

7.5

IBM WebSphere MQ 구성

IBM

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [411 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

구성	5
UNIX, Linux 및 Windows에서 다중 설치 구성	5
다중 설치 환경에서 애플리케이션 연결	5
기본 설치 변경	14
큐 관리자를 설치와 연관	15
시스템에서 IBM WebSphere MQ 설치 찾기	16
큐 관리자 작성 및 관리	17
기본 큐 관리자 작성	20
기존 큐 관리자를 기본값으로 만들기	21
큐 관리자 작성 후 구성 파일 백업	22
큐 관리자 시작	22
큐 관리자 중지	23
큐 관리자 재시작	24
큐 관리자 삭제	24
분산 큐잉을 사용하여 애플리케이션 연결	25
IBM WebSphere MQ 분산 메시징 기술	26
분산 큐 관리 소개	43
UNIX, Linux 및 Windows에서 채널 모니터링 및 제어	66
서버와 클라이언트 간 연결 구성	89
사용할 통신 유형 결정	91
확장된 트랜잭션 클라이언트 구성	93
MQI 채널 정의	101
다른 플랫폼에서 서버 연결 및 클라이언트 연결 정의 작성	102
서버에서 서버 연결 및 클라이언트 연결 정의 작성	105
채널 엑시트	110
큐 공유 그룹에 클라이언트 연결	113
구성 파일을 사용하여 클라이언트 구성	114
환경 변수를 사용한 클라이언트 구성	130
큐된 발행/구독 제어	137
큐된 발행/구독 메시지 속성 설정	137
큐된 발행/구독 시작	139
큐된 발행/구독 중지	139
스트림 추가	140
스트림 삭제	141
구독 지점 추가	141
계층에 큐 관리자 연결	142
계층에서 큐 관리자의 연결 끊기	143
큐 관리자 클러스터 구성	144
액세스 제어 및 다중 클러스터 전송 큐	145
분산 큐잉과 비교	146
클러스터의 컴포넌트	148
전체 저장소를 보유할 클러스터 큐 관리자 선택 방법	160
클러스터 조직화	161
클러스터 이름 지정 규칙	162
중첩 클러스터	163
클러스터링 팁	164
클러스터에서 통신 설정	165
저장소 정보 보유	167
IBM WebSphere MQ 클러스터 관리	167
메시지를 클러스터로(부터) 라우팅	227
워크로드 관리에 클러스터 사용	240
클러스터링: 우수 사례	254

가용성, 복구 및 재시작.....	281
자동 클라이언트 다시 연결.....	282
콘솔 메시지 모니터링.....	287
고가용성 구성.....	287
메시지가 없어지지 않도록 확인(로깅).....	363
IBM WebSphere MQ 큐 관리자 데이터 백업 및 복원.....	376
구성 정보 변경.....	380
UNIX, Linux 및 Windows 시스템에서 구성 정보 변경.....	381
IBM WebSphere MQ 구성 정보를 변경하기 위한 속성.....	386
큐 관리자 구성 정보 변경.....	392
HP Integrity NonStop Server 구성.....	407
게이트웨이 프로세스 개요.....	408
Pathway에서 실행되도록 게이트웨이 구성.....	408
클라이언트 초기화 파일 구성.....	409
채널에 권한 부여.....	410
주의사항.....	411
프로그래밍 인터페이스 정보.....	412
상표.....	412

구성

하나 이상의 컴퓨터에서 하나 이상의 큐 관리자를 작성하고 개발, 테스트, 프로덕션 시스템에서 해당 큐 관리자가 비즈니스 데이터를 포함하는 메시지를 처리하도록 구성합니다.

IBM WebSphere MQ를 구성하기 전에 [IBM WebSphere MQ 기술 개요](#)에서 IBM WebSphere MQ 개념을 읽으십시오. [계획](#)에서 IBM WebSphere MQ 환경을 계획하는 방법에 대해 읽어보십시오.

IBM WebSphere MQ에서 큐 관리자와 관련 자원을 작성, 구성, 관리하기 위해 다양한 방법을 사용할 수 있습니다. 이러한 방법에는 명령행 인터페이스, 그래픽 사용자 인터페이스, 관리 API가 포함됩니다. 이 인터페이스에 대한 자세한 정보는 [IBM WebSphere MQ 관리](#)를 참조하십시오.

큐 관리자 작성, 시작, 중지, 삭제 방법에 대한 지시사항은 [17 페이지](#)의 『[큐 관리자 작성 및 관리](#)』의 내용을 참조하십시오.

IBM WebSphere MQ 설치 및 애플리케이션을 함께 연결하는 데 필요한 컴포넌트를 작성하는 방법에 대한 정보는 [25 페이지](#)의 『[분산 큐잉을 사용하여 애플리케이션 연결](#)』의 내용을 참조하십시오.

다른 방법을 사용하여 IBM WebSphere MQ 서버에 클라이언트를 연결하는 방법에 대한 지시사항은 [89 페이지](#)의 『[클라이언트와 서버 간의 연결 구성](#)』의 내용을 참조하십시오.

큐 관리자 클러스터를 구성하는 방법에 대한 지시사항은 [144 페이지](#)의 『[큐 관리자 클러스터 구성](#)』의 내용을 참조하십시오.

구성 정보를 변경하여 IBM WebSphere MQ 또는 큐 관리자의 동작을 변경할 수 있습니다. 자세한 정보는 [380 페이지](#)의 『[IBM WebSphere MQ 및 큐 관리자 구성 정보 변경](#)』의 내용을 참조하십시오. 이 제품 문서에 명시된 경우를 제외하고는 일반적으로 구성 변경사항을 적용하기 위해 큐 관리자를 재시작하지 않아도 됩니다.

관련 개념

[WebSphere MQ 기술 개요](#)

관련 태스크

[로컬 WebSphere MQ 오브젝트 관리](#)

[원격 WebSphere MQ 오브젝트 관리](#)

[계획 중](#)

UNIX, Linux, and Windows 에서 다중 설치 구성

동일한 시스템에서 다중 설치를 사용하는 경우 설치 및 큐 관리자를 구성해야 합니다.

이 정보는 UNIX, Linux®, and Windows에 적용됩니다.

다음 링크의 정보를 이용하여 설치를 구성하십시오.

- [14 페이지](#)의 『[기본 설치 변경](#)』
- [15 페이지](#)의 『[큐 관리자를 설치와 연관](#)』
- [5 페이지](#)의 『[다중 설치 환경에서 애플리케이션 연결](#)』

관련 개념

[다중 설치](#)

관련 태스크

[기본 설치 선택](#)

[설치 이름 선택](#)

다중 설치 환경에서 애플리케이션 연결

UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ Version 7.1 이상의 라이브러리가 로드되면 추가 조치를 수행할 필요없이 IBM WebSphere MQ가 자동으로 적절한 라이브러리를 사용합니다. IBM WebSphere MQ 애플리케이션이 연결되는 큐 관리자와 연관된 설치의 라이브러리를 사용합니다.

다음 개념은 애플리케이션이 IBM WebSphere MQ에 연결되는 방법을 설명하는 데 사용됩니다.

링크

애플리케이션이 컴파일될 때 IBM WebSphere MQ 라이브러리에 애플리케이션이 링크되어 애플리케이션이 실행될 때 로드되는 기능 내보내기를 가져옵니다.

로드 중

애플리케이션이 실행되면 IBM WebSphere MQ 라이브러리를 찾아 로드됩니다. 라이브러리를 찾는 데 사용되는 특정 메커니즘은 운영 체제 및 애플리케이션 빌드 방법에 따라 다릅니다. 다중 설치 환경에서 라이브러리를 찾아 로드하는 방법에 대한 자세한 정보는 8 페이지의 『IBM WebSphere MQ Version 7.1 이상의 라이브러리 로드』를 참조하십시오.

연결 중

예를 들어 MQCONN 또는 MQCONNX 호출을 사용하여 애플리케이션이 실행 중인 큐 관리자에 연결되면, 로드된 IBM WebSphere MQ 라이브러리를 사용하여 연결됩니다.

서버 애플리케이션이 큐 관리자에 연결되면, 로드된 라이브러리는 큐 관리자와 연관된 설치의 라이브러리어야 합니다. 시스템에 여러 설치가 있는 경우, 이러한 제한사항으로 인해 운영 체제에서 로드할 IBM WebSphere MQ 라이브러리를 찾기 위해 사용하는 메커니즘을 선택할 때 새로운 과제를 야기합니다.

- **setmqm** 명령을 사용하여 큐 관리자와 연관된 설치를 변경하는 경우, 로드해야 하는 라이브러리가 변경됩니다.
- 애플리케이션이 다른 설치에 속하는 다중 큐 관리자에 연결되는 경우, 여러 라이브러리 세트를 로드해야 합니다.

그러나 IBM WebSphere MQ Version 7.1 이상의 라이브러리를 찾아 로드한 경우, IBM WebSphere MQ는 추가 조치를 수행할 필요없이 해당 라이브러리를 로드하여 사용합니다. 애플리케이션이 큐 관리자에 연결되면, IBM WebSphere MQ는 큐 관리자와 연관된 설치의 라이브러리를 로드합니다.

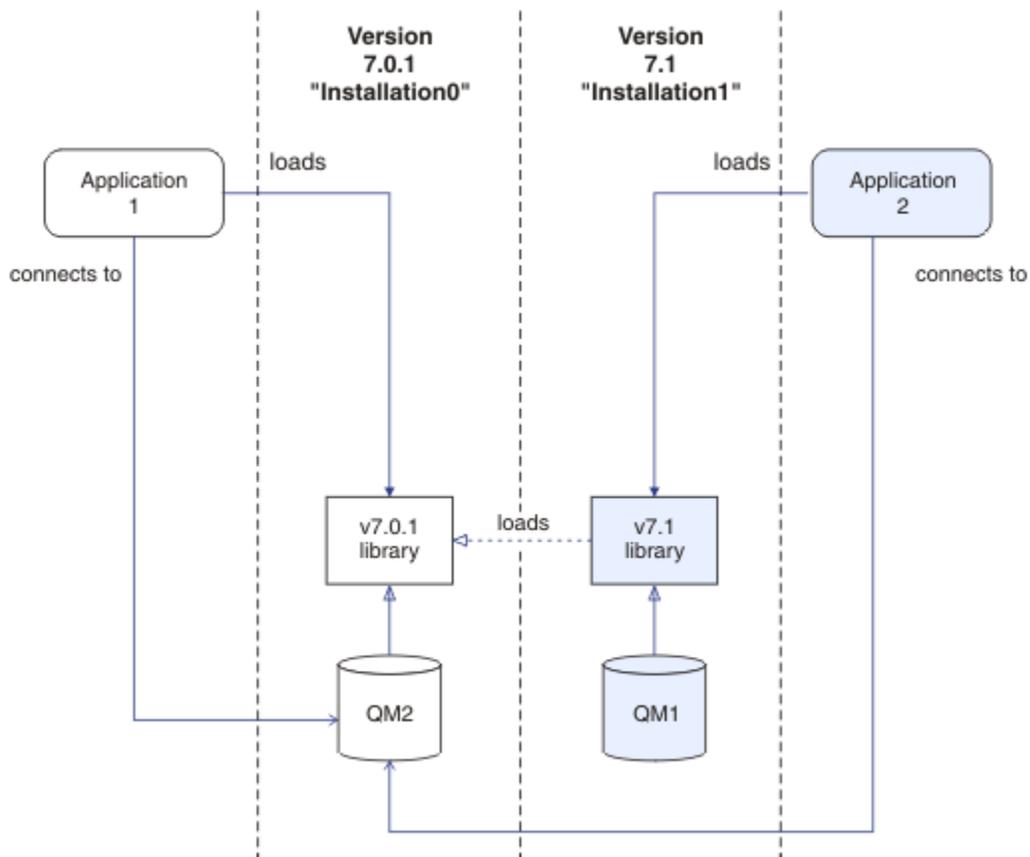


그림 1. 다중 설치 환경에서 애플리케이션 연결

예를 들어, 6 페이지의 [그림 1](#) 는 버전 7.0.1 설치 (Installation0) 및 버전 7.1 설치 (Installation1) 가 있는 다중 설치 환경을 표시합니다. 두 애플리케이션이 이 설치에 연결되어 있지만 서로 다른 라이브러리 버전을 로드합니다.

Application 1은 버전 7.0.1 라이브러리를 직접 로드합니다. application 1이 QM2로 연결되면 버전 7.0.1 라이브러리가 사용됩니다. application 1가 QM1에 연결을 시도하거나 QM2가 Installation1와 연관된 경우 application 1는 2059 (080B) (RC2059): MQR_C_MGR_NOT_AVAILABLE 오류로 실패합니다. 버전 7.0.1 라이브러리는 기타 라이브러리 버전을 로드할 수 없으므로 애플리케이션이 실패합니다. 즉, 버전 7.0.1 라이브러리가 직접 로드되면 이후 버전의 IBM WebSphere MQ에 있는 설치와 연관된 큐 관리자를 사용할 수 없습니다.

Application 2는 버전 7.1 라이브러리를 직접 로드합니다. application 2가 QM2로 연결되면 버전 7.1 라이브러리가 로드되고 버전 7.0.1 라이브러리를 사용합니다. application 2가 QM1에 연결되거나 QM2가 Installation1과 연관되는 경우, 버전 7.1 라이브러리가 로드되고 애플리케이션은 예상 대로 작업합니다.

마이그레이션 시나리오 및 다중 설치와 애플리케이션 연결에 대해서는 [UNIX, Linux 및 Windows에서 다중 설치 큐 관리자 공존하기](#)에서 자세히 설명합니다.

IBM WebSphere MQ Version 7.1 라이브러리 로드 방법에 대한 자세한 정보는 8 페이지의 『[IBM WebSphere MQ Version 7.1 이상의 라이브러리 로드](#)』의 내용을 참조하십시오.

지원 및 제한사항

다음 버전 7.1 이상의 라이브러리를 찾아 로드한 경우 IBM WebSphere MQ는 자동으로 해당 라이브러리를 로드하여 사용할 수 있습니다.

- C 서버 라이브러리
- C++ 서버 라이브러리
- XA 서버 라이브러리
- COBOL 서버 라이브러리
- COM+ 서버 라이브러리
- 비관리 모드에서의 .NET

IBM WebSphere MQ는 바인딩 모드에서 Java 및 JMS 애플리케이션에 적절한 라이브러리를 자동으로 로드하여 사용합니다.

다중 설치를 사용하는 애플리케이션에 대한 여러가지 제한사항이 있습니다. 자세한 정보는 11 페이지의 『[다중 설치를 사용하는 애플리케이션에 대한 제한사항](#)』의 내용을 참조하십시오.

관련 개념

15 페이지의 『[큐 관리자를 설치와 연관](#)』

큐 관리자를 작성하면 `crtmqm` 명령을 실행하는 설치와 자동으로 연관됩니다. UNIX, Linux, and Windows에서 `setmqm` 명령을 사용하여 큐 관리자와 연관된 설치를 변경할 수 있습니다.

11 페이지의 『[다중 설치를 사용하는 애플리케이션에 대한 제한사항](#)』

다중 설치 환경에서 CICS 서버 라이브러리, 빠른 경로 연결, 메시지 핸들, 엑시트를 사용할 경우 제한사항이 있습니다.

8 페이지의 『[IBM WebSphere MQ Version 7.1 이상의 라이브러리 로드](#)』

IBM WebSphere MQ 라이브러리를 로드하는 방법을 결정할 때는 사용자 환경, 기존 애플리케이션을 변경할 수 있는지 여부, 기본 설치를 원하는지 여부, IBM WebSphere MQ 설치 위치, IBM WebSphere MQ 위치가 변경되는지 여부를 포함하여 여러 요소를 고려해야 합니다.

관련 태스크

[기본 설치 선택](#)

14 페이지의 『[기본 설치 변경](#)』

`setmqinst` 명령을 사용하여 설치를 기본 설치로 설정 또는 설정 해제할 수 있습니다.

IBM WebSphere MQ Version 7.1 이상의 라이브러리 로드

IBM WebSphere MQ 라이브러리를 로드하는 방법을 결정할 때는 사용자 환경, 기존 애플리케이션을 변경할 수 있는지 여부, 기본 설치를 원하는지 여부, IBM WebSphere MQ 설치 위치, IBM WebSphere MQ 위치가 변경되는지 여부를 포함하여 여러 요소를 고려해야 합니다.

IBM WebSphere MQ Version 7.1 라이브러리를 찾아 로드하는 방법은 사용자 설치 환경에 따라 다릅니다.

- UNIX and Linux 시스템의 경우, IBM WebSphere MQ Version 7.1의 한 사본이 기본 위치에 설치되어 있으면, 기존 애플리케이션은 이전 버전과 동일한 방식으로 계속 작업합니다. 그러나 애플리케이션에 /usr/lib에 있는 기호 링크가 필요한 경우, 버전 7.1 설치를 기본 설치가 되도록 선택하거나 직접 기호 링크를 작성해야 합니다.
- IBM WebSphere MQ Version 7.1이 기본이 아닌 위치에 설치되는 경우(IBM WebSphere MQ Version 7.0.1도 설치되어 있는 경우), 올바른 라이브러리를 로드하도록 기존 애플리케이션을 변경해야 합니다.

IBM WebSphere MQ Version 7.1 이상의 라이브러리를 찾아 로드하는 방법은 기존 애플리케이션이 라이브러리를 로드하도록 설정된 방법에 따라 다릅니다. 라이브러리 로드 방법에 대한 자세한 정보는 [10 페이지의 『운영 체제 라이브러리 로딩 메커니즘』](#)의 내용을 참조하십시오.

큐 관리자가 운영 체제에서 로드하는 IBM WebSphere MQ 라이브러리와 연관되어 있는 경우가 가장 좋습니다.

IBM WebSphere MQ 라이브러리를 로드하는 방법은 플랫폼마다 다르며 각각의 방법마다 장단점이 있습니다.

플랫폼	옵션	장점	문제점
UNIX and Linux 시스템	<p>애플리케이션의 임베드된 런타임 검색 경로(RPath)를 설정하거나 변경합니다.</p> <p>이 옵션에서는 사용자가 애플리케이션을 다시 컴파일하고 링크해야 합니다. 애플리케이션 컴파일 및 링크에 대한 자세한 정보는 WebSphere MQ 애플리케이션 빌드를 참조하십시오.</p>	<ul style="list-style-type: none"> • 변경 범위가 명확합니다. 	<ul style="list-style-type: none"> • 애플리케이션을 다시 컴파일하고 링크할 수 있어야 합니다. • IBM WebSphere MQ 위치가 변경되면 RPath를 변경해야 합니다.

표 1. 라이브러리 로드 옵션의 장단점 (계속)

플랫폼	옵션	장점	문제점
UNIX and Linux 시스템	setmqenv 또는 crtmqenv를 -k 또는 -l 옵션과 함께 사용하여 <code>LD_LIBRARY_PATH</code> 환경 변수 (AIX®에서는 <code>LIBPATH</code>)를 설정하십시오.	<ul style="list-style-type: none"> • 기존 애플리케이션을 변경할 필요가 없습니다. • 애플리케이션에서 임베드된 RPath를 대체합니다. • IBM WebSphere MQ의 위치가 변경되면 변수를 변경하기가 쉽습니다. 	<ul style="list-style-type: none"> • <code>setuid</code> 및 <code>setgid</code> 애플리케이션 또는 기타 방식으로 빌드된 애플리케이션에서는 보안 상의 이유로 <code>LD_LIBRARY_PATH</code>를 무시합니다. • 환경에 따라 다르므로 애플리케이션이 실행되는 각 환경에서 설정되어야 합니다. • <code>LD_LIBRARY_PATH</code>에 의존하는 기타 애플리케이션에 영향을 줄 수 있습니다. • HP-UX: 애플리케이션이 컴파일되었을 때 사용한 옵션이 <code>LD_LIBRARY_PATH</code>를 사용 불가능으로 설정했을 수 있습니다. 자세한 정보는 HP-UX용 런타임 링크 고려사항을 참조하십시오. • Linux: 애플리케이션을 빌드할 때 사용한 컴파일러가 <code>LD_LIBRARY_PATH</code>를 사용 불가능으로 설정했을 수 있습니다. 자세한 정보는 Linux의 런타임 링크 고려사항을 참조하십시오.
Windows 시스템	setmqenv 또는 crtmqenv를 사용하여 PATH 변수를 설정합니다.	<ul style="list-style-type: none"> • 기존 애플리케이션을 변경할 필요가 없습니다. • IBM WebSphere MQ의 위치가 변경되면 변수를 변경하기가 쉽습니다. 	<ul style="list-style-type: none"> • 환경에 따라 다르므로 애플리케이션이 실행되는 각 환경에서 설정되어야 합니다. • 기타 애플리케이션에 영향을 줄 수 있습니다.
UNIX, Linux, and Windows 시스템	버전 7.1 이상의 설치로 기본 설치를 설정합니다. 14 페이지의 『기본 설치 변경』 의 내용을 참조하십시오. 기본 설치에 대한 자세한 정보는 기본 설치 선택 을 참조하십시오.	<ul style="list-style-type: none"> • 기존 애플리케이션을 변경할 필요가 없습니다. • IBM WebSphere MQ의 위치가 변경되면 기본 설치를 변경하기가 쉽습니다. • 이전 버전의 IBM WebSphere MQ에서와 유사하게 작동합니다. 	<ul style="list-style-type: none"> • WebSphere MQ 버전 7.0.1이 설치되면 기본 설치를 버전 7.1 이상으로 설정할 수 없습니다. • UNIX and Linux: <code>/usr/lib</code>가 기본 검색 경로에 없으면 작동하지 않습니다.

HP-UX의 라이브러리 로딩 고려사항

이전 버전의 IBM WebSphere MQ에 대한 제품 문서의 샘플 컴파일 명령에는 64비트 애플리케이션에 대한 `-w1, +noenvvar` 링크 옵션이 포함되어 있습니다. 이 옵션은 공유 라이브러리를 로드하기 위한 `LD_LIBRARY_PATH` 사용을 불가능하게 합니다. 애플리케이션이 RPath에 지정된 위치 이외의 위치에서 IBM WebSphere MQ 라이브러리를 로드하도록 하려면 애플리케이션을 업데이트해야 합니다. `-w1, +noenvvar` 링크 옵션 없이 다시 컴파일하고 링크함으로써 또는 `chatr` 명령을 사용하여 애플리케이션을 업데이트할 수 있습니다.

애플리케이션이 현재 라이브러리를 로드하는 방법을 확인하려면 10 페이지의 『운영 체제 라이브러리 로딩 메커니즘』의 내용을 참조하십시오.

Linux의 라이브러리 로딩 고려사항

gcc의 일부 버전(예: 버전 3.2.x)을 사용하여 컴파일된 애플리케이션에는 `LD_LIBRARY_PATH` 환경 변수를 사용하여 대체할 수 없는 임베드된 RPath가 있을 수 있습니다. `readelf -d applicationName` 명령을 사용하여 애플리케이션에 영향이 있는지 여부를 판별할 수 있습니다. RPATH 기호가 있고 RUNPATH 기호가 없는 경우에는 RPath를 대체할 수 없습니다.

Solaris의 라이브러리 로딩 고려사항

이전 버전의 IBM WebSphere MQ에 대한 제품 문서의 샘플 컴파일 명령에는 `-lmqmc` `-lmqzse` 링크 옵션이 포함되어 있습니다. 이제 IBM WebSphere MQ가 이러한 라이브러리의 적절한 버전을 자동으로 로드합니다. IBM WebSphere MQ가 기본이 아닌 위치에 설치되거나 시스템에 다중 설치가 있는 경우, 애플리케이션을 업데이트해야 합니다. `-lmqmc` `-lmqzse` 링크 옵션 없이 다시 컴파일하고 링크하여 애플리케이션을 업데이트할 수 있습니다.

운영 체제 라이브러리 로딩 메커니즘

Windows 시스템에서는 라이브러리를 찾기 위해 여러 디렉토리를 검색합니다.

- 애플리케이션이 로드되는 디렉토리
- 현재 디렉토리
- `PATH` 환경 변수(글로벌 `PATH` 변수 및 현재 사용자의 `PATH` 변수 둘 다)의 디렉토리

UNIX and Linux 시스템에서는 다음과 같은 다양한 방법을 사용하여 라이브러리를 찾을 수 있습니다.

- `LD_LIBRARY_PATH` 환경 변수 사용 (AIX에서는 `LIBPATH`, HP-UX에서는 `SHLIB_PATH`). 이 변수가 설정되면, 필수 WebSphere MQ 라이브러리를 위해 검색하는 디렉토리 세트를 정의합니다. 이 디렉토리에서 라이브러리를 찾으면 다른 방법을 사용하여 찾을 수 있는 라이브러리의 환경 설정에서 해당 라이브러리를 사용합니다.
- 임베드된 검색 경로(RPath) 사용. IBM WebSphere MQ 라이브러리를 검색하는 디렉토리 세트가 애플리케이션에 포함될 수 있습니다. `LD_LIBRARY_PATH`가 설정되지 않거나 변수를 사용하여 필요한 라이브러리를 찾지 못한 경우에는 RPath에서 라이브러리를 검색합니다. 기존 애플리케이션에서 RPath를 사용하지만 애플리케이션을 다시 컴파일하고 링크할 수 없는 경우에는 기본 위치에 IBM WebSphere MQ Version 7.1을 설치하거나 다른 방법을 사용하여 라이브러리를 찾아야 합니다.
- 기본 라이브러리 경로 사용. `LD_LIBRARY_PATH` 변수 및 RPath 위치를 검색한 후 WebSphere MQ 라이브러리를 찾지 못하면 기본 라이브러리 경로를 검색합니다. 일반적으로 이 경로에는 `/usr/lib` 또는 `/usr/lib64`가 포함됩니다. 기본 라이브러리 경로를 검색한 후 라이브러리를 찾지 못하면 종속 항목 누락으로 인해 애플리케이션이 시작되지 않습니다.

운영 체제 메커니즘을 사용하여 애플리케이션에 임베드된 검색 경로가 있는지 여부를 확인할 수 있습니다. 예를 들면, 다음과 같습니다.

- AIX: **dump**
- HP-UX: **chatr**
- Linux: **readelf**
- Solaris: **elfdump**

관련 개념

15 페이지의 『큐 관리자를 설치와 연관』

큐 관리자를 작성하면 `crtmqm` 명령을 실행하는 설치와 자동으로 연관됩니다. UNIX, Linux, and Windows에서 `setmqm` 명령을 사용하여 큐 관리자와 연관된 설치를 변경할 수 있습니다.

11 페이지의 『다중 설치를 사용하는 애플리케이션에 대한 제한사항』

다중 설치 환경에서 CICS 서버 라이브러리, 빠른 경로 연결, 메시지 핸들, 엑시트를 사용할 경우 제한사항이 있습니다.

5 페이지의 『다중 설치 환경에서 애플리케이션 연결』

UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ Version 7.1 이상의 라이브러리가 로드되면 추가 조치를 수행할 필요없이 IBM WebSphere MQ가 자동으로 적절한 라이브러리를 사용합니다. IBM WebSphere MQ 애플리케이션이 연결되는 큐 관리자와 연관된 설치의 라이브러리를 사용합니다.

관련 태스크

기본 설치 선택

14 페이지의 『기본 설치 변경』

setmqinst 명령을 사용하여 설치를 기본 설치로 설정 또는 설정 해제할 수 있습니다.

다중 설치를 사용하는 애플리케이션에 대한 제한사항

다중 설치 환경에서 CICS 서버 라이브러리, 빠른 경로 연결, 메시지 핸들, 엑시트를 사용할 경우 제한사항이 있습니다.

CICS 서버 라이브러리

CICS 서버 라이브러리를 사용하는 경우 IBM WebSphere MQ는 올바른 라이브러리 레벨을 자동으로 선택하지 않습니다. 애플리케이션이 연결되는 큐 관리자에 적합한 라이브러리 레벨로 애플리케이션을 컴파일 및 링크해야 합니다. 자세한 정보는 [TXSeries®](#) 를 참조하십시오.

메시지 핸들

특수값 MQHC_UNASSOCIATED_HCONN을 사용하는 메시지 핸들은 프로세스에 로드되는 첫 번째 설치에 사용하도록 제한됩니다. 특정 설치에서 메시지 핸들을 사용할 수 없는 경우 이유 코드 MQRC_HMSG_NOT_AVAILABLE 이 리턴됩니다.

이 제한은 메시지 특성에 영향을 미칩니다. 메시지 핸들을 사용하면 한 설치의 큐 관리자에서 메시지 특성을 가져와 다른 설치의 큐 관리자로 삽입할 수 없습니다. 메시지 핸들에 대한 자세한 정보는 [MQCRTMH - 메시지 핸들 작성](#) 을 참조하십시오.

엑시트

다중 설치 환경에서, IBM WebSphere MQ Version 7.1 이상 설치와 함께 사용하려면 기존 엑시트를 업데이트해야 합니다. **crtmqcvx** 명령을 사용하여 생성된 데이터 변환 엑시트는 업데이트된 명령을 사용하여 다시 생성되어야 합니다.

모든 엑시트는 MQIEP 구조를 사용하여 작성되어야 하며, 임베드된 RPATH를 사용하여 IBM WebSphere MQ 라이브러리를 찾을 수 없고, IBM WebSphere MQ 라이브러리에 링크할 수도 없습니다. 자세한 정보는 [엑시트 및 설치 가능 서비스의 작성 및 컴파일](#) 을 참조하십시오.

바로 가기

다중 설치가 있는 서버에서, IBM WebSphere MQ Version 7.1 이상에 대한 빠른 경로 연결을 사용하는 애플리케이션은 다음 규칙을 준수해야 합니다.

1. 큐 관리자는 애플리케이션이 IBM WebSphere MQ 런타임 라이브러리를 로드할 때 사용한 것과 동일한 설치와 연관되어야 합니다. 애플리케이션은 다른 설치와 연관된 큐 관리자에 대해 빠른 경로 연결을 사용할 수 없습니다. 연결을 시도하면 오류가 발생하고 이유 코드는 MQRC_INSTALLATION_MISMATCH입니다.
2. 애플리케이션이 IBM WebSphere MQ 런타임 라이브러리를 로드할 때 사용한 것과 동일한 설치와 연관된 큐 관리자에 빠르지 않은 경로를 연결하면 애플리케이션에서 빠른 경로 연결을 사용할 수 없게 됩니다. 단, 다음 조건 중 하나에 해당하는 경우는 예외입니다.
 - 애플리케이션이 빠른 경로 연결과 동일한 설치와 연관된 큐 관리자에 처음 연결됩니다.
 - 환경 변수 AMQ_SINGLE_INSTALLATION이(가) 설정됩니다.
3. Version 7.1 이상의 설치와 연관된 큐 관리자로의 빠르지 않은 경로 연결은 애플리케이션이 빠른 경로에 연결할 수 있는지 여부에 영향을 미치지 않습니다.

4. Version 7.0.1 설치와 연관된 큐 관리자로의 연결과, Version 7.1 이상 설치와 연관된 큐 관리자로의 빠른 경로 연결을 조합할 수 없습니다.

AMQ_SINGLE_INSTALLATION을(를) 설정하면 빠른 경로 연결을 큐 관리자에 연결할 수 있습니다. 그렇지 않은 경우 다음과 거의 동일한 제한사항이 적용됩니다.

- 설치하는 IBM WebSphere MQ 런타임 라이브러리가 로드된 설치와 동일해야 합니다.
- 동일한 프로세스에서의 모든 연결은 동일한 설치로의 연결이어야 합니다. 다른 설치와 연관된 큐 관리자에 연결하려고 하면 오류 코드 MQRC_INSTALLATION_MISMATCH과(와) 함께 연결이 실패합니다. AMQ_SINGLE_INSTALLATION 세트를 사용하면 빠른 경로 연결만이 아니라 모든 연결에 이 제한사항이 적용됩니다.
- 빠른 경로 연결과 하나의 큐 관리자만 연결하십시오.

관련 참조

[MQCONN - 연결 큐 관리자\(확장\)](#)

[MQIEP 구조](#)

[2583 \(0A17\) \(RC2583\): MQRC_INSTALLATION_MISMATCH](#)

[2587 \(0A1B\) \(RC2587\): MQRC_HMSG_NOT_AVAILABLE](#)

[2590 \(0A1E\) \(RC2590\): MQRC_FASTPATH_NOT_AVAILABLE](#)

다중 설치 환경에서 .NET 애플리케이션 연결

기본적으로 애플리케이션에서는 기본 설치의 .NET 어셈블리를 사용합니다. 기본 설치가 없거나 기본 설치 어셈블리를 사용하지 않으려면 애플리케이션 구성 파일 또는 *DEVPATH* 환경 변수를 업데이트해야 합니다.

시스템에 기본 설치가 있는 경우, 해당 설치의 정책 파일 및 .NET 어셈블리는 GAC(global assembly cache)에 등록됩니다. 기타 모든 설치의 .NET 어셈블리는 각 설치의 설치 경로에서 찾을 수 있지만, 어셈블리는 GAC에 등록되지 않습니다. 그러므로 기본적으로 애플리케이션은 기본 설치에서 .NET 어셈블리를 사용하여 실행됩니다. 다음과 같은 경우 애플리케이션 구성 파일을 업데이트해야 합니다.

- 기본 설치가 없습니다.
- 애플리케이션에서 기본 설치 어셈블리를 사용하지 않습니다.
- 기본 설치가 애플리케이션이 컴파일된 버전보다 낮은 버전의 IBM WebSphere MQ입니다.

애플리케이션 구성 파일을 업데이트하는 방법에 대한 정보는 [12 페이지](#)의 『애플리케이션 구성 파일을 사용한 .NET 애플리케이션 연결』의 내용을 참조하십시오.

다음과 같은 경우 *DEVPATH* 환경 변수를 업데이트해야 합니다.

- 애플리케이션에서 비기본 설치의 어셈블리를 사용하려고 하지만 기본 설치가 비기본 설치와 동일한 버전입니다.

DEVPATH 변수 업데이트 방법에 대한 자세한 정보는 [13 페이지](#)의 『*DEVPATH*를 사용한 .NET 애플리케이션 연결』의 내용을 참조하십시오.

애플리케이션 구성 파일을 사용한 .NET 애플리케이션 연결

애플리케이션 구성 파일에서, 여러 태그를 설정하여 기본 설치가 아닌 어셈블리를 사용하도록 애플리케이션 경로를 재지정해야 합니다.

다음 표에서는 .NET 애플리케이션이 특정 어셈블리를 사용하여 연결할 수 있도록 애플리케이션 구성 파일에 작성해야 하는 특정 변경사항을 표시합니다.

