합니까?

버전 7.1 이상에서 사용자 설치를 위한 설치 위치를 선택할 수 있습니다. 자세한 정보는 [설치 위치 선택](#)을 참조하십시오.

- 기본 설치가 필요합니까?

기본 설치는 시스템 전체 위치에서 참조하는 설치입니다. 자세한 정보는 109 페이지의 『기본 설치 선택』의 내용을 참조하십시오.

- 애플리케이션이 연결하는 방법은 무엇입니까?

애플리케이션이 적절한 IBM WebSphere MQ 라이브러리를 찾는 방법을 고려해야 합니다. 자세한 정보는 [다중 설치 환경에서 애플리케이션 연결 및 다중 설치 환경에서 .NET 애플리케이션 연결](#)을 참조하십시오.

- 기존 엑시트를 변경해야 합니까?

IBM WebSphere MQ가 기본 위치에 설치되지 않으면, 엑시트를 업데이트해야 합니다. 자세한 정보는 [엑시트 및 설치 가능 서비스의 작성 및 컴파일을 참조](#)하십시오.

- 어느 큐 관리자가 어느 설치와 연관됩니까?

각 큐 관리자가 특정 설치와 연관되어 있습니다. 해당 설치에서의 명령만으로 관리될 수 있도록 큐 관리자가 연관된 설치가 해당 큐 관리자를 제한합니다. 자세한 정보는 [큐 관리자를 설치와 연관](#)을 참조하십시오.

- 각 설치와 작업하도록 사용자 환경을 어떻게 설정합니까?

시스템에 다중 설치가 있는 경우, 특정 설치에 대해 작업하는 방법 및 해당 설치에서 명령을 발행하는 방법을 고려해야 합니다. 명령에 대한 전체 경로를 지정하거나 **setmqenv** 또는 **crtmqenv** 명령을 사용하여 환경 변수를 설정할 수 있습니다. 환경 변수를 설정하면 해당 설치에 대한 명령 경로를 생략할 수 있습니다. 자세한 정보는 [setmqenv 및 crtmqenv](#)를 참조하십시오.

이러한 질문에 응답하면 [IBM WebSphere MQ설치](#)에 제공된 단계를 사용하여 IBM WebSphere MQ를 설치할 수 있습니다.

IBM WebSphere MQ의 기존 설치가 있고 다중 설치 기능을 사용하여 IBM WebSphere MQ의 한 버전에서 다른 버전으로 이주하려는 경우 [UNIX, Linux 및 Windows에서 다중 설치 큐 관리자 공존](#)의 내용을 참조하십시오.

.NET 지원 팩 및 다중 설치를 위한 IBM 메시징 서비스 클라이언트

다중 버전 지원을 위해, *Java and .NET Messaging and Web Services* 기능은 IBM WebSphere MQ 제품과 함께 설치되어야 합니다. 이 기능에는 *IBM Message Service Client for .NET* 지원팩(IA9H)에 포함된 모든 기능이 포함되어 있습니다. 지원 팩이 시스템에 설치되어 있으면, 다중 버전이 지원되지 않습니다. IBM WebSphere MQ를 설치하기 전에 지원팩을 설치 제거해야 합니다. .NET 기능 설치에 대한 자세한 정보는 [.NET용 WebSphere MQ 클래스 설치](#)를 참조하십시오.

관련 개념

[UNIX, Linux 및 Windows: 버전 7.0.1에서 버전 7.5로 단계별 마이그레이션](#)

[UNIX, Linux 및 Windows: 버전 7.0.1에서 버전 7.5로 다중 스테이지 마이그레이션](#)

관련 태스크

[다중 설치 구성](#)

[시스템에서 WebSphere MQ 설치 찾기](#)

기본 설치 선택

다수의 IBM WebSphere MQ(UNIX, Linux, and Windows) 설치를 지원하는 시스템에서 기본 설치에는 IBM WebSphere MQ 시스템 전체 위치가 가리키는 설치입니다. 기본 설치를 포함하는 것은 선택사항이지만, 포함할 경우 편리합니다.

IBM WebSphere MQ Version 7.1이전에는 한 번에 하나의 제품 인스턴스만 설치할 수 있습니다. Windows 는 여러 글로벌 환경 변수가 해당 설치를 가리키도록 설정되었습니다. UNIX and Linux 는 기호 링크가 단일 설치를 가리키는 `/usr/lib`, `/usr/bin` 및 `/usr/include`에 추가되었습니다.

Version 7.1에서 IBM WebSphere MQ 의 다중 버전을 UNIX, Linux, and Windows에 설치할 수 있습니다. 언제 든지 이러한 시스템 중 하나에 둘 이상의 IBM WebSphere MQ 설치가 있을 수 있으며, 선택적으로 이러한 설치 중 하나를 기본 설치로 구성할 수 있습니다. 단일 설치를 가리키는 환경 변수 및 기호 링크는 다중 버전이 존재하는 경우 의미가 없을 수 있습니다. 그러나 일부 기능을 사용하려면 이러한 시스템 전체 위치가 작동해야 합니다. 예를 들어, IBM WebSphere MQ 및 타사 제품을 관리하기 위한 사용자 정의 사용자 스크립트가 있습니다. 이러한 기능은 기본 설치에서만 사용할 수 있습니다.

UNIX and Linux 설치를 기본 설치로 설정하면 외부 라이브러리에 대한 기호 링크 및 해당 설치의 제어 명령 이 `/usr/lib` 및 `/usr/bin`에 추가됩니다. 기본 설치를 사용하고 있지 않는 경우에는 기호 링크가 작성되지 않습니다. 기본 설치에 작성된 기호 링크 목록은 [UNIX and Linux에서 기본 설치에 대한 외부 라이브러리 및 제어 명령 링크](#)를 참조하십시오.

Windows 글로벌 환경 변수는 기본 설치가 설치된 디렉토리를 가리킵니다. 이러한 환경 변수는 IBM WebSphere MQ 라이브러리, 제어 명령 및 헤더 파일을 찾는 데 사용됩니다. 또한 Windows 운영 체제의 일부 기능에는 단일 프로세스로 로드되는 인터페이스 라이브러리의 중앙 등록이 필요합니다. IBM WebSphere MQ의 여러 버전에서는 IBM WebSphere MQ 라이브러리의 충돌 세트가 있습니다. 이 기능은 단일 프로세스로 이러한 라이브러리의 충돌하는 세트를 로드하려고 합니다. 따라서 이러한 기능은 기본 설치에서만 사용할 수 있습니다. 기본 설치와 함께 사용할 수 있는 일부 기능에 대한 세부사항은 [Windows](#)을 참조하십시오.

시스템에 IBM WebSphere MQ Version 7.0.1 설치가 있는 경우 이 설치는 자동으로 기본 설치입니다. Version 7.0.1 이 설치되어 있는 동안에는 기본 설치를 변경할 수 없습니다. 시스템의 모든 설치가 Version 7.1이상인 경우 기본 설치 여부를 선택할 수 있습니다. [109 페이지의 표 11](#)의 옵션을 고려하십시오.

표 11. 기본 설치 옵션.

이 표에서는 기본 설치에 적합한 설치 구성을 보여줍니다. 단일 Version 7.1이상의 경우 기본 또는 비기본이 될 수 있습니다. 다중 설치를 사용하는 경우 Version 7.0.1 에 하나, Version 7.1이상에서 하나 이상, Version 7.0.1 은 기본이어야 하며 다른 설치는 기본이 아니어야 합니다. Version 7.1이상에서 다중 설치를 사용하면 하나의 설치가 기본이거나 모든 설치가 비기본이 될 수 있습니다.

옵션	올바른 설치 구성		자세한 정보
	1차	기본이 아닌 설치	

표 11. 기본 설치 옵션.

이 표에서는 기본 설치에 적합한 설치 구성을 보여줍니다. 단일 Version 7.1이상의 경우 기본 또는 비기본이 될 수 있습니다. 다중 설치를 사용하는 경우 Version 7.0.1 에 하나, Version 7.1이상에서 하나 이상, Version 7.0.1 은 기본이어야 하며 다른 설치는 기본이 아니어야 합니다. Version 7.1이상에서 다중 설치를 사용하면 하나의 설치가 기본이거나 모든 설치가 비기본이 될 수 있습니다.

(계속)

옵션	올바른 설치 구성		자세한 정보
Version 7.1이상의 단일 설치.	Version 7.1 이상.	없음	이전 릴리스와 동일한 방식으로 단일 설치에서 작업을 계속하려면 설치를 기본 설치로 구성하십시오. 이 옵션에 대한 정보는 기본 설치로 구성된 IBM WebSphere MQ Version 7.1 이상의 단일 설치를 참조하십시오.
	없음	Version 7.1 이상.	단일 설치에서 작업을 계속하지만 기호 링크나 글로벌 환경 변수가 자동으로 작성되지 않게 하려면 설치를 기본이 아닌 설치로 구성하십시오. 이 옵션의 의미에 대한 정보는 기본 이외로 구성된 IBM WebSphere MQ Version 7.1 이상의 단일 설치를 참조하십시오.
다중 설치: Version 7.0.1 및 Version 7.1이상	Version 7.0.1	Version 7.1 이상.	버전이 7.0.1인 IBM WebSphere MQ의 다중 설치를 사용하려는 경우 버전 7.0.1 설치가 자동으로 기본 설치입니다. IBM WebSphere MQ 버전 7.0.1 이 설치되어 있는 동안에는 기본 설치인 설치를 변경할 수 없습니다. 이 옵션 및 해당 기능에 대한 정보는 IBM WebSphere MQ의 다중 설치, Version 7.0.1 의 내용을 참조하십시오.
다중 설치: Version 7.1이상.	Version 7.1 이상.	Version 7.1 이상.	버전 7.1에서 WebSphere MQ의 다중 설치를 사용하려는 경우에는 기본 설치 중 하나를 적용할지 여부를 선택할 수 있습니다. 이 옵션에 대한 정보는 IBM WebSphere MQ Version 7.1 이상의 다중 설치를 참조하십시오.
	없음	Version 7.1 이상.	

관련 개념

[기본 설치로 구성된 WebSphere MQ 버전 7.1 이상의 단일 설치](#)

[기본이 아닌 설치로 구성된 WebSphere MQ 버전 7.1 이상의 단일 설치](#)

[WebSphere MQ 버전 7.1 이상의 다중 설치](#)

[WebSphere MQ의 다중 설치, 버전 7.0.1에 하나](#)

관련 태스크

[기본 설치 변경](#)

[설치 위치 선택](#)

[설치 계획](#)

[설치 이름 선택](#)

기본 설치로 구성된 IBM WebSphere MQ Version 7.1 이상의 단일 설치

IBM WebSphere MQ 설치를 기본으로 표시하면 기호 링크 또는 글로벌 환경 변수가 시스템에 추가되므로 애플리케이션에서 사용되는 IBM WebSphere MQ 명령 및 라이브러리가 필요한 최소 시스템 설정으로 자동으로 사용 가능합니다.

IBM WebSphere MQ를 설치할 위치를 결정합니다.

가능한 경우 IBM WebSphere MQ 제어 명령 또는 IBM WebSphere MQ 라이브러리를 찾기 위해 시스템 검색 경로를 사용하도록 애플리케이션 및 스크립트를 구성하십시오. 이 애플리케이션 및 스크립트 구성에서는 IBM WebSphere MQ의 다음 릴리스로 마이그레이션하거나 두 번째 설치를 설치하는 것과 같은 향후 태스크를 이행하는 데 최대한의 유연성을 제공합니다. 애플리케이션을 연결하는 옵션에 대한 자세한 정보는 [다중 설치 환경에서 애플리케이션 연결을 참조하십시오.](#)

Windows에서는 첫 번째 설치가 기본 설치로 자동 구성됩니다. UNIX and Linux 플랫폼에서는 시스템의 첫 번째 설치가 기본 설치가 되도록 수동으로 구성해야 합니다. **setmqinst** 명령을 사용하여 기본 설치를 설정하십시오. 자세한 정보는 [기본 설치 설치 제거, 업그레이드 및 유지보수를 참조하십시오](#).

관련 태스크

[기본 설치 변경](#)

[설치 위치 선택](#)

[설치 계획](#)

[설치 이름 선택](#)

기본이 아닌 설치로 구성된 IBM WebSphere MQ Version 7.1 이상의 단일 설치

IBM WebSphere MQ Version 7.1 이상을 기본이 아닌 설치로 실행하는 경우 IBM WebSphere MQ 라이브러리를 로드하기 위해 애플리케이션에 대한 라이브러리 경로를 구성해야 합니다. Windows에서는 일부 제품 기능이 IBM WebSphere MQ이 기본 설치로 구성된 경우에만 사용 가능합니다.

UNIX 및 Linux 시스템

UNIX and Linux에서 기본이 아닌 설치 실행의 의미는 다음과 같습니다.

- 임베디드 라이브러리 경로 (예: RPATH) 를 사용하여 해당 IBM WebSphere MQ 라이브러리를 찾는 응용프로그램은 다음 조건이 참인 경우 해당 라이브러리를 찾을 수 없습니다.
 - IBM WebSphere MQ 은 RPATH 에 지정된 디렉토리외 다른 디렉토리에 설치됩니다.
 - /usr에 기호 링크가 없는 경우
- 애플리케이션이 외부 라이브러리 경로 (예: LD_LIBRARY_PATH) 를 사용하여 라이브러리를 찾는 경우, **MQ_INSTALLATION_PATH/lib** 또는 **MQ_INSTALLATION_PATH/lib64** 디렉토리를 포함하도록 외부 라이브러리 경로를 구성해야 합니다. **setmqenv** 및 **crtmqenv** 명령은 외부 라이브러리 경로를 포함하여 현재 셸에서 여러 환경 변수를 구성할 수 있습니다.
- 대부분의 IBM WebSphere MQ 프로세스는 setuid/setgid로 실행됩니다. 따라서 사용자 엑시트를 로드하는 경우 외부 라이브러리 경로를 무시합니다. IBM WebSphere MQ 라이브러리를 참조하는 사용자 엑시트는 여기에 임베드된 라이브러리 경로에서 찾을 수 있는 경우에만 해당 라이브러리를 찾을 수 있습니다. /usr에 기호 링크가 있으면 링크가 분석됩니다. IBM WebSphere MQ Version 7.1 이상에서 실행하려는 사용자 엑시트는 이제 빌드 가능합니다. 이를 통해 IBM WebSphere MQ 라이브러리를 참조하지 않습니다. 대신 IBM WebSphere MQ에 의존하여 엑시트가 사용할 수 있는 IBM WebSphere MQ 함수에 함수 포인터를 전달합니다. 자세한 정보는 [엑시트 및 설치 가능 서비스의 작성 및 컴파일을 참조하십시오](#).

애플리케이션을 연결하는 옵션에 대한 자세한 정보는 [다중 설치 환경에서 애플리케이션 연결을 참조하십시오](#).

UNIX and Linux 플랫폼에서는 시스템의 첫 번째 설치가 기본 설치로 자동 구성되지 않습니다. 하지만 **dspmqver** 명령을 찾을 수 있도록 단일 기호 링크가 /usr/bin에 포함됩니다. 기호 링크를 원하지 않을 경우 다음 명령을 사용하여 이 링크를 제거해야 합니다.

```
setmqinst -x -p MQ_INSTALLATION_PATH
```

Windows 시스템

Windows에서 기본이 아닌 설치 실행의 의미는 다음과 같습니다.

- 정상적으로 애플리케이션은 외부 라이브러리 경로(PATH)를 사용하여 해당 라이브러리를 찾습니다. 임베드된 라이브러리 경로 또는 명시적 라이브러리 위치의 개념이 없습니다. 설치가 기본이 아닌 경우 글로벌 PATH 환경 변수는 IBM WebSphere MQ 설치 디렉토리를 포함하지 않습니다. 애플리케이션이 IBM WebSphere MQ 라이브러리를 찾으려면 PATH 환경 변수를 업데이트하여 IBM WebSphere MQ 설치 디렉토리를 참조하십시오. **setmqenv** 및 **crtmqenv** 명령은 외부 라이브러리 경로를 포함하여 현재 셸에서 여러 환경 변수를 구성할 수 있습니다.
- 일부 제품 기능은 설치가 기본 설치로 구성된 경우에만 사용 가능합니다. [Windows 에서 기본 설치에만 사용할 수 있는 기능을 참조하십시오](#).