표 2. 특정 어셈블리를 사용하도록 애플리케이션 구성		
	IBM WebSphere MQ의 하위 버전으로 컴파일된 애플리케이션	IBM WebSphere MQ의 상위 버전을 사용하여 컴파일된 애플리케이션
높은 버전의 IBM WebSphere MQ 기본 설치로 애플리케이션을 실행합니다. (GAC에 있는 높은 버전 어셈블리):	변경이 필요하지 않음	변경이 필요하지 않음
낮은 버전의 IBM WebSphere MQ 기본 설치로 애플리케이션을 실행합니다. (GAC에 있는 낮은 버전 어셈블리):	변경이 필요하지 않음	애플리케이션 구성 파일에서: • GAC에 있는 어셈블리의 하위 버전 사용을 표시하려면 <code><bindingRedirect></code> 태그를 사용하십시오.
높은 버전의 IBM WebSphere MQ 1차가 아닌 설치로 애플리케이션을 실행합니다. (설치 폴더에 있는 높은 버전 어셈블리):	애플리케이션 구성 파일에서: • 상위 버전 어셈블리의 위치를 지시하려면 <code><codebase></code> 태그를 사용하십시오. • <code><bindingRedirect></code> 태그를 사용하여 상위 버전 어셈블리의 사용을 표시하십시오.	애플리케이션 구성 파일에서: • 상위 버전 어셈블리의 위치를 지시하려면 <code><codebase></code> 태그를 사용하십시오.
낮은 버전의 IBM WebSphere MQ 1차가 아닌 설치로 애플리케이션을 실행합니다. (설치 폴더에 있는 낮은 버전 어셈블리):	애플리케이션 구성 파일에서: • <code><codebase></code> 태그를 사용하여 하위 버전 어셈블리의 위치를 지시하십시오. • <code><publisherpolicy Apply=no></code> 태그를 포함하십시오.	애플리케이션 구성 파일에서: • <code><codebase></code> 태그를 사용하여 하위 버전 어셈블리의 위치를 지시하십시오. • <code><bindingRedirect></code> 태그를 사용하여 하위 버전 어셈블리의 사용을 표시하십시오. • <code><publisherpolicy Apply=no></code> 태그를 포함하십시오.

샘플 응용프로그램 구성 파일 `NonPrimaryRedirect.config` 이 `MQ_INSTALLATION_PATH\tools\dotnet\samples\base` 폴더에 제공됩니다. 이 파일은 비기본 설치의 IBM WebSphere MQ 설치 경로로 수정될 수 있습니다. 파일은 `<linkedConfiguration>` 태그를 사용하여 다른 구성 파일에도 직접 포함될 수 있습니다. 샘플은 `nmqsget.exe.config` 및 `nmqsput.exe.config`에 대해 제공됩니다. 두 샘플 모두 `<linkedConfiguration>` 태그를 사용하며 `NonPrimaryRedirect.config` 파일을 포함합니다.

DEVPATH를 사용한 .NET 애플리케이션 연결

`DEVPATH` 환경 변수를 사용하여 어셈블리를 찾을 수 있습니다. `DEVPATH` 변수로 지정하는 어셈블리는 GAC에 있는 어셈블리에 대한 참조에서 사용됩니다. 이 변수 사용 시기에 대한 자세한 정보는 `DEVPATH`에 대한 해당 Microsoft 문서를 참조하십시오.

`DEVPATH` 환경 변수를 사용하여 어셈블리를 찾으려면 사용하려는 어셈블리가 있는 폴더로 `DEVPATH` 변수를 설정해야 합니다. 그런 다음 애플리케이션 구성 파일을 업데이트하고 다음 런타임 구성 정보를 추가해야 합니다.

```
<configuration>
  <runtime>
    <developmentMode developerInstallation="true" />
  </runtime>
</configuration>
```

관련 개념

5 페이지의 『다중 설치 환경에서 애플리케이션 연결』

UNIX, Linux, and Windows 시스템에서 IBM WebSphere MQ Version 7.1 이상의 라이브러리가 로드되면 추가 조치를 수행할 필요없이 IBM WebSphere MQ가 자동으로 적절한 라이브러리를 사용합니다. IBM WebSphere MQ 애플리케이션이 연결되는 큐 관리자와 연관된 설치의 라이브러리를 사용합니다.

다중 설치

관련 태스크

기본 설치 선택

.NET 사용

기본 설치 변경

`setmqinst` 명령을 사용하여 설치를 기본 설치로 설정 또는 설정 해제할 수 있습니다.

이 태스크 정보

이 태스크는 UNIX, Linux, and Windows에 적용됩니다.

기본 설치는 요청되는 시스템 전체 위치에서 참조하는 설치입니다. 기본 설치에 대한 자세한 정보와 기본 설치 선택을 위한 고려사항은 [기본 설치 선택](#)을 참조하십시오.

IBM WebSphere MQ Version 7.1 이상의 설치가 IBM WebSphere MQ Version 7.0.1의 설치와 공존하는 경우 IBM WebSphere MQ Version 7.0.1 설치는 1차여야 합니다. IBM WebSphere MQ Version 7.1 이상 버전이 설치되면 기본으로 플래그가 지정되고 IBM WebSphere MQ Version 7.1 이상 설치는 기본이 될 수 없습니다.

Windows에서 설치 프로세스를 수행하는 동안 이 설치가 기본 설치가 되도록 지정할 수 있습니다. UNIX and Linux 시스템에서 설치를 기본 설치로 설정한 후 `setmqinst` 명령을 실행해야 합니다.

14 페이지의 『기본 설치 설정』.

15 페이지의 『기본 설치 설정 해제』.

기본 설치 설정

프로시저

설치를 기본 설치로 설정하려면 다음을 수행하십시오.

1. 다음 명령을 입력하여 설치가 기본 설치인지 확인하십시오.

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

여기서 `MQ_INSTALLATION_PATH`는 IBM WebSphere MQ Version 7.1 이상 설치의 설치 경로입니다.

2. 기존 IBM WebSphere MQ Version 7.1 이상의 설치가 기본 설치로 설정되어 있는 경우 [15 페이지의 『기본 설치 설정 해제』](#)의 지시사항에 따라 설정을 해제하십시오. 시스템에 IBM WebSphere MQ Version 7.0.1이 설치되어 있으면 기본 설치를 변경할 수 없습니다.
3. UNIX and Linux 시스템의 루트 또는 Windows 시스템의 관리자 그룹 구성원으로 다음 명령 중 하나를 입력하십시오.

- 기본 설치로 사용할 설치의 경로를 사용하여 기본 설치를 설정하려면 다음 명령을 실행하십시오.

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH
```

- 기본 설치로 사용할 설치의 이름을 사용하여 기본 설치를 설정하려면 다음 명령을 실행하십시오.

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -n installationName
```

4. Windows 시스템에서 시스템을 재시작하십시오.

기본 설치 설정 해제

프로시저

설치를 기본 설치로 설정 해제하려면 다음을 수행하십시오.

1. 다음 명령을 입력하여 설치가 기본 설치인지 확인하십시오.

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

여기서 `MQ_INSTALLATION_PATH`는 IBM WebSphere MQ Version 7.1 이상 설치의 설치 경로입니다.

IBM WebSphere MQ Version 7.0.1이 기본 설치이면 기본 설치를 설정 해제할 수 없습니다.

2. UNIX and Linux 시스템의 루트 또는 Windows 시스템의 관리자 그룹 구성원으로 다음 명령 중 하나를 입력하십시오.

- 더 이상 기본 설치로 사용하지 않을 설치의 경로를 사용하여 기본 설치를 설정 해제하려면 다음 명령을 실행하십시오.

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -p MQ_INSTALLATION_PATH
```

- 더 이상 기본 설치로 사용하지 않을 설치의 이름을 사용하여 기본 설치를 설정 해제하려면 다음 명령을 실행하십시오.

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -n installationName
```

관련 개념

Windows에서 기본 설치에서만 사용할 수 있는 기능

[UNIX 및 Linux에서 기본 설치에 대한 외부 라이브러리 및 제어 명령 링크](#)

관련 태스크

[기본 설치의 설치 제거, 업그레이드 및 유지보수](#)

[설치 이름 선택](#)

관련 참조

[setmqinst](#)

큐 관리자를 설치와 연관

큐 관리자를 작성하면 `crtmqm` 명령을 실행하는 설치와 자동으로 연관됩니다. UNIX, Linux, and Windows에서 `setmqm` 명령을 사용하여 큐 관리자와 연관된 설치를 변경할 수 있습니다.

`setmqm` 명령은 다음 방법으로 사용할 수 있습니다.

- 동일한 버전의 WebSphere MQ 간에 개별적인 큐 관리자를 이동합니다. 예를 들어, 테스트 시스템에서 프로덕션 시스템으로 큐 관리자를 이동합니다.
- 개별적인 큐 관리자를 이전 버전의 WebSphere MQ에서 새 버전의 WebSphere MQ로 마이그레이션합니다. 버전 간에 큐 관리자를 마이그레이션할 때 여러 가지 함축된 내용을 알고 있어야 합니다. 마이그레이션에 대한 자세한 정보는 [WebSphere MQ 마이그레이션 및 업그레이드](#)를 참조하십시오.

큐 관리자를 설치와 연관시키려면 다음을 수행하십시오.

1. 현재 큐 관리자와 연관된 설치에서 `endmqm` 명령을 사용하여 큐 관리자를 중지하십시오.
2. 설치에서 `setmqm` 명령을 사용하여 큐 관리자를 다른 설치와 연관시키십시오.

예를 들어 이름이 `Installation2`인 설치와 연관되도록 큐 관리자 `QMB`를 설정하려면 `Installation2`에서 다음 명령을 입력하십시오.

```
MQ_INSTALLATION_PATH/bin/setmqm -m QMB -n Installation2
```

여기서, `MQ_INSTALLATION_PATH`는 `Installation2`가 설치된 경로입니다.

3. 현재 큐 관리자와 연관된 설치에서 `strmqm` 명령을 사용하여 큐 관리자를 시작하십시오.

이 명령은 필요한 큐 관리자 마이그레이션을 수행하며 큐 관리자를 사용 준비 상태로 만듭니다.

해당 설치에서의 명령만으로 관리될 수 있도록 큐 관리자가 연관된 설치가 해당 큐 관리자를 제한합니다. 다음과 같은 세 가지 주요 예외 사항이 있습니다.

- **setmqm**은 큐 관리자와 연관된 설치를 변경합니다. 이 명령은 큐 관리자가 현재 연관된 설치가 아니라 큐 관리자와 연관시키려는 설치에서 실행되어야 합니다. **setmqm** 명령을 통해 지정된 설치 이름은 명령이 실행된 설치와 일치해야 합니다.
- 일반적으로 **strmqm**은 큐 관리자와 연관된 설치에서 실행되어야 합니다. 그러나 V7.0.1 또는 이전 큐 관리자가 V7.1 이상의 설치에서 처음으로 시작될 때 **strmqm**을 사용할 수 있습니다. 이 경우 **strmqm**이 큐 관리자를 시작하여 명령이 실행된 설치와 연관시킵니다.
- **dspmq**는 **dspmq** 명령과 동일한 설치와 연관된 큐 관리자만이 아니라 시스템에 있는 모든 큐 관리자에 대한 정보를 표시합니다. **dspmq -o installation** 명령은 어떤 큐 관리자가 어떤 설치와 연관되어 있는지에 대한 정보를 표시합니다.

HA 환경의 큐 관리자 연관

HA 환경의 경우 **addmqinf** 명령이 자동으로 큐 관리자를 **addmqinf** 명령이 실행된 설치와 연관시킵니다. **strmqm** 명령이 **addmqinf** 명령과 동일한 설치에서 실행되는 동안 추가적인 설정이 필요하지 않습니다. 다른 설치를 사용하여 큐 관리자를 시작하려면 먼저 **setmqm** 명령을 사용하여 연관된 설치를 변경해야 합니다.

삭제된 설치와 연관된 큐 관리자

큐 관리자가 연관된 설치가 삭제되었거나 큐 관리자 상태 정보를 사용할 수 없는 경우 큐 관리자를 다른 설치와 연관시키기 위한 **setmqm** 명령이 실패합니다. 이 경우 다음 조치를 수행하십시오.

1. 시스템에 있는 다른 설치를 보려면 **dspmqinst** 명령을 사용하십시오.
2. 다른 설치를 지정하려면 **mqs.ini**에 있는 QueueManager 스탠자의 **InstallationName** 필드를 수동으로 수정하십시오.
3. 큐 관리자를 삭제하려면 해당 설치에서 **dltmqm** 명령을 사용하십시오.

관련 개념

[16 페이지의 『시스템에서 IBM WebSphere MQ 설치 찾기』](#)

시스템에 다중 IBM WebSphere MQ 설치가 있는 경우 설치된 버전과 설치 위치를 확인할 수 있습니다.

[382 페이지의 『IBM WebSphere MQ 구성 파일, mqs.ini』](#)

IBM WebSphere MQ 구성 파일 **mqs.ini**는 노드에 있는 모든 큐 관리자에 대한 정보를 포함합니다. 이 파일은 설치 중 자동으로 작성됩니다.

관련 태스크

[기본 설치 선택](#)

관련 참조

[setmqm](#)

[strmqm](#)

[dspmq](#)

[dspmqinst](#)

시스템에서 IBM WebSphere MQ 설치 찾기

시스템에 다중 IBM WebSphere MQ 설치가 있는 경우 설치된 버전과 설치 위치를 확인할 수 있습니다.

다음 방법을 사용하여 시스템에서 IBM WebSphere MQ 설치를 찾을 수 있습니다.

- **dspmqver** 명령을 사용하십시오. 이 명령은 Version 7.0.1 설치에서 실행된 경우 시스템의 모든 설치에 대한 자세한 내용을 제공하지 않습니다.
- 플랫폼 설치 도구를 사용하여 IBM WebSphere MQ가 설치된 위치를 조회합니다. 그런 다음 Version 7.1 이상 설치에서 **dspmqver** 명령을 사용하십시오. 다음 명령은 IBM WebSphere MQ가 설치된 위치를 조회하는 데 사용할 수 있는 명령 예제입니다.

- AIX 시스템의 경우 다음과 같이 **lslpp** 명령을 사용할 수 있습니다.

```
lslpp -R ALL -l mqm.base.runtime
```

- HP-UX 시스템의 경우 다음과 같이 **swlist** 명령을 사용할 수 있습니다.

```
swlist -a location -a revision -l product MQSERIES
```

- Linux 시스템의 경우 다음과 같이 **rpm** 명령을 사용할 수 있습니다.

```
rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
```

- Solaris 시스템의 경우 다음과 같이 **pkginfo** 및 **pkgparam** 명령을 사용할 수 있습니다.

1. 다음 명령을 입력하여 설치된 패키지를 나열하십시오.

```
pkginfo | grep -w mqm
```

2. 나열된 각 패키지에 대해 다음 명령을 입력하십시오.

```
pkgparam pkgname BASEDIR
```

- Windows 시스템의 경우 다음과 같이 **wmic** 명령을 사용할 수 있습니다. 이 명령은 wmic 클라이언트를 설치할 수 있습니다.

```
wmic product where "(Name like '%MQ%') AND (not Name like '%bitSupport')" get Name, Version, InstallLocation
```

- UNIX and Linux 시스템의 경우 다음 명령을 실행하여 IBM WebSphere MQ가 설치된 위치를 확인하십시오.

```
cat /etc/opt/mqm/mqinst.ini
```

그런 다음 Version 7.1 이상 설치에서 **dspmqver** 명령을 사용하십시오.

- 32비트 Windows에서 시스템의 설치 세부사항을 표시하려면 다음 명령을 실행하십시오.

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation" /s
```

- 64비트 Windows의 경우 다음 명령을 실행하십시오.

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\WebSphere MQ\Installation" /s
```

참고: **reg.exe** 명령은 Version 7.1 이상 설치에 대한 정보만 표시합니다.

관련 개념

[다중 설치](#)

[관련 참조](#)

[dspmqver](#)

[dspmqinst](#)

큐 관리자 작성 및 관리

메시지 및 큐를 사용하려면 먼저 최소한 하나의 큐 관리자 및 연관된 오브젝트를 작성하고 시작해야 합니다.

큐 관리자 작성

큐 관리자는 연관된 자원, 특히 큐에서 소유하는 자원을 관리합니다. 큐 관리자는 IBM WebSphere MQ 오브젝트 작성, 수정, 표시, 삭제를 위한 메시지 큐잉 인터페이스(MQI) 호출 및 명령에 대한 큐잉 서비스를 애플리케이션에 제공합니다.

큐 관리자를 작성하려면 IBM WebSphere MQ 제어 명령 **crtmqm**(**crtmqm**에서 설명함)을 사용하십시오.

crtmqm 명령은 필수 기본 오브젝트 및 시스템 오브젝트 (시스템 기본 오브젝트에 설명됨) 를 자동으로 작성합니다. 기본 오브젝트는 사용자가 작성하는 오브젝트 정의의 기반을 형성합니다. 큐 관리자 조작에는 시스템 오브

젝트가 필요합니다. 큐 관리자와 해당 오브젝트를 작성한 경우에는 `strmqm` 명령을 사용하여 큐 관리자를 시작하십시오.

참고: IBM WebSphere MQ는 공백을 포함하는 시스템 이름을 지원하지 않습니다. 공백을 포함하는 시스템 이름으로 컴퓨터에 IBM WebSphere MQ를 설치하면 큐 관리자를 작성할 수 없습니다.

On

큐 관리자를 작성하려면 여러 가지 사항을 고려해야 합니다(특히 프로덕션 환경의 경우). 다음 체크리스트를 검토하십시오.

큐 관리자와 연관된 설치

`crtmqm` 명령은 큐 관리자를 `crtmqm` 명령이 실행된 설치와 자동으로 연관시킵니다. 큐 관리자에 작용하는 명령의 경우, 큐 관리자와 연관된 설치에서 명령을 실행해야 합니다. `setmqm` 명령을 사용하여 큐 관리자의 연관된 설치를 변경할 수 있습니다. Windows 설치 프로그램은 `mqm` 그룹에 대한 설치를 수행하는 사용자를 추가하지 않습니다. 자세한 내용은 [Authority to administer IBM WebSphere MQ on UNIX, Linux and 윈도우 systems](#)의 내용을 참조하십시오.

이름 지정 규칙

모든 플랫폼에서 큐 관리자와 통신할 수 있도록 대문자 이름을 사용하십시오. 이름은 입력한 그대로 지정됩니다. 많은 입력에 따른 불편을 방지하려면 불필요하게 긴 이름을 사용하지 마십시오.

고유 큐 관리자 이름 지정

큐 관리자를 작성할 경우 다른 큐 관리자가 네트워크의 임의 위치에서 동일한 이름을 사용하지 않도록 하십시오. 큐 관리자를 작성할 때 큐 관리자 이름을 검사하지 않으므로, 고유하지 않은 이름을 사용하면 분산 큐잉 목적으로 채널을 작성할 수 없습니다.

고유하게 하는 한 가지 방법은 각 큐 관리자 이름에 자체의 고유한 노드 이름으로 접두부를 붙이는 것입니다. 예를 들어, 노드를 ACCOUNTS(이)라고 하는 경우, 큐 관리자 ACCOUNTS.SATURN.QUEUE.MANAGER의 이름을 지정할 수 있습니다. 여기서 SATURN은(는) 특정 큐 관리자를 식별하고 QUEUE.MANAGER은(는) 모든 큐 관리자에게 제공할 수 있는 확장입니다. 또는 이를 생략할 수도 있지만 ACCOUNTS.SATURN과 ACCOUNTS.SATURN.QUEUE.MANAGER는 서로 다른 큐 관리자 이름임을 참고하십시오.

다른 엔터프라이즈와의 통신을 위해 IBM WebSphere MQ를 사용하는 경우에는 사용자 고유의 엔터프라이즈 이름을 접두부로 포함할 수도 있습니다. 이 작업은 수행 방법이 더 어려워 예에서는 다루지 않습니다.

참고: 제어 명령의 큐 관리자 이름은 대소문자를 구분합니다. 이는 이름이 `jupiter.queue.manager` 및 `JUPITER.queue.manager`인 두 개의 큐 관리자를 작성할 수 있음을 의미합니다. 그러나 이러한 조합은 피하는 것이 좋습니다.

큐 관리자 수 제한

자원이 허용하는 한 많은 큐 관리자를 작성할 수 있습니다. 그러나 각 큐 관리자에서는 고유한 자원이 필요하므로 일반적으로 10개 큐 관리자에서 각각 10개 큐를 갖는 것보다 노드에서 하나의 큐 관리자가 100개 큐를 갖는 것이 좋습니다.

프로덕션 시스템에서는 단일 큐 관리자로 많은 프로세서를 이용할 수 있지만 큐 관리자가 많아 서버 시스템이 클수록 보다 효과적으로 실행될 수 있습니다.

기본 큐 관리자 지정

각 노드마다 기본 큐 관리자가 있어야 하지만 기본 큐 관리자 없이 노드에서 IBM WebSphere MQ를 구성할 수도 있습니다. 기본 큐 관리자는 애플리케이션이 MQCONN 호출에서 큐 관리자 이름을 지정하지 않는 경우 연결하는 큐 관리자입니다. 또한 큐 관리자 이름을 지정하지 않고 `runmqsc` 명령을 호출할 때 MQSC 명령을 처리하는 큐 관리자이기도 합니다.

큐 관리자를 기본으로 지정하면 노드에 대한 기존의 기본 큐 관리자 스펙이 바뀝니다.

기본 큐 관리자를 변경하면 다른 사용자 또는 애플리케이션에 영향을 줄 수 있습니다. 현재 연결된 애플리케이션은 추가 MQI 호출에서 원래 연결 호출의 핸들을 사용하므로 변경에 따른 영향을 받지 않습니다. 이 핸들을 사용하면 동일한 큐 관리자에 호출이 전달됩니다. 기본 큐 관리자를 변경한 후 연결하는 모든 애플리케이션은 새 기본 큐 관리자에 연결됩니다. 원하는 결과일지라도 기본값을 변경하기 전에 이를 고려해야 합니다.

기본 큐 관리자 작성은 20 페이지의 『기본 큐 관리자 작성』에서 설명됩니다.

데드-레터 큐 지정

데드-레터 큐는 메시지가 지정된 목적지로 라우트될 수 없을 때 메시지를 넣는 로컬 큐입니다.

네트워크의 각 큐 관리자에 데드-레터 큐를 두어야 합니다. 이 큐를 정의하지 않으면 애플리케이션 프로그램의 오류로 인해 채널이 닫히고 관리 명령에 대한 응답이 수신되지 않을 수 있습니다.

예를 들어 애플리케이션이 다른 큐 관리자의 큐에 메시지를 넣으려고 하지만 큐 이름을 잘못 입력하면 채널은 중지되고 메시지는 전송 큐에 그대로 남습니다. 그러면 다른 애플리케이션이 해당 메시지에 이 채널을 사용할 수 없게 됩니다.

큐 관리자에 데드-레터 큐가 있으면 채널이 영향을 받지 않습니다. 미전달 메시지는 수신측 데드-레터 큐에 놓이고 채널과 해당 전송 큐를 계속 사용할 수 있습니다.

큐 관리자를 작성할 때 -u 플래그를 사용하여 데드-레터 큐의 이름을 지정하십시오. MQSC 명령을 사용하여 사용될 데드-레터 큐를 지정하기 위해 이미 정의한 큐 관리자의 속성을 대체할 수도 있습니다. MQSC 명령 ALTER의 예는 [큐 관리자에 대한 작업을 참조하십시오](#).

기본 전송 큐 지정

전송 큐는 리모트 큐 관리자로 전송 중인 메시지가 전송 이전에 큐잉되는 로컬 큐입니다. 기본 전송 큐는 명시적으로 정의된 전송 큐가 없을 때 사용되는 큐입니다. 각 큐 관리자에 기본 전송 큐가 지정될 수 있습니다.

큐 관리자를 작성할 때 -d 플래그를 사용하여 기본 전송 큐의 이름을 지정하십시오. 실제로 큐를 작성하는 것은 아니며 나중에 명시적으로 수행해야 합니다. 자세한 정보는 [로컬 큐에 대한 작업을 참조하십시오](#).

필요한 로깅 매개변수 지정

로깅 유형과 로그 파일의 경로 및 크기를 포함하는 로깅 매개변수를 `crtmqm` 명령에 지정할 수 있습니다.

개발 환경에서는 기본 로깅 매개변수가 적절해야 합니다. 그러나 예를 들어 다음과 같은 경우 기본값을 변경할 수 있습니다.

- 대용량 로그를 지원할 수 없는 로우엔드 시스템 구성입니다.
- 큐에 동시에 많은 긴 메시지가 있는 것으로 예상합니다.
- 큐 관리자를 통해 많은 지속 메시지가 전달되는 것으로 예상합니다.

로깅 매개변수를 설정한 경우, 일부 매개변수는 큐 관리자를 삭제하고 이름은 같지만 다른 로깅 매개변수로 재작성하는 방법으로도 변경할 수 있습니다.

매개변수 로깅에 대한 자세한 정보는 281 페이지의 [『가용성, 복구 및 재시작』](#)의 내용을 참조하십시오.

유닉스 시스템의 경우 IBM WebSphere MQ 의 경우

`crtmqm` 명령을 사용하기 전에 별도의 로컬 파일 시스템에서도 큐 관리자 디렉토리 `/var/mqm/qmgrs/<qmgr>`를 작성할 수 있습니다. `crtmqm`를 사용하는 경우, `/var/mqm/qmgrs/<qmgr>` 디렉토리가 존재하고 비어 있고 `mqm`이 소유하는 경우 큐 관리자 데이터에 사용됩니다. 디렉토리를 `mqm`이 소유하지 않으면 FFDC (First Failure Support Technology) (파스트) 메시지와 함께 작성에 실패합니다. 디렉토리가 비어 있지 않으면 새 디렉토리가 작성됩니다.

관련 개념

5 페이지의 [『구성』](#)

하나 이상의 컴퓨터에서 하나 이상의 큐 관리자를 작성하고 개발, 테스트, 프로덕션 시스템에서 해당 큐 관리자가 비즈니스 데이터를 포함하는 메시지를 처리하도록 구성합니다.

22 페이지의 [『큐 관리자 작성 후 구성 파일 백업』](#)

IBM WebSphere MQ 구성 정보는 Windows, UNIX and Linux 시스템의 구성 파일에 저장되어 있습니다.

22 페이지의 [『큐 관리자 시작』](#)

큐 관리자를 작성할 때 명령 또는 MQI 호출을 처리하려면 큐 관리자를 시작해야 합니다.

23 페이지의 [『큐 관리자 중지』](#)

큐 관리자를 중지시키는 방법에는 정상 종료(Quiesced shutdown), 즉시 종료, 강제 종료(Preemptive shutdown)의 세 가지가 있습니다.

24 페이지의 [『큐 관리자 재시작』](#)

You can use the `strmqm` command to restart a queue manager, or, on IBM WebSphere MQ for Windows and IBM WebSphere MQ for Linux (x86 and x86-64 platforms) systems, restart a queue manager from IBM WebSphere MQ Explorer.

380 페이지의 [『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』](#)

설치 요구에 맞게 IBM WebSphere MQ의 작동 또는 개별 큐 관리자를 변경합니다.

[시스템 및 기본 오브젝트](#)

관련 태스크

[21 페이지의 『기존 큐 관리자를 기본값으로 만들기』](#)

기존 큐 관리자를 기본 큐 관리자로 만들 수 있습니다. 이 작업을 수행하는 방법은 사용하는 플랫폼에 따라 다릅니다.

[24 페이지의 『큐 관리자 삭제』](#)

WebSphere MQ Explorer를 사용하거나 `dltmqm` 명령을 사용하여 큐 관리자를 삭제할 수 있습니다.

distributed 기본 큐 관리자 작성

기본 큐 관리자는 애플리케이션이 MQCONN 호출에 큐 관리자 이름을 지정하지 않은 경우에 연결하는 큐 관리자입니다. 또한 큐 관리자 이름을 지정하지 않고 `runmqsc` 명령을 호출할 때 MQSC 명령을 처리하는 큐 관리자이기도 합니다. 큐 관리자를 작성하려면 IBM WebSphere MQ 제어 명령 `crtmqm`을 사용하십시오.

시작하기 전에

기본 큐 관리자를 작성하기 전에 [17 페이지의 『큐 관리자 작성 및 관리』](#)에 설명된 고려사항을 읽으십시오.

UNIX `crtmqm`을 사용하여 UNIX and Linux에 큐 관리자를 작성할 때, `mqm`이 소유한 빈 `/var/mqm/qmgrs/<qmgr>` 디렉토리가 있을 경우 이 디렉토리가 큐 관리자 데이터에 사용됩니다. 이 디렉토리를 `mqm`이 소유하고 있지 않은 경우에는 First Failure Support Technology(FFST) 메시지가 표시되면서 큐 관리자 작성이 실패합니다. 디렉토리가 비어 있지 않은 경우 큐 관리자 데이터에 사용할 새 디렉토리가 작성됩니다.

이 고려사항은 별도의 로컬 파일 시스템에 `/var/mqm/qmgrs/<qmgr>` 디렉토리가 이미 있는 경우에도 적용됩니다.

이 태스크 정보

`crtmqm` 명령을 사용하여 큐 관리자를 작성하면 명령이 자동으로 필수 기본 오브젝트 및 시스템 오브젝트를 작성합니다. 기본 오브젝트는 사용자가 작성하는 오브젝트 정의의 기반이 되고, 시스템 오브젝트는 큐 관리자 조작에 필요합니다.

관련 매개변수를 명령에 포함시켜 큐 관리자가 사용할 기본 전송 큐의 이름 및 데드-레터 큐의 이름 등을 정의할 수도 있습니다.

Windows Windows에서 `crtmqm` 명령의 `sax` 옵션을 사용하여 큐 관리자의 다중 인스턴스를 시작할 수 있습니다.

`crtmqm` 명령 및 해당 구문에 대한 자세한 정보는 `crtmqm`의 내용을 참조하십시오.

프로시저

- 기본 큐 관리자를 작성하려면 `crtmqm` 명령을 `-q` 플래그와 함께 사용하십시오.
`crtmqm` 명령의 다음 예에서는 SATURN.QUEUE.MANAGER라는 기본 큐 관리자를 작성합니다.

```
crtmqm -q -d MY.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE SATURN.QUEUE.MANAGER
```

설명:

-q
이 큐 관리자가 기본 큐 관리자임을 나타냅니다.

-d MY.DEFAULT.XMIT.QUEUE
이 큐 관리자가 사용할 기본 전송 큐의 이름입니다.

참고: IBM WebSphere MQ는 기본 전송 큐를 자동으로 작성하지 않습니다. 사용자가 직접 정의해야 합니다.

-u SYSTEM.DEAD.LETTER.QUEUE

IBM WebSphere MQ가 설치에 작성하는 기본 데드-레터 큐의 이름입니다.

SATURN.QUEUE.MANAGER

큐 관리자의 이름입니다. 이 이름은 `crtmqm` 명령에서 지정되는 마지막 매개변수여야 합니다.

다음에 수행할 작업

큐 관리자 및 해당 오브젝트를 작성한 경우 `strmqm` 명령을 사용하여 [큐 관리자를 시작](#)하십시오.

관련 개념

22 페이지의 『[큐 관리자 작성 후 구성 파일 백업](#)』

IBM WebSphere MQ 구성 정보는 Windows, UNIX and Linux 시스템의 구성 파일에 저장되어 있습니다.

[큐 관리자에 대한 작업](#)

[로컬 큐에 대한 작업](#)

관련 참조

[시스템 및 기본 오브젝트](#)

기존 큐 관리자를 기본값으로 만들기

기존 큐 관리자를 기본 큐 관리자로 만들 수 있습니다. 이 작업을 수행하는 방법은 사용하는 플랫폼에 따라 다릅니다.

WebSphere MQ for 윈도우 and WebSphere MQ for Linux (x86 and x86-64 platforms) systems

이 태스크 정보

Use the following instructions to make an existing queue manager the default queue manager on WebSphere MQ for 윈도우 and WebSphere MQ for Linux (x86 and x86-64 platforms) systems:

프로시저

1. IBM WebSphere MQ Explorer를 여십시오.
2. IBM WebSphere MQ을 마우스 오른쪽 단추로 클릭한 후 Properties...을 선택하십시오. WebSphere MQ의 특성 패널이 표시됩니다.
3. 기본 큐 관리자의 이름을 기본 큐 관리자 이름 필드에 입력하십시오.
4. 확인을 클릭하십시오.

UNIX and Linux 시스템

이 태스크 정보

기본 큐 관리자를 작성하면 해당 이름이 WebSphere MQ 구성 파일(mqs.ini)에 있는 DefaultQueueManager 스탠자의 Name 속성에 삽입됩니다. 스탠자 및 해당 콘텐츠는 존재하지 않을 경우 자동으로 작성됩니다.

프로시저

- 기존 큐 관리자를 기본 큐 관리자로 만들려면 Name 속성의 큐 관리자 이름을 새 기본 큐 관리자의 이름으로 변경하십시오. 이 작업은 텍스트 편집기를 사용하여 수동으로 수행할 수 있습니다.
- 노드에 기본 큐 관리자가 없고 기존 큐 관리자를 기본 큐 관리자로 만들려면 필요한 이름으로 *DefaultQueueManager* 스탠자를 작성하십시오.
- 실수로 다른 큐 관리자를 기본 큐 관리자로 만들어 원래 기본 큐 관리자로 복귀하려면 mqs.ini의 DefaultQueueManager 스탠자를 편집하여 원하지 않는 기본 큐 관리자를 원하는 큐 관리자로 바꾸십시오.

다음에 수행할 작업

구성 파일에 대한 정보는 380 페이지의 『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』의 내용을 참조하십시오.

큐 관리자 작성 후 구성 파일 백업

IBM WebSphere MQ 구성 정보는 Windows, UNIX and Linux 시스템의 구성 파일에 저장되어 있습니다.

Windows 및 Linux(x86 및 x86-64) 시스템에서는 IBM WebSphere MQ Explorer를 사용하여 구성 파일을 변경하십시오.

Windows 시스템의 경우에도 amqmdain 명령을 사용하여 구성 파일을 변경할 수 있습니다. [amqmdain](#)을 참조하십시오.

구성 파일은 다음과 같은 두 가지 유형이 있습니다.

- 제품을 설치할 때 IBM WebSphere MQ 구성 파일(mqs.ini)이 작성됩니다. 이 파일에는 큐 관리자를 작성 또는 삭제할 때마다 업데이트되는 큐 관리자 목록이 포함됩니다. 노드마다 하나의 mqs.ini 파일이 있습니다.
- 새 큐 관리자를 작성할 때 새 큐 관리자 구성 파일(qm.ini)이 자동으로 작성됩니다. 이 파일에는 큐 관리자에 대한 구성 매개변수가 포함됩니다.

큐 관리자를 작성한 후 구성 파일을 백업하십시오. 그런 다음 문제점을 초래하는 다른 큐 관리자를 작성하는 경우 문제점의 원인을 제거할 때 백업을 복원할 수 있습니다. 일반적으로 새 큐 관리자를 작성할 때마다 구성 파일을 백업합니다.

구성 파일에 대한 자세한 정보는 380 페이지의 『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』의 내용을 참조하십시오.

큐 관리자 시작

큐 관리자를 작성할 때 명령 또는 MQI 호출을 처리하려면 큐 관리자를 시작해야 합니다.

큐 관리자를 시작하려면 다음과 같이 **strmqm** 명령을 사용하십시오.

참고: **strmqm** 명령은 작업을 수행하는 큐 관리자와 연관된 설치에서 사용해야 합니다. **dspmqs -o installation** 명령을 사용하여 큐 관리자가 연관된 설치를 찾을 수 있습니다.

예를 들어, 큐 관리자 QMB를 시작하려면 다음 명령을 입력하십시오.

```
strmqm QMB
```

On WebSphere MQ for 윈도우 and WebSphere MQ for Linux (x86 and x86-64 platforms) systems, you can start a queue manager as follows:

1. IBM WebSphere MQ Explorer를 여십시오.
2. 네비게이터 보기에서 큐 관리자를 선택하십시오.
3. Start을(를) 클릭하십시오. 큐 관리자가 시작됩니다.

큐 관리자가 시작되는 데 몇 초 이상 걸리면 WebSphere MQ가 시작 진행 상태를 자세히 표시하는 정보 메시지를 간헐적으로 발행합니다.

strmqm 명령은 큐 관리자가 시작되어 연결 요청을 승인할 수 있을 때까지 제어를 리턴하지 않습니다.

큐 관리자 자동 시작

In WebSphere MQ for 윈도우 you can start a queue manager automatically when the system starts using the IBM WebSphere MQ Explorer. 자세한 정보는 [IBM WebSphere MQ Explorer를 사용한 관리](#)를 참조하십시오.

큐 관리자 중지

큐 관리자를 중지시키는 방법에는 정상 종료(Quiesced shutdown), 즉시 종료, 강제 종료(Preemptive shutdown)의 세 가지가 있습니다.

endmqm 명령을 사용하여 큐 관리자를 중지하십시오.

참고: 작업 중인 큐 관리자와 연관된 설치에서 **endmqm** 명령을 사용해야 합니다. `dspmqr -o installation` 명령을 사용하여 큐 관리자가 연관된 설치를 찾을 수 있습니다.

예를 들어, QMB라는 큐 관리자를 중지하려면 다음 명령을 입력하십시오.

```
endmqm QMB
```

On WebSphere MQ for 윈도우 and WebSphere MQ for Linux (x86 and x86-64 platforms) systems, you can stop a queue manager as follows:

1. IBM WebSphere MQ Explorer를 여십시오.
2. 네비게이터 보기에서 큐 관리자를 선택하십시오.
3. Stop...을(를) 클릭하십시오. 큐 관리자 종료 패널이 표시됩니다.
4. 제어됨 또는 즉시를 선택하십시오.
5. OK을(를) 클릭하십시오. 큐 관리자가 중지됩니다.

정상 종료(Quiesced shutdown)

기본적으로 **endmqm** 명령은 지정된 큐 관리자의 정상 종료(Quiesced shutdown)를 수행합니다. 이 명령을 완료하는 데는 약간의 시간이 걸릴 수 있습니다. 정상 종료(Quiesced shutdown)는 연결된 모든 애플리케이션이 끊어질 때까지 대기합니다.

애플리케이션이 중지하도록 알려려면 이 종료 유형을 사용하십시오. 다음을 실행하는 경우,

```
endmqm -c QMB
```

모든 애플리케이션이 중지되었을 때 이를 통보받지 않습니다. (`endmqm -c QMB` 명령은 `endmqm QMB` 명령과 동등합니다.)

그러나, 다음을 실행하는 경우,

```
endmqm -w QMB
```

명령은 모든 애플리케이션이 중지되고 큐 관리자가 종료될 때까지 대기합니다.

즉시 종료

즉시 종료의 경우, 현재 MQI 호출을 모두 완료할 수 있지만 새 호출은 실패합니다. 이 종료 유형은 애플리케이션이 큐 관리자로부터 연결이 끊어질 때까지 대기하지 않습니다.

즉시 종료의 경우, 다음을 입력하십시오.

```
endmqm -i QMB
```

강제 종료(preemptive shutdown)

참고: **endmqm** 명령을 사용하여 큐 관리자를 중지하려는 다른 시도가 모두 실패할 때까지 이 방법을 사용하지 마십시오. 이 방법은 연결된 애플리케이션에 대해 예측할 수 없는 결과를 초래할 수 있습니다.

즉시 종료 작동하지 않을 경우, `-p` 플래그를 지정하여 강제 종료(Preemptive shutdown)를 사용해야 합니다. 예를 들면, 다음과 같습니다.

```
endmqm -p QMB
```

큐 관리자가 즉시 중지됩니다. 이 방법도 작동하지 않는 경우 대체 솔루션은 [큐 관리자 수동 중지](#)를 참조하십시오.

endmqm 명령 및 해당 옵션에 대한 세부 설명은 [endmqm](#)을 참조하십시오.

큐 관리자를 종료하는 데 문제점이 있는 경우

큐 관리자 종료 문제점은 애플리케이션에 그 원인이 있는 경우가 많습니다. 예를 들어, 애플리케이션이 다음과 같은 경우,

- MQI 리턴 코드를 올바르게 검사하지 않는 경우
- 일시정지 알림을 요청하지 않는 경우
- 큐 관리자 연결을 끊지 않고 종료하는 경우(MQDISC 호출 발행)

큐 관리자를 중지할 때 문제점이 발생하면 Ctrl-C를 사용하여 **endmqm** 명령을 중단할 수 있습니다. 그런 다음 다른 **endmqm** 명령을 실행할 수 있지만, 이번에는 필요한 종료의 유형을 지정하는 플래그로 실행할 수 있습니다.

큐 관리자 재시작

You can use the **strmqm** command to restart a queue manager, or, on IBM WebSphere MQ for Windows and IBM WebSphere MQ for Linux (x86 and x86-64 platforms) systems, restart a queue manager from IBM WebSphere MQ Explorer.

큐 관리자를 재시작하려면 다음을 입력하십시오.

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

On IBM WebSphere MQ for Windows and IBM WebSphere MQ for Linux (x86 and x86-64 platforms) systems, you can restart a queue manager in the same way as starting it, as follows:

1. IBM WebSphere MQ Explorer를 여십시오.
2. 네비게이터 보기에서 큐 관리자를 선택하십시오.
3. Start을(를) 클릭하십시오. 큐 관리자가 재시작됩니다.

큐 관리자 재시작을 위해 몇 초 이상이 소요되는 경우 IBM WebSphere MQ는 시작 진행 상황을 나타내는 정보 메시지를 간헐적으로 실행합니다.

큐 관리자 삭제

WebSphere MQ Explorer를 사용하거나 **dltmqm** 명령을 사용하여 큐 관리자를 삭제할 수 있습니다.

시작하기 전에

큐 관리자를 중지합니다.

프로시저

- **dltmqm QMB** 명령을 실행하십시오.

참고: 작업 중인 큐 관리자와 연관된 설치에서 **dltmqm** 명령을 사용해야 합니다. **dspmq -o installation** 명령을 사용하여 큐 관리자가 연관된 설치를 찾을 수 있습니다.

큐 관리자 삭제 단계

이 태스크 정보

On WebSphere MQ for 윈도우 and WebSphere MQ for Linux (x86 and x86-64 platforms) systems, you can delete a queue manager as follows:

프로시저

1. WebSphere MQ Explorer를 여십시오.
2. 네비게이터 보기에서 큐 관리자를 선택하십시오.
3. 큐 관리자가 중지되지 않은 경우 중지하십시오.
 - a) 큐 관리자를 마우스 오른쪽 단추로 클릭하십시오.
 - b) 중지를 누르십시오.
4. 큐 관리자를 마우스 오른쪽 단추로 클릭하십시오.
5. 삭제를 클릭하십시오.

결과

큐 관리자가 삭제됩니다.



주의:

- 큐 관리자 삭제는 극단적인 조치입니다. 모든 큐와 해당 메시지 및 모든 오브젝트 정의를 포함하여 큐 관리자와 연관된 모든 자원이 삭제되기 때문입니다. **dltmqm** 명령을 사용하는 경우, 사용자의 마음을 변경할 수 있는 표시된 프롬프트가 없습니다. Enter키를 누르면 연관된 모든 자원이 손실됩니다.
- In WebSphere MQ for 윈도우, deleting a queue manager also removes the queue manager from the automatic startup list (described in 22 페이지의 『큐 관리자 시작』). 명령이 완료되면 WebSphere MQ queue manager ending 메시지가 표시됩니다. 큐 관리자가 삭제되었음을 알리지 않습니다.
- 클러스터 큐 관리자를 삭제해도 클러스터에서는 제거되지 않습니다. 자세한 정보는 **dltmqm** 설명에서 참고 사항을 참조하십시오.

dltmqm 명령과 해당 옵션에 대한 설명은 **dltmqm**을 참조하십시오. 이 명령은 신뢰할 수 있는 관리자만 사용할 수 있습니다. (보안에 대한 정보는 [Windows, UNIX and Linux 시스템에서 보안 설정을 참조하십시오.](#))

분산 큐잉을 사용하여 애플리케이션 연결

이 절에서는 큐 정의, 채널 정의, 트리거 및 동기점 프로시저를 포함한 WebSphere MQ 설치 간 상호통신에 관한 자세한 정보를 제공합니다.

이 절을 읽기 전에 채널, 큐 및 상호통신 개념에 소개된 기타 개념을 이해하는 것이 좋습니다.

다음 링크의 정보를 사용하여 분산 큐잉을 사용하는 애플리케이션을 연결하십시오.

- [46 페이지의 『다른 큐 관리자로 메시지를 보내는 방법』](#)
- [61 페이지의 『채널 트리거』](#)
- [59 페이지의 『메시지의 안전성』](#)
- [26 페이지의 『IBM WebSphere MQ 분산 메시징 기술』](#)
- [43 페이지의 『분산 큐 관리 소개』](#)
- [Windows](#) [UNIX](#) [Linux](#) [66 페이지의 『UNIX, Linux, and Windows에서 채널 모니터링 및 제어』](#)

관련 개념

[89 페이지의 『클라이언트와 서버 간의 연결 구성』](#)

WebSphere MQ MQI 클라이언트와 서버 사이의 통신 링크를 구성하려면 통신 프로토콜을 결정하고 링크의 양 끝에서의 연결을 정의하고 리스너를 시작한 다음 채널을 정의하십시오.

[380 페이지의 『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』](#)

설치 요구에 맞게 IBM WebSphere MQ의 작동 또는 개별 큐 관리자를 변경합니다.

관련 태스크

[144 페이지의 『큐 관리자 클러스터 구성』](#)

클러스터가 작업하는 방법, 클러스터 구성을 설계하는 방법을 찾고 단순한 클러스터를 설정하는 방법의 예를 알려면 이 토픽의 링크를 사용하십시오.

IBM WebSphere MQ 분산 메시징 기술

이 절의 하위 주제에서는 채널 계획 시 사용되는 기술을 설명합니다. 이 하위 주제에서는 큐 관리자와 함께 연결하는 방법을 계획하고 애플리케이션 간 메시지 플로우를 관리하는 데 도움이 되는 기술을 설명합니다.

메시지 채널 계획 예는 다음을 참조하십시오.

- [분산 플랫폼에 대한 메시지 채널 계획 예](#)

관련 개념

[25 페이지의 『분산 큐잉을 사용하여 애플리케이션 연결』](#)

이 절에서는 큐 정의, 채널 정의, 트리거 및 동기점 프로시저를 포함한 WebSphere MQ 설치 간 상호통신에 관한 자세한 정보를 제공합니다.

채널

[메시지 큐잉 소개](#)

[상호통신의 개념](#)

관련 참조

[구성 정보 예](#)

메시지 플로우 제어

메시지 플로우 제어는 큐 관리자 사이의 메시지 라우트 설정 및 유지보수를 처리하는 태스크로, 많은 큐 관리자를 통해 멀티호핑하는 라우트에 중요합니다. 이 절에서는 메시지 플로우를 제어하기 위해 시스템에서 큐, 알리어스 큐 정의, 메시지 채널을 사용하는 방법을 설명합니다.

[25 페이지의 『분산 큐잉을 사용하여 애플리케이션 연결』](#)에 소개되어 있는 여러 기술을 사용하여 메시지 플로우를 제어할 수 있습니다. 큐 관리자가 클러스터에 있으면 다른 기술을 사용하여 메시지 플로우가 제어됩니다([26 페이지의 『메시지 플로우 제어』](#)의 설명 참조).

메시지 플로우를 제어하기 위해 사용할 수 있는 오브젝트는 다음과 같습니다.

- 전송 큐
- 메시지 채널
- 리모트 큐 정의
- 큐 관리자 알리어스 정의
- 응답 대상 큐 알리어스 정의

큐 관리자 및 큐 오브젝트는 [오브젝트](#)에 설명되어 있습니다. 메시지 채널은 [분산 큐잉 컴포넌트](#)에 설명되어 있습니다. 다음 기술은 위의 오브젝트를 사용하여 시스템에서 메시지 플로우를 작성합니다.

- 리모트 큐에 메시지 넣기
- 특정 전송 큐를 통한 라우팅
- 메시지 수신
- 사용자 시스템을 통한 메시지 전달
- 메시지 플로우 분리
- 메시지 플로우를 다른 목적지로 전환
- 응답 대상 큐 이름을 알리어스 이름으로 해석

참고

이 절에서 설명하는 모든 개념은 네트워크의 모든 노드와 관련이 있으며 메시지 채널의 송신측과 수신측을 포함합니다. 따라서 대부분의 예에는 하나의 노드만 나와 있습니다. 예외는 메시지 채널의 다른 측에서 관리자의 명시적인 협업이 필요한 경우입니다.

개별 기술에 대한 내용으로 진행하기 전에 이름 분석의 개념과 리모트 큐 정의를 사용하는 세 가지 방법에 대한 기본 사항을 정리하는 것이 좋습니다. [상호통신의 개념](#)을 참조하십시오.

관련 개념

27 페이지의 『전송 헤더의 큐 이름』

목적지 큐 이름은 목적지 큐에 도달할 때까지 전송 헤더에서 메시지와 함께 이동합니다.

27 페이지의 『큐 관리자 및 응답 대상 알리어스 작성 방법』

이 주제는 리모트 큐 정의를 작성할 수 있는 세 가지 방법을 설명합니다.

전송 헤더의 큐 이름

목적지 큐 이름은 목적지 큐에 도달할 때까지 전송 헤더에서 메시지와 함께 이동합니다.

애플리케이션이 사용하는 큐 이름인 논리적 큐 이름은 큐 관리자가 목적지 큐 이름으로 해석합니다(즉, 물리적 큐 이름). 이 목적지 큐 이름은 목적지 큐에 도달할 때까지 개별 데이터 영역인 전송 헤더에서 메시지와 함께 이동합니다. 그런 다음 전송 헤더가 제거됩니다.

서비스의 병렬 클래스를 작성할 때 이 큐 이름의 큐 관리자 파트를 변경합니다. 서비스 클래스 전환 끝에 도달하면 큐 관리자 이름을 원래 이름으로 되돌려야 합니다.

큐 관리자 및 응답 대상 알리어스 작성 방법

이 주제는 리모트 큐 정의를 작성할 수 있는 세 가지 방법을 설명합니다.

리모트 큐 정의의 오브젝트는 세 가지 다른 방식으로 사용됩니다. 27 페이지의 표 3 다음 세 가지 방법을 정의하는 방법에 대해 설명합니다.

- 리모트 큐 정의를 사용하여 로컬 큐 이름 재정의.

애플리케이션이 큐를 열 때 큐 이름만 제공하며 이 큐 이름은 리모트 큐 정의의 이름입니다.

리모트 큐 정의에는 대상 큐와 큐 관리자의 이름이 포함됩니다. 선택적으로, 이 정의에는 사용될 전송 큐의 이름이 포함됩니다. 전송 큐 이름이 제공되지 않는 경우에는 큐 관리자가 전송 큐 이름에 대해 리모트 큐 정의에서 가져온 큐 관리자 이름을 사용합니다. 이 이름의 전송 큐가 정의되지 않지만 기본 전송 큐가 정의된 경우에는 기본 전송 큐를 사용합니다.

- 리모트 큐 정의를 사용하여 큐 관리자 이름 재정의.

애플리케이션 또는 채널 프로그램은 큐를 열 때 리모트 큐 관리자 이름과 함께 큐 이름을 제공합니다.

큐 관리자 이름과 동일한 이름을 리모트 큐 정의에 제공하고 정의의 큐 이름을 공백으로 남겨둔 경우에는 큐 관리자가 열린 호출의 큐 관리자 이름을 정의의 큐 관리자 이름으로 대체합니다.

또한 정의에는 사용될 전송 큐의 이름이 포함될 수 있습니다. 전송 큐 이름이 제공되지 않는 경우에는 큐 관리자가 전송 큐 이름에 대해 리모트 큐 정의에서 가져온 큐 관리자 이름을 사용합니다. 이 이름의 전송 큐가 정의되지 않지만 기본 전송 큐가 정의된 경우에는 기본 전송 큐를 사용합니다.

- 리모트 큐 정의를 사용하여 응답 대상 큐 이름 재정의.

애플리케이션이 큐에 메시지를 둘 때마다 응답 메시지에 대한 응답 대상 큐 이름을 제공할 수 있지만 큐 관리자 이름은 공백으로 둡니다.

응답 대상 큐와 동일한 이름을 가진 리모트 큐 정의를 제공하는 경우, 로컬 큐 관리자는 응답 대상 큐 이름을 정의의 큐 이름으로 바꿉니다.

정의에서 큐 관리자 이름은 제공할 수 있지만 전송 큐 이름은 제공할 수 없습니다.

표 3. 리모트 큐 정의의 오브젝트를 사용하는 세 가지 방법			
사용법	큐 관리자 이름	큐 이름	전송 큐 이름
1. 리모트 큐 정의(OPEN 호출 시)			
호출에 제공	공백 또는 로컬 QM	(*) 필수	-
정의에 제공	필수	필수	선택사항
2. 큐 관리자 알리어스(OPEN 호출 시)			
호출에 제공	(*) 필수 및 비로컬 QM	필수	-
정의에 제공	필수	공백	선택사항

표 3. 리모트 큐 정의 오브젝트를 사용하는 세 가지 방법 (계속)			
사용법	큐 관리자 이름	큐 이름	전송 큐 이름
3. 응답 대상 큐 알리어스(PUT 호출 시)			
호출에 제공	공백	(*) 필수	-
정의에 제공	선택사항	선택사항	공백
참고: (*)는 이 이름이 정의 오브젝트의 이름임을 의미			

정식 설명은 [큐 이름 해석](#)을 참조하십시오.

리모트 큐에 메시지 넣기

리모트 큐 정의 오브젝트를 사용하여 큐 이름을 인접 큐 관리자의 전송 큐로 해석할 수 있습니다.

분산 큐잉 환경에서 전송 큐와 채널은 메시지가 로컬 시스템에 있는 애플리케이션에서 수신되거나 또는 인접 시스템으로부터 채널을 통해 도착하거나 한 위치로의 모든 메시지에 대한 중심점입니다. 28 페이지의 [그림 2](#)는 'QA_norm' 이라는 논리적 큐에 메시지를 배치하는 애플리케이션을 보여준다. 이름 해석은 리모트 큐 정의 'QA_norm'을 사용하여 전송 큐 QMB를 선택합니다. 그런 다음 'QA_norm at QMB' 메시지에 전송 헤더를 추가합니다.

예를 들어, 'Channel_back'의 인접 시스템에서 도착하는 메시지의 전송 헤더에는 물리적 큐 이름 'QA_norm at QMB'가 포함됩니다. 이러한 메시지는 변경되지 않은 채로 전송 큐 QMB에 놓여집니다.

채널은 메시지를 인접 큐 관리자로 이동합니다.

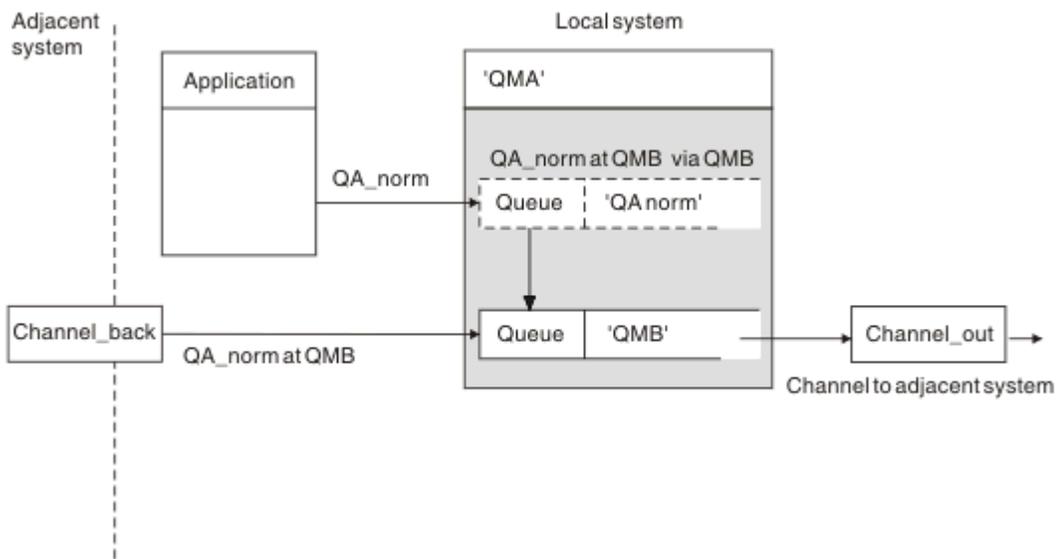


그림 2. 리모트 큐 정의는 큐 이름을 인접 큐 관리자의 전송 큐로 해석하는 데 사용됩니다.

WebSphere MQ 시스템 관리자는 다음을 수행해야 합니다.

- 인접 시스템으로부터의 메시지 채널 정의
- 인접 시스템으로의 메시지 채널 정의
- 전송 큐 QMB 작성
- 애플리케이션이 사용하는 큐 이름을 목적지 큐 이름, 목적지 큐 관리자 이름, 전송 큐 이름으로 해석하려면 리모트 큐 오브젝트 'QA_norm' 정의

클러스터링 환경의 경우, 로컬 큐 관리자에서만 클러스터 수신자 채널을 정의하면 됩니다. 전송 큐 또는 리모트 큐 오브젝트는 정의하지 않아도 됩니다. 자세한 정보는 [클러스터](#)를 참조하십시오.

이름 해석 정보

리모트 큐를 정의하면 물리적 목적지 큐 이름과 큐 관리자 이름이 정의됩니다. 이러한 이름은 메시지의 전송 헤더에 놓입니다.

인접 시스템으로부터의 수신 메시지는 원래 큐 관리자가 수행한 이름 분석 유형을 이미 갖고 있습니다. 따라서 전송 헤더에 물리적 목적지 큐 이름과 큐 관리자 이름이 표시됩니다. 이러한 메시지는 리모트 큐 정의의 영향을 받지 않습니다.

전송 큐 선택

리모트 큐 정의를 사용하여 다른 전송 큐에서 동일한 인접 큐 관리자로 메시지를 보내도록 허용할 수 있습니다.

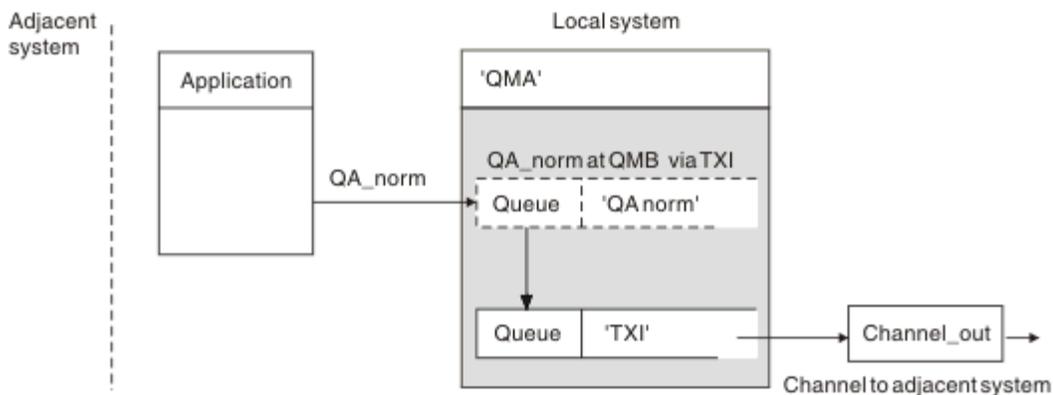


그림 3. 리모트 큐 정의에서는 다른 전송 큐를 사용할 수 있습니다.

분산 큐잉 환경에서 특정 채널에서 다른 채널로의 메시지 플로우를 변경해야 하는 경우, 28 페이지의 『리모트 큐에 메시지 넣기』의 28 페이지의 그림 2에 표시된 것과 동일한 시스템 구성을 사용하십시오. 이 주제의 29 페이지의 그림 3에서는 리모트 큐 정의를 사용하여 다른 전송 큐와 그에 따라 다른 채널을 통해 동일한 인접 큐 관리자로 메시지를 보내는 방법을 보여줍니다.

29 페이지의 그림 3에 표시된 구성의 경우, 리모트 큐 오브젝트 'QA_norm'과 전송 큐 'TX1'을 제공해야 합니다. 리모트 큐 관리자의 'QA_norm' 큐, 전송 큐 'TX1', 큐 관리자 'QMB_priority'를 선택하려면 'QA_norm'을 제공해야 합니다. 시스템에 인접한 채널의 정의에 'TX1'을 지정하십시오.

메시지는 전송 헤더에 'QA_norm at QMB_priority'가 포함되는 전송 큐 'TX1'에 배치되며 채널을 통해 인접 시스템으로 보내집니다.

이 예에서는 큐 관리자 알리어스가 필요하므로 channel_back이 제외되었습니다.

클러스터링 환경에서는 전송 큐 또는 리모트 큐 정의를 정의하지 않아도 됩니다. 추가 정보는 150 페이지의 『클러스터 큐』의 내용을 참조하십시오.

메시지 수신

큐 관리자가 다른 큐 관리자로부터 메시지를 수신하도록 구성할 수 있습니다. 의도하지 않은 이름 해석이 발생하지 않는지 확인해야 합니다.

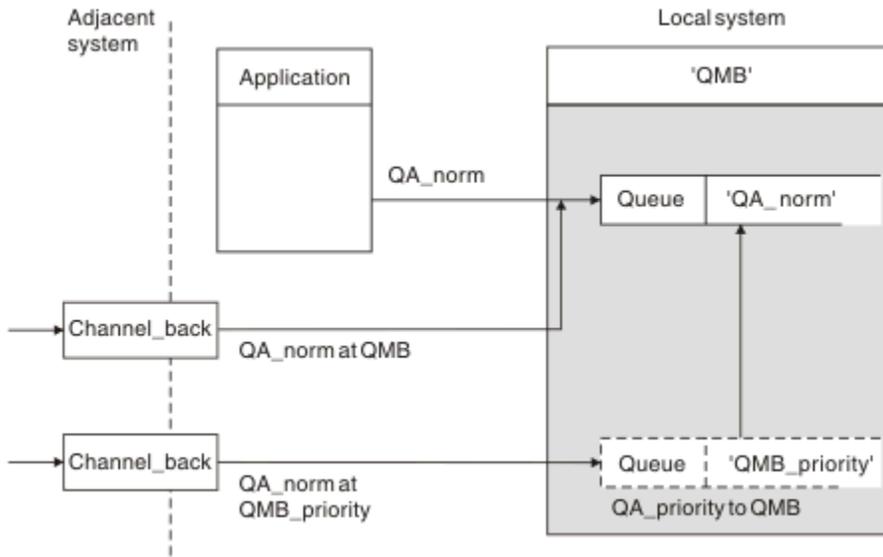


그림 4. 직접적인 메시지 수신과 알리어스 큐 관리자 이름 해석

시스템 관리자는 송신할 메시지에 대한 정렬은 물론, 인접 큐 관리자로부터 수신할 메시지도 정렬해야 합니다. 수신된 메시지의 전송 헤더에는 목적지 큐 관리자 및 큐의 물리적 이름이 포함됩니다. 이 메시지는 큐 관리자 이름과 큐 이름을 둘 다 지정하는 로컬 애플리케이션으로부터의 메시지와 동일한 것으로 간주됩니다. 따라서 시스템에 들어가는 메시지에 고의적이지 않은 이름 해석이 수행되지 않는지 확인해야 합니다. 이 시나리오에 대해서는 30 페이지의 그림 4의 내용을 참조하십시오.

이 구성의 경우 다음 준비가 필요합니다.

- 인접 큐 관리자로부터 메시지를 수신할 메시지 채널
- 수신 메시지 플로우, 'QMB_priority'를 로컬 큐 관리자 이름, 'QMB'로 해석하기 위한 큐 관리자 알리어스 정의
- 로컬 큐, 'QA_norm'(존재하지 않는 경우)

알리어스 큐 관리자 이름 수신

이 예에서 큐 관리자 알리어스 정의 사용 시 다른 목적지 큐 관리자를 선택하지 않았습니다. 이 로컬 큐 관리자를 통해 전달되고 'QMB_priority'에 주소 지정된 메시지는 큐 관리자 'QMB'용입니다. 알리어스 큐 관리자 이름은 별도의 메시지 플로우를 작성하는 데 사용됩니다.

사용자 시스템을 통한 메시지 전달

세 가지 방법(위치 이름 사용, 큐 관리자의 알리어스 사용 또는 전송 큐 선택)으로 사용자 시스템을 통해 메시지를 전달할 수 있습니다.

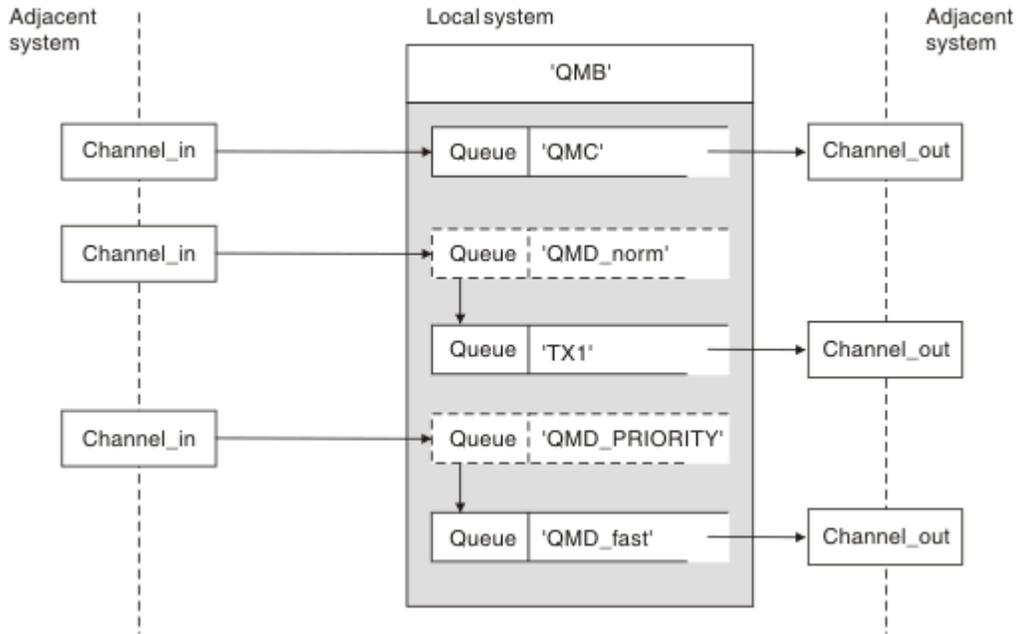


그림 5. 시스템을 통한 세 가지 메시지 전달 방법

29 페이지의 『메시지 수신』의 30 페이지의 그림 4 에 표시된 기술은 별명 플로우가 캡처되는 방법을 보여줍니다. 31 페이지의 그림 5 는 앞서 설명한 기술들을 함께 제공함으로써 네트워크들이 구축되는 방법들을 예시한다.

구성은 목적이 다른 다음 세 개 메시지를 전달하는 채널을 보여줍니다.

1. QMC의 QB
2. QMD_norm의 QB
3. QMD_PRIORITY의 QB

첫 번째 메시지 플로우는 변경되지 않은 시스템을 통해 전달해야 합니다. 두 번째 메시지 플로우는 다른 전송 큐와 채널을 통해 전달해야 합니다. 두 번째 메시지 플로우의 경우, 또한 알리어스 큐 관리자 이름 QMD_norm에 대한 메시지를 큐 관리자 QMD로 해석해야 합니다. 세 번째 메시지 플로우는 다른 변경사항 없이 다른 전송 큐를 선택합니다.

클러스터링 환경에서는 클러스터 전송 큐를 통해 메시지가 전달됩니다. 일반적으로 단일 전송 큐, SYSTEM.CLUSTER.TRANSMIT.QUEUE가 큐 관리자가 멤버인 모든 클러스터에 속하는 모든 큐 관리자로 모든 메시지를 전달합니다(큐 관리자 클러스터 참조). 큐 관리자가 멤버인 클러스터에 속하는 일부 또는 모든 큐 관리자에 개별 전송 큐를 정의할 수 있습니다.

다음 방법은 분산 큐잉 환경에 적용 가능한 기술을 설명합니다.

다음 방법 사용

이러한 구성의 경우 다음을 준비해야 합니다.

- 입력 채널 정의
- 출력 채널 정의
- 전송 큐:
 - QMC
 - TX1
 - QMD_fast
- 큐 관리자 알리어스 정의:
 - TX1을 통한 QMD_norm에서 QMD까지의 QMD_norm

- QMD_fast를 통한 QMD_PRIORITY에서 QMD_PRIORITY까지의 QMD_PRIORITY

참고: 이 예에 표시된 어떤 메시지 플로우도 목적지 큐를 변경하지 않습니다. 큐 관리자 이름 알리어스는 메시지 플로우를 분리하기 위해 제공됩니다.

방법 1: 수신 위치 이름 사용

다른 위치 이름(예: QMC)을 포함하는 전송 헤더로 메시지를 수신하려고 합니다. 가장 단순한 구성은 해당 이름, QMC를 사용하는 전송 큐를 작성하는 것입니다. 전송 큐를 지원하는 채널은 메시지를 변경 없이 다음 목적지로 전달합니다.

방법 2: 큐 관리자 알리어스 사용

두 번째 방법은 큐 관리자 알리어스 오브젝트 정의를 사용하되 새 위치 이름, QMD와 특정 전송 큐, TX1을 지정하는 것입니다. 이 조치는 다음과 같이 수행됩니다.