기본적으로 Windows에서는 첫 번째 설치가 1차로 자동 구성됩니다. 이 설치를 기본 설치로 수동으로 선택 취소해야 합니다.

관련 태스크

[기본 설치 변경](#)
[설치 위치 선택](#)
[설치 계획](#)
[설치 이름 선택](#)

관련 참조

[setmqenv](#)
[crtmqenv](#)

IBM WebSphere MQ Version 7.1 이상의 다중 설치

IBM WebSphere MQ Version 7.1 이상 설치 중 하나를 기본 설치로 구성하도록 선택할 수 있습니다. 선택은 애플리케이션이 라이브러리를 찾는 방식에 따라 달라집니다.

IBM WebSphere MQ Version 7.1 로 제공되는 IBM WebSphere MQ 라이브러리 (예: mqm) 는 연결 중인 큐 관리자에 필요한 레벨의 라이브러리를 자동으로 사용합니다. 즉, 애플리케이션이 IBM WebSphere MQ 라이브러리를 IBM WebSphere MQ Version 7.1 설치에서 찾는 경우 해당 시스템의 큐 관리자에 연결할 수 있습니다. 하나의 IBM WebSphere MQ Version 7.1 설치를 기본으로 구성하면 애플리케이션이 IBM WebSphere MQ 인터페이스 라이브러리를 찾는 경우 애플리케이션이 모든 큐 관리자에 연결할 수 있도록 보장합니다.

다중 설치 환경에서 애플리케이션을 연결하는 데 대한 자세한 정보는 [다중 설치 환경에서 애플리케이션 연결을 참조하십시오](#).

기본 설치가 설치 제거되어도 기본 설치는 자동으로 변경되지 않습니다. 다른 설치가 기본 설치가 되도록 하려는 경우, **setmqinst** 명령을 사용하여 기본 설치를 직접 설정해야 합니다. 자세한 정보는 [기본 설치 설치 제거, 업그레이드 및 유지보수를 참조하십시오](#).

관련 개념

[다중 설치](#)

관련 태스크

[기본 설치 변경](#)
[설치 위치 선택](#)
[설치 계획](#)
[설치 이름 선택](#)

Version 7.0.1 에 있는 IBM WebSphere MQ의 다중 설치

IBM WebSphere MQ Version 7.1 이상은 약간의 제한사항이 있지만 IBM WebSphere MQ Version 7.0.1과 공존할 수 있습니다.

- UNIX and Linux 시스템에서 Version 7.0.1 은 고정된 기본 위치에만 설치할 수 있으므로 해당 기본 위치에 Version 7.1 이상을 설치할 수 없습니다.
- IBM WebSphere MQ Version 7.0.1은 자동으로 기본 설치로 구성됩니다. UNIX and Linux 시스템의 경우 해당 IBM WebSphere MQ 디렉토리로 기호 링크가 자동 작성됩니다. Windows의 경우 제품이 제공하는 모든 항목이 글로벌로 등록됩니다. IBM WebSphere MQ Version 7.0.1 작업하려면 이 방법으로 설치해야 합니다. 따라서 IBM WebSphere MQ Version 7.0.1이 설치된 경우 IBM WebSphere MQ Version 7.1 이상 설치는 기본이 될 수 없습니다.

IBM WebSphere MQ Version 7.1 라이브러리는 IBM WebSphere MQ Version 7.0.1 이상에서 실행 중인 큐 관리자에 대해 작업할 수 있습니다. 응용프로그램이 Version 7.0.1 아래에서 실행 중인 큐 관리자에 연결해야 하는 경우, 다음 조건이 충족되면 정상적으로 계속 작동할 수 있습니다.

- 런타임 시에 IBM WebSphere MQ Version 7.1 이상의 라이브러리를 찾습니다.
- Version 7.0.1에서 사용 가능한 기능만 사용합니다.

다중 설치 환경에서 애플리케이션을 연결하는 데 대한 자세한 정보는 [다중 설치 환경에서 애플리케이션 연결을 참조하십시오](#).

IBM WebSphere MQ Version 7.0.1을 설치 제거해도 기본 설치가 자동으로 변경되지 않습니다. 다른 설치가 기본 설치가 되도록 하려는 경우, **setmqinst** 명령을 사용하여 기본 설치를 직접 설정해야 합니다. 자세한 정보는 [기본 설치 설치 제거, 업그레이드 및 유지보수를 참조하십시오.](#)

관련 개념

[다중 설치](#)

[관련 태스크](#)

[설치 위치 선택](#)

[설치 계획](#)

[설치 이름 선택](#)

스토리지 및 성능 요구사항 계획

IBM WebSphere MQ 시스템에 맞는 실제적이고 도달 가능한 스토리지 및 성능 목표를 설정해야 합니다. 링크를 사용하여 플랫폼의 스토리지 및 성능에 영향을 미치는 요인을 알아보십시오.

요구사항은 IBM WebSphere MQ 를 사용 중인 시스템 및 사용할 컴포넌트에 따라 다릅니다.

지원되는 하드웨어 및 소프트웨어 환경에 대한 최신 정보는 [IBM WebSphere MQ에 대한 시스템 요구사항](#) 웹 사이트를 참조하십시오.

www.ibm.com/software/integration/wmq/requirements/

IBM WebSphere MQ는 파일 시스템에 큐 관리자 데이터를 저장합니다. IBM WebSphere MQ에서 사용할 디렉토리 구조 계획 및 구성에 대해 알아보려면 다음 링크를 사용하십시오.

- [115 페이지의 『파일 시스템 지원 계획』](#)
- [115 페이지의 『공유 파일 시스템에 대한 요구사항』](#)
- [124 페이지의 『IBM WebSphere MQ 파일 공유』](#)
- [126 페이지의 『UNIX and Linux 시스템의 디렉토리 구조』](#)
- [136 페이지의 『Windows 시스템의 디렉토리 구조』](#)

UNIX and Linux에서 시스템 자원, 공유 메모리 및 프로세스 우선순위에 대한 정보는 다음 링크를 사용하십시오.

- [139 페이지의 『IBM WebSphere MQ 및 UNIX System V IPC 자원』](#)
- [140 페이지의 『AIX의 공유 메모리』](#)
- [140 페이지의 『WebSphere MQ 및 UNIX 프로세스 우선순위』](#)

관련 개념

[5 페이지의 『계획 중』](#)

IBM WebSphere MQ 환경을 계획하는 경우 구성하려는 IBM WebSphere MQ 아키텍처, 자원 요구사항, 로깅에 대한 필요 및 백업 기능을 고려해야 합니다. 이 토픽의 링크를 사용하여 IBM WebSphere MQ가 실행되는 환경을 계획하십시오.

[13 페이지의 『IBM WebSphere MQ 아키텍처 설계』](#)

[포인트-투-포인트 및 Pub/Sub 메시징 스타일에 대해 IBM WebSphere MQ에서 지원하는 서로 다른 아키텍처에 대해 확인하십시오.](#)

[UNIX 및 Linux 에서 하드웨어 및 소프트웨어 요구사항](#)

[Windows에서의 하드웨어 및 소프트웨어 요구사항](#)

디스크 공간 요구사항



WebSphere MQ의 스토리지 요구사항은 설치하는 컴포넌트와 필요한 작업 공간의 양에 따라 달라집니다.

디스크 스토리지는 필요한 필수조건 컴포넌트를 포함하여 설치하도록 선택한 선택적 컴포넌트에 필수적입니다. 총 스토리지 요구사항은 사용하는 큐의 수, 큐에 있는 메시지의 수와 크기, 메시지가 지속적인지 여부에 따라 다릅니다. 자체 애플리케이션 프로그램에 대한 공간 외에 디스크, 테이프 또는 기타 매체의 아카이브 용량도 필요합니다.

다음 표에는 여러 다른 플랫폼에 다양한 제품 조합을 설치할 때 필요한 대략적인 디스크 공간이 표시되어 있습니다. (값은 5MB 근사값으로 반올림되며 1MB는 1048576바이트입니다.)

플랫폼	클라이언트 설치 ¹	서버 설치 ²	WebSphere MQ MFT 설치 ³	전체 설치 ⁴
AIX	145MB	190MB	705MB	915MB
HP-UX	225MB	310MB	1075MB	1340MB
IBM i	215MB	450MB	80MB	655MB
System x (32비트) 의 경우 Linux	85MB	N/A	N/A	120MB
System x (64비트) 의 경우 Linux	125MB	170MB	575MB	935MB
Linux on POWER Systems-빅 엔디안 (Big Endian)	130MB	170MB	565MB	715MB
Solaris x86-64, AMD64, EM64T 및 호환 가능 프로세서	105MB ⁵	150MB ⁵	695MB	860MB
Solaris SPARC	105MB ⁵	150MB ⁵	680MB	820MB
Windows(32비트 설치) ⁶	390MB	N/A	N/A	475MB
Windows(64비트 설치) ⁶	445MB	555MB	710MB	1005MB

사용시 참고사항

- 클라이언트 설치하는 다음 컴포넌트를 포함합니다.
 - 런타임
 - 클라이언트
- 서버 설치하는 다음 컴포넌트를 포함합니다.
 - 런타임
 - SERVER
- IBM WebSphere MQ Managed File Transfer 설치하는 다음 컴포넌트를 포함합니다.
 - IBM WebSphere MQ Managed File Transfer 서비스, 로거, 에이전트, 도구, 기본 컴포넌트
 - 런타임
 - SERVER
 - Java
 - JRE
- 전체 설치하는 사용 가능한 모든 컴포넌트를 포함합니다.
-  Solaris 플랫폼에서는 이러한 컴포넌트 조합을 가져오기 위해 자동으로 설치해야 합니다.
-  여기에 나열된 모든 컴포넌트가 Windows 시스템에 설치 가능한 기능이 아닙니다. 일부 기능은 때로 기타 기능에 포함됩니다. Windows 시스템의 WebSphere MQ 기능을 참조하십시오.

관련 태스크

[설치 항목 선택](#)

파일 시스템 지원 계획

큐 관리자 데이터는 파일 시스템에 저장됩니다. 큐 관리자는 파일 시스템 잠금을 이용하여 다중 인스턴스 큐 관리자의 다중 인스턴스가 동시에 활성화되지 않게 합니다.

공유 파일 시스템

공유 파일 시스템을 통해 여러 시스템이 동일한 물리적 스토리지 디바이스에 동시에 액세스할 수 있습니다. 잠금 및 동시성 제어를 시행할 어떤 수단 없이 여러 시스템이 동일한 물리적 스토리지 디바이스에 직접 액세스한 경우에는 손상이 발생합니다. 운영 체제는 로컬 파일 시스템에 로컬 프로세스에 대한 잠금 및 동시성 제어를 제공하고 네트워크 파일 시스템은 분산 시스템에 대한 잠금 및 동시성 제어를 제공합니다.

네트워크 파일 시스템은 과거에 로깅 메시지의 요구사항을 충족시킬 정도로 충분히 빠르게 수행되지 않았거나 충분한 잠금 및 동시성 제어를 제공하지 않았습니다. 하지만 현재는, 네트워크 파일 시스템이 우수한 성능 및 신뢰할 수 있는 네트워크 파일 시스템 프로토콜(예: *RFC 3530*, *NFS(Network File System)* 버전 4 프로토콜)의 구현을 제공하고 로깅 메시지에 대한 요구사항을 확실히 충족시킬 수 있습니다.

공유 파일 시스템 및 WebSphere MQ

다중 인스턴스 큐 관리자의 큐 관리자 데이터는 공유 네트워크 파일 시스템에 저장됩니다. Microsoft Windows, UNIX and Linux 시스템에서 큐 관리자의 데이터 파일 및 로그 파일은 공유 네트워크 파일 시스템에 배치되어야 합니다.

릴리스 v7.0.1 이전 버전의 WebSphere MQ는 공유 파일 시스템으로 액세스하는 네트워크 스토리지에 저장된 큐 관리자 데이터를 지원하지 않습니다. 큐 관리자 데이터가 공유 네트워크 스토리지에 있으면 동시에 실행 중인 큐 관리자의 다른 인스턴스가 큐 관리자 데이터에 액세스하지 않는지 확인해야 합니다.

v7.0.1 부터 WebSphere MQ는 잠금을 사용하여 동일한 멀티 인스턴스 큐 관리자의 멀티 인스턴스가 동시에 활성이 되는 것을 방지합니다. 동일한 잠금을 통해 두 개의 개별 큐 관리자가 실수로 동일한 큐 관리자 데이터 파일 세트를 사용할 수 없게 하기도 합니다. 한 번에 큐 관리자의 한 인스턴스만 잠금을 보유할 수 있습니다. 따라서 WebSphere MQ는 공유 파일 시스템으로 액세스하는 네트워크 스토리지에 저장된 큐 관리자 데이터를 지원합니다.

네트워크 파일 시스템의 모든 잠금 프로토콜이 견고하지는 않으며 파일 시스템이 데이터 무결성보다는 성능에 맞게 구성되었을 수 있으므로 **amqmfsc** 명령을 실행하여 네트워크 파일 시스템이 큐 관리자 데이터 및 로그에 대한 액세스를 제대로 제어하는지 테스트해야 합니다. 이 명령은 UNIX 및 IBM i 시스템에만 적용됩니다. Microsoft Windows에서는 지원되는 네트워크 파일 시스템이 한 개뿐이며 **amqmfsc** 명령은 필요하지 않습니다.

관련 태스크

117 페이지의 『공유 파일 시스템 작동 확인』

amqmfsc를 실행하여 UNIX 시스템의 표준 파일 시스템에서 멀티 인스턴스 큐 관리자의 큐 관리자 데이터를 저장하는 데 필요한 요구사항을 충족하는지 여부를 확인합니다. IBM WebSphere MQ MQI client 샘플 프로그램 **amqsfhac**를 **amqmfsc**와 병렬로 실행하여 큐 관리자가 실패 중에 메시지 무결성을 유지하는지 증명합니다.

공유 파일 시스템에 대한 요구사항

공유 파일 시스템은 IBM WebSphere MQ에 대한 신뢰할 수 있는 작업을 위해 데이터 쓰기 무결성, 보장된 독점 파일 액세스, 실패 시의 잠금 해제를 제공해야 합니다.

공유 파일 시스템이 충족해야 하는 요구사항

공유 파일 시스템이 신뢰할 수 있는 메시지 기록을 위해 충족시켜야 하는 세 가지 기본 요구사항이 있습니다.

1. 데이터 쓰기 무결성

데이터 쓰기 무결성을 비울 때 디스크에 쓰기라고도 합니다. 큐 관리자가 물리적 디바이스에 커밋되는 데이터와 동기화시킬 수 있어야 합니다. 트랜잭션 시스템에서는 다른 처리를 계속하기 전에 일부 쓰기가 안전하게 커밋되었는지 확인해야 합니다.

특히, UNIX 플랫폼의 IBM WebSphere MQ 는 *O_SYNC* 열기 옵션을 사용하고 **fsync()** 시스템 호출은 명시적으로 복구 가능한 매체에 쓰기를 강제 실행하며 이러한 옵션이 올바르게 작동하는 것에 종속됩니다.



주의: Linux 동기 쓰기 옵션을 여전히 지원하며 sync 옵션보다 더 나은 성능을 제공하는 async 옵션으로 파일 시스템을 마운트해야 합니다.

하지만 파일 시스템을 Linux에서 내보낸 경우에는 여전히 sync 옵션을 사용하여 파일 시스템을 내보내야 함에 유의하십시오.

2. 보장된 독점 파일 액세스

여러 큐 관리자를 동기화하려면 파일에 대한 배타적 잠금을 확보할 큐 관리자의 메커니즘이 있어야 합니다.

3. 실패 시 잠금 해제

큐 관리자가 실패하는 경우나 파일 시스템과의 통신 장애가 있는 경우 큐 관리자가 잠근 파일을 잠금 해제해서 큐 관리자가 파일 시스템에 다시 연결할 때까지 대기하지 않고 기타 프로세스에 사용 가능하게 해야 합니다.

IBM WebSphere MQ가 확실히 작동하려면 공유 파일 시스템이 이 요구사항을 충족시켜야 합니다. 그렇지 않으면 다중 인스턴스 큐 관리자 구성에 공유 파일 시스템을 사용할 때 큐 관리자 데이터와 로그가 손상됩니다.

Microsoft Windows의 다중 인스턴스 큐 관리자의 경우 Microsoft Windows 네트워크에 사용된 CIFS(Common Internet File System) 프로토콜을 통해 네트워크 스토리지에 액세스해야 합니다. CIFS (Common Internet File System) 클라이언트는 Microsoft Windows 이외의 플랫폼에서의 잠금 시맨틱에 대한 IBM WebSphere MQ의 요구사항을 충족하지 않으므로, Microsoft Windows 이외의 플랫폼에서 실행되는 다중 인스턴스 큐 관리자는 공통 인터넷 파일 시스템 (CIFS) 을 공유 파일 시스템으로 사용하지 않아야 합니다.

기타 지원 플랫폼의 다중 인스턴스 큐 관리자의 경우에는 Posix를 준수하며 임대형 잠금을 지원하는 네트워크 파일 시스템 프로토콜을 통해 스토리지에 액세스해야 합니다. NFS(Network File System) 버전 4와 같은 최신 파일 시스템은 리스된 잠금을 사용하여 장애를 감지한 후 장애에 따라 잠금을 해제합니다. NFS 버전 3과 같은 이전 파일 시스템은 장애 후 잠금을 해제하는 확실한 메커니즘을 가지고 있지 않으며 멀티 인스턴스 큐 관리자에 사용해서는 안 됩니다.

공유 파일 시스템이 요구사항을 충족하는지에 대한 확인

사용할 계획인 공유 파일 시스템이 이 요구사항을 충족시키는지 확인해야 합니다. 파일 시스템이 신뢰할 수 있게 제대로 구성되었는지도 확인해야 합니다. 공유 파일 시스템은 때로 신뢰성 대신 성능을 개선할 구성 옵션을 제공합니다.

정상 상황에서는 IBM WebSphere MQ가 속성 캐싱으로 제대로 작동하며 예를 들어, NFS 마운트에 NOAC를 설정해서 캐싱을 사용 불가능하게 할 필요가 없습니다. 여러 파일 시스템 클라이언트가 파일 시스템 서버의 동일한 파일에 대한 쓰기 액세스를 두고 경합할 때 각 클라이언트에 사용되는 캐싱된 속성이 서버의 속성과 같지 않을 수 있기 때문에 속성 캐싱은 문제를 야기할 수 있습니다. 이러한 방식으로 액세스하는 파일의 예로는 다중 인스턴스 큐 관리자에 대한 큐 관리자 오류 로그가 있습니다. 큐 관리자 오류 로그는 활성 및 대기 큐 관리자 인스턴스 모두에 기록될 수 있으며 캐싱된 파일 속성으로 인해 파일 롤오버가 발생하기 전에 오류 로그가 예상보다 더 커질 수 있습니다.

파일 시스템을 확인하려면 117 페이지의 『공유 파일 시스템 작동 확인』 태스크를 실행하십시오. 이 태스크는 공유 파일 시스템이 요구사항 2 및 3을 충족하는지 확인합니다. 공유 파일 시스템 문서에서 또는 디스크에 로그 데이터 기록을 시험해서 요구사항 1을 확인해야 합니다.

디스크 결함은 디스크에 쓸 때 IBM WebSphere MQ가 FFDC(First Failure Data Capture) 오류로 보고하는 오류를 야기할 수 있습니다. 운영 체제의 파일 시스템 검사기를 실행하여 공유 파일 시스템의 디스크 결함을 확인할 수 있습니다. 예를 들어, UNIX 및 Linux 플랫폼에서 파일 시스템 검사기를 fsck라고 합니다. Windows 플랫폼에서는 파일 시스템 검사기가 CHKDSK 또는 SCANDISK입니다.

NFS 서버 보안

참고: NFS(Network File System) 서버에는 큐 관리자 데이터만 두어야 합니다. NFS에서는 마운트 명령과 함께 다음 세 가지 옵션을 사용하여 시스템에 보안을 설정하십시오.

noexec

이 옵션을 사용하면 NFS에서 2진 파일의 실행을 중지하여 원격 사용자가 시스템에서 원하지 않는 코드를 실행하지 못하게 합니다.

nosuid

이 옵션을 사용하면 set-user-identifier 및 set-group-identifier 비트 사용을 방지하여 원격 사용자가 상위 권한을 얻지 못하게 합니다.

nodev

이 옵션을 사용하면 문자 및 블록 특수 디바이스의 사용 또는 정의를 중지하여 원격 사용자가 chroot에서 벗어나지 않게 합니다.

공유 파일 시스템 작동 확인

amqmfscck를 실행하여 UNIX 시스템의 표준 파일 시스템에서 멀티 인스턴스 큐 관리자의 큐 관리자 데이터를 저장하는 데 필요한 요구사항을 충족하는지 여부를 확인합니다. IBM WebSphere MQ MQI client 샘플 프로그램 **amqsfhac**를 **amqmfscck**와 병렬로 실행하여 큐 관리자가 실패 중에 메시지 무결성을 유지하는지 증명합니다.

시작하기 전에

네트워크 스토리지가 있는 하나의 서버와 이 서버에 연결되고 WebSphere MQ가 설치된 두 개의 다른 서버가 필요합니다. 파일 시스템을 구성하려면 관리자(루트) 권한이 있어야 하며 **amqmfscck**를 실행하려면 WebSphere MQ 관리자여야 합니다.

이 태스크 정보

115 페이지의 『공유 파일 시스템에 대한 요구사항』에 다중 인스턴스 큐 관리자에 공유 파일 시스템 사용을 위한 파일 시스템 요구사항이 설명되어 있습니다. IBM WebSphere MQ 기술 노트 [WebSphere MQ 멀티 인스턴스 큐 관리자에 대한 테스트 및 지원 명령](#)에는 IBM이 이미 테스트한 공유 파일 시스템이 나열됩니다. 이 태스크의 프로시저는 나열되지 않은 파일 시스템이 데이터 무결성을 유지하는지 평가하는 데 도움이 되는 파일 시스템 테스트 방법을 설명합니다.

다중 인스턴스 큐 관리자의 장애 복구는 큐 관리자의 데이터 또는 로그 파일 쓰기를 차단하는 네트워킹 문제점을 포함하여, 하드웨어 또는 소프트웨어 장애를 통해 트리거될 수 있습니다. 사용자는 주로 파일 서버의 장애 요인에 관심이 있습니다. 하지만 잠금이 제대로 해제되는지 테스트하려면 IBM WebSphere MQ 서버의 실패도 야기해야 합니다. 공유 파일 시스템에서 확신을 가지려면 다음의 장애와 환경 고유의 다른 장애를 모두 테스트하십시오.

1. 디스크 동기화를 포함하여 파일 서버의 운영 체제 종료.
2. 디스크 동기화를 포함하지 않은 파일 서버의 운영 체제 정지.
3. 각 서버에서 재설정 단추 누르기.
4. 각 서버의 네트워크 케이블 분리.
5. 각 서버의 전원 케이블 분리.
6. 각 서버 끄기.

큐 관리자 데이터와 로그를 공유하기 위해 사용할 네트워크 스토리지에 디렉토리를 작성하십시오. 디렉토리 소유자는 WebSphere MQ 관리자이거나 UNIX의 mqm 그룹 멤버여야 합니다. 테스트를 실행하는 사용자는 WebSphere MQ 관리자 권한이 있어야 합니다.

Linux에서 멀티 인스턴스 큐 관리자 작성에 있는 파일 시스템 내보내기 및 마운트 예제를 사용하면 파일 시스템 구성에 도움이 됩니다. 여러 다른 파일 시스템에 각기 다른 구성 단계가 필요합니다. 파일 시스템 문서를 읽으십시오.

프로시저

각 검사에서 파일 시스템 검사가 실행되는 동안 이전 목록의 모든 장애를 유발하십시오. **amqsfhac**를 **amqmfscck**와 동시에 실행하려면 122 페이지의 『메시지 무결성을 테스트하기 위해 amqsfhac 실행』 태스크를 이 태스크와 병렬로 수행하십시오.

1. 두 개의 IBM WebSphere MQ 서버에 내보낸 디렉토리를 마운트하십시오.

파일 시스템 서버에서 공유 디렉토리 **shared** 및 다중 인스턴스 큐 관리자에 대한 데이터를 저장할 서브디렉토리 **qmdata**를 작성하십시오. Linux에서 다중 인스턴스 큐 관리자에 대한 공유 디렉토리를 설정하는 예는 [Linux에서 다중 인스턴스 큐 관리자 작성의 예제](#)를 참조하십시오.

2. 기본 파일 시스템 작동을 검사하십시오.

하나의 IBM WebSphere MQ 서버에서 매개변수 없이, 파일 시스템 검사기를 실행하십시오.

```
amqmfsc /shared/qmdata
```

그림 40. IBM WebSphere MQ 서버 1에서

3. 두 개의 IBM WebSphere MQ 서버 모두에서 동일한 디렉토리에 동시 쓰기를 검사하십시오.
두 개의 IBM WebSphere MQ 서버 모두에서 -c 옵션과 함께 파일 시스템 검사기를 동시에 실행하십시오.

```
amqmfsc -c /shared/qmdata
```

그림 41. IBM WebSphere MQ 서버 1에서

```
amqmfsc -c /shared/qmdata
```

그림 42. IBM WebSphere MQ 서버 2에서

4. 두 개의 IBM WebSphere MQ 서버 모두에서 잠금 대기 및 해제를 검사하십시오.
두 개의 IBM WebSphere MQ 서버 모두에서 -w 옵션과 함께 파일 시스템 검사기를 동시에 실행하십시오.

```
amqmfsc -w /shared/qmdata
```

그림 43. IBM WebSphere MQ 서버 1에서

```
amqmfsc -w /shared/qmdata
```

그림 44. IBM WebSphere MQ 서버 2에서

5. 데이터 무결성을 확인하십시오.

- a) 테스트 파일을 포맷하십시오.

테스트 중인 디렉토리에 큰 파일을 작성하십시오. 후속 단계가 정상적으로 완료될 수 있도록 파일이 포맷됩니다. 장애 복구를 시뮬레이트하기 위해 두 번째 단계를 인터럽트할 시간이 충분하도록 파일이 충분히 커야 합니다. 기본값 262144페이지(1GB)를 시도해 보십시오. 느린 파일 시스템에서는 포맷이 약 60초 안에 완료되도록 프로그램이 자동으로 이 기본값을 줄입니다.

```
amqmfsc -f /shared/qmdata
```

서버는 다음 메시지로 응답합니다.

```
Formatting test file for data integrity test.  
Test file formatted with 262144 pages of data.
```

그림 45. IBM WebSphere MQ 서버 1에서

- b) 장애를 유발시키는 동안 파일 시스템 검사기를 사용하여 테스트 파일에 데이터를 쓰십시오.

두 개 서버에서 동시에 테스트 프로그램을 실행하십시오. 실패가 발생할 서버에서 테스트 프로그램을 시작한 후 실패에서 생존할 서버에서 테스트 프로그램을 시작하십시오. 조사하고 있는 장애를 유발하십시오.

첫 번째 테스트 프로그램이 오류 메시지로 중지됩니다. 두 번째 테스트 프로그램은 테스트 파일에 대한 잠금을 확보하고 첫 번째 테스트 프로그램이 중단한 위치에서 시작하여 테스트 파일에 데이터를 씁니다. 두 번째 테스트 프로그램이 완료할 때까지 실행되게 하십시오.

표 12. 두 개의 서버에서 동시에 데이터 무결성 검사기 실행	
IBM WebSphere MQ 서버 1	IBM WebSphere MQ 서버 2
amqmfscck -a /shared/qmdata	
<pre>Please start this program on a second machine with the same parameters. File lock acquired. Start a second copy of this program with the same parameters on another server. Writing data into test file. To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power. To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power.</pre>	<pre>amqmfscck -a /shared/qmdata Waiting for lock... Waiting for lock... Waiting for lock... Waiting for lock... Waiting for lock... Waiting for lock...</pre>
Turn the power off here	
	<pre>File lock acquired. Reading test file Checking the integrity of the data read. Appending data into the test file after data already found. The test file is full of data. It is ready to be inspected for data integrity.</pre>

테스트 타이밍은 파일 시스템의 작동에 따라 다릅니다. 예를 들어, 첫 번째 프로그램이 가동 중단 후에 확보한 파일 잠금을 파일 시스템이 해제하려면 일반적으로 30 - 90초가 걸립니다. 첫 번째 테스트 프로그램이 파일을 채우기 전에 실패를 유발할 시간이 없으면 **amqmfscck**의 -x 옵션을 사용하여 테스트 파일을 삭제하십시오. 더 큰 테스트 파일로 처음부터 테스트를 시도하십시오.

c) 테스트 파일에서 데이터의 무결성을 확인하십시오.

```
amqmfscck -i /shared/qmdata
```

서버는 다음 메시지로 응답합니다.

```
File lock acquired
Reading test file checking the integrity of the data read.
The data read was consistent.
The tests on the directory completed successfully.
```

그림 46. IBM WebSphere MQ 서버 2에서

6. 테스트 파일을 삭제하십시오.

```
amqmfscck -x /shared/qmdata
Test files deleted.
```

그림 47. IBM WebSphere MQ 서버 2에서

서버는 다음 메시지로 응답합니다.

```
Test files deleted.
```

결과

이 프로그램은 테스트가 정상적으로 완료되면 엑시트 코드 0을 리턴하고 그렇지 않을 경우 0 이외의 값을 리턴합니다.

예:

세 가지 예의 첫 번째 세트는 최소 출력을 생성하는 명령을 보여줍니다.

한 서버의 기본 파일 잠금 테스트 성공

```
> amqmfscck /shared/qmdata
The tests on the directory completed successfully.
```

한 서버의 기본 파일 잠금 테스트 실패

```
> amqmfscck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck' error '2'.
```

두 개의 서버에서 잠금 테스트 성공

IBM WebSphere MQ 서버 1	IBM WebSphere MQ 서버 2
<pre>> amqmfscck -w /shared/qmdata Please start this program on a second machine with the same parameters. Lock acquired. Press Return or terminate the program to release the lock.</pre>	
	<pre>> amqmfscck -w /shared/qmdata Waiting for lock...</pre>
<pre>[Return pressed] Lock released.</pre>	

표 13. 두 개의 서버에서 잠금 성공 (계속)	
IBM WebSphere MQ 서버 1	IBM WebSphere MQ 서버 2
	Lock acquired. The tests on the directory completed successfully

세 가지 예의 두 번째 세트는 상세 모드를 사용하여 동일한 명령을 표시합니다.