- 큐 관리자 이름 알리어스 QMD_norm으로 설정된 알리어스 메시지 플로우를 종료합니다(즉, 이름 지정된 서비스 클래스 QMD_norm).
- 이 메시지의 전송 헤더를 QMD_norm에서 QMD로 변경합니다.

방법 3: 전송 큐 선택

세 번째 방법은 목적지 위치, QMD_PRIORITY와 동일한 이름으로 큐 관리자 알리어스 오브젝트를 정의하는 것입니다. 특정 전송 큐, QMD_fast와 그에 따라 다른 채널을 선택하려면 큐 관리자 알리어스 정의를 사용하십시오. 이러한 메시지의 전송 헤더는 변경되지 않습니다.

메시지 플로우 분리

큐 관리자 알리어스를 사용하여 동일한 큐 관리자로 메시지를 보내는 별도 메시지 플로우를 작성할 수 있습니다.

분산 큐잉 환경에서 여러 가지 이유로 인해 동일한 큐 관리자에 대한 메시지를 서로 다른 메시지 플로우로 분리해야 할 수 있습니다. 예를 들면, 다음과 같습니다.

- 대형, 중형, 소형 메시지에 개별 플로우를 제공해야 할 수 있습니다. 이 요구 사항은 또한 클러스터링 환경에서 적용되며 이 경우 중첩되는 클러스터를 작성할 수 있습니다. 이렇게 해야 하는 이유는 다음과 같은 여러 가지가 있습니다.
 - 서로 다른 조직이 각자 고유의 관리를 할 수 있도록 하기 위해
 - 독립 애플리케이션을 개별적으로 관리할 수 있도록 하기 위해
 - 서비스 클래스를 작성하기 위해. 예를 들어, STUDENTS 클러스터의 서브세트인 STAFF 클러스터가 있을 수 있습니다. STAFF 클러스터에서 알려진 큐에 메시지를 넣을 때는 제한된 채널이 사용됩니다. STUDENTS 클러스터에서 알려진 큐에 메시지를 넣을 때는 일반 채널 또는 제한된 채널을 사용할 수 있습니다.
 - 테스트 환경 및 프로덕션 환경을 생성하기 위해
- 로컬로 생성된 메시지의 경로에서 다른 경로가 수신 메시지를 라우트해야 할 수 있습니다.
- 설치에서 특정 시점(예를 들어, 야간)에 메시지 이동을 스케줄링해야 할 수 있으며 이 경우 메시지는 스케줄링 될 때까지 예약된 큐에 저장되어야 합니다.

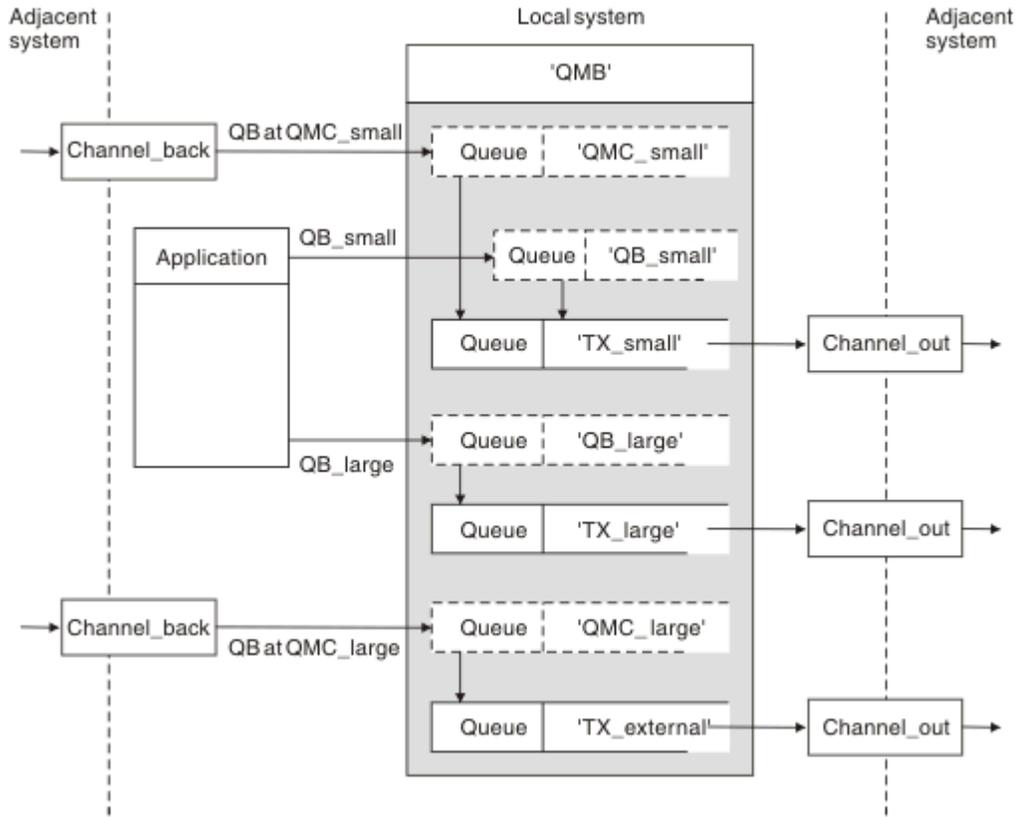


그림 6. 메시지 플로우 분리

33 페이지의 그림 6에 표시된 예에는 알리어스 큐 관리자 이름 'QMC_small' 및 'QMC_large'로의 두 개 수신 플로우가 있습니다. 이러한 플로우에는 로컬 큐 관리자를 위해 플로우를 캡처하기 위한 큐 관리자 알리어스 정의를 제공합니다. 두 개 리모트 큐에 대한 주소를 지정하는 애플리케이션이 있으며 이 메시지 플로우를 별도로 유지해야 합니다. 동일한 위치, 'QMC'를 지정하지만 다른 전송 큐를 지정하는 두 가지 리모트 큐 정의를 제공합니다. 이 정의는 플로우를 개별적으로 유지하므로 다른 한쪽에서 추가 요소가 필요하지 않습니다. 전송 헤더에 동일한 목적지 큐 관리자 이름이 있기 때문입니다. 다음을 제공합니다.

- 수신 채널 정의
- 두 개의 리모트 큐 정의(QB_small과 QB_large)
- 두 개의 큐 관리자 알리어스 정의(QMC_small과 QMC_large)
- 세 개의 송신 채널 정의
- 세 개의 전송 큐: TX_small, TX_large, TX_external

인접 시스템과의 조정

큐 관리자 알리어스를 사용하여 별도 메시지 플로우를 작성하는 경우, 메시지 채널의 원격 끝 시스템 관리자와 함께 이 활동을 조정하여 해당 큐 관리자 알리어스를 사용할 수 있도록 해야 합니다.

다양한 위치로 분산될 메시지 집중

다양한 위치를 목적으로 하는 메시지를 단일 채널에 집중시킬 수 있습니다.

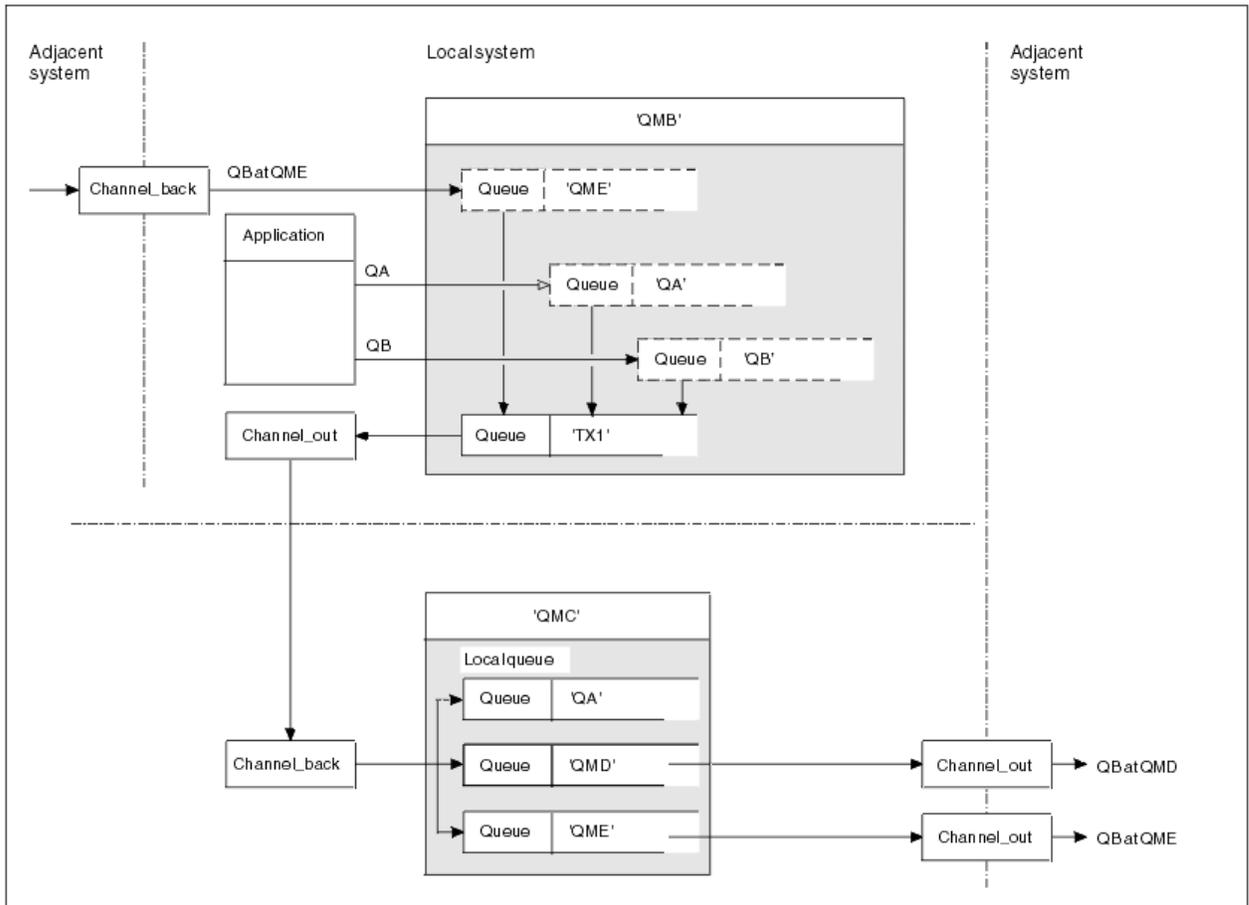


그림 7. 채널에 메시지 플로우 결합

34 페이지의 그림 7에서는 다양한 위치를 목적지로 하는 메시지를 단일 채널로 집중시키기 위한 분산 큐잉 기법을 보여줍니다. 두 가지 가능한 방법은 다음과 같습니다.

- 게이트웨이를 통해 메시지 트래픽 집중
- 노드 간 광대역 하이웨이 사용

이 예에서는 다른 목적지 큐와 큐 관리자가 다른 로컬 및 인접 소스의 메시지가 전송 큐 'TX1'을 통해 큐 관리자 QMC로 이동합니다. 큐 관리자 QMC는 목적지에 따라 메시지를 전달합니다. 한 메시지는 큐 관리자 QMD로 전송되는 전송 큐 'QMD'로 설정되고 또 다른 메시지는 큐 관리자 QME로 전송되는 전송 큐 'QME'로 설정됩니다. 다른 메시지는 로컬 큐 'QA'에 넣습니다.

다음은 제공해야 합니다.

- 채널 정의
- 전송 큐 TX1
- 리모트 큐 정의:
 - QA - 'TX1'을 통한 QMC의 QA'
 - QB - 'TX1'을 통한 QMD의 QB'
- 큐 관리자 알리어스 정의:
 - QME - 'TX1'을 통한 QME'

QMC를 구성하는 보조 관리자는 다음을 제공해야 합니다.

- 채널 이름이 같은 수신 채널 정의
- 연관된 전송 채널 정의가 있는 전송 큐 QMD
- 연관된 전송 채널 정의가 있는 전송 큐 QME

- 로컬 큐 오브젝트 QA

메시지 플로우를 다른 목적지로 전환

큐 관리자 알리어스와 전송 큐를 사용하여 특정 메시지의 목적지를 재정의할 수 있습니다.

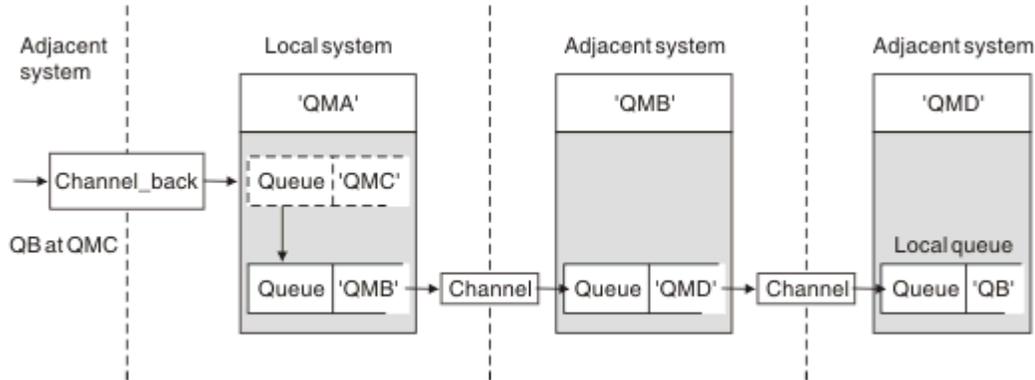


그림 8. 메시지 스트림을 다른 목적지로 전환

35 페이지의 그림 8에서는 특정 메시지의 목적지를 재정의할 수 있는 방법을 보여줍니다. QMA에 대한 수신 메시지는 'QMC의 QB'용으로 지정됩니다. 이러한 메시지는 일반적으로 QMA에 도달하며 QMC 채널의 일부인 전송 큐 QMC에 배치됩니다. QMA는 QMD로 메시지를 보내야 하지만 이 경우 QMB를 통해서만 QMD에 도달할 수 있습니다. 이 방법은 한 위치에서 다른 위치로 서비스를 이동해야 하고 메시지를 새 주소로 조정할 때까지 신청자에게 임시로 메시지 송신을 계속하도록 허용해야 할 때 유용합니다.

특정 큐 관리자를 수신지로 하여 들어오는 메시지를 다른 큐 관리자로 다시 라우팅하는 메소드는 다음이 사용됩니다.

- 목적지 큐 관리자를 다른 큐 관리자로 변경하고, 인접 시스템으로 향하는 전송 큐를 선택하기 위해 큐 관리자 알리어스 사용
- 인접 큐 관리자에게 서비스를 제공하기 위해 전송 큐 사용
- 목적지 큐 관리자로의 지속적인 라우팅을 위해 인접 큐 관리자에 있는 전송 큐 사용

다음을 제공해야 합니다.

- Channel_back 정의
- QMB를 통한 QMD에 있는 QB를 가진 큐 관리자 별칭 오브젝트 정의 QMC
- Channel_out 정의
- 연관된 전송 큐 QMB

QMB를 구성하는 보조 관리자는 다음을 제공해야 합니다.

- 해당 channel_back 정의
- 전송 큐, QMD
- QMD에 대한 관련 채널 정의

클러스터링 환경에서 알리어스를 사용할 수 있습니다. 관련 정보는 237 페이지의 『큐 관리자 알리어스 및 클러스터』의 내용을 참조하십시오.

분배 목록으로 메시지 송신

단일 MQPUT 호출을 사용하여 애플리케이션이 여러 대상으로 메시지를 보내도록 할 수 있습니다.

z/OS®를 제외한 모든 플랫폼의 WebSphere MQ 에서 애플리케이션은 단일 MQPUT 호출로 여러 대상에 메시지를 전송할 수 있습니다. 이 기능은 분산 큐잉 환경과 클러스터링 환경에서 모두 수행할 수 있습니다. 목적지는 분배 목록에 정의해야 합니다(분배 목록의 설명 참조).

모든 큐 관리자가 분배 목록을 지원하는 것은 아닙니다. MCA가 파트너와의 연결을 설정할 때 파트너가 분배 목록을 지원하는지 여부를 판별하며 그에 따라 전송 큐에 플래그를 설정합니다. 애플리케이션이 분배 목록을 목적지로 하는 메시지를 보내려고 시도하지만 파트너가 분배 목록을 지원하지 않는 경우에는 송신 MCA가 메시지를 가로채 정해진 각 목적지에 맞는 전송 큐에 메시지를 한번 넣습니다.

수신 MCA는 분배 목록으로 보낸 메시지가 지정된 모든 목적지에서 안전하게 수신되는지 확인합니다. 목적지가 실패하면 MCA가 실패한 목적지를 설정합니다. 그런 다음 해당 목적지에 대한 예외 보고서를 생성하고 다시 메시지를 보내도록 시도할 수 있습니다.

응답 대상 큐

응답 대상 큐를 사용하여 전체 리모트 큐 처리 루프를 작성할 수 있습니다.

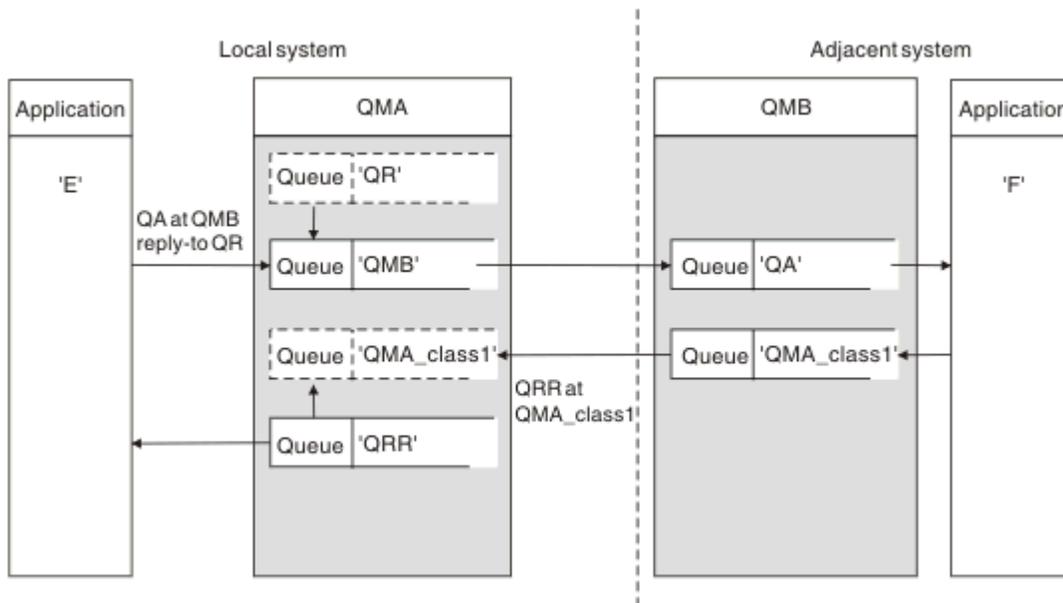


그림 9. PUT 호출 시 응답 대상 큐 이름 대체

응답 대상 큐를 사용하는 전체 리모트 큐 처리 루프가 36 페이지의 그림 9에 나와 있습니다. 이 루프는 분산 큐잉 환경과 클러스터링 환경에서 모두 적용됩니다. 자세한 내용은 42 페이지의 표 7에 나와 있습니다.

애플리케이션은 QMB에서 QA를 열고 해당 큐에 메시지를 넣습니다. 큐 관리자 이름이 지정되지 않고 응답 대상 큐 이름 QR이 메시지에 제공됩니다. 큐 관리자 QMA는 응답 대상 큐 오브젝트 QR을 찾고, 여기서 알리어스 이름 QRR과 큐 관리자 이름 QMA_class1을 추출합니다. 이러한 이름은 메시지의 응답 대상 필드에 들어갑니다.

QMB의 애플리케이션에서 들어온 응답 메시지는 QMA_class1의 QRR로 지정됩니다. 큐 관리자는 큐 관리자 알리어스 이름 정의의 QMA_class1을 사용하여 메시지를 자신 또는 큐 QRR로 이동합니다.

이 시나리오는 응답 메시지에 대한 서비스 클래스를 선택하는 기능을 애플리케이션에 제공합니다. 이 클래스는 QMA의 큐 관리자 알리어스 정의, QMA_class1과 함께 QMB의 전송 큐 QMA_class1로 구현됩니다. 이 방식으로, 애플리케이션과 관계 없이 플로우가 분리될 수 있도록 애플리케이션의 응답 대상 큐를 변경할 수 있습니다. 애플리케이션은 이 특정 서비스 클래스에 항상 QR을 선택합니다. 응답 대상 큐 정의의 QR로 서비스 클래스를 변경할 수 있습니다.

다음을 작성해야 합니다.

- 응답 대상 큐 정의 QR
- 전송 큐 오브젝트 QMB
- Channel_out 정의
- Channel_back 정의
- 큐 관리자 알리어스 정의 QMA_class1
- 로컬 큐 오브젝트가 존재하지 않을 경우, 로컬 큐 오브젝트 QRR

인접 시스템의 보조 관리자는 다음을 작성해야 합니다.

- 수신 채널 정의
- 전송 큐 오브젝트 QMA_class1
- 연관된 송신 채널
- 로컬 큐 오브젝트 QA

애플리케이션은 프로그램은 다음을 사용합니다.

- 넣기 호출에서 응답 대상 큐 이름 QR
- 가져오기 호출에서 큐 이름 QRR

이 방식을 사용하면 애플리케이션과 관계 없이 필요에 따라 서비스 클래스를 변경할 수 있습니다. 전송 큐 'QMA_class1' 및 큐 관리자 알리어스 'QMA_class1'과 함께 응답 대상 알리어스 'QR'을 변경할 수 있습니다.

큐에 메시지를 넣을 때 응답 대상 알리어스 오브젝트가 없으면 비어 있는 응답 대상 큐 관리자 이름 필드에 로컬 큐 관리자 이름이 삽입됩니다. 응답 대상 큐 이름은 변경되지 않습니다.

이름 해석 제한사항

이름 해석은 원래 메시지를 넣을 때 'QMA'의 응답 대상 큐에 대해 수행되므로 'QMB'에서는 더 이상 이름 해석이 허용되지 않습니다. 응답 애플리케이션이 응답 대상 큐의 실제 이름으로 메시지를 넣습니다.

애플리케이션은 응답 대상 큐에 사용하는 이름이 리턴 메시지를 갖게 될 실제 큐의 이름과 다르다는 사실을 알아야 합니다.

예를 들어, 응답 대상 큐 알리어스 이름이 'C1_alias' 및 'C2_alias'인 애플리케이션을 사용하기 위해 두 가지 서비스 클래스가 제공되는 경우 애플리케이션은 해당 이름을 메시지 넣기 호출에서 응답 대상 큐 이름으로 사용합니다. 그러나 실제로는 애플리케이션이 큐 'C1'('C1_alias'의 경우) 및 'C2'('C2_alias'의 경우)에 메시지가 나타나는 것으로 예상합니다.

그러나 애플리케이션은 응답 대상 알리어스 큐에서 조회 호출을 하여 응답 메시지를 가져오는 데 사용해야 하는 실제 큐의 이름을 직접 검사할 수 있습니다.

관련 개념

[27 페이지의 『큐 관리자 및 응답 대상 알리어스 작성 방법』](#)

이 주제는 리모트 큐 정의를 작성할 수 있는 세 가지 방법을 설명합니다.

[37 페이지의 『응답 대상 큐 알리어스 예』](#)

이 예는 응답 대상 큐 알리어스를 사용하여 리턴된 메시지에 서로 다른 라우트(전송 큐)를 선택하는 경우를 보여줍니다. 이 기능을 사용하려면 애플리케이션과 함께 응답 대상 큐 이름이 변경되어야 합니다.

[39 페이지의 『이 예의 작동 방법』](#)

큐 관리자가 응답 대상 큐 알리어스를 사용하는 방법과 그 예에 대한 설명입니다.

[40 페이지의 『응답 대상 큐 알리어스 개요』](#)

리모트 큐에 메시지를 넣는 애플리케이션에서 알리어스 응답 대상 큐에서 응답 메시지를 제거하는 동일한 애플리케이션까지의 프로세스 개요

응답 대상 큐 알리어스 예

이 예는 응답 대상 큐 알리어스를 사용하여 리턴된 메시지에 서로 다른 라우트(전송 큐)를 선택하는 경우를 보여줍니다. 이 기능을 사용하려면 애플리케이션과 함께 응답 대상 큐 이름이 변경되어야 합니다.

[38 페이지의 그림 10](#)와 같이, 전송 큐, 채널, 큐 관리자 알리어스를 포함하여 응답 메시지에 리턴 라우트를 사용할 수 있어야 합니다.

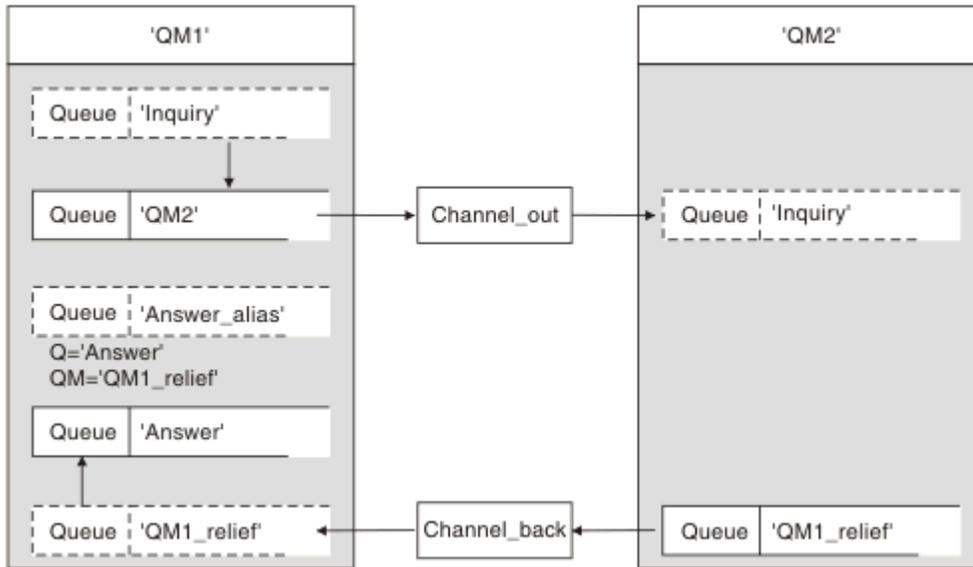


그림 10. 응답 대상 큐 알리어스 예

이 예는 'QM2'의 서버 애플리케이션으로 메시지를 보내는 'QM1'의 요청자 애플리케이션에 대한 것입니다. 서버의 메시지는 전송 큐 'QM1_relief'를 사용하여 대체 채널을 통해 리턴됩니다. 기본 리턴 채널은 전송 큐 'QM1'과 함께 제공됩니다.

응답 대상 큐 알리어스는 리모트 큐 정의 'Answer_alias'의 특정 용도입니다. QM1의 애플리케이션은 'Inquiry' 큐에 넣는 모든 메시지의 응답 대상 필드에 이 이름, 'Answer_alias'를 포함합니다.

응답 대상 큐 정의 'Answer_alias'는 'Answer at QM1_relief'로 정의됩니다. QM1의 애플리케이션은 해당 응답이 로컬 큐 'Answer'에 나타나는 것으로 예상합니다.

QM2의 서버 애플리케이션은 수신된 메시지의 응답 대상 필드를 사용하여 QM1의 요청자의 응답 메시지에 대한 큐 및 큐 관리자 이름을 획득합니다.

이 예에서 QM1에 사용된 정의

QM1의 WebSphere 시스템 MQ 시스템 관리자는 다른 오브젝트와 함께 응답 대상 큐 'Answer'가 작성되도록 해야 합니다. '*' 표시가 있는 큐 관리자 알리어스의 이름은 또한 '*' 표시가 있는 응답 대상 큐 알리어스 정의의 큐 관리자 이름과 일치해야 합니다.

오브젝트	정의	
로컬 전송 큐	QM2	
리모트 큐 정의	오브젝트 이름	조회
	리모트 큐 관리자 이름	QM2
	리모트 큐 이름	조회
	전송 큐 이름	QM2 (DEFAULT)
큐 관리자 알리어스	오브젝트 이름	QM1_relief *
	큐 관리자 이름	QM1
	큐 이름	(공백)
응답 대상 큐 알리어스	오브젝트 이름	Answer_alias
	리모트 큐 관리자 이름	QM1_relief *
	리모트 큐 이름	Answer

QM1에 정의 지정

애플리케이션은 응답 대상 필드를 응답 대상 큐 알리어스 이름으로 채우고 큐 관리자 이름 필드는 공백으로 둡니다.

필드	컨텐츠
큐 이름	조회
큐 관리자 이름	(공백)
응답 대상 큐 이름	Answer_alias
응답 대상 큐 관리자	(공백)

이 예에서 QM2에 사용된 정의

QM2의 WebSphere MQ 시스템 관리자는 들어오는 메시지에 대해 로컬 큐가 존재하는지, 그리고 응답 메시지에 대해 제대로 이름 지정된 전송 큐를 사용할 수 있는지를 확인해야 합니다.

오브젝트	정의
로컬 큐	조회
전송 큐	QM1_relief

QM2에 정의 지정

QM2의 애플리케이션은 원래 메시지에서 응답 대상 큐 이름과 큐 관리자 이름을 검색하고 응답 대상 큐에 응답 메시지를 둘 때 해당 이름을 사용합니다.

필드	컨텐츠
큐 이름	Answer
큐 관리자 이름	QM1_relief

이 예의 작동 방법

큐 관리자가 응답 대상 큐 알리어스를 사용하는 방법과 그 예에 대한 설명입니다.

이 예에서는 QM1의 요청자 애플리케이션이 항상 'Answer_alias'를 넣기 호출의 관련 필드에서 응답 대상 큐로 사용합니다. 애플리케이션은 항상 'Answer' 큐에서 메시지를 검색합니다.

응답 대상 큐 알리어스 정의는 QM1 시스템 관리자가 응답 대상 큐 'Answer'와 응답 라우트 'QM1_relief'의 이름을 변경하는 데 사용할 수 있습니다.

QM1 애플리케이션은 이 큐에서 응답을 예상하므로 큐 이름 'Answer'를 변경하는 것은 일반적으로 유용하지 않습니다. 그러나 QM1 시스템 관리자는 필요에 따라 리턴 라우트(서비스 클래스)를 변경할 수 있습니다.

큐 관리자가 응답 대상 큐 알리어스를 사용하는 방식

애플리케이션의 넣기 호출에 포함된 응답 대상 큐 이름이 응답 대상 큐 알리어스와 동일하고 큐 관리자 파트가 공백인 경우 큐 관리자 QM1이 응답 대상 큐 알리어스에서 정의를 검색합니다.

큐 관리자는 넣기 호출의 응답 대상 큐 이름을 정의의 큐 이름으로 바꿉니다. 또한 넣기 호출의 공백 큐 관리자 이름을 정의의 큐 관리자 이름으로 바꿉니다.

이러한 이름은 메시지 디스크립터에서 메시지와 함께 이동합니다.

표 4. 응답 대상 큐 알리어스		
필드 이름	넣기(put) 호출	전송 헤더
응답 대상 큐 이름	Answer_alias	Answer
응답 대상 큐 관리자 이름	(공백)	QM1_relief

응답 대상 큐 알리어스 개요

리모트 큐에 메시지를 넣는 애플리케이션에서 알리어스 응답 대상 큐에서 응답 메시지를 제거하는 동일한 애플리케이션까지의 프로세스 개요

이 예를 완료하려면 프로세스를 확인하십시오.

1. 애플리케이션은 'Inquiry' 큐를 열고 메시지를 큐에 넣습니다. 애플리케이션은 메시지 디스크립터의 응답 대상 큐를 다음으로 설정합니다.

응답 대상 큐 이름

응답 대상 큐 관리자 이름

Answer_alias

(공백)

2. 이름 'Answer_alias'로 리모트 큐 정의를 확인하여 큐 관리자 'QM1'이 공백 큐 관리자 이름에 응답합니다. 해당 항목이 없으면 큐 관리자가 자체 이름 'QM1'을 메시지 디스크립터의 응답 대상 큐 관리자 필드에 넣습니다.
3. 큐 관리자가 이름 'Answer_alias'로 리모트 큐 정의를 찾으면 정의에서 큐 이름과 큐 관리자 이름을 추출합니다(큐 이름 = 'Answer', 큐 관리자 이름 = 'QM1_relief'). 그런 다음 해당 이름을 메시지 디스크립터의 응답 대상 필드에 넣습니다.
4. 큐 관리자 'QM1'은 리모트 큐 정의 '조회'를 사용하여 의도한 목적지 큐가 큐 관리자 'QM2'에 있고 메시지가 전송 큐 'QM2'에 배치되었는지 판별합니다. 'QM2'는 큐 관리자 'QM2'에서 큐로 예정된 메시지의 기본 전송 큐 이름입니다.
5. 큐 관리자 'QM1'이 전송 큐에 메시지를 넣으면 메시지에 전송 헤더를 추가합니다. 이 헤더에는 목적지 큐, 'Inquiry'의 이름과 목적지 큐 관리자, 'QM2'가 포함됩니다.
6. 메시지가 큐 관리자 'QM2'에 도착하고 'Inquiry' 로컬 큐에 배치됩니다.
7. 애플리케이션이 이 큐에서 메시지를 가져와 처리합니다. 애플리케이션은 응답 메시지를 준비하고 해당 응답 메시지를 원래 메시지의 메시지 디스크립터에서 응답 대상 큐에 넣습니다.

응답 대상 큐 이름

응답 대상 큐 관리자 이름

Answer

QM1_relief

8. 큐 관리자 'QM2'가 넣기 명령을 수행합니다. 큐 관리자 이름, 'QM1_relief'가 리모트 큐 관리자임을 발견한 후 동일한 이름 'QM1_relief'로 전송 큐에 메시지를 배치합니다. 목적지 큐 이름, 'Answer'와 목적지 큐 관리자, 'QM1_relief'를 포함하는 전송 헤더가 메시지에 제공됩니다.
9. 메시지는 큐 관리자 'QM1'로 전송됩니다. 큐 관리자가 큐 관리자 이름 'QM1_relief'가 알리어스임을 인식하고 알리어스 정의 'QM1_relief'에서 물리적 큐 관리자 이름 'QM1'을 추출합니다.
10. 그런 다음 큐 관리자 'QM1'이 전송 헤더에 포함된 큐 이름, 'Answer'에 메시지를 넣습니다.
11. 애플리케이션은 큐 'Answer'에서 응답 메시지를 추출합니다.