한 서버의 기본 파일 잠금 테스트 성공

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")'
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd2, F_SETLK, F_RDLCK)
System call: close(fd2)
System call: write(fd1)
System call: close(fd1)
The tests on the directory completed successfully.
```

한 서버의 기본 파일 잠금 테스트 실패

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fdSameFile, F_SETLK, F_RDLCK)
System call: close(fdSameFile)
System call: write(fd)
AMQxxxx: Error calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck', errno 2
(Permission denied).
```

두 개의 서버에서 잠금 테스트 성공

표 14. 두 개의 서버에서 잠금 성공 - 상세 모드	
IBM WebSphere MQ 서버 1	IBM WebSphere MQ 서버 2
<pre>> amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/amqmfscck.lkw", O_EXCL O_CREAT O_RDWR, 0666)'' Calling 'fchmod(fd, 0666)'' Calling 'fstat(fd)'' Please start this program on a second machine with the same parameters. Calling 'fcntl(fd, F_SETLK, F_WRLCK)'' Lock acquired. Press Return or terminate the program to release the lock.</pre>	

표 14. 두 개의 서버에서 잠금 성공 - 상세 모드 (계속)	
IBM WebSphere MQ 서버 1	IBM WebSphere MQ 서버 2
	<pre>> amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL O_CREAT O_RDWR,0666)' Calling 'fd = open("/shared/qmdata/amqmfscck.lkw, O_RDWR, 0666)' Calling 'fcntl(fd, F_SETLK, F_WRLCK) 'Waiting for lock...</pre>
<pre>[Return pressed] Calling 'close(fd)' Lock released.</pre>	
	<pre>Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. The tests on the directory completed successfully</pre>

관련 참조

[amqmfscck\(파일 시스템 검사\)](#)

메시지 무결성을 테스트하기 위해 **amqsfhac** 실행

amqsfhac는 네트워크 스토리지를 사용하는 큐 관리자가 실패에 따른 데이터 무결성을 유지보수하는지 확인합니다.

시작하기 전에

이 테스트를 위해 네 개의 서버가 필요합니다. 다중 인스턴스 큐 관리자에 두 개의 서버, 파일 시스템에 하나, **amqsfhac**를 IBM WebSphere MQ MQI client 애플리케이션으로 실행하는 데 하나의 서버가 필요합니다.

프로시저의 단계 117 페이지의 『1』에 따라 다중 인스턴스 큐 관리자에 대한 파일 시스템을 설정하십시오.

이 태스크 정보

프로시저

1. 프로시저의 단계 117 페이지의 『1』에서 작성한 파일 시스템을 사용하여 또 다른 서버 QM1에 다중 인스턴스 큐 관리자를 작성하십시오.

다중 인스턴스 큐 관리자 작성을 참조하십시오.

2. 큐 관리자의 가용성을 높게 하여 두 서버 모두에서 큐 관리자를 시작하십시오.

서버 1에서:

```
strmqm -x QM1
```

서버 2에서:

```
strmqm -x QM1
```

3. **amqsfhac**를 실행할 클라이언트 연결을 설정하십시오.
 - a) [클라이언트 설치 확인](#)의 프로시저를 사용하여 클라이언트 연결을 설정하거나 [다시 연결할 수 있는 클라이언트 샘플](#)의 예제 스크립트를 설정하십시오.
 - b) 클라이언트 채널을 IP 주소가 두 개가 되도록 수정하십시오. 이 주소는 QM1을 실행하는 두 개의 서버에 해당합니다.

예 스크립트에서 다음을

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
```

대상:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
```

여기서, `server1` 및 `server2`는 두 개 서버의 호스트 이름이고 `2345`는 채널 리스너가 대기 중인 포트입니다. 일반적으로 기본값은 `1414`입니다. `1414`를 기본 리스너 구성에 사용할 수 있습니다.

4. 테스트를 위해 QM1에 두 개의 로컬 큐를 작성하십시오.
다음 MQSC 스크립트를 실행하십시오.

```
DEFINE QLOCAL(TARGETQ) REPLACE
DEFINE QLOCAL(SIDEQ) REPLACE
```

5. `amqsfhac`로 구성을 테스트하십시오.

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

6. 파일 시스템의 무결성을 테스트하는 동안 메시지 무결성을 테스트하십시오.

[프로시저의 단계 118 페이지의 『5』](#) 중에 `amqsfhac`를 실행하십시오.

```
amqsfhac QM1 TARGETQ SIDEQ 10 20 0
```

활성 큐 관리자 인스턴스를 중지하면 `amqsfhac`가 활성화된 후 다른 큐 관리자 인스턴스에 다시 연결합니다. 다음 테스트에서 실패를 되돌릴 수 있도록 중지된 큐 관리자 인스턴스를 다시 시작하십시오. 장애 복구가 발생하기에 충분한 시간 동안 테스트 프로그램이 실행하도록 환경에 대한 시험을 기반으로 반복 수를 늘려야 합니다.

결과

단계 [123 페이지의 『6』](#)의 `amqsfhac` 실행 예는 [124 페이지의 그림 48](#)에 표시됩니다. 테스트는 성공입니다.

테스트에서 문제가 감지된 경우 출력에 실패가 보고됩니다. 일부 테스트 실행 `MQRC_CALL_INTERRUPTED`에서 "Resolving to backed out"을 보고할 수 있습니다. 결과는 차이가 없습니다. 실패가 발생하기 이전 또는 이후에 네트워크 파일 스토리지에서 디스크 쓰기가 커밋되었는지 여부에 따라 결과가 다릅니다.

```

Sample AMQSFHAC start
qmname = QM1
qname = TARGETQ
sidename = SIDEQ
transize = 10
iterations = 20
verbose = 0
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Resolving MQRC_CALL_INTERRUPTED
MQGET browse side tranid=14 pSideinfo->tranid=14
Resolving to committed
Iteration 7
Iteration 8
Iteration 9
Iteration 10
Iteration 11
Iteration 12
Iteration 13
Iteration 14
Iteration 15
Iteration 16
Iteration 17
Iteration 18
Iteration 19
Sample AMQSFHAC end

```

그림 48. *amqsfhac* 실행 성공 후 결과

관련 참조

[고가용성 샘플 프로그램](#)

IBM WebSphere MQ 파일 공유

일부 IBM WebSphere MQ 파일은 활성 큐 관리자가 독점 액세스하고 다른 파일은 공유됩니다.

WebSphere MQ 파일은 프로그램 파일과 데이터 파일로 나뉩니다. 프로그램 파일은 일반적으로 WebSphere MQ를 실행하는 각 서버에 로컬로 설치됩니다. 큐 관리자는 기본 데이터 디렉토리의 데이터 파일과 디렉토리에 대한 액세스를 공유합니다. [124 페이지의 그림 49](#)에 표시된 각 *qmgrs* 및 *log* 디렉토리에 포함된 자체 큐 관리자 디렉토리 트리에 대한 독점 액세스를 요구합니다.

[124 페이지의 그림 49](#)은 WebSphere MQ 디렉토리 구조의 상위 레벨 보기입니다. 큐 관리자 간에 공유 가능하며 원격으로 할 수 있는 디렉토리를 표시합니다. 플랫폼별로 세부사항이 다릅니다. 점선은 구성 가능한 경로를 표시합니다.

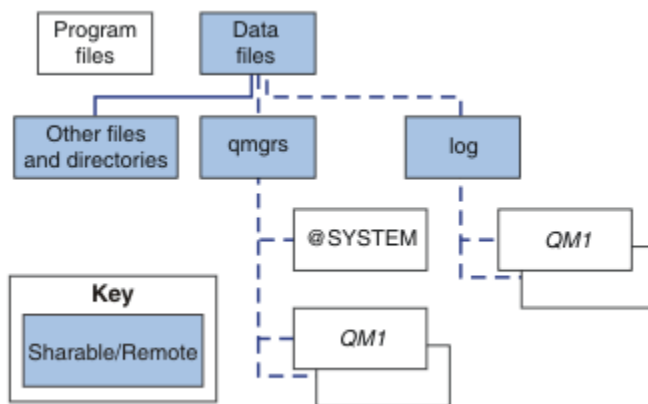


그림 49. WebSphere MQ 디렉토리 구조 전체 보기

프로그램 파일

프로그램 파일 디렉토리는 일반적으로 기본 위치이고 로컬이며 서버의 모든 큐 관리자에 공유됩니다.

데이터 파일

데이터 파일 디렉토리는 일반적으로 기본 위치, UNIX and Linux 시스템의 /var/mqm에서 로컬이며 Windows의 설치에서 구성 가능합니다. 큐 관리자 간에 공유됩니다. 기본 위치를 원격으로 만들 수 있지만 서로 다른 WebSphere MQ 설치 간에는 공유할 수 없습니다. WebSphere MQ 구성의 DefaultPrefix 속성이 이 경로를 가리킵니다.

qmgrs

v7.0.1부터는 큐 관리자 데이터의 위치를 지정할 두 가지 대체 방법이 있습니다.

Prefix 사용

Prefix 속성은 qmgrs 디렉토리의 위치를 지정합니다. WebSphere MQ는 큐 관리자 이름에서 큐 관리자 디렉토리 이름을 구성하며 이것을 qmgrs 디렉토리의 서브디렉토리로 작성합니다.

Prefix 속성은 QueueManager 스탠자에 위치하며 DefaultPrefix 속성의 값에서 상속됩니다. 기본적으로 간편한 관리를 위해 큐 관리자는 일반적으로 동일한 qmgrs 디렉토리를 공유합니다.

QueueManager 스탠자는 mqs.ini 파일에 있습니다.

큐 관리자의 qmgrs 디렉토리 위치를 변경할 경우 Prefix 속성의 값을 변경해야 합니다.

UNIX and Linux 플랫폼의 경우 [124 페이지의 그림 49](#)에서 QM1 디렉토리에 대한 Prefix 속성은 다음과 같습니다.

```
Prefix=/var/mqm
```

DataPath 사용

DataPath 속성은 큐 관리자 데이터 디렉토리의 위치를 지정합니다.

DataPath 속성은 큐 관리자 데이터 디렉토리의 이름을 포함하여 전체 경로를 지정합니다. DataPath 속성은 큐 관리자 데이터 디렉토리에 대한 불완전한 경로를 지정하는 Prefix 속성과 다릅니다.

DataPath 속성은 지정될 경우 QueueManager 스탠자에 위치합니다. 지정된 경우에는 Prefix 속성의 값보다 우선합니다.

QueueManager 스탠자는 mqs.ini 파일에 있습니다.

큐 관리자의 큐 관리자 데이터 디렉토리 위치를 변경할 경우 DataPath 속성의 값을 변경해야 합니다.

UNIX 또는 Linux 플랫폼의 경우 [124 페이지의 그림 49](#)에서 QM1 디렉토리에 대한 DataPath 속성은 다음과 같습니다.

```
DataPath=/var/mqm/qmgrs/QM1
```

log

로그 디렉토리는 큐 관리자 구성에서 Log 스탠자의 각 큐 관리자마다 별도로 지정됩니다. 큐 관리자 구성은 qm.ini에 있습니다.

DataPath/QmgrName/@IPCC 서브디렉토리

DataPath/QmgrName/@IPCC 서브디렉토리는 공유 디렉토리 경로에 있습니다. IPC 파일 시스템 오브젝트에 대한 디렉토리 경로를 구성하는 데 사용됩니다. 시스템에서 큐 관리자가 공유될 때 큐 관리자의 네임스페이스를 구별해야 합니다. V7.0.1 이전에는 큐 관리자가 한 시스템에만 사용되었습니다. 서브디렉토리 한 세트는 IPC 파일 시스템 오브젝트에 대한 디렉토리 경로를 정의하기에 충분했습니다([125 페이지의 그림 50](#) 참조).

```
DataPath/QmgrName/@IPCC/esem
```

그림 50. 예 IPC 서브디렉토리, V7.0.1 이전

V7.0.1 이후부터는 시스템에서 IPC 파일 시스템 오브젝트를 구별해야 합니다. 큐 관리자가 실행하는 각 시스템에 대한 서브디렉토리가 디렉토리 경로에 추가됩니다(126 페이지의 [그림 51](#) 참조).

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

그림 51. 예 IPC 서브디렉토리, V7.0.1 및 후속 릴리스

*myHostName*은 운영 체제에서 리턴하는 호스트 이름의 처음 20자로 결정됩니다. 일부 시스템에서는 자르기 전의 호스트 이름 길이가 최대 64자일 수 있습니다. *myHostName*의 생성된 값은 다음과 같은 두 가지 이유로 문제를 유발할 수 있습니다.

1. 처음 20자가 고유하지 않습니다.
2. 시스템에 항상 동일한 호스트 이름을 할당하지는 않는 DHCP 알고리즘을 통해 호스트 이름이 생성됩니다.

이러한 경우 환경 변수 MQC_IPC_HOST를 사용하여 *myHostName*을 설정하십시오. [126 페이지의 그림 52](#)를 보십시오.

```
export MQC_IPC_HOST=myHostName
```

그림 52. 예: MQC_IPC_HOST 설정

기타 파일 및 디렉토리

추적 파일을 포함한 디렉토리와 같은 기타 파일 및 디렉토리와 공통 오류 로그는 일반적으로 공유되며 로컬 파일 시스템에 보관됩니다.

V7.0.1까지 WebSphere MQ는 외부 관리에 의존하여 큐 관리자가 큐 관리자 데이터 및 로그 파일에 독점 액세스할 수 있도록 했습니다. V7.0.1 이후부터 공유 파일 시스템의 지원으로 WebSphere MQ가 파일 시스템 잠금을 사용하여 이들 파일에 대한 독점 액세스를 관리합니다. 파일 시스템 잠금을 사용하면 한 번에 특정 큐 관리자의 한 인스턴스만 활성화할 수 있습니다.

특정 큐 관리자의 첫 번째 인스턴스를 시작하면 이 큐 관리자가 큐 관리자 디렉토리의 소유권을 가져옵니다. 두 번째 인스턴스를 시작하는 경우 첫 번째 인스턴스가 중지되어 있으면 두 번째가 소유권을 가져갈 수 있습니다. 첫 번째 큐 관리자가 여전히 실행 중이면 두 번째 인스턴스는 시작에 실패하고 큐 관리자가 실행 중이라고 보고합니다. 첫 번째 큐 관리자가 중지된 경우에는 두 번째 큐 관리자가 큐 관리자 파일의 소유권한을 가져가고 실행 큐 관리자가 됩니다.

두 번째 큐 관리자가 첫 번째 큐 관리자로부터 인계받는 프로시저를 자동화할 수 있습니다. 다른 큐 관리자가 이 큐 관리자에서 인계하도록 허용하는 `strmqm -x` 옵션을 사용하여 첫 번째 큐 관리자를 시작하십시오. 그러면 두 번째 큐 관리자는 큐 관리자 파일의 소유권 인계를 시도하기 전에 큐 관리자 파일이 잠금 해제될 때까지 대기한 후에 시작합니다.

UNIX and Linux 시스템의 디렉토리 구조

UNIX and Linux 시스템의 WebSphere MQ 디렉토리 구조는 좀 더 쉬운 관리와 더 나은 성능 및 신뢰도를 위해 다른 파일 시스템으로 맵핑될 수 있습니다.

WebSphere MQ의 유연한 디렉토리 구조를 사용하여 멀티 인스턴스 큐 관리자를 실행하기 위한 공유 파일 시스템의 장점을 이용하십시오.

`crtmqm QM1` 명령을 사용하여 [127 페이지의 그림 53](#)에 표시된 디렉토리 구조를 작성하십시오. 여기서 R는 제품의 릴리스입니다. 이는 WebSphere MQ 시스템 v7.0.1 이상에서 작성된 큐 관리자의 일반적 디렉토리 구조입니다. 일부 디렉토리, 파일, .ini 속성 설정은 명확한 표시를 위해 생략되며 다른 큐 관리자 이름은 맵핑을 통해 변경될 수 있습니다. 파일 시스템의 이름은 시스템에 따라 다릅니다.

일반 설치에서는 작성한 모든 큐 관리자가 로컬 파일 시스템의 공용 log 및 qmgrs 디렉토리를 가리킵니다. 멀티 인스턴스 구성에서 log 및 qmgrs 디렉토리는 WebSphere MQ의 다른 설치와 공유하는 네트워크 파일 시스템에 있습니다.

[127 페이지의 그림 53](#) shows the default configuration for WebSphere MQ v7.R on AIX where R is the release of the product. 대체 멀티 인스턴스 구성에 대한 예로는 [131 페이지의 『UNIX and Linux 시스템의 디렉토리 구성 예제』](#)의 내용을 참조하십시오.

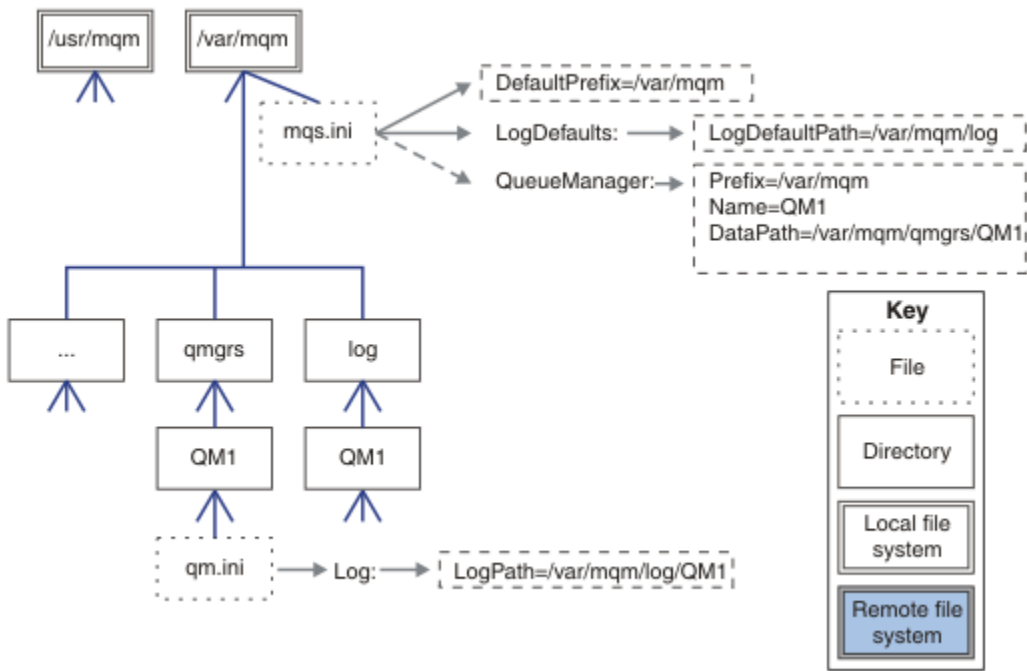


그림 53. 예제 기본 WebSphere MQ v7.R UNIX and Linux 시스템의 디렉토리 구조

이 제품은 AIX 상의 /usr/mqm와 다른 시스템 상의 /opt/mqm에 기본으로 설치됩니다. 작업 디렉토리는 /var/mqm 디렉토리에 설치됩니다.

참고: IBM WebSphere MQ를 설치하기 전에 /var/mqm 파일 시스템을 작성한 경우, mqm 사용자에게 전체 디렉토리 권한이 있는지 확인하십시오 (예: 파일 모드 755).

log 및 qmgrs 디렉토리는 mqs.ini 파일에서 LogDefaultPath 및 DefaultPrefix 속성의 기본값에 정의된 기본 위치에 표시됩니다. 큐 관리자가 작성되면 기본적으로 큐 관리자 데이터 디렉토리가 DefaultPrefix/qmgrs에 작성되고 로그 파일 디렉토리는 LogDefaultPath /log에 작성됩니다. LogDefaultPath 및 DefaultPrefix는 기본적으로 큐 관리자 및 로그 파일이 작성되는 위치에만 영향을 줍니다. 큐 관리자 디렉토리의 실제 위치는 mqs.ini 파일에 저장되고 로그 파일 디렉토리의 위치는 qm.ini 파일에 저장됩니다.

큐 관리자의 로그 파일 디렉토리는 qm.ini 파일의 LogPath 속성에 정의됩니다. **crtmqm** 명령에 -ld 옵션을 사용하여 큐 관리자에 대한 LogPath 속성을 설정하십시오 (예: **crtmqm -ld LogPath QM1**). ld 매개변수를 생략하면 LogDefaultPath 값이 대신 사용됩니다.

큐 관리자 데이터 디렉토리는 mqs.ini 파일에 있는 QueueManager 스탠자의 DataPath 속성에 정의됩니다. **crtmqm** 명령에서 -md 옵션을 사용하여 큐 관리자에 대한 DataPath를 설정하십시오 (예: **crtmqm -md DataPath QM1**). md 매개변수를 생략하면 대신 DefaultPrefix 또는 접두부 속성 값이 사용됩니다. 접두부는 DefaultPrefix보다 우선합니다.

보통은, 로그와 데이터 디렉토리를 모두 단일 명령으로 지정하여 QM1을 작성하십시오.

```
crtmqm
-md DataPath -ld
LogPath QM1
```

큐 관리자가 중지될 때 qm.ini 파일의 DataPath 및 LogPath 속성을 편집하여 기존 큐 관리자의 큐 관리자 로그 및 데이터 디렉토리 위치를 수정할 수 있습니다.

errors 디렉토리에 대한 경로는 /var/mqm에 있는 다른 모든 디렉토리에 대한 경로와 마찬가지로 수정 불가능합니다. 하지만 이 디렉토리를 다른 파일 시스템에 마운트하거나 다른 디렉토리에 기호 링크할 수 있습니다.

UNIX and Linux 시스템의 디렉토리 콘텐츠

큐 관리자와 연관된 디렉토리의 콘텐츠

제품 파일 위치에 대한 자세한 정보는 [설치 위치 선택](#)을 참조하십시오.

대체 디렉토리 구성에 대한 정보는 [115 페이지의 『파일 시스템 지원 계획』](#)의 내용을 참조하십시오.

[129 페이지의 그림 54](#)에서 레이아웃은 큐 관리자를 일정 시간 동안 사용한 후의 WebSphere MQ를 나타냅니다. 실제 구조는 큐 관리자에 발생한 조작에 따라 다릅니다.

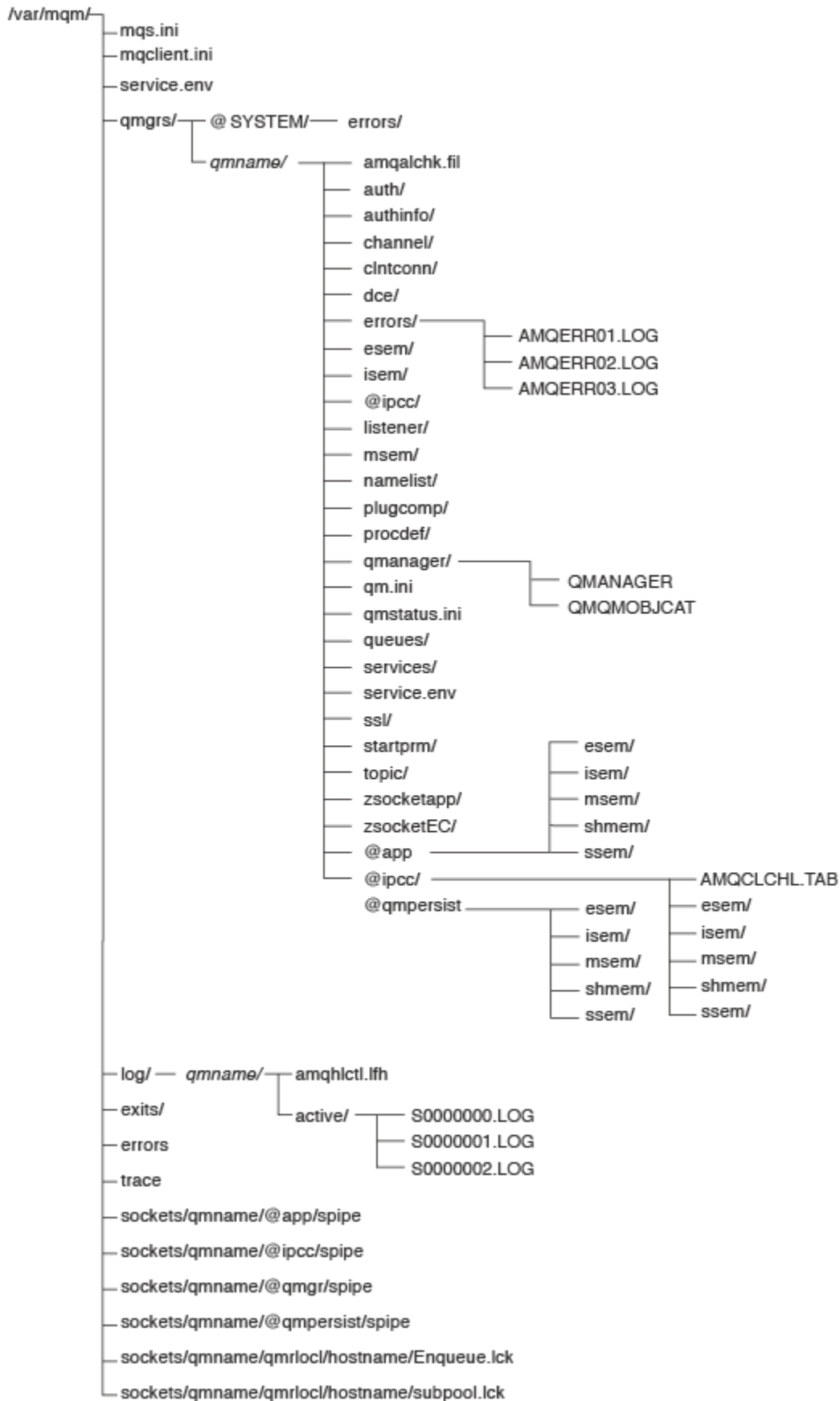


그림 54. 큐 관리자가 시작된 후 기본 디렉토리 구조(UNIX 시스템)

/var/mqm/

/var/mqm 디렉토리에는 개별 큐 관리자가 아닌 전체로 WebSphere MQ 설치에 적용되는 구성 파일 및 출력 디렉토리가 포함되어 있습니다.

표 15. 시스템 /var/mqm 디렉토리의 문서화된 콘텐츠	
<u>mqs.ini</u>	WebSphere MQ 설치 범위 구성 파일. 큐 관리자가 시작될 때 읽습니다. AMQ_MQS_INI_LOCATION 환경 변수를 사용하여 수정 가능한 파일 경로. strmqm 명령이 실행되는 셸에서 설정되고 내보내는지 확인하십시오.
<u>mqclient.ini</u>	WebSphere MQ MQI 클라이언트 프로그램이 읽는 기본 클라이언트 구성 파일. MQCLNTCF 환경 변수를 사용하여 수정 가능한 파일 경로.
<u>service.env</u>	서비스 프로세스에 대한 시스템 범위 환경 변수를 포함합니다. 수정된 파일 경로.
<u>errors/</u>	시스템 범위 오류 로그 및 FFST 파일. 수정된 디렉토리 경로. 파스트: IBM WebSphere MQ for UNIX and Linux systems도 참조하십시오.
<u>sockets/</u>	시스템 용도로만 각 큐 관리자에 대한 정보를 포함합니다.
<u>trace/</u>	추적 파일. 수정된 디렉토리 경로.
<u>exits/</u>	사용자 채널 엑시트 프로그램을 포함한 기본 디렉토리. mqs.ini 파일의 ApiExit 스탠자에서 수정 가능한 위치.
<u>exits64/</u>	

/var/mqm/qmgrs/qmname/

/var/mqm/qmgrs/qmname/에는 큐 관리자에 대한 디렉토리 및 파일이 포함되어 있습니다. 활성 큐 관리자 인스턴스의 독점 액세스를 위해 디렉토리가 잠깁니다. 디렉토리 경로는 mqs.ini 파일에서 직접 수정 가능하거나 **crtmqm** 명령의 **md** 옵션을 사용하여 수정할 수 있습니다.

표 16. 시스템 /var/mqm/qmgrs/qmname 디렉토리의 문서화된 콘텐츠	
<u>qm.ini</u>	큐 관리자가 시작될 때 읽는 큐 관리자 구성 파일.
<u>errors/</u>	큐 관리자 범위 오류 로그. <i>qmname</i> = @system에는 알 수 없거나 사용 불가능한 큐 관리자의 채널 관련 메시지가 포함됩니다.
<u>@ipcc/AMQCLCHL.TAB</u>	기본 클라이언트 채널 제어 파일. WebSphere MQ 서버가 작성하며 WebSphere MQ MQI 클라이언트 프로그램이 읽습니다. MQCHLLIB 및 MQCHLTAB 환경 변수를 사용하여 수정 가능한 파일 경로.
<u>qmanager</u>	큐 관리자 오브젝트 파일: QMANAGER 오브젝트 관리자 오브젝트 카탈로그: QMQMOBJCAT

표 16. 시스템 <code>/var/mqm/qmgrs/qmname</code> 디렉토리의 문서화된 콘텐츠 (계속)	
authinfo/	큐 관리자 내에 정의된 각 오브젝트는 이 디렉토리의 파일과 연관됩니다. 파일 이름은 정의 이름과 거의 일치합니다. WebSphere MQ 파일 이름 이해 를 참조하십시오.
채널 /	
clntconn/	
리스너 /	
이름 목록	
procdef/	
큐 /	
services/	
topics/	
...	WebSphere MQ에서 사용되는 기타 디렉토리(예: @ipcc)는 WebSphere MQ만 수정합니다.

`/var/mqm/log/qmname/`

`/var/mqm/log/qmname/`에는 큐 관리자 로그 파일이 포함되어 있습니다. 활성 큐 관리자 인스턴스의 독점 액세스를 위해 디렉토리가 잠깁니다. 디렉토리 경로는 `qm.ini` 파일에서 수정 가능하거나 `crtmqm` 명령의 `ld` 옵션을 사용하여 수정할 수 있습니다.

표 17. 시스템 <code>/var/mqm/log/qmname</code> 디렉토리의 문서화된 콘텐츠	
amqhlctl.lfh	로그 제어 파일.
활성 /	이 디렉토리는 S0000000.LOG, S0000001.LOG, S0000002.LOG 등으로 번호가 지정된 로그 파일을 포함합니다.

UNIX and Linux 시스템의 디렉토리 구성 예제

UNIX and Linux 시스템의 대체 파일 시스템 구성 예.

여러 가지 목표를 달성하기 위해 WebSphere MQ 디렉토리 구조를 다양한 방법으로 사용자 정의할 수 있습니다.

- 다중 인스턴스 큐 관리자를 구성하려면 `qmgrs` 및 `log` 디렉토리를 원격 공유 파일 시스템에 두십시오.
- 입출력 경합을 줄임으로써 성능을 향상시키려면 데이터 및 로그 디렉토리에 분리된 파일 시스템을 사용하고 디렉토리를 서로 다른 디스크에 할당하십시오.
- 성능에 큰 영향을 미치는 디렉토리에 대해 보다 빠른 스토리지 디바이스를 사용하십시오. 종종 디바이스가 로컬 또는 원격으로 마운트되었는지 여부보다 물리적 디바이스 대기 시간이 지속 메시징 성능에 있어서 보다 중요한 요인입니다. 다음 목록은 성능에 미치는 영향이 최소 및 최대인 디렉토리를 보여줍니다.
 1. `log`
 2. `qmgrs`
 3. `/usr/mqm`을 포함하는 기타 디렉토리
- 중복 디스크 어레이처럼 탄력적인 스토리지에 할당된 파일 시스템에 `qmgrs` 및 `log` 디렉토리를 작성하십시오.
- 공통 오류 로그는 네트워크 파일 시스템 보다는 로컬로 `var/mqm/errors`에 저장하여 네트워크 파일 시스템에 관련된 오류가 로그될 수 있도록 하는 것이 더 좋습니다.

132 페이지의 그림 55은 대체 WebSphere MQ 디렉토리 구조가 도출되는 템플릿입니다. 템플릿에서 점선은 구성 가능한 경로를 나타냅니다. 예에서 점선은 `AMQ_MQS_INI_LOCATION` 환경 변수와 `mqs.ini` 및 `qm.ini` 파일에 저장된 구성 정보에 해당하는 실선으로 대체됩니다.

참고: 경로 정보는 `mqs.ini` 또는 `qm.ini` 파일에서처럼 표시됩니다. `crtmqm` 명령에 경로 매개변수를 제공한 경우 큐 관리자 디렉토리 이름을 생략하십시오. 큐 관리자 이름은 망글링(mangling)된 후 WebSphere MQ에 의해 경로에 추가됩니다.