네트워킹 고려사항

분산 큐잉 환경에서는 큐 이름과 큐 관리자 이름만으로 메시지 목적지의 주소를 지정하므로 특정 규칙이 적용됩니다.

1. 큐 관리자 이름이 지정되는 경우 해당 이름은 로컬 큐 관리자 이름과 다릅니다.
 - 이름이 같은 전송 큐를 사용할 수 있어야 합니다. 이 전송 큐는 다른 큐 관리자로 메시지를 이동하는 메시지 채널의 일부여야 합니다. 또는
 - 큐 관리자 이름을 동일하거나 다른 큐 관리자 이름과 선택적 전송 큐로 해석하기 위해서는 큐 관리자 알리어스 정의가 존재해야 합니다. 또는
 - 전송 큐 이름을 해석할 수 없고 기본 전송 큐가 정의된 경우에는 기본 전송 큐를 사용합니다.
2. 큐 이름만 제공된 경우 동일한 이름을 사용하는 임의 유형의 큐를 로컬 큐 관리자에서 사용할 수 있어야 합니다. 이 큐는 인접 큐 관리자로의 전송 큐, 큐 관리자 이름, 선택적 전송 큐로 해석되는 리모트 큐 정의일 수 있습니다.

클러스터 환경에서 작동하는 방법을 알려면, 제품 문서의 [클러스터 작동 방법](#) 절의 해당 토픽을 참조하십시오.

분산 큐잉 환경 내 특정 큐 관리자에서 다른 큐 관리자로 메시지를 이동하는 메시지 채널 시나리오를 고려하십시오.

이동하는 메시지는 네트워크의 다른 큐 관리자에서 송신된 것이며 알 수 없는 큐 관리자 이름을 목적지로 갖는 일부 메시지가 도착할 수 있습니다. 이 문제는 예를 들어, 큐 관리자 이름이 변경되었거나 시스템에서 제거된 경우 발생할 수 있습니다.

채널 프로그램은 해당 메시지의 전송 큐를 찾을 수 없을 때 이 상황을 인식하고 미전달 메시지(데드-레터) 큐에 메시지를 놓습니다. 해당 메시지를 찾아 올바른 목적지로 전달되도록 조정해야 합니다. 또는 진원지를 알 수 있는 경우 진원지로 돌려보냅니다.

원래 메시지에서 보고 메시지가 요청된 경우에는 이러한 상황에서 예외 보고서가 생성됩니다.

이름 해석 규칙

목적지 큐의 ID를 변경(즉, 논리적 이름을 물리적 이름으로 변경)하는 이름 해석은 원래 큐 관리자에서 한 번만 발생합니다.

다양한 알리어스 가능성의 후속 사용은 메시지 플로우를 분리하고 결합할 때만 수행됩니다.

리턴 라우팅

메시지에는 큐 및 큐 관리자 이름 양식의 리턴 주소가 포함될 수 있습니다. 이 리턴 주소 양식은 분산 큐잉 환경과 클러스터링 환경에서 모두 사용할 수 있습니다.

이 주소는 일반적으로 메시지를 작성하는 애플리케이션으로 지정됩니다. 이 주소는 사용자 엑시트 애플리케이션을 포함하여 메시지를 처리하는 애플리케이션으로 수정될 수 있습니다.

이 주소의 소스에 관계 없이, 메시지를 처리하는 애플리케이션이 원래 애플리케이션으로 응답, 상태 또는 보고서 메시지를 리턴하기 위해 이 주소를 사용하도록 선택할 수 있습니다.

이러한 응답 메시지의 경로 방식은 원래의 메시지 경로 방식과 동일합니다. 다른 큐 관리자에 작성하는 메시지 플로우에 해당 리턴 플로우가 필요함에 유의하십시오.

물리적 이름 충돌

이 목적지 응답 대상 큐 이름은 원래 큐 관리자에서 물리적 큐 이름으로 해석되었습니다. 이 이름은 응답 큐 관리자에서 다시 해석되어서는 안 됩니다.

물리적 큐 이름과 논리적 큐 이름에 대한 네트워크 범위 동의로만 방지될 수 있는 이름 충돌 문제점이 존재할 수 있습니다.

큐 이름 변환 관리

큐 관리자 알리어스 정의 또는 리모트 큐 정의를 작성하는 경우, 해당 이름을 포함하는 모든 메시지에 대해 이름 분석이 수행됩니다. 이 상황은 관리가 필요합니다.

이 설명은 인접 시스템에 대한 메시지 채널이 있는 개별 시스템에 관심이 있는 채널 기획자와 애플리케이션 설계자에게 제공됩니다. 채널 기획과 제어에 대한 로컬 관점을 갖습니다.

큐 관리자 알리어스 정의 또는 리모트 큐 정의를 작성하면, 메시지의 소스에 관계 없이 해당 이름을 사용하는 모든 메시지에 대해 이름 해석이 수행됩니다. 큐 관리자 네트워크에 여러 개의 큐가 있는 이러한 상황을 감독하기 위해 다음 내용의 테이블을 유지해야 합니다.

- 해석된 큐 이름, 해석된 큐 관리자 이름, 그리고 해석된 전송 큐 이름과 관련하여 소스 큐 및 소스 큐 관리자의 이름과 그 해석 방법
- 다음과 관련된 소스 큐의 이름
 - 해석된 목적지 큐 이름
 - 해석된 목적지 큐 관리자 이름
 - 전송 큐
 - 메시지 채널 이름
 - 인접 시스템 이름

- 응답 대상 큐 이름

참고: 이 컨텍스트에서 소스라는 용어의 사용은 애플리케이션이 제공하는 큐 이름이나 큐 관리자 이름 또는 메시지를 넣기 위해 큐를 열 때의 채널 프로그램을 의미합니다.

이들 표에 대한 각각의 예는 42 페이지의 표 5, 42 페이지의 표 6 및 42 페이지의 표 7에 나타나 있습니다.

이들 표의 이름은 이 절의 예에서 도출된 것이며 이 표는 단일 노드의 실제 큐 이름 분석 예로 작성되지 않았습니다.

큐를 열 때 지정된 소스 큐	큐를 열 때 지정된 소스 큐 관리자	해석된 큐 이름	해석된 큐 관리자 이름	해석된 전송 큐 이름	해석 유형
QA_norm	-	QA_norm	QMB	QMB	리모트 큐
(임의)	QMB	-	-	QMB	(없음)
QA_norm	-	QA_norm	QMB	TX1	리모트 큐
QB	QMC	QB	QMD	QMB	큐 관리자 알리어스

큐를 열 때 지정된 소스 큐	큐를 열 때 지정된 소스 큐 관리자	해석된 큐 이름	해석된 큐 관리자 이름	해석된 전송 큐 이름	해석 유형
QA_norm	-	QA_norm	QMB	-	(없음)
QA_norm	QMB	QA_norm	QMB	-	(없음)
QA_norm	QMB_PRIORITY	QA_norm	QMB	-	큐 관리자 알리어스
(임의)	QMC	(임의)	QMC	QMC	(없음)
(임의)	QMD_norm	(임의)	QMD_norm	TX1	큐 관리자 알리어스
(임의)	QMD_PRIORITY	(임의)	QMD_PRIORITY	QMD_fast	큐 관리자 알리어스
(임의)	QMC_small	(임의)	QMC_small	TX_small	큐 관리자 알리어스
(임의)	QMC_large	(임의)	QMC_large	TX_external	큐 관리자 알리어스
QB_small	QMC	QB_small	QMC	TX_small	리모트 큐
QB_large	QMC	QB_large	QMC	TX_large	리모트 큐
(임의)	QME	(임의)	QME	TX1	큐 관리자 알리어스
QA	QMC	QA	QMC	TX1	리모트 큐
QB	QMD	QB	QMD	TX1	리모트 큐

애플리케이션 설계		응답 대상 알리어스 정의	
로컬 QMGR	메시지의 큐 이름	응답 대상 큐 알리어스 이름	다음으로 재정의
QMA	QRR	QR	QMA_class1의 QRR

채널 메시지 순서 매기기

채널이 순서 번호를 사용함으로써 메시지가 중복되지 않고 전달되며, 전송 큐에서 수신된 동일한 순서대로 저장됩니다.

순서 번호는 채널의 송신측에서 생성되며, 사용 전에 1씩 증가됩니다. 이는 현재 순서 번호가 마지막으로 송신된 메시지의 번호임을 의미합니다. 이 정보는 DISPLAY CHSTATUS를 사용하여 표시할 수 있습니다(MQSC 참조 참조). 순서 번호와 ID, LUWID는 배치에서 전송된 마지막 메시지에 대한 영구 스토리지에 저장됩니다. 채널 시작 시 이 값을 사용함으로써 링크 양측이 메시지가 성공적으로 전송되었음을 확인할 수 있습니다.

순차적 메시지 검색

애플리케이션이 일련의 메시지를 동일한 목적지 큐에 넣는 경우, 다음 조건이 만족되면 이들 메시지는 MQGET 조작의 순서대로 **단일** 애플리케이션에 의해 검색될 수 있습니다.

- 모든 넣기 요청이 동일한 애플리케이션에서 수행되었습니다.
- 모든 넣기 요청이 동일한 작업 단위에서 시작되었거나 모든 넣기 요청이 작업 단위 밖에서 작성되었습니다.
- 모든 메시지는 우선순위가 같습니다.
- 메시지의 지속성이 모두 같습니다.
- 리모트 큐잉의 경우, 넣기 요청을 작성하는 애플리케이션부터 해당 큐 관리자와 상호통신을 통해 목적지 큐 관리자 및 대상 큐까지 한 가지 경로만 존재하도록 구성됩니다.
- 데드-레터 큐에는 메시지를 넣지 않습니다(예를 들어, 큐가 일시적으로 가득 찬 경우).
- 메시지를 가져오는 애플리케이션은 특정 *MsgId* 또는 *CorrelId*를 지정하거나, 메시지 우선순위를 사용하는 등의 방법을 사용하여 고의로 검색의 순서를 변경하지 않습니다.
- 목적지 큐에서 메시지를 검색하는 가져오기 조작은 하나의 애플리케이션만 수행합니다. 애플리케이션이 여러 개인 경우, 해당 애플리케이션은 송신 애플리케이션이 각 순서의 모든 메시지를 가져오도록 설계되어야 합니다.

참고: 다른 태스크와 작업 단위로부터의 메시지는 단일 작업 단위 내에서 시퀀스를 넣은 경우라도 시퀀스와 함께 배치될 수 있습니다.

이러한 조건이 충족되지 않고 대상 큐의 메시지 순서가 중요한 경우에는 자체 메시지 순서 번호를 메시지의 일부로 사용하도록 애플리케이션을 코드화하여 메시지 순서를 유지할 수 있습니다.

빠른, 비지속 메시지의 검색 순서

빠른 채널의 비지속 메시지는 동일한 채널의 지속 메시지를 추월하지 못해 순서가 유지되지 않을 수 있습니다. 수신 MCA는 목적지 큐에 즉시 비지속 메시지를 넣어 표시 가능하게 만듭니다. 지속 메시지는 다음 동기점까지 표시되지 않습니다.

루프백 테스트

루프백 테스트는 실제로 다른 시스템에 링크하지 않고 통신 링크를 테스트할 수 있는 비 z/OS 플랫폼의 기술입니다.

두 큐 관리자가 개별 시스템에 있는 것처럼 연결을 설정하지만 동일한 시스템의 다른 프로세스로 루프백하여 연결을 테스트합니다. 이 기술은 활성 네트워크 없이 통신 코드를 테스트할 수 있음을 의미합니다.

이 방법은 사용하는 제품과 프로토콜에 따라 다릅니다.

Windows 시스템에서는 "루프백" 어댑터를 사용할 수 있습니다.

자세한 정보는 사용하고 있는 제품 설명서를 참조하십시오.

라우트 추적 및 활동 레코드

메시지 라우트가 두 가지 방식으로 일련의 큐 관리자를 통해 수행되는지 확인할 수 있습니다.

WebSphere MQ 디스플레이 라우트 애플리케이션을 사용하여 제어 명령 dspmqrte를 통해 사용 가능하거나 활동 기록을 사용할 수 있습니다. 이 두 주제는 모두 [모니터링 참조](#)에서 설명됩니다.

분산 큐 관리 소개

분산 큐 관리(DQM)는 큐 관리자 간 통신을 정의하고 제어하는 데 사용됩니다.

분산 큐 관리:

- 큐 관리자 간 통신 채널을 정의하고 제어할 수 있습니다.
- 메시지 채널 서비스를 제공하여 전송 큐로 알려진 로컬 큐의 유형에서 로컬 시스템의 통신 링크로, 통신 링크에서 목적지 큐 관리자의 로컬 큐로 메시지를 이동합니다.
- 패널, 명령, 프로그램을 사용하여 채널 작업을 모니터링하고 문제점을 진단하는 기능을 제공합니다.

채널 정의는 채널 이름을 전송 큐, 통신 링크 ID, 채널 속성과 연관시킵니다. 다른 플랫폼에서는 채널 정의가 다른 방식으로 구현됩니다. 메시지 송신 및 수신은 메시지 채널 에이전트(MCA)로 알려진 프로그램에서 제어되며, 이 프로그램은 채널 정의를 사용하여 통신을 시작하고 제어합니다.

MCA는 차례로 DQM이 직접 제어합니다. 구조는 플랫폼 의존적이지만 일반적으로 운영자 명령 및 패널과 함께 리스너와 트리거 모니터를 포함합니다.

메시지 채널은 특정 큐 관리자에서 다른 큐 관리자로 메시지를 이동시키는 단방향 파이프입니다. 따라서 메시지 채널은 MCA 쌍으로 표시되는 두 개의 엔드 포인트를 갖습니다. 각 엔드 포인트에는 메시지 채널 끝에 대한 정의가 있습니다. 예를 들어, 한쪽 끝은 송신자를 정의하고 다른 측은 수신자를 정의합니다.

채널을 정의하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

-  66 페이지의 『UNIX, Linux, and Windows에서 채널 모니터링 및 제어』

메시지 채널 계획 예는 다음을 참조하십시오.

-  분산 플랫폼에 대한 메시지 채널 계획 예

채널 엑시트에 대한 정보는 [메시지 채널을 위한 채널 엑시트 프로그램](#)을 참조하십시오.

관련 개념

44 페이지의 『메시지 송신 및 수신』

다음 그림은 메시지가 전송될 때 엔티티 간의 관계를 나타내는 분산 큐 관리 모델을 보여줍니다. 또한 제어를 위한 플로우를 보여줍니다.

49 페이지의 『채널 제어 기능』

채널 제어 기능은 채널을 정의, 모니터, 제어할 수 있는 기능을 제공합니다.

60 페이지의 『메시지를 전달할 수 없을 경우의 메시지 처리』

메시지를 전달할 수 없는 경우 MCA가 여러 가지 방법으로 메시지를 처리할 수 있습니다. 다시 시도하거나 송신자로 돌아가거나 데드-레터 큐에 넣을 수 있습니다.

63 페이지의 『초기화 및 구성 파일』

채널 초기화 데이터 핸들링 양식은 사용자의 WebSphere MQ 플랫폼에 따라 달라집니다.

64 페이지의 『메시지에 대한 데이터 변환』

다른 큐 관리자의 큐 사이에서 송신할 때 WebSphere MQ 메시지에서 데이터 변환이 필요할 수 있습니다.

64 페이지의 『사용자 고유의 메시지 채널 에이전트 작성』

WebSphere MQ를 사용하여 고유의 MCA(Message Channel Agent) 프로그램을 쓰거나 독립 소프트웨어 벤더에서 하나를 설치할 수 있습니다.

65 페이지의 『분산 큐 관리를 위해 고려해야 할 기타 사항』

분산 큐 관리를 위한 WebSphere MQ 준비 시 고려할 다른 주제. 이 주제에서는 미전달 메시지 큐, 사용 중인 큐, 시스템 확장 기능, 사용자 엑시트 프로그램과 실행 중인 채널 및 리스너를 신뢰성 있는 애플리케이션으로 다룹니다.

관련 참조

[구성 정보 예](#)

메시지 송신 및 수신

다음 그림은 메시지가 전송될 때 엔티티 간의 관계를 나타내는 분산 큐 관리 모델을 보여줍니다. 또한 제어를 위한 플로우를 보여줍니다.

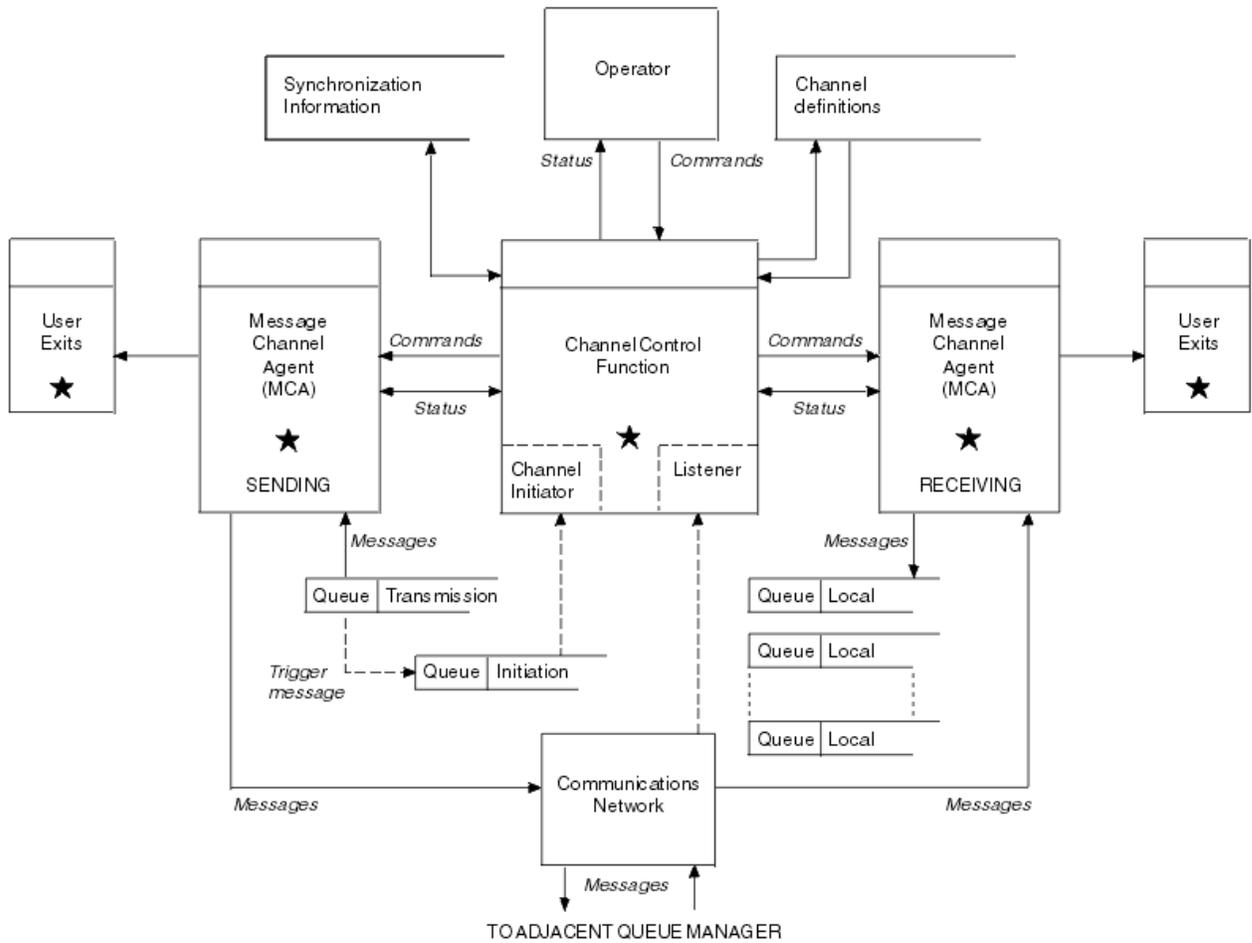


그림 11. 분산 큐 관리 모델

참고:

1. 플랫폼에 따라 채널당 MCA가 하나 있습니다. 특정 큐 관리자의 경우 하나 이상의 채널 제어 기능이 존재할 수 있습니다.
2. MCA 및 채널 제어 기능 구현은 플랫폼에 따라 크게 다릅니다. 이는 프로그램이나 프로세스 또는 스레드일 수 있으므로 여러 독립 또는 링크된 파트로 구성되는 단일 또는 다수의 엔티티가 될 수 있습니다.
3. 별표로 표시된 모든 구성요소는 MQI를 사용할 수 있습니다.

채널 매개변수

MCA는 다음 몇 가지 방법 중 하나로 매개변수를 수신합니다.

- 명령으로 시작되면, 채널 이름이 데이터 영역으로 전달됩니다. 그런 다음 MCA가 직접 채널 정의를 읽어 속성을 확보합니다.
- 송신자의 경우, 또한 몇몇 상황에서는 서버 채널의 경우, 큐 관리자 트리거에 의해 MCA가 자동으로 시작될 수 있습니다. 채널 이름은 사용 가능한 경우 트리거 프로세스 정의로부터 검색되며, MCA로 전달됩니다. 나머지 처리는 이전에 설명한 것과 같습니다. 서버 채널은 완전히 규정된 경우, 즉 연결될 CONNAME을 지정하는 경우에만 트리거되도록 설정되어야 합니다.
- 송신자, 서버, 요청자 또는 클라이언트 연결이 원격으로 시작된 경우, 채널 이름은 파트너 메시지 채널 에이전트로부터 초기 데이터로 전달됩니다. MCA는 그 속성을 확보하기 위해 채널 정의를 직접 읽습니다.

또한 채널 정의에 정의되어 있지 않은 특정 속성도 조정 가능합니다.

메시지 분할

한쪽이 분할 메시지를 지원하지 않으면 분할 메시지를 보내지 않습니다.

변환 기능

한쪽에서 필요한 코드 페이지 변환이나 숫자 인코딩 변환을 필요할 때 수행하지 못하면 다른 쪽이 이를 처리해야 합니다. 필요할 때 어느 한쪽에서도 지원하지 않으면 채널을 시작할 수 없습니다.

분배 목록 지원

한쪽에서 분배 목록을 지원하지 않으면 파트너 MCA가 알 수 있도록 전송 큐에서 플래그를 설정하여 여러 목적지를 대상으로 하는 메시지를 가로챍니다.

채널 상태 및 순서 번호

메시지 채널 에이전트 프로그램은 현재 순서 번호와 각 채널의 논리 작업 단위 그리고 채널의 일반 상태 레코드를 보관합니다. 일부 플랫폼은 이 상태 정보를 표시하여 패널 제어에 도움을 줍니다.

다른 큐 관리자로 메시지를 보내는 방법

이 절에서는 전제조건, 필요한 권한 등 큐 관리자 간에 메시지를 보내는 가장 간단한 방법을 설명합니다. 리모트 큐 관리자로 메시지를 보내기 위해 다른 방법을 사용할 수도 있습니다.

특정 큐 관리자에서 다른 큐 관리자로 메시지를 보내려면 다음 단계를 수행해야 합니다.

1. 선택한 통신 프로토콜을 사용할 수 있는지 확인하십시오.
2. 큐 관리자를 시작하십시오.
3. 채널 시작기를 시작하십시오.
4. 리스너를 시작하십시오.

또한 필요한 오브젝트를 작성하려면 올바른 WebSphere MQ 보안 권한도 있어야 합니다.

특정 큐 관리자에서 다른 관리자로 메시지를 보내려면 다음을 수행하십시오.

- 소스 큐 관리자에서 다음 오브젝트를 정의하십시오.
 - 송신자 채널
 - 리모트 큐 정의
 - 이니시에이션 큐(선택적)
 - 전송 큐
 - 데드-레터 큐
- 대상 큐 관리자에서 다음 오브젝트를 정의하십시오.
 - 수신자 채널
 - 대상 큐
 - 데드-레터 큐

사용자의 WebSphere MQ 플랫폼에 따라 몇 가지 다른 방법을 사용하여 이러한 오브젝트를 정의할 수 있습니다.

- 모든 플랫폼에서 [MQSC 명령](#)에 설명된 WebSphere MQ 스크립트 명령 (MQSC) 을 사용하여 [관리 태스크 자동화](#) 또는 WebSphere MQ 탐색기에 설명된 PCF (programmable command format) 명령을 사용할 수 있습니다.

다른 큐 관리자로 메시지를 보내기 위한 컴포넌트 작성에 대한 자세한 정보는 다음 하위 주제를 참조하십시오.

관련 개념

[17 페이지의 『큐 관리자 작성 및 관리』](#)

메시지 및 큐를 사용하려면 먼저 최소한 하나의 큐 관리자 및 연관된 오브젝트를 작성하고 시작해야 합니다.

[26 페이지의 『IBM WebSphere MQ 분산 메시징 기술』](#)

이 절의 하위 주제에서는 채널 계획 시 사용되는 기술을 설명합니다. 이 하위 주제에서는 큐 관리자와 함께 연결하는 방법을 계획하고 애플리케이션 간 메시지 플로우를 관리하는 데 도움이 되는 기술을 설명합니다.

[43 페이지의 『분산 큐 관리 소개』](#)

분산 큐 관리(DQM)는 큐 관리자 간 통신을 정의하고 제어하는 데 사용됩니다.

[61 페이지의 『채널 트리거』](#)

WebSphere MQ는 큐의 특정 조건이 충족되면 자동으로 애플리케이션을 시작하는 기능을 제공합니다. 이 기능을 트리거라고 합니다.

59 페이지의 『메시지의 안전성』

WebSphere MQ의 일반 복구 기능 외에 분산 큐 관리는 메시지 채널의 두 끝 간에 통합된 동기점 프로시저를 사용하여 메시지가 적절하게 전달되었는지를 확인합니다. 이 프로시저가 오류를 감지하면, 채널이 닫혀 문제를 조사하고 채널이 다시 시작될 때까지 메시지를 전송 큐에 안전하게 유지할 수 있습니다.

66 페이지의 『UNIX, Linux, and Windows에서 채널 모니터링 및 제어』

DQM의 경우 리모트 큐 관리자에 대한 채널을 작성, 모니터, 제어해야 합니다. 명령, 프로그램, IBM WebSphere MQ Explorer, 채널 정의 파일, 동기화 정보 스토리지 영역을 사용하여 채널을 제어할 수 있습니다.

89 페이지의 『클라이언트와 서버 간의 연결 구성』

WebSphere MQ MQI 클라이언트와 서버 사이의 통신 링크를 구성하려면 통신 프로토콜을 결정하고 링크의 양 끝에서의 연결을 정의하고 리스너를 시작한 다음 채널을 정의하십시오.

관련 태스크

144 페이지의 『큐 관리자 클러스터 구성』

클러스터가 작업하는 방법, 클러스터 구성을 설계하는 방법을 찾고 단순한 클러스터를 설정하는 방법의 예를 알려면 이 토픽의 링크를 사용하십시오.

채널 정의

특정 큐 관리자에서 다른 큐 관리자로 메시지를 보내려면 두 개 채널을 정의해야 합니다. 소스 큐 관리자에서 한 채널을 정의하고 대상 큐 관리자에 한 채널을 정의해야 합니다.

소스 큐 관리자에서

SENDER 유형의 채널을 정의하십시오. 다음을 지정해야 합니다.

- 사용될 전송 큐의 이름(XMITQ 속성)
- 파트너 시스템의 연결 이름(CONNAME 속성)
- 사용 중인 통신 프로토콜의 이름(TRPTYPE 속성). On WebSphere MQ for z/OS, the protocol must be TCP or LU6.2. 다른 플랫폼에서는 이를 지정하지 않아도 됩니다. 그대로 두어 기본 채널 정의에서 값을 선택할 수 있습니다.

모든 채널 속성에 대한 세부사항은 [채널 속성](#)에 제공됩니다.

대상 큐 관리자에서

채널 유형이 RECEIVER이고 송신자 채널과 이름이 같은 채널을 정의하십시오.

사용 중인 통신 프로토콜의 이름을 지정하십시오(TRPTYPE 속성). On WebSphere MQ for z/OS, the protocol must be TCP or LU6.2. 다른 플랫폼에서는 이를 지정하지 않아도 됩니다. 그대로 두어 기본 채널 정의에서 값을 선택할 수 있습니다.

수신자 채널 정의는 일반일 수 있습니다. 이는 동일 수신자와 통신하는 여러 큐 관리자가 있는 경우, 송신 채널이 수신자에 동일한 이름을 모두 지정할 수 있으며 하나의 수신자 정의가 모두에 적용된다는 것을 의미합니다.

참고: TRPTYPE 매개변수의 값은 응답 메시지 채널 에이전트에서 무시됩니다. 예를 들어, 송신자 채널 정의에 있는 TCP의 TRPTYPE은 수신자 채널 정의에 있는 LU62의 TRPTYPE을 파트너로 사용하여 시작됩니다.

채널을 정의하면 PING CHANNEL 명령을 사용하여 테스트할 수 있습니다. 이 명령은 송신자 채널에서 수신자 채널로 특별한 메시지를 보내고 리턴되는지 확인합니다.

큐 정의

특정 큐 관리자에서 다른 큐 관리자로 메시지를 송신하려면 최대 여섯 개의 큐를 정의해야 합니다. 소스 큐 관리자에서 최대 네 개의 큐, 대상 큐 관리자에서 최대 두 개의 큐를 정의해야 합니다.

소스 큐 관리자에서

- 리모트 큐 정의

이 정의에서는 다음을 지정하십시오.

리모트 큐 관리자 이름

대상 큐 관리자의 이름

리모트 큐 이름

대상 큐 관리자의 대상 큐 이름

전송 큐 이름

전송 큐의 이름. 이 전송 큐 이름은 지정하지 않아도 됩니다. 이 이름을 지정하지 않으면 대상 큐 관리자와 이름이 같은 전송 큐를 사용합니다. 이 큐가 없으면 기본 전송 큐를 사용합니다. 기본적으로 큐가 표시되도록 전송 큐에 대상 큐 관리자와 동일한 이름을 지정하는 것이 좋습니다.

- 이니시에이션 큐 정의

z/OS에서는 필수이고 기타 플랫폼에서는 선택적입니다. 이니시에이션 큐 이름으로 SYSTEM.CHANNEL.INITQ를 지정할 수 있습니다. 다른 플랫폼에서는

- 전송 큐 정의

USAGE 속성이 XMITQ로 설정된 로컬 큐.

- 데드-레터 큐 정의

미전달 메시지를 쓸 수 있는 데드-레터 큐를 정의하십시오.

대상 큐 관리자에서

- 로컬 큐 정의

대상 큐. 이 큐의 이름은 소스 큐 관리자에 있는 리모트 큐 정의의 리모트 큐 이름 필드에 지정된 것과 같아야 합니다.

- 데드-레터 큐 정의

미전달 메시지를 쓸 수 있는 데드-레터 큐를 정의하십시오.

관련 개념

48 페이지의 『전송 큐 작성』

채널(요청자 채널 이외)을 시작하려면 이 절의 설명대로 전송 큐가 정의되어야 합니다. 전송 큐는 채널 정의에 이름이 지정되어야 합니다.

전송 큐 작성

채널(요청자 채널 이외)을 시작하려면 이 절의 설명대로 전송 큐가 정의되어야 합니다. 전송 큐는 채널 정의에 이름이 지정되어야 합니다.

각 송신 메시지 채널에 대해 USAGE 속성을 XMITQ로 설정하여 로컬 큐를 정의합니다. 리모트 큐 정의에서 특정 전송 큐를 사용하려면 표시된 대로 리모트 큐를 작성하십시오.

전송 큐를 작성하려면 다음 예제에서와 같이 WebSphere MQ Commands(MQSC)를 사용하십시오.

전송 큐 작성 예

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') USAGE(XMITQ)
```

리모트 큐 작성 예

```
DEFINE QREMOTE(PAYROLL) DESCR('Remote queue for QM2') +  
XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

예제에서와 같이 전송 큐를 원격 시스템의 큐 관리자 이름으로 지정해 보십시오.

채널 시작

소스 큐 관리자에 정의되어 있는 리모트 큐에 메시지를 넣으면 채널을 시작할 때까지 전송 큐에 저장됩니다. 채널이 시작되면 메시지가 리모트 큐 관리자의 대상 큐로 전달됩니다.

START CHANNEL 명령을 사용하여 송신 큐 관리자에서 채널을 시작하십시오. 송신 채널을 시작하면, 수신 채널이 자동으로(리스너에 의해) 시작되며 메시지가 대상 큐로 송신됩니다. 메시지를 전송하려면 메시지 채널의 양측이 실행 중이어야 합니다.

채널의 양측이 서로 다른 큐 관리자에 있으므로 서로 다른 속성으로 정의되어 있을 수 있습니다. 차이점을 해결하기 위해, 채널이 시작할 때 양쪽 사이에 초기 데이터 조정이 이루어집니다. 일반적으로 채널의 양측은 보다 적은 자원이 필요한 속성으로 작동합니다. 따라서 보다 큰 시스템이 메시지 채널의 다른 한쪽에서 작은 시스템의 적은 자원을 수용할 수 있습니다.

송신 MCA는 채널을 통해 대형 메시지를 송신하기 전에 이를 작게 분할합니다. 그리고 리모트 큐 관리자에서 이 메시지를 다시 조합합니다. 이는 사용자에게 표시되지 않습니다.

MCA는 다중 스레드를 사용하여 메시지를 전송할 수 있습니다. 파이프라이닝이라고 하는 이 프로세스는 MCA가 더 적게 대기하면서 더 효율적으로 메시지를 전송할 수 있도록 합니다. 파이프라이닝은 채널 성능을 향상시킵니다.

채널 제어 기능

채널 제어 기능은 채널을 정의, 모니터, 제어할 수 있는 기능을 제공합니다.

명령은 패널이나 프로그램을 통해 또는 명령행에서 채널 제어 기능으로 실행됩니다. 패널 인터페이스에는 또한 채널 상태와 채널 정의 데이터가 표시됩니다. PCF(Programmable Command Format) 또는 MQSC(WebSphere MQ 명령) 및 66 페이지의 『UNIX, Linux, and Windows에서 채널 모니터링 및 제어』에서 자세히 설명된 제어 명령을 사용할 수 있습니다.

명령은 다음 그룹으로 분류됩니다.

- 채널 관리
- 채널 제어
- 채널 상태 모니터링

채널 관리 명령은 채널 정의를 처리합니다. 이 명령은 다음 기능을 수행합니다.

- 채널 정의 작성
- 채널 정의 복사
- 채널 정의 대체
- 채널 정의 삭제

채널 제어 명령은 채널 조작을 관리합니다. 이 명령은 다음 기능을 수행합니다.

- 채널 시작
- 채널 중지
- 파트너와 재동기화(일부 구현에서)
- 메시지 순서 번호 재설정
- 메시지의 인다우트(in-doubt) 배치 해석
- ping, 채널에서 테스트 통신 송신

채널 모니터링은 다음과 같은 채널의 상태를 표시합니다.

- 현재 채널 설정
- 채널이 활성 또는 비활성인지 여부
- 채널이 동기화된 상태로 종료되었는지 여부

채널 정의, 제어, 모니터링에 대한 자세한 정보는 다음 하위 주제를 참조하십시오.

채널 준비

메시지 채널 또는 MQI 채널을 시작하려고 시도하기 전에 채널을 준비해야 합니다. 로컬 및 원격 채널 정의의 모든 속성이 올바르고 호환 가능한지 확인해야 합니다.

채널 속성은 채널 정의와 속성을 설명합니다.

명시적 채널 정의를 설정하더라도 채널이 시작될 때 수행된 채널 조정은 정의된 하나 이상의 다른 값을 대체할 수 있습니다. 이 동작은 정상적인 것으로 사용자에게 표시되지 않으며 호환되지 않는 정의가 함께 작동할 수 있도록 이 방식으로 배열되었습니다.

수신자 및 서버 연결 채널의 자동 정의

z/OS를 제외한 모든 플랫폼의 WebSphere MQ에서 적절한 채널 정의가 없으면, 자동 정의가 작동 가능화되어 있는 수신자 또는 서버 연결 채널에 대해서는 자동으로 정의가 작성됩니다. 정의는 다음을 사용하여 작성됩니다.

1. 적합한 모델 채널 정의, SYSTEM.AUTO.RECEIVER 또는 SYSTEM.AUTO.SVRCONN. 자동 정의를 위한 모델 채널 정의는 설명 필드("Auto-defined by" 다음에 49개 공백이 있음)를 제외하고는 시스템 기본값인 SYSTEM.DEF.RECEIVER, SYSTEM.DEF.SVRCONN과 동일합니다. 시스템 관리자는 제공된 모델 채널 정의의 일부를 변경할 수 있습니다.
2. 파트너 시스템에서 얻은 정보. 파트너의 값은 채널 이름과 순서 번호 줄바꿈 값에 사용됩니다.
3. 자동 정의로 작성된 값을 대체하는 데 사용할 수 있는 채널 엑시트 프로그램. [채널 자동 정의 엑시트 프로그램](#)을 참조하십시오.

그런 다음 설명을 확인하여 자동 정의 엑시트에 의해 대체되었는지 또는 모델 정의가 변경되었는지 여부를 판별합니다. 처음 44개 문자가 계속 "Auto-defined by"와 그 다음의 29개 공백인 경우에는 큐 관리자 이름이 추가됩니다. 마지막 20자가 계속 모두 공백이면 현지 시간과 날짜가 추가됩니다.

정의가 작성 및 저장되면 정의가 항상 존재한 것처럼 채널 시작이 진행됩니다. 배치 크기, 전송 크기, 메시지 크기는 파트너와 함께 조정됩니다.

다른 오브젝트 정의

메시지 채널을 시작하려면 해당 큐 관리자에서 양쪽이 모두 정의되어야 합니다(또는 자동 정의에 사용 가능). 지원되는 전송 큐는 송신측에서 큐 관리자에 정의되어야 합니다. 통신 링크가 정의되고 사용 가능해야 합니다. 리모트 큐 정의, 큐 관리자 알리어스 정의 및 응답 대상 큐 알리어스 정의와 같은 다른 WebSphere MQ 오브젝트를 준비해야 25 페이지의 『분산 큐잉을 사용하여 애플리케이션 연결』에 설명된 시나리오를 수행할 수 있습니다.

MQI 채널 정의에 대한 정보는 101 페이지의 『MQI 채널 정의』의 내용을 참조하십시오.

전송 큐당 복수 메시지 채널

전송 큐당 여러 채널을 정의할 수 있지만 한 번에 해당 채널 중 하나만 활성화될 수 있습니다. 트래픽 밸런싱 및 링크 실패 정정 조치를 위해 큐 관리자 간에 대체 라우트를 제공하는 경우 이 옵션을 고려하십시오. 전송 큐를 사용하려는 이전의 채널이 송신측에서 메시지 배치가 인다우트(in-doubt)인 채로 종료되면 다른 채널에서 전송 큐를 사용할 수 없습니다. 자세한 정보는 58 페이지의 『인다우트(in-doubt) 채널』의 내용을 참조하십시오.

채널 시작

채널을 사용하여 다음 네 가지 방법 중 하나로 메시지 전송을 시작할 수 있습니다. 채널은 다음과 같습니다.

- 운영자가 시작할 수 있습니다. 수신자, 클러스터 수신자 또는 서버 연결 채널은 시작할 수 없습니다.
- 전송 큐에서 트리거될 수 있습니다. 이 방법은 송신자 채널과 완전히 규정된 서버 채널(CONNAME을 지정하는 채널)에만 적용됩니다. 채널 트리거에 필요한 오브젝트를 준비해야 합니다.
- 애플리케이션 프로그램에서 시작할 수 있습니다. 수신자, 클러스터 수신자 또는 서버 연결 채널은 시작할 수 없습니다.
- 송신자, 클러스터 송신자, 요청자, 서버 또는 클라이언트 연결 채널에 의해 네트워크로부터 원격으로 시작됩니다. 수신자, 클러스터 수신자, 서버 및 요청자 채널 전송은 이 방식으로 시작됩니다. 서버 연결 채널도 마찬가지로입니다. 채널은 이미 시작된 상태여야 합니다(즉, 사용 가능한 상태).

참고: 채널이 '시작'되었으므로 반드시 메시지를 전송하지는 않습니다. 대신 이전에 설명한 네 가지 이벤트 중 하나가 발생할 때 전송을 시작할 수 있도록 '설정'될 수 있습니다. 채널 사용과 사용 불가능은 START 및 STOP 운영자 명령을 사용하여 수행됩니다.

채널 상태

채널은 언제든지 여러 상태 중 하나일 수 있습니다. 일부 상태에는 하위 상태도 있습니다. 채널은 제공된 상태에서 다른 상태로 이동할 수 있습니다.

51 페이지의 그림 12에서는 가능한 모든 채널 상태의 계층과 각 채널 상태에 적용되는 하위 상태를 보여줍니다.

52 페이지의 그림 13에서는 채널 상태 간 링크를 보여줍니다. 이들 링크는 모든 유형의 메시지 채널과 서버 연결 채널에 적용됩니다.

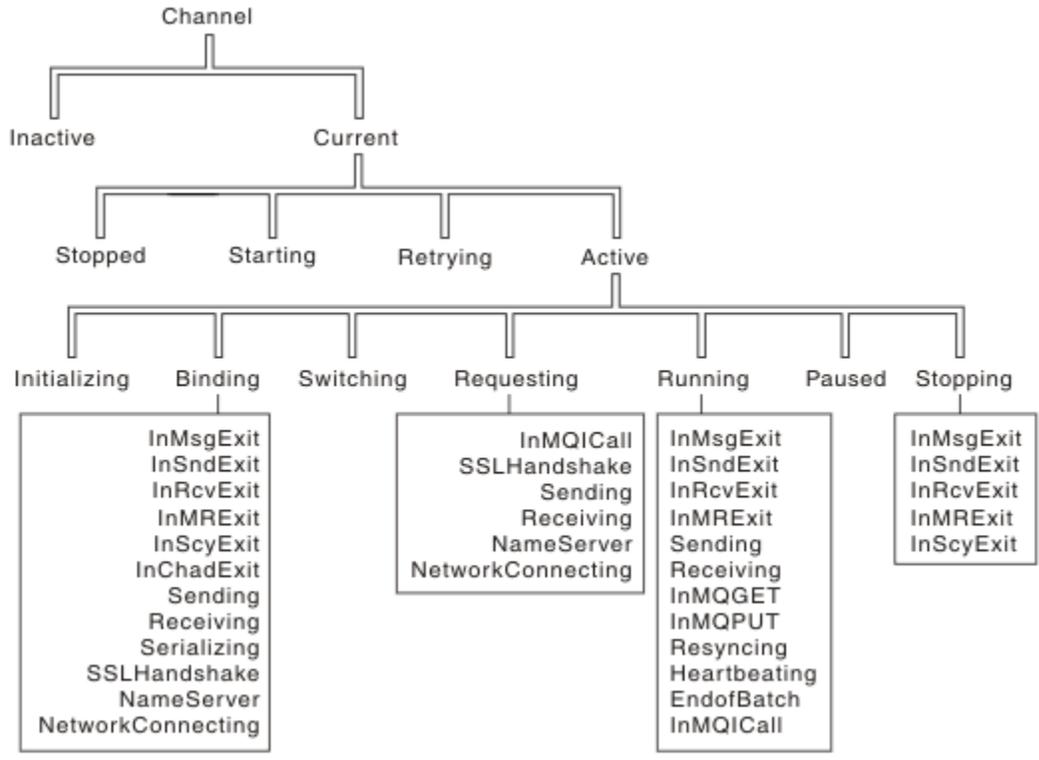


그림 12. 채널 상태 및 하위 상태

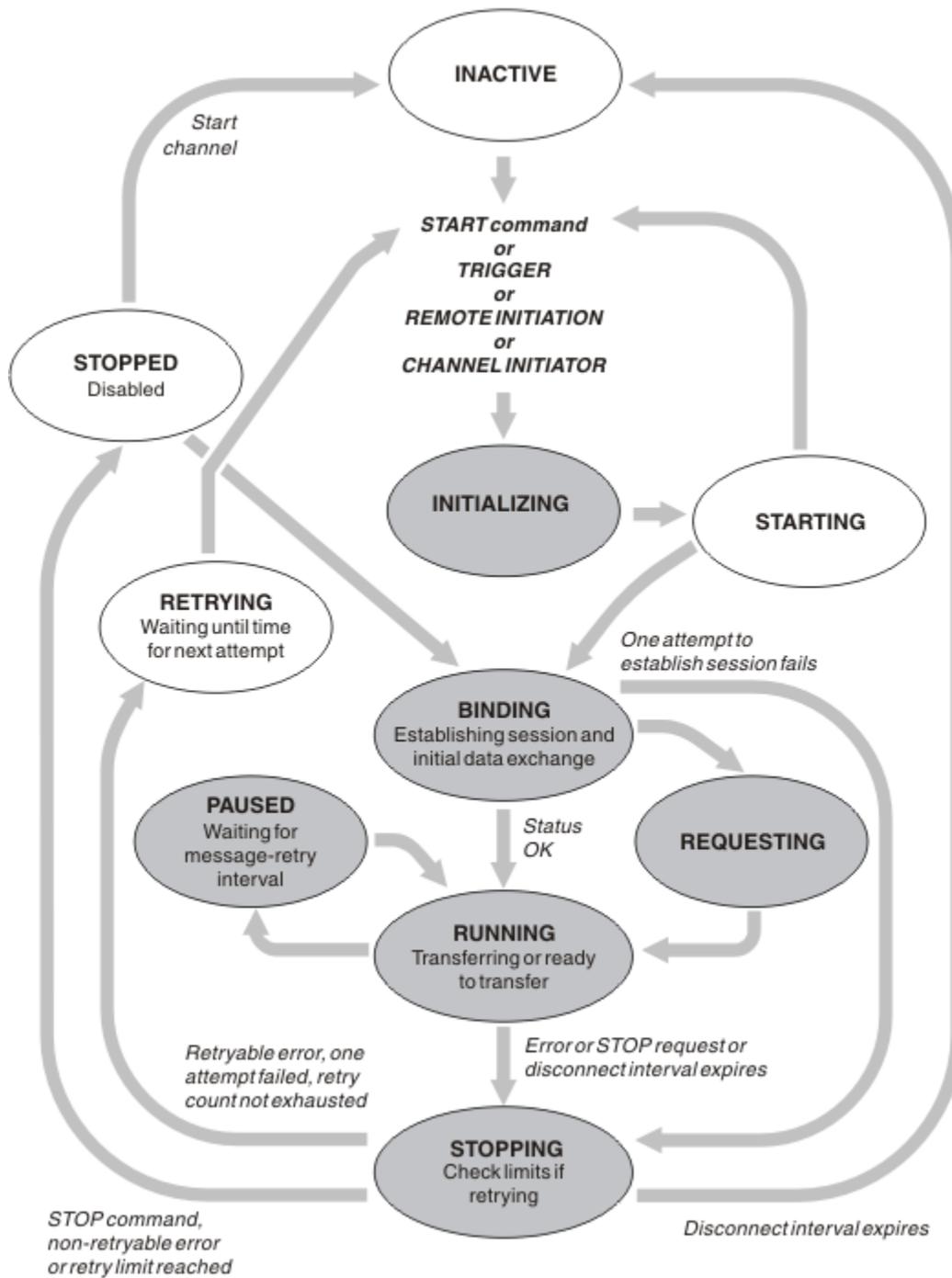


그림 13. 채널 상태 간 플로우

현재 및 활성화

비활성 이외의 상태인 경우 채널은 현재 상태입니다. RETRYING, STOPPED 또는 STARTING 상태에 있지 않은 현재 채널은 활성화입니다. 채널이 활성화 상태이면 자원을 이용하고 프로세스나 스레드가 실행 중입니다. 활성화 채널의 일곱 가지 가능한 상태(INITIALIZING, BINDING, SWITCHING, REQUESTING, RUNNING, PAUSED 또는 STOPPING)가 52 페이지의 그림 13에 강조표시되어 있습니다.

활성 채널은 채널이 수행하고 있는 작업의 자세한 정보를 정확하게 제공하는 하위 상태를 표시할 수도 있습니다. 각 상태에 대한 하위 상태는 51 페이지의 그림 12에서 볼 수 있습니다.

현재 및 활성화

비활성 이외의 상태인 경우 채널은 "현재" 상태입니다. RETRYING, STOPPED 또는 STARTING 상태에 있지 않는 한 현재 채널은 "활성"입니다.

"활성" 채널은 채널이 수행하고 있는 내용의 세부사항을 자세히 제공하는 하위 상태를 표시할 수도 있습니다.

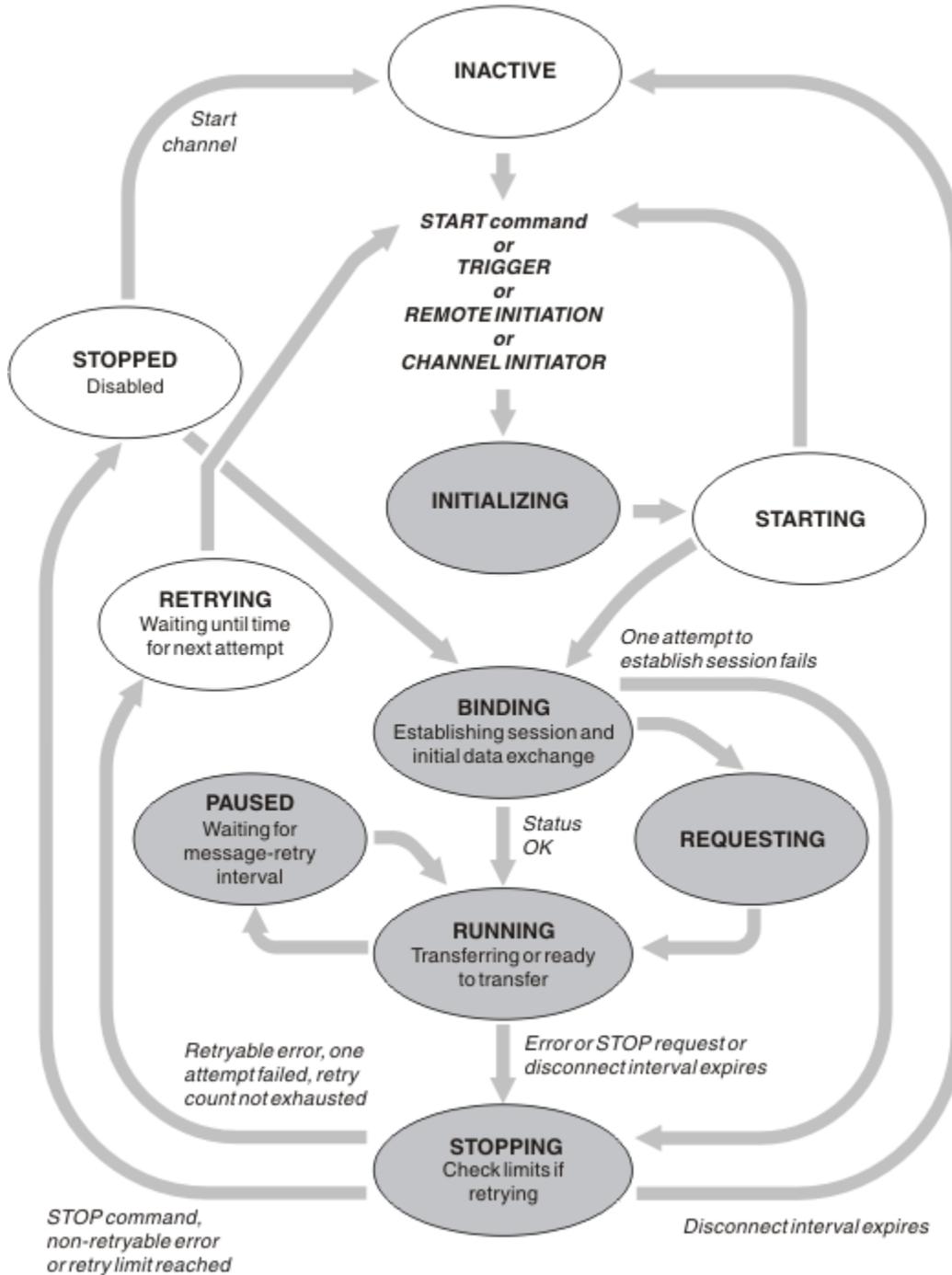


그림 14. 채널 상태 간 플로우

참고:

1. 채널이 53 페이지의 그림 14에서 강조표시된 6가지 상태(초기화, 바인딩, 요청, 실행, 일시정지 또는 정지)에 있는 경우에 이는 자원을 소비하고 있으며, 프로세스나 스레드가 실행 중입니다. 채널은 활성화 되어 있습니다.
2. 채널이 STOPPED 상태이면 다음 상태를 아직 알지 못해 세션이 활성화 상태일 수 있습니다.

최대 현재 채널 수 지정

한 번에 현재 상태일 수 있는 최대 채널 수를 지정할 수 있습니다. 이 수는 재시도 중인 채널과 중지된 채널을 포함하여 채널 상태 테이블에 항목이 있는 채널의 수입니다. Specify this using the queue manager configuration file for UNIX and Linux systems, or the WebSphere MQ Explorer. 초기화 또는 구성 파일을 사용하여 설정되는 값에 대한 자세한 정보는 [분산 큐잉에 대한 구성 파일 스탠자를 참조하십시오](#). For more information about specifying the maximum number of channels, see [IBM WebSphere MQ 관리](#) for WebSphere MQ for UNIX and Linux systems, and 윈도우 systems.

참고:

1. 이 수에는 서버 연결 채널이 포함됩니다.
2. 채널이 활성화하려면 먼저 현재 상태여야 합니다. 채널이 시작되었지만 현재가 될 수 없으면 시작이 실패합니다.

최대 활성 채널 수 지정

또한 많은 시작 채널로 인한 시스템 과부하를 방지하기 위해 최대 활성 채널 수를 지정할 수 있습니다. 이 방법을 사용하는 경우, 연결 끊기 간격 속성을 낮은 값으로 설정하여 다른 채널이 종료되는 즉시 대기 채널이 시작될 수 있도록 하십시오.

재시도하는 채널이 파트너와의 연결 설정 시도할 때마다 활성 채널이 되어야 합니다. 시도가 실패하면 다음 시도 전까지 활성 상태가 아닌 현재 채널로 남게 됩니다. 채널 재시도 횟수와 빈도는 재시도 횟수 및 재시도 간격 채널 속성으로 결정됩니다. 이 속성에는 둘 다 짧은 값과 긴 값이 있습니다. 자세한 정보는 [채널 속성을 참조하십시오](#).

START 명령이 실행 또는 트리거되었거나 다른 재시도 시간이 되어 채널이 활성 채널이 되었지만 활성 채널이 이미 최대값에 도달하여 활성화될 수 없는 경우, 채널은 다른 채널 인스턴스가 활성화를 멈추어 활성 슬롯이 비워질 때까지 대기합니다. 그러나 채널이 시작되고(원격으로 시작되기 때문에) 해당 시점에 사용 가능한 활성 슬롯이 없으면 원격 시작이 거부됩니다.

요청자 채널 이외의 채널이 활성화를 시도할 때마다 때마다 STARTING 상태가 됩니다. 이 상태는 즉시 사용 가능한 활성 슬롯이 있는 경우에도 발생하며 STARTING 상태는 잠시 동안만 유지됩니다. 그러나 채널이 활성 슬롯을 대기해야 하는 경우에는 대기 중에 STARTING 상태가 됩니다.

요청자 채널은 시작 상태가 되지 않습니다. 활성 채널 수가 이미 한계에 도달하여 요청자 채널을 시작할 수 없는 경우에는 채널이 비정상적으로 종료됩니다.

요청자 채널 이외의 채널이 활성 슬롯을 얻지 못하여 이를 기다리게 될 때마다, 메시지가 로그 에 기록되며 이벤트가 생성됩니다. 나중에 슬롯이 비워져 채널이 확보할 수 있으면 다른 메시지와 이벤트가 생성됩니다. 채널이 슬롯을 즉시 확보할 수 없으면 이러한 이벤트와 메시지 모두 생성되지 않습니다.

채널이 활성화될 때까지 대기하는 동안 STOP CHANNEL 명령을 실행하면 채널이 STOPPED 상태가 됩니다. 채널-중지 이벤트가 생성됩니다.

서버 연결 채널은 최대 활성 채널 수에 포함됩니다.

For more information about specifying the maximum number of active channels, see [IBM WebSphere MQ 관리](#) for WebSphere MQ for UNIX and Linux systems, and 윈도우 systems.

채널 오류

채널 오류가 발생하면 채널이 더 이상 전송 작업을 수행하지 않습니다. 채널이 송신자 또는 서버인 경우에는 문제점이 명확할 수 있으므로 재시도 상태로 전환됩니다. 재시도 상태로 전환할 수 없는 경우에는 채널이 STOPPED 상태가 됩니다.

송신 채널의 경우, 연관된 전송 큐가 GET(DISABLED)로 설정되고 트리거가 꺼집니다. (STATUS(STOPPED) 상태의 STOP 명령은 명령을 실행한 측을 STOPPED 상태로 전환합니다. 연결 끊기 간격이 만료되거나 STATUS(INACTIVE) 상태의 STOP 명령은 정상적으로 종료하여 비활성화됩니다.) STOPPED 상태의 채널은 운영자가 개입해야 재시작할 수 있습니다(57 페이지의 『중지된 채널 재시작』 참조).

참고: UNIX, Linux 및 Windows 시스템의 경우, 채널 시작기는 재시도하기 위해 실행 중이어야 합니다. 채널 시작기를 사용할 수 없으면 채널이 비활성화되므로 수동으로 재시작해야 합니다. 스크립트를 사용하여 채널을 시작하는 경우에는 스크립트 실행을 시도하기 전에 채널 시작기가 실행 중인지 확인하십시오.

긴 재시도 횟수(LONGRTY)는 재시도 과정을 설명합니다. 오류가 제거되면 채널은 자동으로 재시작되며, 전송 큐를 다시 사용할 수 있습니다. 오류를 제거하지 않고 재시도 한계에 도달하면, 채널이 STOPPED 상태가 됩니다. 중지된 채널은 운영자가 수동으로 재시작해야 합니다. 오류가 해결되지 않으면 다시 재시도하지 않습니다. 성공적으로 시작되면 전송 큐를 다시 사용합니다.

채널이 재시도(RETRYING) 또는 중지됨(STOPPED) 상태에 있는 동안 큐 관리자가 중지하면 큐 관리자가 재시작될 때 채널 상태가 기억됩니다. 그러나 채널이 중지됨(STOPPED) 상태에 있는 동안 큐 관리자가 중지하면 SVRCONN 채널 유형의 채널 상태는 재설정됩니다.

대상 큐가 가득 차거나 금지 상태가 되어 채널이 해당 큐에 메시지를 넣을 수 없으면 채널이 시간 간격(메시지 재시도 간격 속성에 지정)에 따라 조작을 여러 번(메시지 재시도 횟수 속성에 지정됨) 재시도할 수 있습니다. 또는 재시도를 야기하는 상황과 시도 횟수를 판별하는 메시지 재시도 엑시트를 작성할 수 있습니다. 채널은 메시지 재시도 간격이 완료할 때까지 기다리는 동안 일시정지 상태가 됩니다.

채널 속성에 대한 정보는 [채널 속성](#)을, 메시지 재시도 엑시트에 대한 정보는 [메시징 채널을 위한 채널 엑시트 프로그래밍](#)을 참조하십시오.

서버 연결 채널 한계

서버 연결 채널 한계를 설정하여 클라이언트 애플리케이션이 큐 관리자 채널 자원, MAXINST를 이용하는 것을 방지하고 단일 클라이언트 애플리케이션이 서버 연결 채널 용량, MAXINSTC를 이용하는 것을 방지할 수 있습니다.

개별 큐 관리자에서 언제라도 최대 총 채널 수가 활성화될 수 있습니다. 총 서버 연결 채널 인스턴스 수는 최대 활성 채널 수에 포함됩니다.

시작될 수 있는 서버 연결 채널의 최대 동시 인스턴스 수를 지정하지 않으면 단일 클라이언트 애플리케이션이 단일 서버 연결 채널에 연결되어 사용 가능한 최대 활성 채널 수를 이용할 수 있습니다. 최대 활성 채널 수에 도달하면 다른 채널이 큐 관리자에서 시작될 수 없습니다. 이러한 상황을 피하려면 채널을 시작한 클라이언트에 관계 없이, 시작될 수 있는 개별 서버 연결 채널의 동시 인스턴스의 수를 제한해야 합니다.

한계 값이 현재 실행 중인 서버 연결 채널의 인스턴스 수 미만으로 감소하면(0까지) 실행 중인 채널이 영향을 받지 않습니다. 충분한 수의 기존 인스턴스가 실행을 중지하여 현재 실행 중인 인스턴스 수가 한계 값 미만이 될 때까지 새 인스턴스가 시작될 수 없습니다.

또한 다수의 서로 다른 클라이언트 연결 채널을 개별 서버 연결 채널에 연결할 수 있습니다. 채널을 시작한 클라이언트에 관계 없이 시작될 수 있는 개별 서버 연결 채널의 동시 인스턴스 수 한계로 인해 클라이언트가 큐 관리자의 최대 활성 채널 용량을 이용할 수 없게 됩니다. 또한 개별 클라이언트에서 시작될 수 있는 개별 서버 연결 채널의 동시 인스턴스 수를 제한하지 않으면, 결합이 있는 단일 클라이언트 애플리케이션이 너무 많은 연결을 열어 개별 서버 연결 채널에 할당된 채널 용량을 이용하고 결과적으로 채널을 사용해야 하는 다른 클라이언트가 연결하지 못하게 됩니다. 이러한 상황을 피하려면 개별 클라이언트에서 시작될 수 있는 개별 서버 연결 채널의 동시 인스턴스의 수를 제한해야 합니다.

개별 클라이언트 한계 값이 현재 개별 클라이언트에서 실행 중인 서버 연결 채널의 인스턴스 수 미만으로 감소하면(0까지) 실행 중인 채널이 영향을 받지 않습니다. 그러나 해당 클라이언트로부터의 충분한 수의 기존 인스턴스가 실행을 중지하여 현재 실행 중인 인스턴스 수가 이 매개변수의 값보다 적어질 때까지 새 한계를 초과하는 개별 클라이언트에서 서버 연결 채널의 새 인스턴스를 시작할 수 없습니다.

채널의 다른 쪽이 여전히 사용 가능한지 검사

하트비트 간격, 활성 유지(keepalive) 간격, 수신 제한시간을 사용하여 채널의 다른 한 쪽을 사용할 수 있는지 확인할 수 있습니다.

하트비트

하트비트 간격 채널 속성을 사용하여 전송 큐에 메시지가 없을 때 송신 MCA에서 플로우가 전달되는 것으로 지정할 수 있습니다(하트비트 간격(HBINT)의 설명 참조).

활성 유지(keepalive)

WebSphere MQ 에서 UNIX, Linux 및 시스템 경우, 전송 프로토콜로 TCP를 사용하는 경우 keepalive=yes를 설정할 수 있습니다. 이 옵션을 지정하면 TCP가 연결의 다른 한 쪽을 계속 사용할 수 있는지 주기적으로 확인합니다. 사용할 수 없으면 채널이 종료됩니다. 이 옵션에 대한 설명은 [활성 유지\(keepalive\) 간격\(KAINT\)](#)을 참조하십시오.

TCP 오류를 보고하는 신뢰할 수 없는 채널이 있는 경우 **Keepalive** 옵션을 사용하면 채널을 복구할 수 가능성이 높음을 의미합니다.

Keepalive 옵션 작동을 제어하는 시간 간격을 지정할 수 있습니다. 시간 간격을 변경하면 변경 이후 시작된 TCP/IP 채널만 영향을 받습니다. 시간 간격에 선택하는 값은 채널의 연결 끊기 간격 값보다 작아야 합니다.

Keepalive 옵션 사용에 대한 자세한 정보는 `DEFINE CHANNEL` 명령의 `KAINT` 매개변수를 참조하십시오.

수신 제한시간

TCP를 전송 프로토콜로 사용하는 경우 일정 기간 동안 데이터가 수신되지 않으면 유틸 비MQI 채널 연결의 수신 측 또한 닫힙니다. 이 기간에 해당하는 `receive time-out` 값은 HBINT(하트비트 간격) 값에 따라 판별됩니다.

In WebSphere MQ for UNIX, Linux, and 윈도우 systems, the 타임아웃을 받다. value is set as follows:

1. 초기 플로우 수의 경우, 조정이 수행되기 전 `receive time-out` 값은 채널 정의의 HBINT 값의 두 배입니다.
2. HBINT가 60초 미만으로 설정되는 경우 채널이 HBINT 값을 조정하면 `receive time-out` 값이 이 값의 두 배로 설정됩니다. HBINT가 60초 이상으로 설정되면 `receive time-out` 값이 HBINT 값보다 60초 더 큰 값으로 설정됩니다.

참고:

1. 다음 값 중 하나가 0이면 제한시간이 없습니다.
2. 하트비트를 지원하지 않는 연결의 경우, HBINT 값이 2단계에서 0으로 조정되어 제한시간이 없으므로 TCP/IP KEEPALIVE를 사용해야 합니다.
3. 공유 대화를 사용하는 클라이언트 연결의 경우, 하트비트가 MQGET이 미해결 상태인 경우뿐 아니라 항상 채널 양측에서 전달될 수 있습니다.
4. 공유 대화를 사용하지 않는 클라이언트 연결의 경우, 클라이언트가 대기과 함께 MQGET 호출을 발행하는 경우에만 서버에서 하트비트가 전달됩니다. 따라서 클라이언트 채널의 경우 하트비트 간격을 너무 작게 설정하는 것은 바람직하지 않습니다. 예를 들어, 하트비트가 10초로 설정되면 커밋 시간이 20초보다 길게 소요되는 경우 MQRC_CONNECTION_BROKEN과 함께 MQCMIT 호출이 실패합니다. 이 시간 동안 데이터가 전달되지 않기 때문입니다. 이는 작업 단위가 큰 경우에 해당할 수 있습니다. 그러나 대기가 있는 MQGET만 상당한 시간이 소요되므로 하트비트 간격에 적절한 값을 선택하면 이에 해당되지 않습니다.

SHARECNV가 0이 아니면 클라이언트가 양방향 전송 연결을 사용하며 이는 클라이언트가 모든 MQI 호출 중에 하트비트를 수행할 수 있고 실제로 수행함을 의미합니다.

5. WebSphere MQ 버전 7 클라이언트 채널에서 하트비트는 서버 및 클라이언트 측 모두에서 흐를 수 있습니다. 양측의 제한시간은 $2 * HBINT$ (HBINT가 60초 미만인 경우) 및 $HBINT + 60$ (HBINT가 60초보다 긴 경우)을 기반으로 합니다.
6. 최소한 모든 하트비트 간격에서 데이터 또는 하트비트 플로우가 예상되므로 두 배의 하트비트 간격 후 연결을 취소하는 것은 올바르지 않습니다. 그러나 하트비트 간격을 너무 작게 설정하면 특히 채널 엑시트를 사용하는 경우 문제가 발생할 수 있습니다. 예를 들어, HBINT 값이 1초이고 송신 또는 수신 엑시트를 사용하는 경우에는 채널을 취소하기 전에 2초 동안만 수신측이 대기합니다. MCA가 메시지 암호화와 같은 태스크를 수행하는 경우에는 이 값이 너무 짧을 수도 있습니다.

MCA 채택

IBM WebSphere MQ Explorer는 MCA 채택 기능으로 수신자 채널을 취소하고 올바른 위치에서 새 채널을 시작할 수 있습니다.

채널에 통신 장애가 발생하면 수신자 채널은 '통신 수신' 상태로 유지됩니다. 통신이 재설정되면 송신자 채널이 다시 연결을 시도합니다. 리모트 큐 관리자가 수신자 채널이 이미 실행 중임을 발견하는 경우, 이는 동일한 수신자 채널의 다른 버전이 시작되지 않도록 합니다. 이 문제를 해결하려면 사용자가 개입하여 문제를 수정하거나 시스템 활성 유지(keepalive)를 사용해야 합니다.

MCA 채택 기능은 문제점을 자동으로 해결합니다. IBM WebSphere MQ Explorer는 이 기능으로 수신자 채널을 취소하고 올바른 위치에서 새 채널을 시작할 수 있습니다.

기능은 다양한 옵션으로 설정될 수 있습니다. **distributed** 분산 플랫폼의 경우, [관리](#)를 참조하십시오.

채널 중지 및 일시정지

이 주제에서는 연결 끊기 시간 간격이 만기되기 전에 채널을 중지하고 일시정지하는 방법에 대해 설명합니다.

메시지 채널은 큐 관리자 간에 오래 실행되는 연결이 되도록, 또한 질서 있는 종료는 연결 끊기 간격 채널 속성으로만 제어되도록 설계되었습니다. 이 메커니즘은 연결 끊기 시간 간격이 만기되기 전에 운영자가 채널을 종료하지 않아도 되는 경우에도 적용됩니다. 이러한 요구는 다음 상황에서 발생할 수 있습니다.

- 시스템 일시정지
- 자원 유지
- 채널의 한쪽 끝에서의 일방적인 조치

이 경우 채널을 중지할 수 있습니다. 다음을 사용하여 이를 수행할 수 있습니다.