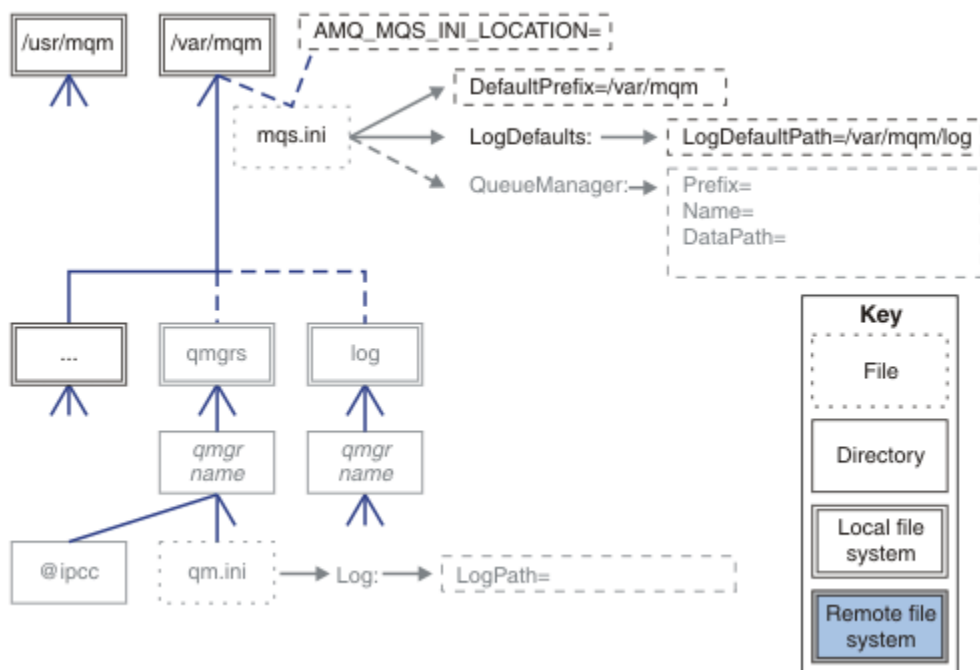


그림 55. 디렉토리 구조 패턴 템플릿

구성된 디렉토리 구조의 예는 다음과 같습니다. 첫 번째 예는 `crtmqm QM1` 명령을 실행하여 작성한 WebSphere MQ V7.0.1의 일반적인 기본 디렉토리 구조를 보여줍니다. 두 번째 예는 WebSphere MQ V7.0.1 이전 릴리스를 사용하여 작성한 큐 관리자에 대해 일반 디렉토리 구조가 어떻게 표시되는지를 보여줍니다. 디렉토리 구조는 변경되지 않습니다.

버전 7.0.1에서 새로 작성된 큐 관리자는 V7 이전 릴리스와 다른 구성 파일을 가집니다. v7.0.0.2로 되돌리기 위해 v7.0.1 수정팩을 제거해야 하는 경우 구성 파일을 재작성할 필요가 있습니다. Prefix 속성을 사용하여 새 큐 관리자 데이터 디렉토리의 경로를 정의하기만 하면 되거나 큐 관리자 데이터 디렉토리 및 로그 디렉토리를 다른 위치로 이동시켜야 할 수 있습니다. 큐 관리자를 재구성하는 가장 안전한 방법은 큐 관리자 데이터 및 로그 디렉토리를 저장하고 큐 관리자를 재작성한 후 새 위치의 데이터 및 로그 디렉토리를 저장된 디렉토리로 바꾸는 것입니다.

v7.0.1 이상 릴리스의 일반 디렉토리 구조

133 페이지의 그림 56는 `crtmqm QM1` 명령을 발행하여 v7.0.1에 작성된 기본 디렉토리 구조입니다.

The `mqs.ini` 파일에는 `DefaultPrefix`의 값을 참조하여 작성한 QM1 큐 관리자에 대한 스탠자가 있습니다. `qm.ini` 파일에 있는 `Log` 스탠자에는 `LogPath`의 값이 있으며 `mqs.ini`의 `LogDefaultPath`에 의거하여 설정됩니다.

선택적 `crtmqm` 매개변수를 사용하여 `DataPath` 및 `LogPath`의 기본값을 대체하십시오.

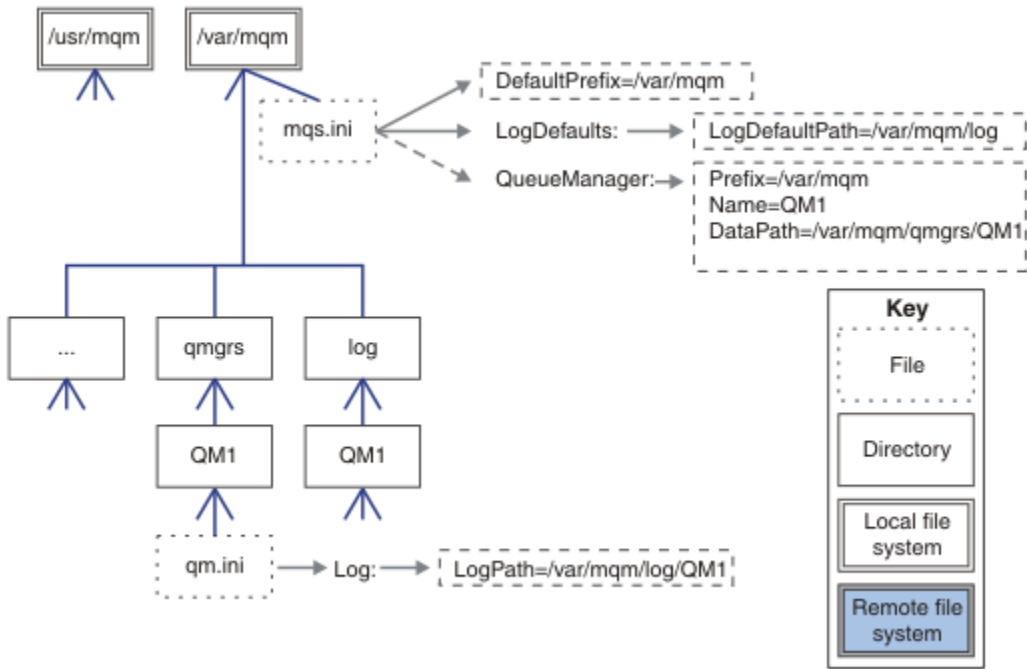


그림 56. 예제 기본 WebSphere MQ v7.R UNIX and Linux 시스템의 디렉토리 구조

v7.0.1 이전 릴리스의 일반 디렉토리 구조

WebSphere MQ V7.0.1 이전에는 DataPath 속성이 존재하지 않았으므로 `mqs.ini` 파일에는 이 속성이 없습니다. `qmgrs` 디렉토리의 위치가 Prefix 속성을 사용하여 구성되었습니다. 개별 디렉토리의 위치는 상이한 파일 시스템 위치를 가리키는 기호 링크를 사용하여 구성할 수 있습니다.

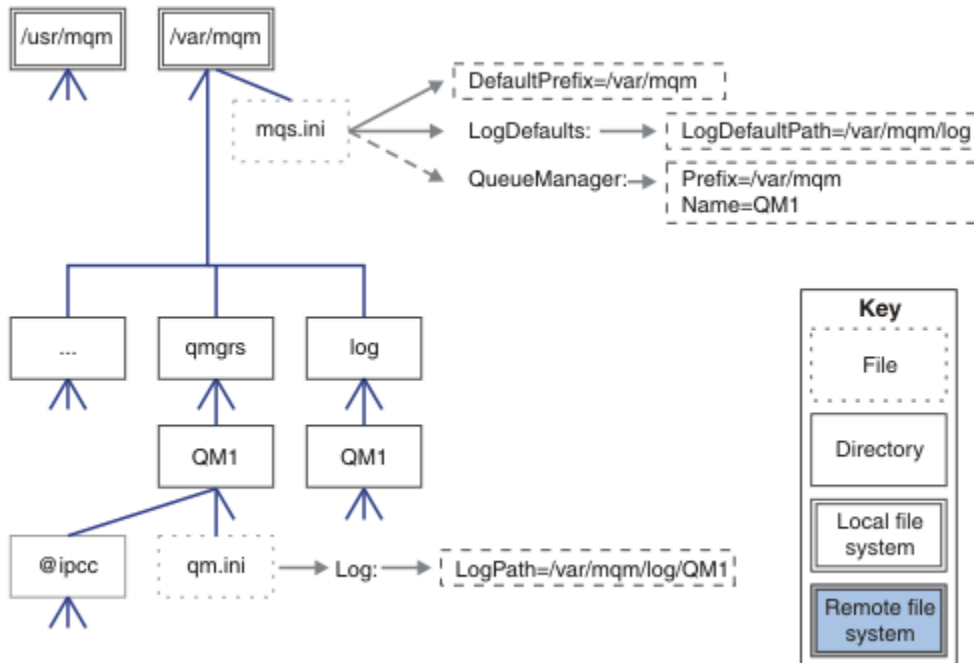


그림 57. v7.0.1 이전 릴리스의 일반 디렉토리 구조

기본 qmgrs 및 log 디렉토리 공유 (릴리스 v7.0.1 이상)

135 페이지의 『모두 공유(v7.0.1 릴리스부터)』의 대안은 `qmgrs` 및 `log` 디렉토리를 별도로 공유하는 것입니다 (134 페이지의 그림 58). 이 구성에서는 기본 `mqs.ini` 이 로컬 `/var/mqm` 파일 시스템에 저장되므로

AMQ_MQS_INI_LOCATION 를 설정할 필요가 없습니다. mqclient.ini 및 mqserver.ini 와 같은 파일과 디렉토리로 공유되지 않습니다.

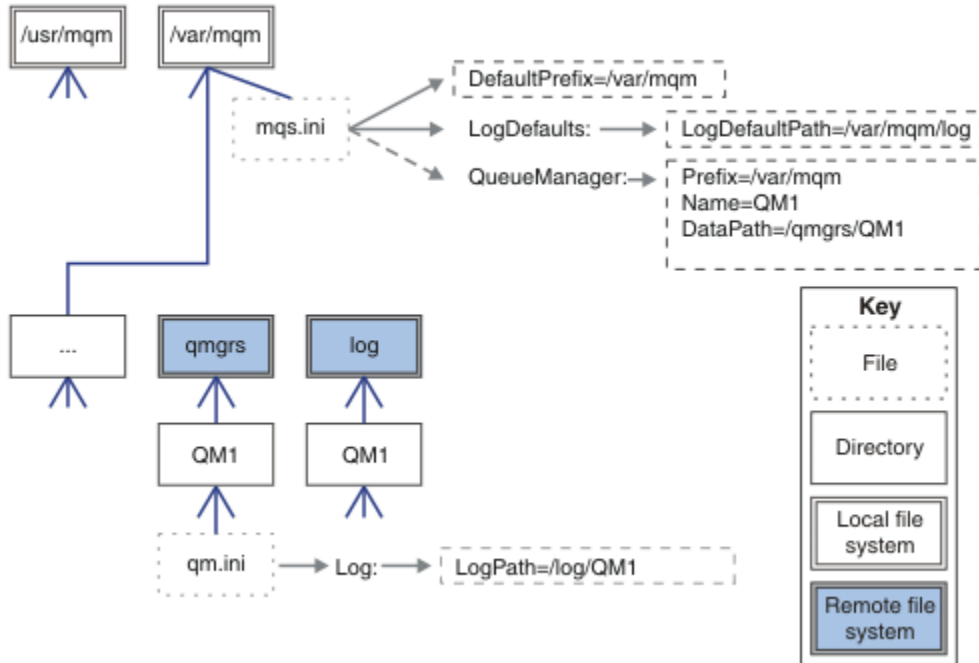


그림 58. qmgrs 및 log 디렉토리

이름 지정된 qmgrs 및 log 디렉토리 (릴리스 v7.0.1 이상)

135 페이지의 그림 59의 구성은 log 및 qmgrs를 /ha라는 공통 이름 지정된 원격 공유 파일 시스템에 배치합니다. 동일한 실제 구성을 두 가지 다른 방식으로 작성할 수 있습니다.

1. LogDefaultPath=/ha 를 설정한 후 **crtmqm -md /ha/qmgrs QM1** 명령을 실행하십시오. 결과는 [135 페이지의 그림 59](#)의 설명과 정확히 같습니다.
2. 기본 경로를 변경하지 않은 채로 두고 명령을 실행하십시오. **crtmqm -ld /ha/log - md /ha/qmgrs QM1**.

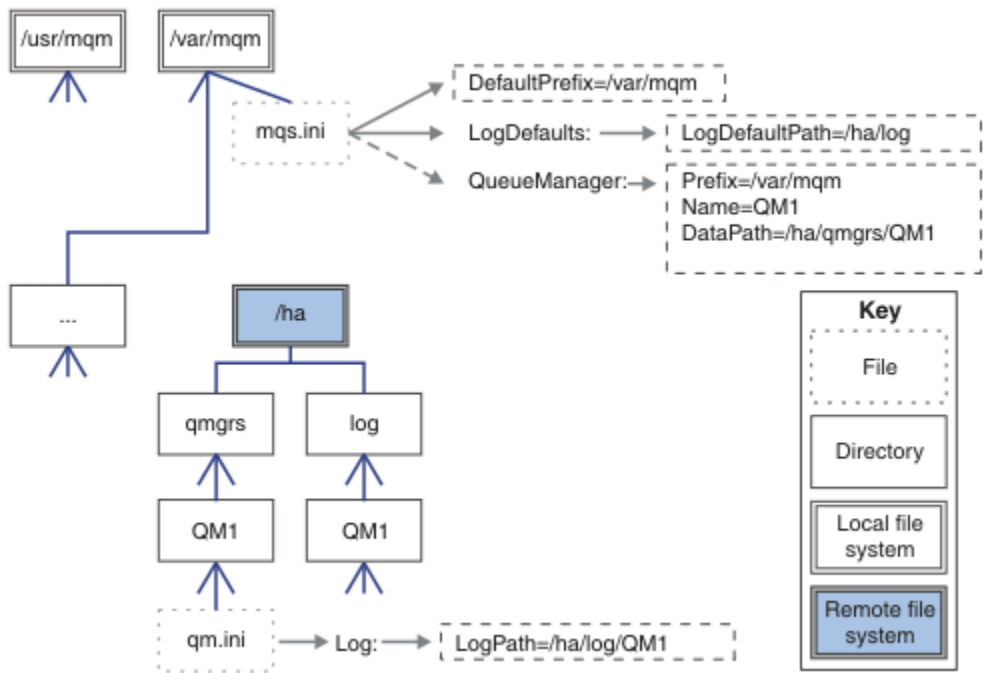


그림 59. 이름 지정된 *qmgrs* 및 *log* 디렉토리 공유

모두 공유(v7.0.1 릴리스부터)

136 페이지의 그림 60는 빠른 네트워크 파일 스토리지가 있는 시스템의 단순 구성입니다.

`/var/mqm`을 원격 공유 파일 시스템으로 마운트하십시오. 기본적으로 QM1을 시작하면 `/var/mqm`을 검색하여 공유 파일 시스템에서 이를 찾습니다. 그리고 `/var/mqm`에 있는 `mqs.ini` 파일을 읽습니다. 모든 서버의 큐 관리자에 대해 단일 `/var/mqm/mqs.ini` 파일을 사용하는 대신, 각 서버에서 `AMQ_MQS_INI_LOCATION` 환경 변수를 설정하여 서로 다른 `mqs.ini` 파일을 지시할 수 있습니다.

참고: `/var/mqm/errors/`의 일반 오류 파일 콘텐츠는 여러 서버의 큐 관리자 간에 공유됩니다.

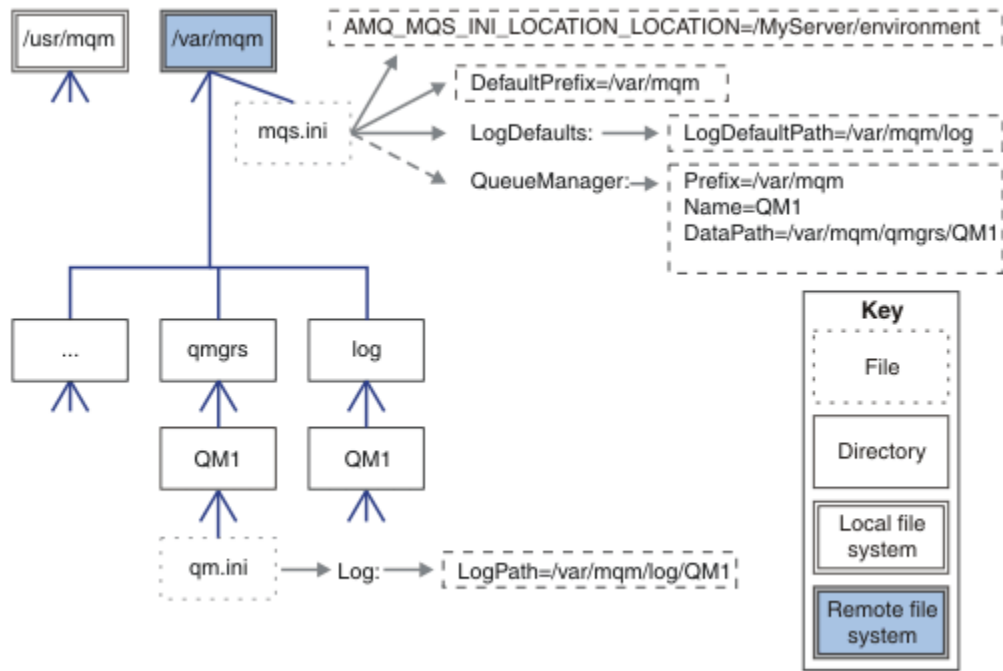


그림 60. 모두 공유

다중 인스턴스 큐 관리자에는 이를 사용할 수 없음을 유의하십시오. 다중 인스턴스 큐 관리자의 각 호스트는 세마 포어 및 공유 메모리와 같은 로컬 데이터를 추적하기 위해 `/var/mqm`의 자체 로컬 사본을 보유해야 하기 때문입니다. 이 엔티티는 호스트에서 공유할 수 없습니다.

Windows 시스템의 디렉토리 구조

Windows의 큐 관리자 구성 정보 및 디렉토리를 찾는 방법을 설명합니다.

Windows 설치를 위한 IBM WebSphere MQ의 기본 디렉토리는 다음과 같습니다.

32비트

`C:\Program Files\IBM\WebSphere MQ`

64비트

`C:\Program Files (x86)\IBM\WebSphere MQ`

설치 정보는 Windows 레지스트리에 저장됩니다. IBM WebSphere MQ 정보가 저장된 레지스트리 키는 다음과 같습니다.

32비트

`My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\`

64비트

`My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\IBM\WebSphere MQ\`

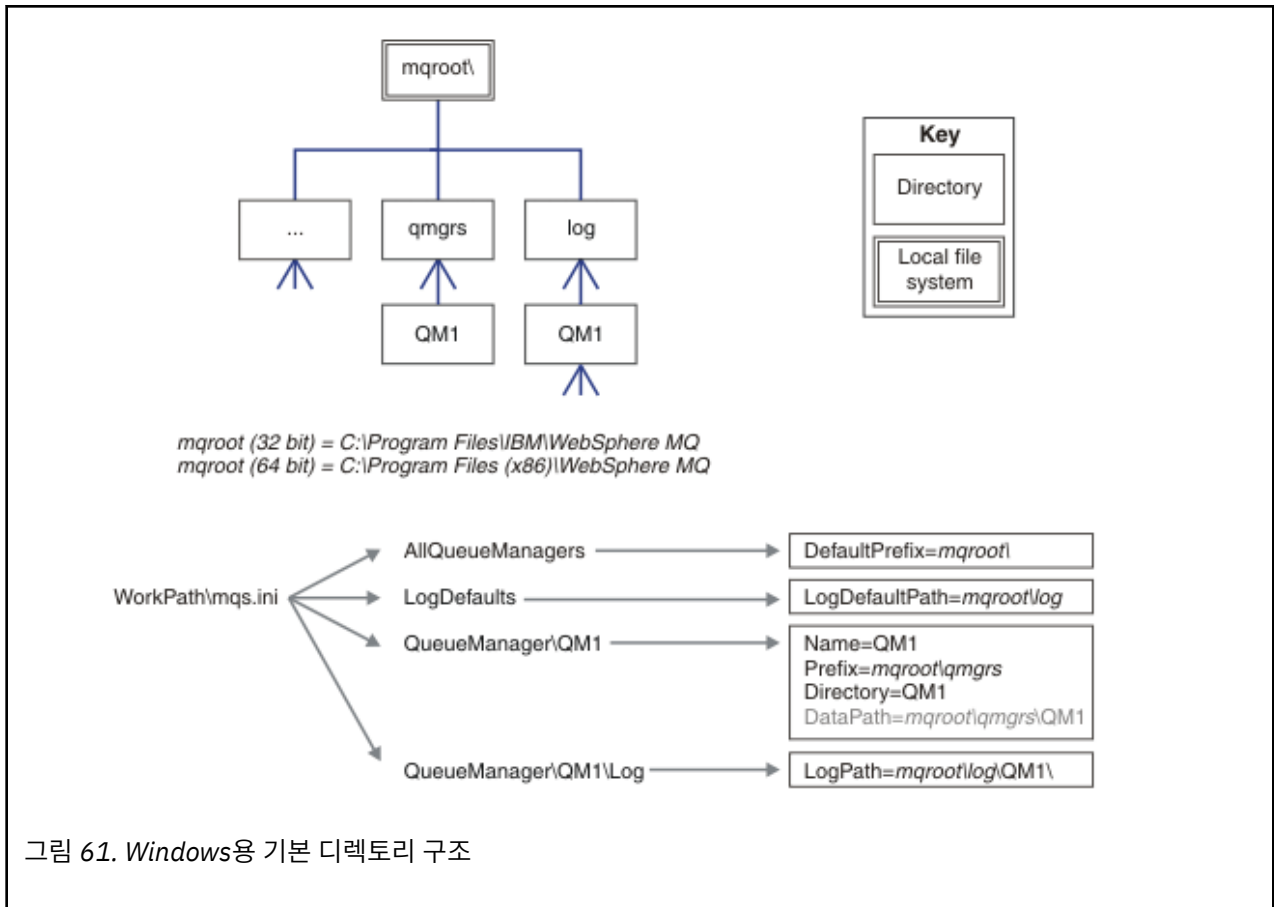
각 설치에는 특정 하위 키를 갖습니다.

`Installation\<InstallationName>\`

IBM WebSphere MQ 데이터 디렉토리를 가리키는 경로는 `WorkPath`로 이름 지정된 문자열 값에 저장되고 로그에 대한 기본 디렉토리는 `LogDefaultPath`에 저장됩니다. 큐 관리자 데이터 디렉토리는 `WorkPath\qmgrs\Qmgrname`에 작성됩니다. 큐 관리자 로그는 `LogDefaultPath\QmgrName`에 작성됩니다. 137 페이지의 그림 61를 참조하십시오.

IBM WebSphere MQ를 설치할 때 큐 관리자 데이터 및 로그 디렉토리를 정의하는 경우 `WorkPath` 및 `LogDefaultPath`가 사용자 정의된 경로 정보로 업데이트됩니다.

`WorkPath` 및 `LogDefaultPath`는 큐 관리자를 작성하는 경우에만 사용됩니다.



다중 인스턴스 큐 관리자

다중 인스턴스 큐 관리자를 구성하려면 로그 및 데이터 디렉토리가 큐 관리자의 인스턴스를 실행 중인 서버와 다른 서버의 네트워크 스토리지에 있어야 합니다.

큐 관리자 데이터와 로그 디렉토리의 위치를 쉽게 지정할 수 있도록 **crtmqm** 명령에 두 개의 매개변수, **-md** 및 **-ld**가 제공됩니다. **-md** 매개변수를 지정하는 효과는 네 배입니다.

1. mqs.ini 스탠자 QueueManager\QmgrName에 새로운 변수 DataPath가 포함되며, 이는 큐 관리자 데이터 디렉토리를 가리킵니다. Prefix 변수와 다르게, 큐 관리자 디렉토리의 이름이 경로에 포함됩니다.
2. mqs.ini 파일에 저장된 큐 관리자 구성 정보는 Name, Prefix, Directory, DataPath로 감소합니다.

디렉토리 콘텐츠

WebSphere MQ 디렉토리의 위치 및 콘텐츠를 나열합니다.

WebSphere MQ 구성에는 세 개의 기본 세트의 파일과 디렉토리가 있습니다.

1. 실행 파일 및 readme 파일, WebSphere MQ Explorer 플러그인 및 도움말 파일 및 라이선스 파일과 같은 기타 읽기 전용 파일. 이들 파일은 유지보수가 적용될 때만 업데이트됩니다. 이 파일은 [138 페이지의 표 18](#)에 설명되어 있습니다.
2. 부분 큐 관리자에 고유하지 않은 잠재적으로 수정 가능한 파일 및 디렉토리. 이 파일과 디렉토리는 [138 페이지의 표 19](#)에 설명되어 있습니다.
3. 서버의 각 큐 관리자에 고유한 파일 및 디렉토리. 이러한 파일 및 디렉토리는 [138 페이지의 표 18](#)에 설명되어 있습니다.

자원 디렉토리 및 파일

자원 디렉토리와 파일은 큐 관리자를 실행하기 위한 모든 실행 코드와 자원을 포함합니다. 설치 특정 IBM WebSphere MQ 구성 레지스트리 키에 있는 FilePath변수는 자원 디렉토리에 대한 경로를 포함합니다.

표 18. <i>FilePath</i> 디렉토리에 있는 디렉토리 및 파일	
파일 경로	컨텐츠
<i>FilePath</i> \bin	명령 및 DLL
<i>FilePath</i> \bin64	명령 및 DLL(64비트)
<i>FilePath</i> \conv	데이터 변환 테이블
<i>FilePath</i> \doc	마법사 도움말 파일
<i>FilePath</i> \MQExplorer	탐색기 및 탐색기 도움말 Eclipse 플러그인
<i>FilePath</i> \gskit8	글로벌 보안 키
<i>FilePath</i> \java	Java 자원(JRE 포함)
<i>FilePath</i> \licenses	라이선스 정보
<i>FilePath</i> \Non_IBM_License	라이선스 정보
<i>FilePath</i> \properties	내부적으로 사용됨
<i>FilePath</i> \Tivoli	
<i>FilePath</i> \tools	개발 자원 및 샘플
<i>FilePath</i> \Uninst	내부적으로 사용됨
<i>FilePath</i> \README.TXT	Readme 파일

큐 관리자에 고유하지 않은 디렉토리

일부 디렉토리는 추적 파일 및 오류 로그와 같이 특정 큐 관리자에 고유하지 않은 파일을 포함합니다. *DefaultPrefix* 변수는 이러한 디렉토리의 경로를 포함합니다. *DefaultPrefix*는 *AllQueueManagers* 스탠자의 일부입니다.

표 19. <i>DefaultPrefix</i> 디렉토리에 있는 디렉토리 및 파일	
파일 경로	컨텐츠
<i>DefaultPrefix</i> \Config	내부적으로 사용됨
<i>DefaultPrefix</i> \conv	ccsid.tbl 데이터 변환 제어 파일(데이터 변환에 설명됨)
<i>DefaultPrefix</i> \errors	큐 관리자에 해당되지 않은 오류 로그 AMQERRnn.LOG
<i>DefaultPrefix</i> \exits	채널 엑시트 프로그램
<i>DefaultPrefix</i> \exits64	채널 엑시트 프로그램(64비트)
<i>DefaultPrefix</i> \ipc	사용 안함
<i>DefaultPrefix</i> \Qmgrs	139 페이지의 표 20에 설명됨
<i>DefaultPrefix</i> \trace	추적 파일
<i>DefaultPrefix</i> \amqmjpse.txt	내부적으로 사용됨

큐 관리자 디렉토리

큐 관리자를 작성하면 큐 관리자에 고유한 디렉토리의 새로운 세트가 작성됩니다.

-md filepath 매개변수를 사용하여 큐 관리자를 작성한 경우 경로는 *mqs.ini* 파일의 큐 매니저 스탠자에 있는 *DataPath* 변수에 저장됩니다. **-md filepath** 매개변수를 설정하지 않고 큐 관리자를 작성하면 큐 관리자 디렉

토리가 *DefaultPrefix*에 저장된 경로에 작성되고 *mqs.ini* 파일의 큐 관리자 스탠자 내 *Prefix* 변수에 경로가 복사됩니다.

표 20. <i>DataPath</i> 및 <i>Prefix/Qmgrs/QmgrName</i> 디렉토리에 있는 디렉토리 및 파일	
파일 경로	컨텐츠
<i>DataPath</i> \@ipcc	클라이언트 연결 테이블, AMQCLCHL.TAB의 기본 위치
<i>DataPath</i> \authinfo	내부적으로 사용됨
<i>DataPath</i> \channel	
<i>DataPath</i> \clntconn	
<i>DataPath</i> \errors	오류 로그 AMQERRnn.LOG
<i>DataPath</i> \listener	내부적으로 사용됨
<i>DataPath</i> \namelist	
<i>DataPath</i> \plugcomp	
<i>DataPath</i> \procdef	
<i>DataPath</i> \qmanager	
<i>DataPath</i> \queues	
<i>DataPath</i> \services	
<i>DataPath</i> \ssl	
<i>DataPath</i> \startprm	
<i>DataPath</i> \topic	
<i>DataPath</i> \active	
<i>DataPath</i> \active.dat	
<i>DataPath</i> \amqalchk.fil	
<i>DataPath</i> \master	
<i>DataPath</i> \master.dat	
<i>DataPath</i> \qm.ini	큐 관리자 구성
<i>DataPath</i> \qmstatus.ini	큐 관리자 상태
<i>Prefix</i> \Qmgrs\QmgrName	내부적으로 사용됨
<i>Prefix</i> \Qmgrs\@SYSTEM	사용 안함
<i>Prefix</i> \Qmgrs\@SYSTEM\errors	

IBM WebSphere MQ 및 UNIX System V IPC 자원

큐 관리자는 일부 IPC 자원을 사용합니다. **ipcs -a**를 사용하면 사용 중인 자원을 찾을 수 있습니다.

이 정보는 **UNIX and Linux** 시스템에서만 실행 중인 **IBM WebSphere MQ**에 적용됩니다.

IBM WebSphere MQ는 System V 프로세스 간 통신(IPC) 자원(*semaphores* 및 공유 메모리 세그먼트)을 사용하여 시스템 컴포넌트 사이에서 데이터를 저장하고 전달합니다. 이러한 자원은 큐 관리자에 연결하는 애플리케이션 및 큐 관리자 프로세스에 사용됩니다. IBM WebSphere MQ MQI 클라이언트는 IBM WebSphere MQ 추적 제어를 제외한 IPC 자원을 사용하지 않습니다. UNIX 명령 **ipcs -a**를 사용하여 시스템에서 현재 사용 중인 IPC 자원의 수 및 크기에 대한 상세한 정보를 가져오십시오.

AIX의 공유 메모리

특정 애플리케이션 유형이 AIX 메모리 제한 때문에 연결에 실패한 경우 대부분 환경 변수 EXTSHM=ON을 설정하여 해결할 수 있습니다.

AIX에서 일부 32비트 프로세스가 WebSphere MQ 큐 관리자에 연결할 수 있는 기능에 영향을 주는 운영 체제 한계에 부딪힐 수 있습니다. WebSphere MQ에 대한 모든 표준 연결은 공유 메모리를 사용하지만 다른 UNIX and Linux 플랫폼과는 다르게 AIX에서는 32비트 프로세스가 11개의 공유 메모리 세트만 접속할 수 있습니다.