- STOP CHANNEL MQSC 명령
- Stop Channel PCF 명령
- IBM WebSphere MQ Explorer

이 명령을 사용한 채널 중지 옵션은 다음과 같은 세 가지가 있습니다.

정지

QUIESCE 옵션은 채널을 중지하기 전에 현재 메시지 배치를 종료하려고 시도합니다.

FORCE

FORCE 옵션은 채널을 즉시 종료하려고 시도하며, 채널이 인다우트(in-doubt) 상태로 남아 있을 수 있으므로 재시작 시 채널을 재동기화해야 할 수 있습니다.

종료

TERMINATE 옵션은 채널을 즉시 중지하려고 시도하며 채널의 스레드 또는 프로세스를 종료합니다.

이러한 옵션은 모두 채널을 STOPPED 상태로 유지하므로 재시작하려면 운영자가 개입해야 합니다.

송신측에서 채널을 중지할 수 있지만 재시작하려면 운영자 개입이 필요합니다. 채널의 수신측에서는 MCA가 송신측에서 들어오는 데이터를 기다리고 수신측에서 질서 있는 채널 종료를 시작할 수 없으므로 상황이 훨씬 더 어려워집니다. 중지 명령은 MCA가 데이터 대기에서 리턴할 때까지 보류 상태가 됩니다.

결과적으로 필요한 조작 특성에 따라 다음과 같은 세 가지 채널 사용 방법이 권장됩니다.

- 채널을 오래 실행하려면 송신측의 질서 있는 종료만 존재할 수 있다는 점에 유의하십시오. 채널이 인터럽트, 즉 중지되는 경우 재시작하려면 운영자 개입(START CHANNEL 명령)이 필요합니다.
- 전송할 메시지가 있는 경우에만 채널을 활성화하려면 연결 끊기 간격을 아주 낮은 값으로 설정하십시오. 기본 설정은 높은 값이므로 이 제어 레벨이 필요한 채널에는 권장되지 않습니다. 수신 채널은 인터럽트하기 어려우므로 가장 경제적인 옵션은 채널 연결이 자동으로 끊어졌다가 워크로드 요구에 따라 다시 연결되도록 하는 것입니다. 대부분 채널의 경우, 연결 끊기 간격의 적절한 설정은 경험적으로 설정할 수 있습니다.
- 하트비트 간격 속성을 사용하면 송신할 메시지가 없는 기간 동안 송신 MCA가 수신 MCA로 하트비트 플로우를 송신할 수 있습니다. 이 조치는 수신 MCA를 대기 상태에서 해제하며, 연결 끊기 간격이 만기될 때까지 기다리지 않고 채널을 일시정지할 수 있습니다. 하트비트 간격에는 연결 끊기 간격 값보다 작은 값을 지정하십시오.

참고:

1. 서버 채널의 경우 연결 끊기 간격을 낮은 값으로 설정하거나 하트비트를 사용하는 것이 좋습니다. 이 낮은 값은 서버 채널이 보낼 메시지가 없을 때 요청자 채널이 비정상적으로 종료되는 경우(예를 들어 채널이 취소되어)를 허용합니다. 연결 끊기 간격이 높게 설정되고 하트비트를 사용하지 않으면, 요청자가 종료된 사실을 서버가 감지하지 못합니다. 다음에 요청자에게 메시지를 보내려고 시도할 때만 감지합니다. 서버가 아직 실행 중인 경우 큐에 도달하는 메시지를 더 가져오기 위해 독점적 입력에 대해 전송 큐를 열린 상태로 유지합니다. 요청자로부터의 채널을 재시작하려고 시도하면 서버가 독점 입력을 위해 전송 큐를 계속 열린 상태로 유지하므로 시작 요청자가 오류를 수신합니다. 서버 채널을 중지한 다음 요청자로부터 다시 채널을 재시작해야 합니다.

중지된 채널 재시작

채널이 STOPPED 상태가 되면 채널을 수동으로 재시작해야 합니다.

채널을 재시작하려면 다음 명령 중 하나를 실행하십시오.

- START CHANNEL MQSC 명령

- Start Channel PCF 명령
- IBM WebSphere MQ Explorer

송신자 또는 서버 채널의 경우 채널에 STOPPED 상태를 입력하면, 연관된 전송 큐가 GET(DISABLED)로 설정되었으며 트리거 설정이 해제되었습니다. 시작 요청이 수신되면 이러한 속성이 자동으로 재설정됩니다.

채널이 재시도(RETRYING) 또는 중지됨(STOPPED) 상태에 있는 동안 큐 관리자(분산 플랫폼에서)가 중지하면 큐 관리자가 재시작될 때 채널 상태가 기억됩니다. 그러나 채널이 중지됨 상태에 있는 동안 큐 관리자가 중지하면 SVRCONN 채널 유형의 채널 상태는 재설정됩니다.

인다우트(in-doubt) 채널

인다우트(in-doubt) 채널은 원격 채널이 송수신된 메시지를 알지 못하는 상태의 채널입니다.

이 채널과 큐에 커밋되어야 하는 메시지를 알지 못하는 큐 관리자의 차이에 유의하십시오.

배치 하트비트 채널 매개변수(BATCHHB)를 사용하여 채널이 인다우트(in-doubt) 상태가 되는 기회를 줄일 수 있습니다. 이 매개변수의 값을 지정하면, 송신자 채널이 추가 조치를 수행하기 전에 원격 채널이 계속 활성화 상태인지 검사합니다. 응답이 수신되지 않으면 수신자 채널이 더 이상 활성화 상태가 아닌 것으로 간주됩니다. 메시지를 롤백하고 다시 라우트할 수 있으므로 송신자 채널은 인다우트(in-doubt) 상태가 되지 않습니다. 따라서 송신자 채널이 수신자 채널이 계속 활성화 상태인지 확인하고 수신자 채널이 송신된 메시지를 수신했는지 확인하는 동안 채널이 인다우트(in-doubt) 상태가 되어야 하는 시간을 줄일 수 있습니다. 배치 하트비트 매개변수에 대한 자세한 정보는 [채널 속성](#)을 참조하십시오.

인다우트(in-doubt) 채널 문제점은 일반적으로 자동으로 해결됩니다. 통신이 끊어지고 채널이 송신측에서 수신 상태를 알 수 없는 메시지 배치로 인해 채널이 인다우트(in-doubt) 상태가 되더라도 통신이 다시 설정되면 상황이 해결됩니다. 순서 번호 및 LUWID 레코드는 이런 목적을 위해 유지됩니다. 채널은 LUWID 정보가 교환될 때까지 인다우트(in-doubt) 상태이고, 하나의 메시지 배치만 채널에 대해 인다우트(in-doubt) 상태가 될 수 있습니다.

필요한 경우 채널을 수동으로 재동기화할 수 있습니다. 수동이라는 용어는 WebSphere MQ 시스템 관리 명령을 포함하는 프로그램 또는 연산자를 사용함을 의미합니다. 수동 재동기화 프로세스는 다음과 같이 진행됩니다. 이 설명은 MQSC 명령을 사용하지만 PCF 해당 명령도 사용할 수 있습니다.

1. DISPLAY CHSTATUS 명령을 사용하여 채널의 각 측에 대해 마지막으로 커밋된 논리 작업 단위 ID(LUWID)를 찾으십시오. 이 경우 다음 명령을 사용하십시오.

- 채널의 인다우트(in-doubt)측의 경우:

```
DISPLAY CHSTATUS(name) SAVED CURLUWID
```

CONNNAME 및 XMITQ 매개변수를 사용하여 채널을 더 식별할 수 있습니다.

- 채널의 수신측의 경우:

```
DISPLAY CHSTATUS(name) SAVED LSTLUWID
```

CONNNAME 매개변수를 사용하여 채널을 더 식별할 수 있습니다.

채널의 송신측만이 인다우트(in-doubt) 상태일 수 있으므로 명령은 서로 다릅니다. 수신측은 절대로 인다우트(in-doubt) 상태가 되지 않습니다.

IBM i용 WebSphere MQ 에서, STRMQMMQSC 명령 또는 MQM 채널 상태 CL 명령, WRKMQMCHST에 대한 작업을 사용하여 파일에서 DISPLAY CHSTATUS 명령을 실행할 수 있습니다.

2. 두 LUWID가 같으면 수신측이 송신자가 인다우트(in-doubt)로 간주하는 작업 단위를 커밋한 것입니다. 이제 송신측은 전송 큐에서 인다우트(in-doubt) 메시지를 제거한 후 다시 사용할 수 있습니다. 이는 다음 채널 RESOLVE 명령으로 수행됩니다.

```
RESOLVE CHANNEL(name) ACTION(COMMIT)
```

3. 두 개 LUWID가 다르면 수신측이 송신자가 인다우트(in-doubt) 상태로 간주하는 작업 단위를 커밋하지 않은 것입니다. 송신측은 인다우트(in-doubt) 메시지를 전송 큐에 보유했다가 재전송해야 합니다. 이는 다음 채널 RESOLVE 명령으로 수행됩니다.

이 프로세스가 완료되면 채널이 더 이상 인다우트(in-doubt) 상태가 아닙니다. 필요한 경우 이제 다른 채널이 전송 큐를 사용할 수 있습니다.

문제점 판별

문제점 판별에는 중요한 두 가지 구분되는 측면(명령이 제출될 때 발견되는 문제점인지 채널 조작 중에 발견되는 문제점인지)이 있습니다.

명령 유효성 검증

명령 및 패널 데이터가 처리 승인을 받으려면 먼저 오류가 없어야 합니다. 유효성 검사에 의해 발견된 모든 오류는 즉시 오류 메시지에 의해 사용자에게 통보됩니다.

문제점 진단 작업은 이러한 오류 메시지를 해석하고 정정 조치를 취하는 것에서 시작합니다.

문제점 처리

채널의 일반 조작중에 발견된 문제점은 시스템 콘솔 또는 시스템 로그로 통지됩니다. 문제점 진단은 로그에서 관련된 모든 정보를 수집하는 것으로 시작하여 문제점 식별을 위한 분석을 계속합니다.

확정 및 오류 메시지는 가능한 경우 명령을 초기화한 터미널로 리턴됩니다.

WebSphere MQ는 계정 및 통계 데이터를 생성하며, 사용 및 성능에서 추세를 식별하는 데 사용할 수 있습니다.

distributed 분산 플랫폼에서 이 정보는 PCF 레코드로 생성됩니다. 자세한 내용은 [구조 데이터 유형](#) 을 참조하십시오.

메시지 및 코드

문제점의 기본 진단에 도움이 되는 메시지 및 코드에 대한 정보는 [진단 메시지 및 이유 코드](#)를 참조하십시오.

메시지의 안전성

WebSphere MQ의 일반 복구 기능 외에 분산 큐 관리는 메시지 채널의 두 끝 간에 통합된 동기점 프로시저를 사용하여 메시지가 적절하게 전달되었는지를 확인합니다. 이 프로시저가 오류를 감지하면, 채널이 닫혀 문제를 조사하고 채널이 다시 시작될 때까지 메시지를 전송 큐에 안전하게 유지할 수 있습니다.

동기점 프로시저에는 채널이 시작될 때 *in-doubt* 상황을 복구하려고 시도한다는 점에서 추가된 이점이 있습니다. (*in-doubt*는 동기점이 요청된 복구 단위의 상태이지만 요청 결과는 아직 알려지지 않았습니다.) 이 기능은 또한 다음 두 기능과 연관되어 있습니다.

1. 커미트 또는 백아웃으로 해석
2. 순서 번호 재설정

채널은 대부분의 경우 자동으로 복구되므로 이러한 기능은 예외적인 상황에서만 사용됩니다.

빠른, 비지속 메시지

비지속 메시지 속도(NPMSPEED) 채널 속성을 사용하여 채널의 비지속 메시지가 보다 빠르게 전달되도록 지정할 수 있습니다. 이 속성에 대한 자세한 정보는 [비지속 메시지 속도\(NPMSPEED\)](#)를 참조하십시오.

빠른, 비지속 메시지가 전송되는 동안 채널이 종료되면 메시지가 손실될 수 있으며 필요한 경우 애플리케이션이 복구를 조율해야 합니다.

수신 채널이 메시지를 목적지 큐에 넣지 못하는 경우, 데드 레터 큐가 정의되어 있으면 메시지가 이 큐에 위치하게 됩니다. 그렇지 않으면 메시지가 제거됩니다.

참고: 채널의 다른 측이 이 옵션을 지원하지 않는 경우에는 채널이 보통 속도로 실행됩니다.

미전달 메시지

메시지가 전달될 수 없을 때 발생하는 상황에 대해서는 60 페이지의 『메시지를 전달할 수 없을 경우의 메시지 처리』를 참조하십시오.

메시지를 전달할 수 없을 경우의 메시지 처리

메시지를 전달할 수 없는 경우 MCA가 여러 가지 방법으로 메시지를 처리할 수 있습니다. 다시 시도하거나 송신자로 돌아가거나 데드-레터 큐에 넣을 수 있습니다.

60 페이지의 그림 15에서는 MCA가 메시지를 수신지 큐에 넣을 수 없을 때 발생하는 프로세싱을 보여줍니다. 표시된 옵션이 모든 플랫폼에 적용되는 것은 아닙니다.

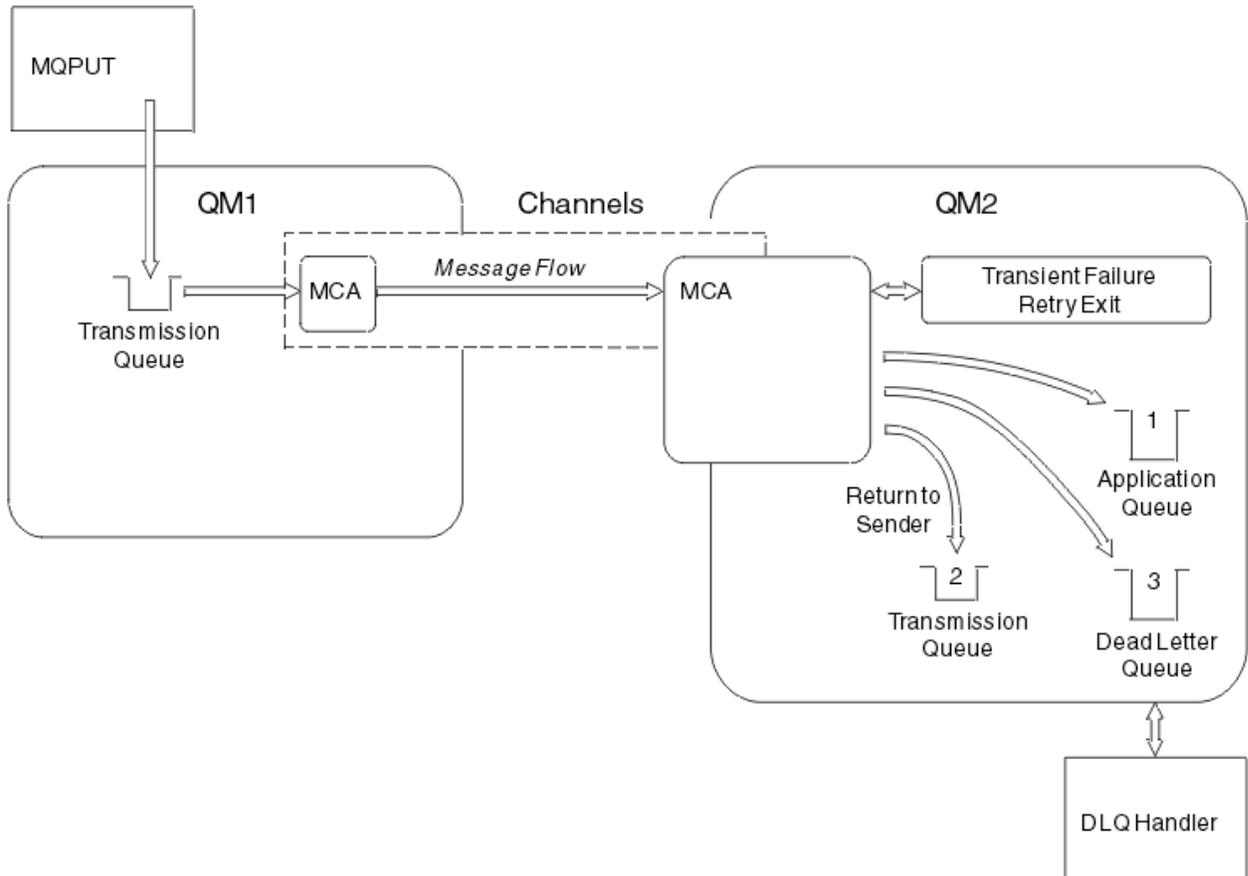


그림 15. 메시지를 전달할 수 없는 경우 발생하는 결과

그림과 같이, MCA는 전달할 수 없는 메시지에 대해 몇 가지 작업을 수행할 수 있습니다. 수행되는 조치는 채널이 정의될 때 지정된 옵션과 메시지에 대한 MQPUT 보고서 옵션으로 결정됩니다.

1. 메시지 재시도

일시적인 어떤 이유로(예를 들어, 대상 큐가 가득 찬 경우) MCA가 대상 큐에 메시지를 넣을 수 없는 경우, MCA는 대기 후 나중에 조작을 다시 시도할 수 있습니다. MCA가 대기하는지 여부, 대기 시간, 시도 횟수를 판별할 수 있습니다.

- 사용자는 채널을 정의할 때 MQPUT 오류에 대비하여 메시지 재시도 시간과 간격을 지정할 수 있습니다. 큐가 가득 찼거나 또는 넣기가 금지되어 메시지를 목적지 큐에 넣을 수 없는 경우, MCA는 지정된 시간 간격에서 지정된 횟수만큼 조작을 시도합니다.
- 사용자 고유의 메시지 재시도 엑시트를 작성할 수 있습니다. 이 엑시트를 사용하여 MCA가 MQPUT 또는 MQOPEN 조작을 다시 시도하는 조건을 지정할 수 있습니다. 채널을 정의할 때 엑시트 이름을 지정하십시오.

2. 송신자로 돌아가기

메시지 재시도가 성공하지 못했거나 다른 유형의 오류가 발생한 경우, MCA는 메시지를 작성자에게 다시 송신할 수 있습니다. 송신자로 돌아가기를 사용하려면 원래 큐에 메시지를 넣을 때 메시지 디스크립터에서 다음 옵션을 지정해야 합니다.

- MQRO_EXCEPTION_WITH_FULL_DATA 보고 옵션
- MQRO_DISCARD_MSG 보고 옵션
- 응답 대상 큐 및 응답 대상 큐 관리자의 이름

MCA가 메시지를 목적지 큐에 넣을 수 없을 경우, 원래 메시지를 수록한 예외 보고를 생성하고 이를 전송 큐에 넣어 원래 메시지에 지정된 응답 대상 큐로 송신되게 합니다 (응답 대상 큐가 MCA와 같은 큐 관리자에 있으면, 메시지를 전송 큐가 아닌 해당 큐에 바로 넣음).

3. 데드-레터 큐

메시지를 전달하거나 리턴할 수 없는 경우, 이를 데드-레터 큐(DLQ)에 넣습니다. DLQ 핸들러를 사용하여 메시지를 처리할 수 있습니다. [This processing is described in WebSphere MQ 데드-레터 큐 핸들러를 사용하여 전달되지 않은 메시지 처리 for IBM WebSphere MQ for UNIX, Linux.](#) 데드-레터 큐를 사용할 수 없는 경우, 송신 MCA가 전송 큐에 메시지를 남겨 두고 채널은 중지됩니다. 고속 채널에서는 데드-레터 큐에 쓸 수 없는 비지속 메시지가 손실됩니다.

IBM WebSphere MQ Version 7.0에서 로컬 데드-레터 큐가 정의되어 있지 않고 리모트 큐가 사용 불가능하거나 정의되어 있지 않으며 리모트 데드-레터 큐가 없으면, 송신자 채널은 재시도 상태가 되고 메시지는 전송 큐에 자동으로 롤백됩니다.

관련 참조

[데드-레터 큐 사용\(USEDLQ\)](#)

채널 트리거

WebSphere MQ는 큐의 특정 조건이 충족되면 자동으로 애플리케이션을 시작하는 기능을 제공합니다. 이 기능을 트리거라고 합니다.

이 설명은 트리거 개념의 개요로 제공됩니다. 전체 설명은 [트리거를 사용하여 WebSphere MQ 애플리케이션 시작](#)을 참조하십시오.

플랫폼별 정보는 다음을 참조하십시오.

- Windows의 경우, UNIX and Linux 시스템, 62 페이지의 『UNIX, Linux 및 Windows 시스템에서의 채널 트리거』를 참조하십시오.

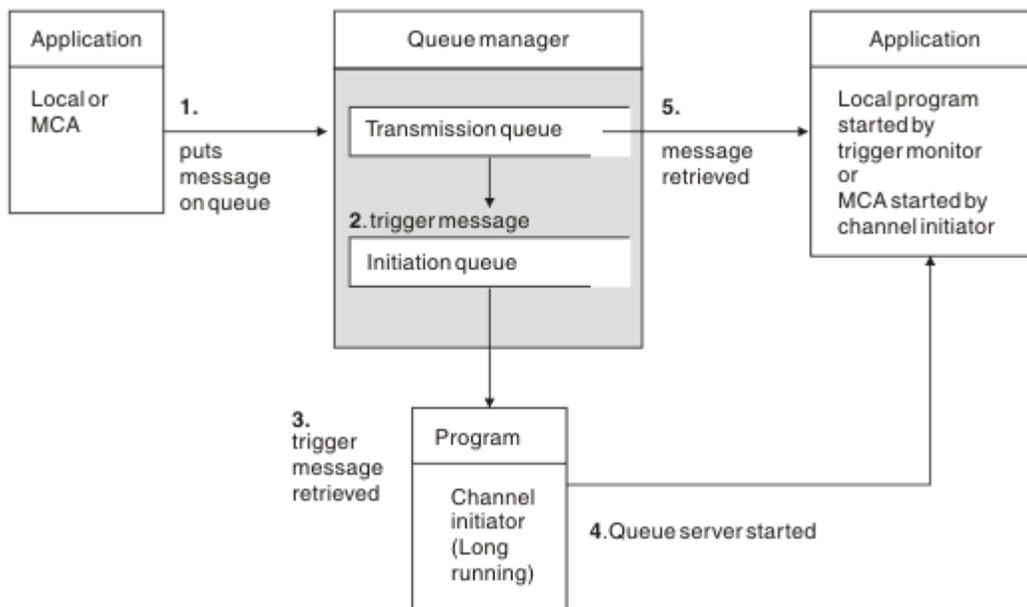


그림 16. 트리거의 개념

61 페이지의 그림 16에서는 트리거링에 필요한 오브젝트를 보여줍니다. 다음과 같은 순서의 이벤트를 보여줍니다.

1. 로컬 큐 관리자가 애플리케이션 또는 메시지 채널 에이전트(MCA)의 메시지를 전송 큐에 넣습니다.
2. 트리거 조건이 충족되면 로컬 큐 관리자는 트리거 메시지를 이니시에이션 큐에 넣습니다.
3. 장기 실행 채널 시작기 프로그램이 시작 큐를 모니터하고 도착한 메시지를 검색합니다.
4. 채널 시작기가 메시지에 포함된 정보에 따라 트리거 메시지를 처리합니다. 이 정보는 채널 이름을 포함할 수 있으며 이 경우 해당 MCA가 시작됩니다.
5. 트리거된 로컬 애플리케이션 또는 MCA가 전송 큐에서 메시지를 검색합니다.

이 시나리오를 설정하려면 다음을 수행해야 합니다.

- 해당 속성에 시작 큐(즉, SYSTEM.CHANNEL.INITQ)의 이름을 지정하여 전송 큐를 작성합니다.
- 시작 큐(SYSTEM.CHANNEL.INITQ)가 있는지 확인합니다.
- 채널 시작기 프로그램이 사용 가능하고 실행 중인지 확인합니다. 채널 시작기 프로그램에는 해당 시작 명령에 시작 큐의 이름이 제공되어야 합니다.
- 선택적으로, 트리거를 위한 프로세스 정의를 작성하십시오. 존재하지 않는 경우, *UserData* 필드에 제공되는 채널의 이름이 포함되었는지 확인하십시오. 프로세스 정의를 작성하는 대신 전송 큐의 *TriggerData* 속성에 채널 이름을 지정할 수 있습니다. WebSphere MQ for UNIX, Linux and Windows systems, allow the channel name to be specified as blank, in which case the first available channel definition with this transmission queue is used.
- 전송 큐 정의에 제공할 프로세스 정의의 이름(해당되는 경우), 이니시에이션 큐 이름, 가장 적합하다고 생각되는 트리거 특성이 포함되는지 확인합니다. 트리거 제어 속성은 필요에 따라 트리거 사용 여부를 허용합니다.

참고:

1. 채널 시작기 프로그램은 채널을 시작하는 데 사용되는 이니시에이션 큐를 모니터하는 '트리거 모니터' 역할을 합니다.
2. 이니시에이션 큐와 트리거 프로세스를 사용하여 원하는 수의 채널을 트리거할 수 있습니다.
3. 원하는 수의 이니시에이션 큐와 트리거 프로세스를 정의할 수 있습니다.
4. 시스템에 과도하게 채널이 시작되지 않게 하기 위해 FIRST 트리거 유형이 권장됩니다.

UNIX, Linux 및 Windows 시스템에서의 채널 트리거

트리거할 프로세스를 정의하는 WebSphere MQ에서 프로세스 정의를 작성할 수 있습니다. 메시지가 전송 큐에 도달할 때 트리거될 프로세스의 이름을 지정하는 프로세스 정의를 작성하려면 MQSC 명령 DEFINE PROCESS를 사용하십시오. 프로세스 정의의 USERDATA 속성은 전송 큐에서 제공되는 채널의 이름을 포함합니다.

채널(QM3.TO.QM4)을 시작하는 애플리케이션을 트리거하기 위한 트리거 메시지를 이니시에이션 큐(IQ)에 기록하도록 지정하여 로컬 큐(QM4)를 정의하십시오.

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(P1) USAGE(XMITQ)
```

시작될 애플리케이션(프로세스 P1)을 정의하십시오.

```
DEFINE PROCESS(P1) USERDATA(QM3.TO.QM4)
```

Alternatively, for WebSphere MQ for UNIX, Linux and 윈도우 systems, you can eliminate the need for a process definition by specifying the channel name in the TRIGDATA attribute of the transmission queue.

로컬 큐(QM4)를 정의하십시오. 트리거 메시지를 기본 이니시에이션 큐 SYSTEM.CHANNEL.INITQ에 작성하여 채널(QM3.TO.QM4)을 시작하는 애플리케이션(프로세스 P1)을 트리거하도록 지정하십시오.

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ)
USAGE(XMITQ) TRIGDATA(QM3.TO.QM4)
```

채널 이름을 지정하지 않으면 채널 시작기가 이름 지정된 전송 큐와 관련된 채널을 찾을 때까지 채널 정의 파일을 검색합니다.

관련 개념

63 페이지의 『채널 시작기 시작 및 중지』

트리거는 채널 시작기 프로세스를 사용하여 구현됩니다.

25 페이지의 『분산 큐잉을 사용하여 애플리케이션 연결』

이 절에서는 큐 정의, 채널 정의, 트리거 및 동기점 프로시저를 포함한 WebSphere MQ 설치 간 상호통신에 관한 자세한 정보를 제공합니다.

관련 참조

[UNIX, Linux 및 Windows 시스템의 채널 프로그램](#)

채널 시작기 시작 및 중지

트리거는 채널 시작기 프로세스를 사용하여 구현됩니다.

이 채널 이니시에이터 프로세스는 MQSC 명령 START CHINIT로 시작됩니다. 기본 이니시에이션 큐를 사용하지 않는 경우 명령에서 이니시에이션 큐의 이름을 지정하십시오. 예를 들어, 기본 큐 관리자에 대해 큐 IQ를 시작하기 위해 START CHINIT 명령을 사용하려면 다음을 입력하십시오.

```
START CHINIT INITQ(IQ)
```

기본적으로 기본 이니시에이션 큐, SYSTEM.CHANNEL.INITQ를 사용하여 자동으로 채널 이니시에이터가 시작됩니다. 모든 채널 시작기를 수동으로 시작하려면 다음 단계를 수행하십시오.

1. 큐 관리자를 작성한 후 시작하십시오.
2. 큐 관리자의 SCHINIT 등록 정보를 MANUAL로 변경하십시오.
3. 큐 관리자를 종료한 후 재시작하십시오.

In Linux and 윈도우 systems, a channel initiator is started automatically. 시작할 수 있는 채널 시작기 수는 제한적입니다. 기본값 및 최대값은 3입니다. UNIX and Linux 시스템의 qm.ini 파일에서 MAXINITIATORS를 사용하고 Windows 시스템의 경우 레지스트리에서 이를 변경할 수 있습니다.

채널 시작기 명령 **runmqchi** 및 기타 제어 명령의 실행에 대한 자세한 내용은 [WebSphere MQ 제어 명령](#)을 참조하십시오.

채널 시작기 중지

큐 관리자를 시작하면 기본 채널 시작기가 자동으로 시작됩니다. 큐 관리자가 중지되면 모든 채널 시작기가 자동으로 중지됩니다.

초기화 및 구성 파일

채널 초기화 데이터 핸들링 양식은 사용자의 WebSphere MQ 플랫폼에 따라 달라집니다.

Windows, UNIX and Linux 시스템

In WebSphere MQ for Windows, UNIX and Linux systems, there are 구성 파일 to hold basic configuration information about the WebSphere MQ installation.

두 개의 구성 파일이 있는데, 하나는 시스템에 적용되고 다른 하나는 개별 큐 관리자에 적용됩니다.

WebSphere MQ 구성 파일

이 파일에는 WebSphere MQ 시스템의 모든 큐 관리자에 관련된 정보가 들어 있습니다. 이 파일의 이름은 mqs.ini입니다. It is fully described in the [관리](#) for WebSphere MQ for Windows, UNIX and Linux systems.

큐 관리자 구성 파일

이 파일에는 특정 단일 큐 관리자와 관련된 구성 정보가 포함됩니다. 이 파일을 `qm.ini`라고 합니다.

이 파일은 큐 관리자 작성 중에 작성되며 큐 관리자의 모든 측면과 관련된 구성 정보를 저장할 수 있습니다. 파일에 보관된 정보에는 로그의 구성이 WebSphere MQ 구성 파일의 기본값과 얼마나 다른지에 대한 자세한 내용이 포함됩니다.

큐 관리자 구성 파일은 큐 관리자가 있는 디렉토리 트리의 루트에 저장됩니다. 예를 들어, `DefaultPath` 속성의 경우 큐 관리자 `QMNAME`의 큐 관리자 구성 파일은 다음과 같습니다.

UNIX and Linux 시스템의 경우:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

`qm.ini` 파일의 일부는 다음과 같습니다. 여기서는 TCP/IP 리스너가 포트 2500에서 대기하고 최대 현재 채널 수가 200이며 최대 활성 채널 수는 100인 것으로 지정됩니다.

```
TCP:
  Port=2500
CHANNELS:
  MaxChannels=200
  MaxActiveChannels=100
```

아웃바운드 채널이 사용할 TCP/IP 포트 범위를 지정할 수 있습니다. 한 가지 방법은 `qm.ini` 파일을 사용하여 포트 값 범위의 시작과 끝을 지정하는 것입니다. 다음 예는 채널의 범위를 지정하는 `qm.ini` 파일을 표시합니다.

```
TCP:
  StrPort=2500
  EndPort=3000
CHANNELS:
  MaxChannels=200
  MaxActiveChannels=100
```

`StrPort` 또는 `EndPort`의 값을 지정할 경우, 둘 다에 대해 값을 지정해야 합니다. `EndPort`의 값은 항상 `StrPort`의 값보다 커야 합니다.

채널은 지정된 범위의 각 포트 값을 사용하려고 합니다. 성공적으로 연결되면, 포트 값은 채널이 사용할 수 있는 포트가 됩니다.

Windows 시스템의 경우:

```
C:\Program Files\IBM\WebSphere MQ\qmgrs\QMNAME\qm.ini
```

`qm.ini` 파일에 대한 자세한 정보는 [분산 큐잉에 대한 구성 파일 스탠자](#)를 참조하십시오.

메시지에 대한 데이터 변환

다른 큐 관리자의 큐 사이에서 송신할 때 WebSphere MQ 메시지에서 데이터 변환이 필요할 수 있습니다.

WebSphere MQ 메시지는 다음의 두 부분으로 구성됩니다.

- 메시지 디스크립터 내의 제어 정보
- 애플리케이션 데이터

다른 큐 관리자의 큐 사이에 보내려면 두 파트 중 하나에 데이터 변환이 필요할 수 있습니다. 애플리케이션 데이터 변환에 대한 정보는 [애플리케이션 데이터 변환](#)을 참조하십시오.

사용자 고유의 메시지 채널 에이전트 작성

WebSphere MQ를 사용하여 고유의 MCA(Message Channel Agent) 프로그램을 쓰거나 독립 소프트웨어 벤더에서 하나를 설치할 수 있습니다.

고유 MCA 프로그램을 작성하여 WebSphere MQ가 고유 독점 통신 프로토콜에 대해 상호 운용하거나 WebSphere MQ가 지원하지 않는 프로토콜에 메시지를 송신할 수 있습니다. (다른 측의 WebSphere MQ 제공 MCA와 상호 운영하기 위해 사용자의 MCA를 작성할 수는 없습니다.)

WebSphere MQ에서 제공되지 않은 MCA를 사용하도록 결정한 경우 다음 사항을 고려해야 합니다.

메시지 송신 및 수신

애플리케이션이 메시지를 넣는 어느 위치에서나(예를 들어, 전송 큐에서) 메시지를 가져오고 통신하려는 프로토콜에서 전송하는 송신 애플리케이션을 작성해야 합니다. 이 프로토콜에서 메시지를 가져와 목적지 큐에 넣는 수신 애플리케이션도 작성해야 합니다. 송신 및 수신 애플리케이션은 특별한 인터페이스가 아닌 메시지 큐 인터페이스(MQI) 호출을 사용합니다.

메시지는 한 번만 전달되어야 합니다. 동기점 조정을 사용하여 이 전달을 도울 수 있습니다.

채널 제어 기능

채널을 제어하는 고유 관리 기능을 제공해야 합니다. 채널 구성(예: DEFINE CHANNEL 명령) 또는 모니터링(예: DISPLAY CHSTATUS)에 WebSphere MQ 채널 관리 기능을 사용할 수 없습니다.

초기화 파일

필요한 경우 고유 초기화 파일을 제공해야 합니다.

애플리케이션 데이터 변환

다른 시스템으로 보내는 메시지의 데이터 변환을 허용할 수 있습니다. 이러한 경우 애플리케이션이 메시지를 넣는 위치에 관계없이(예를 들어, 전송 큐) 메시지를 검색할 때 MQGET 호출에서 MQGMO_CONVERT 옵션을 사용하십시오.

사용자 엑시트

사용자 엑시트가 필요한지 여부를 고려하십시오. 필요하다면, WebSphere MQ가 사용한 것과 동일한 인터페이스 정의를 사용할 수 있습니다.

트리거

애플리케이션이 메시지를 전송 큐에 넣으면 메시지가 큐에 도달할 때 송신 MCA가 트리거되도록 전송 큐 속성을 설정할 수 있습니다.

채널 시작기

사용자 고유의 채널 시작기를 제공해야 할 수도 있습니다.

분산 큐 관리를 위해 고려해야 할 기타 사항

분산 큐 관리를 위한 WebSphere MQ 준비 시 고려할 다른 주제. 이 주제에서는 미전달 메시지 큐, 사용 중인 큐, 시스템 확장 기능, 사용자 엑시트 프로그램과 실행 중인 채널 및 리스너를 신뢰성 있는 애플리케이션으로 다룹니다.

미전달 메시지 큐

미전달 메시지 큐에 도착하는 메시지(데드-레터 큐 또는 DLQ라고도 함)가 처리되었는지 확인하려면, 이러한 메시지를 처리하기 위해 정기적으로 실행 또는 트리거될 수 있는 프로그램을 작성하십시오. DLQ 핸들러는 UNIX and Linux 시스템의 WebSphere MQ와 함께 제공됩니다. 자세한 정보는 [샘플 DLQ 핸들러](#), `amqsdlq`를 참조하십시오.

사용 중인 큐

메시지가 전송 중이 아니더라도 수신자 채널용 MCA는 목적지 큐를 계속 열어둘 수 있습니다. 이 경우 큐가 "사용 중"으로 표시됩니다.

최대 채널 수

[분산 큐잉의 구성 파일 스탠자](#)를 참조하십시오.

시스템 확장 및 사용자 엑시트 프로그램

메시지 처리 중 정의된 시간에 기타 프로그램을 실행할 수 있는 기능이 채널 정의에 제공됩니다. 이 프로그램은 WebSphere MQ와 함께 제공되지 않지만, 로컬 요구사항에 따라 각 설치에서 제공될 수 있습니다.

이러한 사용자 엑시트 프로그램을 실행하려면 사전정의된 이름이 있어야 하며 채널 프로그램에 대한 호출에서 사용 가능해야 합니다. 사용자 엑시트 프로그램의 이름은 메시지 채널 정의에 포함되어 있습니다.

이 프로그램에 대한 제어 핸들링 및 이 프로그램으로부터의 제어 리턴 핸들링을 위해 정의된 제어 블록 인터페이스가 있습니다.

이러한 프로그램이 호출되는 정확한 위치 및 제어 블록 및 이름의 세부사항은 [메시징 채널의 채널 엑시트 프로그램](#)에서 찾을 수 있습니다.

채널 및 리스너를 신뢰 애플리케이션으로 실행

환경에서 성능이 중요한 고려사항이고 환경이 안정적인 경우, FASTPATH 바인딩을 사용하여 채널과 리스너를 신뢰할 수 있는 것으로 실행할 수 있습니다. 채널 및 리스너를 신뢰 상태로 실행할지 여부에 영향을 주는 요인은 두 가지가 있습니다.

- 환경 변수 MQ_CONNECT_TYPE=FASTPATH 또는 MQ_CONNECT_TYPE=STANDARD. 대소문자를 구분합니다. 올바르게 않은 값을 지정하면 무시됩니다.
- qm.ini 또는 레지스트리 파일의 Channels 스탠자의 MQIBindType. 이를 FASTPATH 또는 STANDARD로 설정하고 대소문자를 구분하지 않습니다. 기본값은 STANDARD입니다.

MQIBindType을 환경 변수와 함께 사용하면 다음과 같이 필요한 결과를 얻을 수 있습니다.

MQIBindType	환경 변수	결과
STANDARD	UNDEFINED	STANDARD
FASTPATH	UNDEFINED	FASTPATH
STANDARD	STANDARD	STANDARD
FASTPATH	STANDARD	STANDARD
STANDARD	FASTPATH	STANDARD
FASTPATH	FASTPATH	FASTPATH
STANDARD	클라이언트	클라이언트
FASTPATH	클라이언트	STANDARD
STANDARD	LOCAL	STANDARD
FASTPATH	LOCAL	STANDARD

즉, 채널과 리스너를 실제로 신뢰 상태로 실행하는 방법은 다음과 같은 두 가지가 있습니다.

1. qm.ini 또는 레지스트리에 MQIBindType=FASTPATH를 지정하고 환경 변수는 지정하지 않습니다.
2. qm.ini 또는 레지스트리에 MQIBindType=FASTPATH를 지정하고 환경 변수를 FASTPATH로 설정합니다.

리스너는 안정적인 프로세스이므로 리스너를 신뢰 상태로 실행할 수 있습니다. 불안정한 채널 엑시트나 명령 STOP CHANNEL MODE(TERMINATE)를 사용하지 않는 한 채널을 신뢰 상태로 실행할 수 있습니다.

UNIX, Linux, and Windows에서 채널 모니터링 및 제어

DQM의 경우 리모트 큐 관리자에 대한 채널을 작성, 모니터, 제어해야 합니다. 명령, 프로그램, IBM WebSphere MQ Explorer, 채널 정의 파일, 동기화 정보 스토리지 영역을 사용하여 채널을 제어할 수 있습니다.

다음 유형의 명령을 사용할 수 있습니다.

IBM WebSphere MQ 명령(MQSC)

Windows, UNIX and Linux 시스템의 MQSC 세션에 있는 단일 명령으로 MQSC를 사용할 수 있습니다. 보다 복잡한 명령이나 복수 명령을 실행하려면 MQSC를 파일에 빌드하여 명령행에서 해당 파일을 실행할 수 있습니다. 자세한 내용은 [MQSC 명령](#)을 참조하십시오. 이 절에서는 분산 큐잉을 위해 MQSC를 사용하는 간단한 몇 가지 예를 제공합니다.

채널 명령은 IBM WebSphere MQ 명령(MQSC)의 서브세트입니다. MQSC 및 제어 명령을 사용하여 다음을 수행할 수 있습니다.

- 채널 정의 작성, 복사, 표시, 변경, 삭제
- 채널 시작 및 중지, ping, 채널 순서 번호 재설정, 링크를 재설정할 수 없는 경우 인다우트(in-doubt) 메시지 해결
- 채널에 대한 상태 정보 표시

제어 명령

일부 기능의 경우 명령행으로 제어 명령을 실행할 수도 있습니다. 자세한 내용은 [제어 명령](#)을 참조하십시오.

프로그래밍 가능 명령 형식 명령

자세한 내용은 PCF 명령을 참조하십시오.

IBM WebSphere MQ Explorer

UNIX, Linux 및 시스템은 IBM WebSphere MQ Explorer를 사용할 수 있습니다. 제어 명령이나 MQSC 명령을 사용하지 않고 관리 태스크를 수행하기 위해 그래픽 관리 인터페이스를 제공합니다. 채널 정의는 큐 관리자 오브젝트로 보유됩니다.

각 큐 관리자에는 호환 가능한 리모트 큐 관리자에 대한 상호 연결을 제어하기 위한 DQM 컴포넌트가 있습니다. 스토리지 영역에는 순서 번호와 논리적 작업 단위(LUW) ID가 저장됩니다. 이러한 영역은 채널 동기화 목적으로 사용됩니다.

서로 다른 유형의 명령을 사용하여 메시지 채널을 설정하고 제어할 때 사용하는 기능들의 목록에 대해서는 [68 페이지의 표 8](#)의 내용을 참조하십시오.

관련 개념

[69 페이지의 『오브젝트 시작하기』](#)

채널을 시작하려면 채널이 정의되고 연관된 오브젝트가 존재해야 하며 해당 채널을 사용할 수 있어야 합니다. 이 절은 그 방법을 보여줍니다.

[75 페이지의 『Windows용 통신 설정』](#)

분산 큐잉 관리 채널이 시작되면 이 채널에서 채널 정의에 지정된 연결을 사용하려 합니다. 이를 위해서는 연결이 정의되고 사용 가능해야 합니다. This section explains how to do this using one of the four forms of communication for WebSphere MQ for 윈도우 systems.

[84 페이지의 『UNIX and Linux 시스템에서 통신 설정』](#)

DQM은 IBM WebSphere MQ의 리모트 큐잉 기능입니다. 이 기능은 시스템 운영자가 제어할 수 있는 통신 링크에 대한 인터페이스를 형성하는 큐 관리자용 채널 제어 프로그램을 제공합니다. 분산 큐잉 관리로 유지되는 채널 정의에서 이러한 연결을 사용합니다.

관련 참조

[UNIX, Linux 및 Windows 시스템의 채널 프로그램](#)

[분산 플랫폼용 메시지 채널 계획 예](#)

[구성 정보 예](#)

[채널 속성](#)

채널 설정 및 제어에 필요한 기능

채널을 설정하고 제어하려면 여러 가지 IBM WebSphere MQ 기능이 필요합니다. 이 주제에서는 채널 기능을 설명합니다.

IBM WebSphere MQ가 제공하는 기본값으로 채널 이름, 작성하는 채널 유형, 사용할 통신 방법, 전송 큐 이름과 연결 이름을 지정하여 채널 정의를 작성할 수 있습니다.

채널 이름은 채널 양쪽에서 동일하고 네트워크 내에서 고유해야 합니다. 그러나 사용되는 문자를 IBM WebSphere MQ 오브젝트 이름에 유효한 문자로 제한해야 합니다.

다른 채널 관련 기능은 다음 주제를 참조하십시오.

- [69 페이지의 『오브젝트 시작하기』](#)
- [70 페이지의 『연관된 오브젝트 작성』](#)
- [70 페이지의 『기본 오브젝트 작성』](#)

- 70 페이지의 『채널 작성』
- 71 페이지의 『채널 표시』
- 71 페이지의 『채널 상태 표시』
- 72 페이지의 『Ping을 사용하여 링크 검사』
- 72 페이지의 『채널 시작』
- 73 페이지의 『채널 중지』
- 74 페이지의 『채널 이름 바꾸기』
- 74 페이지의 『채널 재설정』
- 75 페이지의 『채널상의 인다우트(in-doubt) 메시지 해결』

68 페이지의 표 8은 필요한 IBM WebSphere MQ 기능의 전체 목록을 보여줍니다.

표 8. UNIX, Linux, and Windows 시스템에 필요한 기능			
Function	제어 명령	MQSC	WebSphere MQ Explorer 상응 여부
큐 관리자 기능			
큐 관리자 변경		ALTER QMGR	예
큐 관리자 작성	crtmqm		예
큐 관리자 삭제	dlmqm		예
큐 관리자 표시		DISPLAY QMGR	예
큐 관리자 종료	endmqm		예
큐 관리자 ping		PING QMGR	아니오
큐 관리자 시작	strmqm		예
명령 서버 기능			
명령 서버 표시	dspmqcsv		아니오
명령 서버 종료	endmqcsv		아니오
명령 서버 시작	strmqcsv		아니오
큐 기능			
큐 변경		ALTER QALIAS ALTER QLOCAL ALTER QMODEL ALTER QREMOTE ALTER 큐를 참조하십시오.	예
큐 지우기		CLEAR QLOCAL	예
큐 작성		DEFINE QALIAS DEFINE QLOCAL DEFINE QMODEL DEFINE QREMOTE DEFINE 큐를 참조하십시오.	예

표 8. UNIX, Linux, and Windows 시스템에 필요한 기능 (계속)

Function	제어 명령	MQSC	WebSphere MQ Explorer 상응 여부
큐 삭제		DELETE QALIAS DELETE QLOCAL DELETE QMODEL DELETE QREMOTE DELETE 큐를 참조하십시오.	예
큐 표시		DISPLAY QUEUE	예
프로세스 기능			
프로세스 변경		ALTER PROCESS	예
프로세스 작성		DEFINE PROCESS	예
프로세스 삭제		프로세스 삭제	예
프로세스 표시		DISPLAY PROCESS	예
채널 기능			
채널 변경		ALTER CHANNEL	예
채널 작성		DEFINE CHANNEL	예
채널 삭제		채널 삭제	예
채널 표시		DISPLAY CHANNEL	예
채널 상태 표시		DISPLAY CHSTATUS	예
채널 종료		STOP CHANNEL	예
채널 ping		PING CHANNEL	예
채널 재설정		RESET CHANNEL	예
채널 해결		RESOLVE CHANNEL	예
채널 실행	runmqchl	START CHANNEL	예
채널 시작기 실행	runmqchi	START CHINIT	아니오
리스너 실행 ¹	runmqlsr	START LISTENER	아니오
리스너 종료	endmqlsr(시스템, AIX, HP-UX 및 Solaris 전용)		아니오
참고:			
1. 큐 관리자가 시작될 때 리스너가 자동으로 시작될 수 있습니다.			

오브젝트 시작하기

채널을 시작하려면 채널이 정의되고 연관된 오브젝트가 존재해야 하며 해당 채널을 사용할 수 있어야 합니다. 이 절은 그 방법을 보여줍니다.

WebSphere MQ 명령(MQSC) 또는 IBM WebSphere MQ Explorer를 사용하여 다음을 수행하십시오.

1. 메시지 채널 및 연관된 오브젝트 정의
2. 메시지 채널 모니터 및 제어

정의해야 하는 연관된 오브젝트는 다음과 같습니다.

- 전송 큐
- 리모트 큐 정의
- 큐 관리자 알리어스 정의
- 응답 대상 큐 알리어스 정의
- 응답 대상 로컬 큐
- 트리거용 프로세스(MCA)
- 메시지 채널 정의

채널을 실행하려면 각 채널에 대한 특정 통신 링크를 정의하고 사용 가능해야 합니다. LU 6.2, TCP/IP, NetBIOS, SPX, DECnet 링크 정의 방법에 대한 설명은 사용자 설치에 맞는 특정 통신 안내서를 참조하십시오. [구성 정보 예](#)도 참조하십시오.

오브젝트 작성 및 관련 작업에 대한 자세한 정보는 다음 하위 주제를 참조하십시오.

연관된 오브젝트 작성

MQSC를 사용하여 연관된 오브젝트를 작성할 수 있습니다.

MQSC를 사용하여 큐 및 알리어스 오브젝트(전송 큐, 리모트 큐 정의, 큐 관리자 알리어스 정의, 응답 대상 큐 알리어스 정의, 응답 대상 로컬 큐)를 작성하십시오.

또한 유사한 방법으로 트리거(MCA) 프로세스 정의를 작성하십시오.

필수 오브젝트를 모두 작성하는 방법을 보여주는 예제는 [분산 플랫폼의 메시지 채널 계획 예제](#)를 참조하십시오.

기본 오브젝트 작성

큐 관리자가 작성될 때 기본 오브젝트가 자동으로 작성됩니다. 해당 오브젝트는 큐, 채널, 프로세스 정의, 관리 큐입니다. 기본 오브젝트가 작성된 후에는 언제든지 -c 옵션과 strmqm 명령을 실행하여 오브젝트를 바꿀 수 있습니다.

crtmqm 명령을 사용하여 큐 관리자를 작성하는 경우 이 명령은 또한 기본 오브젝트 세트를 작성하는 프로그램을 시작합니다.

1. 각 기본 오브젝트가 차례로 작성됩니다. 프로그램은 성공적으로 정의되는 오브젝트 수, 바뀐 기존 오브젝트 수와 시도 실패 횟수를 보존합니다.
2. 프로그램은 그 결과를 사용자에게 표시하며 오류가 발생하면 세부사항을 나타내는 적절한 오류 로그를 제시합니다.

프로그램 실행이 완료되면 strmqm 명령을 사용하여 큐 관리자를 시작할 수 있습니다.

crtmqm 및 strmqm 명령에 대한 자세한 정보는 [제어 명령](#)을 참조하십시오.

기본 오브젝트 변경

-c 옵션을 지정하는 경우 큐 관리자는 오브젝트가 작성되는 동안 임시로 시작되고 다시 종료됩니다. -c 옵션을 사용하여 strmqm을 실행하면 기존 시스템 오브젝트가 기본값으로 새로 고쳐집니다(예를 들어, 채널 정의의 MCAUSER 속성이 공백으로 설정됨). 큐 관리자를 시작하려면 -c 옵션 없이 strmqm 명령을 다시 사용해야 합니다.

기본 오브젝트를 변경하려는 경우 이전 amqscoma.tst 파일의 사용자 고유 버전을 작성하여 편집하면 됩니다.

채널 작성

연결의 각 측에 하나씩 두 개의 채널 정의를 작성할 수 있습니다. 첫 번째 큐 관리자에서 첫 번째 채널 정의를 작성합니다. 그런 다음 링크의 다른 측에 있는 두 번째 큐 관리자에서 두 번째 채널 정의를 작성합니다.

양 측을 모두 **동일한** 채널 이름을 사용해서 정의해야 합니다. 양 측에 **호환 가능한** 채널 유형(예: 송신자 및 수신자)이 있어야 합니다.

링크의 한쪽 끝에 대해 채널 정의를 작성하려면 MQSC 명령 DEFINE CHANNEL을 사용하십시오. 채널 이름, 연결의 이 측에 대한 채널 유형, 연결 이름, 설명(필요한 경우), 전송 큐의 이름(필요한 경우), 전송 프로토콜을 포함하

십시오. 또한 이전에 수집한 정보를 사용하여 필요한 채널 유형에 대해 시스템 기본값과 다르게 설정하려는 모든 속성을 포함하십시오.

채널 속성 값을 결정하기 위한 도움말이 [채널 속성](#)에 제공됩니다.

참고: 네트워크에서 모든 채널의 이름을 고유하게 지정하는 것이 좋습니다. 채널 이름에 소스 및 대상 큐 관리자 이름을 포함하는 것이 효과적인 방법입니다.

채널 작성 예

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) +
DESCR('Sender channel to QM2') +
CONNNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

모든 MQSC 예에서, 명령은 명령 파일에 나타나는 대로, 그리고 Windows 또는 UNIX 또는 Linux 시스템에 입력하는 대로 표시됩니다. 명령을 대화식으로 실행하기 위해서는 먼저 MQSC 세션을 시작해야 한다는 것을 제외하고는 두 가지 방법이 동일하게 보입니다. 기본 큐 관리자의 경우 `runmqsc`, 그렇지 않을 경우 `runmqsc qmname`(여기서 `qmname`은 필요한 큐 관리자의 이름)을 입력하십시오. 그런 다음, 예에 나타난 것처럼 명령을 입력하십시오.

이식성을 위해 명령의 행 길이는 72자로 제한하십시오. 두 개 이상의 행을 계속하려면 표시된 대로 연결 문자 `+`을 사용하십시오. Windows에서는 Ctrl-z를 사용하여 명령행에서의 입력 항목을 종료하십시오. UNIX and Linux 시스템에서는 Ctrl-d를 사용하십시오. UNIX, Linux 또는 Windows 시스템에서 **end** 명령을 사용하십시오.

채널 표시

MQSC 명령 `DISPLAY CHANNEL`을 사용하여 채널 속성을 표시할 수 있습니다.

특정 속성이 필요하지 않고 지정된 채널 이름이 일반적이지 않은 경우 기본적으로 `DISPLAY CHANNEL` 명령의 `ALL` 매개변수를 가정합니다.

속성은 [채널 속성](#)에 설명되어 있습니다.

채널 표시 예

```
DISPLAY CHANNEL(QM1.TO.QM2) TRPTYPE,CONVERT
DISPLAY CHANNEL(QM1.TO.*) TRPTYPE,CONVERT
DISPLAY CHANNEL(*) TRPTYPE,CONVERT
DISPLAY CHANNEL(QM1.TO.QMR34) ALL
```

채널 상태 표시

채널 이름과, 현재 채널 상태 또는 저장된 정보 상태가 필요한지 여부를 지정하여 MQSC 명령 `DISPLAY CHSTATUS`를 사용합니다.

`DISPLAY CHSTATUS`는 모든 메시지 채널에 적용됩니다. 서버-연결 채널 이외의 MQI 채널에는 적용되지 않습니다.

표시되는 정보는 다음과 같습니다.

- 채널 이름
- 통신 연결 이름
- 채널의 인다우트(In-doubt) 상태(해당되는 경우)
- 마지막 순서 번호
- 전송 큐 이름(해당되는 경우)
- 인다우트(in-doubt) ID(해당되는 경우)
- 마지막으로 커미트된 순서 번호
- 논리적 작업 단위 ID
- 프로세스 ID

- 스레드 ID ()

채널 상태 표시 예

```
DISPLAY CHSTATUS(*) CURRENT
DISPLAY CHSTATUS(QM1.TO.*) SAVED
```

저장된 상태는 최소 하나의 메시지 배치가 채널에서 전송될 때까지 적용되지 않습니다. 채널이 중지되고(STOP CHL 명령 사용) 큐 관리자가 종료될 때에도 상태가 저장됩니다.

Ping을 사용하여 링크 검사

MQSC 명령 PING CHANNEL을 사용하여 고정 데이터 메시지를 원격 끝과 교환할 수 있습니다.

ping은 시스템 감독자에게 링크를 사용할 수 있고 작동한다는 확신을 줄 수 있습니다.

Ping은 전송 큐와 대상 큐를 사용하지 않습니다. 채널 정의, 관련 통신 링크, 네트워크 설정을 사용합니다. 채널이 현재 활성 상태가 아닌 경우에만 사용할 수 있습니다.

이는 송신자 및 서버 채널에서만 사용 가능합니다. 해당 채널은 링크의 먼 측에서 시작되며 시동 매개변수 조정을 수행합니다. 오류는 정상적으로 보고됩니다.

메시지 교환 결과는 Ping complete 또는 오류 메시지로 표시됩니다.

LU 6.2에서의 Ping

ping이 호출될 때 기본적으로 사용자 ID 또는 비밀번호가 수신측으로 이동하지 않습니다. 사용자 ID와 비밀번호가 필요한 경우에는 채널 정의의 시작하는 한쪽 끝에서 작성될 수 있습니다. 비밀번호가 채널 정의에 입력되면, 저장하기 전에 WebSphere MQ가 이를 암호화합니다. 그런 다음 대화에서 이동하기 전에 암호화를 해제합니다.

채널 시작

MQSC 명령 START CHANNEL은 송신자, 서버, 요청자 채널에 사용합니다. 애플리케이션이 메시지를 교환하려면 인바운드 연결을 위해 리스너 프로그램을 시작해야 합니다.

큐 관리자 트리거로 채널이 설정된 경우에는 START CHANNEL이 필요하지 않습니다.

시작 시 송신 MCA가 채널 정의를 읽고 전송 큐를 엽니다. 채널 시작 순서가 발행되어 수신자 또는 송신자 채널의 해당 MCA가 원격으로 시작됩니다. 시작된 후에는 송신자 및 서버 프로세스가 전송 큐에 도착하는 메시지를 대기 하고 도착 시 메시지를 전송합니다.

트리거를 사용하거나 채널을 스레드로 실행하는 경우, 시작 큐를 모니터링하기 위해 채널 시작기를 사용할 수 있는지 확인하십시오. 기본적으로 채널 시작기는 큐 관리자의 일부분으로 시작됩니다.

그러나 TCP와 LU 6.2는 다른 기능을 제공합니다.

- UNIX and Linux 시스템의 TCP의 경우, 채널을 시작하도록 inetd가 구성될 수 있습니다. inetd는 별도 프로세스로 시작됩니다.
- UNIX and Linux 시스템의 LU 6.2의 경우, SNA 제품이 LU 6.2 응답자 프로세스를 시작하도록 구성하십시오.
- SNA Server를 사용하는 Windows 시스템의 LU 6.2의 경우, TpStart (SNA Server와 함께 제공되는 유틸리티)를 사용하여 채널을 시작할 수 있습니다. TpStart는 별도 프로세스로 시작됩니다.

필요한 경우 시작 옵션을 사용하면 항상 채널이 재동기화됩니다.

정상적으로 시작하려면,

- 로컬 및 원격 채널 정의가 존재해야 합니다. 수신자 또는 서버 연결 채널에 대해 적합한 채널 정의가 없으면, 채널이 자동 정의될 때 기본 정의가 자동으로 작성됩니다. [채널 자동 정의 엑시트 프로그램을 참조하십시오.](#)
- 전송 큐가 존재해야 하며 해당 큐를 사용하는 다른 채널이 없어야 합니다.
- 로컬 및 원격 MCA가 존재해야 합니다.
- 통신 링크가 사용 가능해야 합니다.

- 로컬 및 리모트 큐 관리자가 실행 중이어야 합니다.
- 메시지 채널이 이미 실행되고 있으면 안됩니다.

채널 시작 요청이 승인되었음을 확인하는 메시지가 화면에 리턴됩니다. 시작 명령이 성공했는지 확인하려면 오류 로그를 검사하거나 DISPLAY CHSTATUS를 사용하십시오. 오류 로그는 다음과 같습니다.

Windows

`MQ_INSTALLATION_PATH\mqgrs\qmname\errors\AMQERR01.LOG`(qmname이라는 각 큐 관리자의 경우)

`MQ_INSTALLATION_PATH\mqgrs\@SYSTEM\errors\AMQERR01.LOG` (일반 오류의 경우)

`MQ_INSTALLATION_PATH`는 WebSphere MQ가 설치되어 있는 상위 레벨 디렉토리를 표시합니다.

참고: Windows 시스템에서는 Windows 시스템 애플리케이션 이벤트 로그에도 메시지가 표시됩니다.

UNIX and Linux 시스템

`/var/mqm/qmgrs/qmname/errors/AMQERR01.LOG`(qmname이라는 각 큐 관리자의 경우)

`/var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG` (일반 오류의 경우)

Windows, UNIX and Linux 시스템에서는 `runmqtsr` 명령을 사용하여 WebSphere MQ 리스너 프로세스를 시작하십시오. 기본적으로 채널 연결에 대한 인바운드 요청으로 인해 리스너 프로세스가 MCA를 `amqrmppa` 프로세스의 스레드로 시작합니다.

```
runmqtsr -t tcp -m QM2
```

아웃바운드 연결의 경우, 다음 세 가지 방법 중 하나로 채널을 시작해야 합니다.

1. MQSC 명령 `START CHANNEL`로 채널 이름을 지정하여 `MCATYPE` 매개변수에 따라 채널을 프로세스 또는 스레드로 시작하십시오. 채널이 스레드로 시작되면 채널은 채널 시작기의 스레드가 됩니다.

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

2. 제어 명령 `runmqchl`을 사용하여 채널을 프로세스로 시작하십시오.

```
runmqchl -c QM1.TO.QM2 -m QM1
```

3. 채널 시작기를 사용하여 채널을 트리거하십시오.

채널 중지

MQSC 명령 `STOP CHANNEL`을 사용하여 채널이 활동을 중지하도록 요청할 수 있습니다. 운영자가 채널을 다시 시작할 때까지 채널이 새 메시지 배치를 시작하지 않습니다.

정지된 채널 재시작에 대해서는 57 페이지의 『중지된 채널 재시작』을 참조하십시오.

이 명령은 `MQCHT_CLNTCONN`을 제외한 모든 채널 유형에 실행될 수 있습니다.

필요한 중지 유형을 선택할 수 있습니다.

일시정지 예

```
STOP CHANNEL(QM1.TO.QM2) MODE(QUIESCE)
```

이 명령은 채널을 순서대로 단도록 요청합니다. 현재 메시지 배치가 완료되고 채널의 다른 한쪽 끝과 함께 동기점 프로시저가 수행됩니다. 채널이 유힬 상태이면 이 명령이 수신 채널을 종료하지 않습니다.

강제 중지 예

```
STOP CHANNEL(QM1.TO.QM2) MODE(FORCE)
```

이 옵션은 채널을 즉시 중지하지만 채널의 스레드 또는 프로세스는 종료하지 않습니다. 채널이 현재 메시지 배치 처리를 완료하지 않으므로 채널을 인다우트(in-doubt) 상태로 남겨둘 수 있습니다. 일반적으로 일시정지 옵션 사용을 고려하십시오.

종료 중지 예

```
STOP CHANNEL(QM1.TO.QM2) MODE(TERMINATE)
```

이 옵션은 채널을 즉시 중지하며 채널의 스레드 또는 프로세스를 종료합니다.

중지됨 상태의 중지(일시정지) 예

```
STOP CHANNEL(QM1.TO.QM2) STATUS(STOPPED)
```

이 명령은 MODE를 지정하지 않으므로 기본값은 MODE(QUIESCE)입니다. 채널이 중지되어 자동으로 재시작할 수 없고 수동으로 시작해야 하는 것으로 요청합니다.

비활성 상태의 중지(일시정지) 예

```
STOP CHANNEL(QM1.TO.QM2) STATUS(INACTIVE)
```

이 명령은 MODE를 지정하지 않으므로 기본값은 MODE(QUIESCE)입니다. 채널이 비활성화되어 필요할 때 자동으로 재시작되는 것으로 요청합니다.

채널 이름 바꾸기

MQSC를 사용하여 메시지 채널의 이름을 바꿀 수 있습니다.

MQSC를 사용하여 다음 단계를 수행하십시오.

1. STOP CHANNEL을 사용하여 채널을 중지하십시오.
2. DEFINE CHANNEL을 사용하여 새 이름으로 중복 채널 정의를 작성하십시오.
3. DISPLAY CHANNEL을 사용하여 올바르게 작성되었는지 확인하십시오.
4. DELETE CHANNEL을 사용하여 원래 채널 정의를 삭제하십시오.

메시지 채널의 이름을 변경하기로 한 경우, 채널이 한 측에 하나씩 두 개의 채널 정의를 가지고 있다는 점을 기억하십시오. 양쪽에서 동시에 채널의 이름을 바꾸었는지 확인하십시오.

채널 재설정

MQSC 명령 RESET CHANNEL을 사용해서 메시지 순서 매기기 번호를 변경할 수 있습니다.

RESET CHANNEL 명령은 메시지 채널에는 사용할 수 있지만 MQI 채널(클라이언트-연결 또는 서버-연결)에는 사용할 수 없습니다. 채널이 시작된 후에 첫 메시지가 새로운 순서를 시작합니다.

이 명령이 송신자 또는 서버 채널에서 실행되면 채널이 재시작될 때 다른 측에 변경사항을 알려줍니다.

관련 개념

[69 페이지의 『오브젝트 시작하기』](#)

채널을 시작하려면 채널이 정의되고 연관된 오브젝트가 존재해야 하며 해당 채널을 사용할 수 있어야 합니다. 이 절은 그 방법을 보여줍니다.

[49 페이지의 『채널 제어 기능』](#)

채널 제어 기능은 채널을 정의, 모니터, 제어할 수 있는 기능을 제공합니다.

[25 페이지의 『분산 큐잉을 사용하여 애플리케이션 연결』](#)

이 절에서는 큐 정의, 채널 정의, 트리거 및 동기점 프로시저를 포함한 WebSphere MQ 설치 간 상호통신에 관한 자세한 정보를 제공합니다.

관련 참조

[RESET CHANNEL](#)

채널상의 인다우트(*in-doubt*) 메시지 해결

송신자 또는 서버가 메시지를 인다우트(*in-doubt*) 상태로 유지할 때 MQSC 명령 RESOLVE CHANNEL을 사용합니다. 예를 들어, 링크의 한쪽이 종료되고 복구될 가능성이 없는 경우입니다.

RESOLVE CHANNEL 명령은 BACKOUT이나 COMMIT의 두 매개변수 중 하나를 수용합니다. 커미트가 메시지를 제거하는 반면 백아웃은 메시지를 전송 큐로 복원합니다.

채널 프로그램은 파트너와의 세션 설정을 시도하지 않습니다. 대신 인다우트(*in-doubt*) 메시지를 표시하는 논리적 작업 단위 ID(LUWID)를 판별합니다. 그런 다음 필요할 경우 다음 중 하나를 내보냅니다.

- BACKOUT: 전송 큐로 메시지를 복원합니다.
- COMMIT: 전송 큐에서 메시지를 삭제합니다.

해석이 성공하려면 다음 조건이 충족되어야 합니다.

- 채널이 비활성화되어 있어야 합니다.
- 채널이 인다우트(*in-doubt*) 상태여야 합니다.
- 채널 유형이 송신자 또는 서버여야 합니다.
- 로컬 채널 정의가 존재해야 합니다.
- 로컬 큐 관리자가 실행 중이어야 합니다.

관련 개념

[69 페이지의 『오브젝트 시작하기』](#)

채널을 시작하려면 채널이 정의되고 연관된 오브젝트가 존재해야 하며 해당 채널을 사용할 수 있어야 합니다. 이 절은 그 방법을 보여줍니다.

[49 페이지의 『채널 제어 기능』](#)

채널 제어 기능은 채널을 정의, 모니터, 제어할 수 있는 기능을 제공합니다.

[25 페이지의 『분산 큐잉을 사용하여 애플리케이션 연결』](#)

이 절에서는 큐 정의, 채널 정의, 트리거 및 동기점 프로시저를 포함한 WebSphere MQ 설치 간 상호통신에 관한 자세한 정보를 제공합니다.

관련 참조

[RESOLVE CHANNEL](#)

Windows용 통신 설정

분산 큐잉 관리 채널이 시작되면 이 채널에서 채널 정의에 지정된 연결을 사용하려 합니다. 이를 위해서는 연결이 정의되고 사용 가능해야 합니다. This section explains how to do this using one of the four forms of communication for WebSphere MQ for 윈도우 systems.

[Example configuration - IBM WebSphere MQ for 윈도우를 참조하는 것이 도움이 될 수 있습니다.](#)

UNIX and Linux 시스템의 경우에는 [84 페이지의 『UNIX and Linux 시스템에서 통신 설정』](#)의 내용을 참조하십시오.

연결 결정

Choose from the following four forms of communication for WebSphere MQ for 윈도우 systems:

- [76 페이지의 『Windows에서 TCP 연결 정의』](#)
- [77 페이지의 『Windows에서 LU 6.2 연결 정의』](#)
- [79 페이지의 『Windows에서 NetBIOS 연결 정의』](#)
- [82 페이지의 『Windows에서 SPX 연결 정의』](#) (윈도우 XP 및 윈도우 2003 Server 전용)

각 채널 정의는 하나의 프로토콜만 전송 프로토콜(전송 유형) 속성으로 지정해야 합니다. 큐 관리자는 하나 이상의 프로토콜을 사용할 수 있습니다.

WebSphere MQ 클라이언트의 경우, 다른 전송 프로토콜을 사용하는 대체 채널이 있는 것이 유용할 수 있습니다. WebSphere MQ 클라이언트에 대한 자세한 정보는 [클라이언트 개요](#)를 참조하십시오.

관련 개념

25 페이지의 『분산 큐잉을 사용하여 애플리케이션 연결』

이 절에서는 큐 정의, 채널 정의, 