대부분의 32비트 프로세스는 이 한계에 부딪치지 않지만 메모리 요구사항이 높은 애플리케이션은 WebSphere MQ에 연결할 때 이유 코드 2102: MQRC_RESOURCE_PROBLEM으로 실패할 수 있습니다. 다음 애플리케이션 유형이 이 오류를 표시할 수 있습니다.

- 32비트 JVM 에서 실행되는 프로그램
- 대용량 또는 초대용량 메모리 모델을 사용하는 프로그램
- 많은 큐 관리자 또는 데이터베이스에 연결하는 프로그램
- 자체적으로 공유 메모리 세트를 첨부한 프로그램

AIX는 32비트 프로세스가 더 많은 공유 메모리에 접속할 수 있도록 확장 공유 메모리 기능을 제공합니다. 이 기능과 함께 애플리케이션을 실행하려면 큐 관리자 및 프로그램을 시작하기 전에 환경 변수 EXTSHM=ON을 내보내십시오. EXTSHM=ON 기능은 대부분의 경우에 이 오류를 차단하지만 shmctl 함수의 SHM_SIZE 옵션을 사용하는 프로그램에는 호환되지 않습니다.

WebSphere MQ MQI 클라이언트 애플리케이션 및 모든 64비트 프로세스는 이 제한의 영향을 받지 않습니다. EXTSHM 설정 여부에 관계없이 WebSphere MQ 큐 관리자에 연결할 수 있습니다.

WebSphere MQ 및 UNIX 프로세스 우선순위

프로세스 우선순위 *nice* 값을 설정할 때의 우수 사례입니다.

이 정보는 UNIX and Linux 시스템에서 실행 중인 WebSphere MQ 에만 적용됩니다.

백그라운드에서 프로세스를 실행하는 경우 셸 호출을 통해 해당 프로세스에 보다 높은 *nice* 값(및 그에 따른 낮은 우선순위)이 제공될 수 있습니다. 여기에는 일반 WebSphere MQ 성능 분석이 있을 수 있습니다. 스트레스가 심한 상황에서, 높은 우선순위의 여러 실행 준비된 스레드가 있고 낮은 우선순위의 일부 스레드가 있으면 운영 체제 스케줄링 특성이 낮은 우선순위 스레드의 프로세서 시간을 빼앗을 수 있습니다.

큐 관리자와 연관된 독립적으로 시작된 프로세스(예: runmqtsr)가 연관된 큐 관리자와 같은 *nice* 값을 가지는 것은 좋은 사례입니다. 셸이 이러한 백그라운드 프로세스에 높은 *nice* 값을 지정하지 않는지 확인하십시오. 예를 들어 ksh에서는 "set +o bgnice" 설정을 사용하여 ksh가 백그라운드 프로세스의 *nice* 값을 올리지 못하도록 하십시오. "ps -efl" 목록의 NI 열을 검토하여 실행 프로세스의 *nice* 값을 확인할 수 있습니다.

또한 큐 관리자와 동일한 *nice* 값을 사용하여 WebSphere MQ 응용프로그램 프로세스를 시작하십시오. 상이한 *nice* 값으로 실행할 경우 애플리케이션 스레드가 큐 관리자 스레드를 차단하여(또는 그 반대로) 성능이 저하될 수 있습니다.

HP Integrity NonStop Server 에서 IBM WebSphere MQ 클라이언트 환경 계획

IBM WebSphere MQ 환경을 계획할 때는 HP Integrity NonStop Server 환경 및 HP NonStop TMF를 고려해야 합니다. 정보를 사용하여 HP Integrity NonStop Server 용 IBM WebSphere MQ 클라이언트가 실행되는 환경을 계획하십시오.

HP Integrity NonStop Server 아키텍처의 IBM WebSphere MQ 클라이언트를 계획하기 전에 HP Integrity NonStop Server 의 기본 IBM WebSphere MQ 클라이언트에 익숙해지십시오. [HP Integrity NonStop Server 용 IBM WebSphere MQ 클라이언트 기술 개요](#)의 주제를 참조하십시오.

HP Integrity NonStop Server 환경 준비

설치하기 전에, 설치를 즉시 확인해야 하는지 여부에 따라 환경을 준비해야 합니다.

설치하려면 다음 항목이 필요합니다.

- 요구사항을 충족하는 사용자 ID. 사용자 ID 요구사항에 대한 자세한 내용은 [HP Integrity NonStop Server에서 사용자 및 그룹 설정을 참조하십시오.](#)
- 설치 파일에 사용할 수 있는 OSS 및 Guardian 파일 시스템의 확인된 위치.
- 운영 OSS 셸 및 OSS 파일 시스템. 다음 태스크를 수행하여 파일 시스템을 확인할 수 있습니다.
 - OSS 환경(셸)에 로그인. 사용하고자 하는 OSS 설치 루트 디렉토리에 대한 쓰기 액세스 권한이 있는지 확인하십시오.
 - MQM 그룹의 사용자 ID를 사용하여 TACL 환경에 로그인. 사용할 볼륨이 요구사항을 충족하며 액세스 가능 여부 및 하위 볼륨의 부재 여부를 확인하십시오.

알리어스를 사용하거나(있는 경우) 전체 프린시펄을 사용하여 OSS 또는 TACL 모두에 로그인할 수 있습니다.

설치를 사용할 수 있는지에 대한 확인을 즉시 진행하려면 다음의 선택적 항목도 필요합니다.

- OSS 환경의 운영 및 액세스 가능한 로컬 소켓 서브시스템.
- 운영 TCP/IP 서브시스템.

TMF 통합 글로벌 작업 단위를 사용하려면 다음 항목이 필요합니다.

- 운영 TMF 서브시스템.
- 운영 Pathway(TS/MP) 서브시스템.

이 중요 서브시스템의 상태에 대해 조금이라도 의심스러운 점이 있으면 시스템 관리자에게 지원을 요청하십시오.

IBM WebSphere MQ 및 HP NonStop TMF

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트는 HP NonStop NonStop TMF (Transaction Management Facility) 의 조정된 작업 단위에 참여할 수 있습니다. HP NonStop TMF에서 통합 트랜잭션은 큐 관리자가 IBM WebSphere MQ Version 7.1 이상인 경우에만 지원됩니다.

IBM WebSphere MQ 제공 TMF/게이트웨이는 리모트 큐 관리자와 통신하기 위해 TMF 통합의 트랜잭션을 XA(eXtended Architecture) 트랜잭션 통합으로 변환합니다. IBM WebSphere MQ 제공 TMF/게이트웨이는 HP NonStop TMF에서 제공하는 서비스를 사용하는 TMF 및 큐 관리자 트랜잭션 간의 브릿지이며, Pathway 환경에서 실행되도록 설계되었습니다.

HP NonStop TMF 소프트웨어는 수요가 많은 환경에서 트랜잭션 보호와 데이터베이스 일관성을 제공합니다. HP NonStop TMF에 대한 자세한 정보는 [HP NonStop TMF 소개](#)를 참조하십시오.

IBM WebSphere MQ에서 TMF/게이트웨이 구성 방법에 대한 정보는 [HP Integrity NonStop Server 구성](#)을 참조하십시오.

HP NonStop TMF 사용

HP NonStop TMF(Transaction Management Facility)는 HP Integrity NonStop Server의 고유 트랜잭션 관리자이며, 파일 시스템 및 관계형 데이터베이스 관리자, SQL/MP, SQL/MX와 통합됩니다.

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트는 TMF를 사용하여 글로벌 작업 단위를 조정할 수 있습니다.

글로벌 작업 단위를 통합하기 위해 TMF는 트랜잭션 관리자의 역할을 수행하며, 애플리케이션은 TMF가 제공하는 API를 사용하여 글로벌 작업 단위를 시작, 커밋하고 백아웃해야 합니다. 애플리케이션은 BEGINTRANSACTION을 호출하여 글로벌 작업 단위를 시작하며, 동기점 제어 내에서 MQPUT, MQPUT1, MQGET 호출을 실행하여 글로벌 작업 단위 내의 IBM WebSphere MQ 자원을 업데이트합니다. 그런 다음 애플리케이션은 ENDTRANSACTION을 호출하여 글로벌 작업 단위를 커밋하거나 ABORTTRANSACTION을 호출하여 글로벌 작업 단위를 백아웃할 수 있습니다.

TMF 트랜잭션을 사용 중인 애플리케이션은 한 시점에 하나의 트랜잭션에서만 활성으로 작동할 수 있지만 RESUMETRANSACTION을 사용하면 애플리케이션이 이전의 활성 트랜잭션을 완료하거나 중지하지 않고도 TMF가 아닌 트랜잭션과 연관되거나 하나의 활성 트랜잭션에서 다른 활성 트랜잭션으로 전환할 수 있습니다. MQPUT, MQPUT1 또는 MQGET에 대한 호출은 현재의 활성 TMF 트랜잭션에서 작성되거나(있는 경우) 로컬 작업 단위에서 작성됩니다(없는 경우). 따라서 이 호출이 올바른 작업 단위 내에서 작성되는지 확인할 수 있도록 애플리케이션 내에서 주의가 필요합니다.

글로벌 작업 단위 내에서는 IBM WebSphere MQ 자원의 업데이트는 물론 애플리케이션이 Enscribe 파일, SQL/MP 데이터베이스 또는 SQL/MX 데이터베이스를 업데이트할 수 있습니다.

글로벌 작업 단위 사용

글로벌 작업 단위는 TMF 트랜잭션으로 구현됩니다. 애플리케이션은 BEGINTRANSACTION을 호출하여 글로벌 작업 단위를 시작하고, ENDTRANSACTION을 호출하여 작업 단위를 커밋하거나 ABORTTRANSACTION을 호출하여 작업 단위를 백아웃합니다. 애플리케이션은 다른 TMF API 호출도 사용할 수 있습니다.

애플리케이션은 다른 애플리케이션에서 TMF 트랜잭션을 상속할 수 있습니다. 예를 들어, 애플리케이션(첫 번째 애플리케이션)은 추가적인 처리를 위해 두 번째 애플리케이션으로 트랜잭션을 전달하고 응답하기 전에 트랜잭션 내에서 작업을 수행할 수 있습니다. 따라서 첫 번째 및 두 번째 애플리케이션은 모두 IBM WebSphere MQ 큐에 대한 업데이트 및 파일과 데이터베이스에 대한 업데이트가 포함된 동일한 글로벌 작업 단위에 참여할 수 있습니다. 애플리케이션 간에 TMF 트랜잭션을 전달하는 기능은 여러 IBM WebSphere MQ 애플리케이션이 동일한 글로벌 작업 단위 내에서 메시징 조작을 수행할 수 있음을 의미합니다.

애플리케이션은 여러 활성 TMF 트랜잭션을 동시에 관리하고 제어할 수 있습니다. 트랜잭션은 애플리케이션 자체에 의해 시작, 다른 애플리케이션에서 상속 또는 둘 다일 수 있습니다. 즉, 애플리케이션은 동시에 여러 글로벌 작업 단위에 참여할 수 있습니다.

프로세스당 최대 동시 활성 TMF 트랜잭션 수는 1000이며 이는 아키텍처의 한계입니다. 애플리케이션이 여러 TMF 트랜잭션을 관리하는 경우, 한 시점에 하나의 트랜잭션만 현재 트랜잭션일 수 있습니다. 또는, 현재 사용되는 트랜잭션이 없을 수도 있습니다. 애플리케이션은 TMF API 호출(예: RESUMETRANSACTION, ACTIVATERECEIVETRANSID 및 TMF_SET_TX_ID)을 사용하여 현재인 상태를 한 트랜잭션에서 다른 트랜잭션으로 이동하거나 현재 트랜잭션이 없도록 지정할 수 있습니다. 애플리케이션은 이 제어 레벨을 사용하여 메시징 조작을 로컬 작업 단위나 글로벌 작업 단위 내에서 수행하는지 또는 동기점 제어 외부에서 수행하는지를 판별합니다.

- 현재인 TMF 트랜잭션이 없을 때 애플리케이션이 동기점 제어 내에서 MQPUT, MQPUT1 또는 MQGET을 호출하는 경우, IBM WebSphere MQ는 로컬 작업 단위 내에서 이 호출을 처리합니다.
- 애플리케이션에 현재 TMF 트랜잭션이 있을 때 애플리케이션이 동기점 제어 내에서 MQPUT, MQPUT1 또는 MQGET를 호출하는 경우, IBM WebSphere MQ는 현재 TMF 트랜잭션에 의해 구현되는 글로벌 작업 단위 내에서 이 호출을 처리합니다.
- 애플리케이션이 동기점 제어 외부에서 MQPUT, MQPUT1 또는 MQGET를 호출하는 경우, IBM WebSphere MQ는 호출 시점에 애플리케이션에 현재 TMF 트랜잭션이 있는지 여부와 무관하게 동기점 제어 외부에서 이 호출을 처리합니다.

처리 중에 소프트웨어나 하드웨어 장애가 발생하는 경우와 IBM WebSphere MQ 또는 운영 체제가 데이터 무결성 유지를 위해 트랜잭션의 백아웃을 결정하는 경우가 아니면, IBM WebSphere MQ는 MQI 호출 중에 애플리케이션의 TMF 트랜잭션 상태를 절대 변경하지 않습니다. 모든 MQI 호출은 애플리케이션에 제어를 리턴하기 직전에 애플리케이션의 트랜잭션 상태를 복원합니다.

장기 실행 트랜잭션 방지

수십 초가 넘도록 TMF 트랜잭션이 활성 상태를 유지하는 애플리케이션은 설계하지 마십시오. 트랜잭션을 오래 실행하면 TMF의 순환 감사 추적이 만기될 수 있습니다. TMF가 중요한 시스템 전체 자원이므로, TMF는 너무 오랜 동안 활성화된 애플리케이션 트랜잭션을 백아웃하여 스스로를 보호합니다.

애플리케이션 내의 처리가 큐에서 메시지를 가져옴으로써 구동되며 애플리케이션이 큐에서 메시지를 가져오고 작업 단위 내에서 메시지를 처리한다고 가정하십시오. 일반적으로 애플리케이션은 동기점 제어 내에서 대기 옵션으로 MQGET을 호출하여 큐에서 메시지를 가져옵니다.

애플리케이션이 글로벌 작업 단위를 대신 사용하는 경우, MQGET 호출의 지정된 대기 간격은 장기 실행 트랜잭션이 생기지 않도록 짧아야 합니다. 이는 메시지를 검색하기 전에 애플리케이션이 MQGET 호출을 두 번 이상 실행해야 함을 의미합니다.

주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

150-945
서울특별시 영등포구
국제금융로 10, 3IFC
한국 아이.비.엠 주식회사
U.S.A.

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

지적 재산권 라이선스 부여
2-31 Roppongi 3-chome, Minato-Ku
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 명시적 또는 묵시적인 일체의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

서울특별시 영등포구
서울특별시 강남구 도곡동 467-12,
군인공제회관빌딩
한국 아이.비.엠 주식회사
U.S.A.

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정

통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다.

이 정보를 소프트웨어로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

프로그래밍 인터페이스 정보

프로그래밍 인터페이스 정보는 본 프로그램과 함께 사용하기 위한 응용프로그램 소프트웨어 작성을 돕기 위해 제공됩니다.

This book contains information on intended programming interfaces that allow the customer to write programs to obtain the services of IBM WebSphere MQ.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용프로그램 소프트웨어의 디버깅을 돕기 위해 제공된 것입니다.

중요사항: 이 진단, 수정 및 튜닝 정보는 변경될 수 있으므로 프로그래밍 인터페이스로 사용하지 마십시오.

상표

IBM, IBM 로고, [ibm.com](http://www.ibm.com)®는 전세계 여러 국가에 등록된 IBM Corporation의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보"(www.ibm.com/legal/copytrade.shtml)에 있습니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다.

Microsoft 및 Windows는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

이 제품에는 Eclipse 프로젝트 (<http://www.eclipse.org/>)에서 개발한 소프트웨어가 포함되어 있습니다.

Java 및 모든 Java 기반 상표와 로고는 Oracle 및/또는 그 계열사의 상표 또는 등록상표입니다.



부품 번호:

(1P) P/N: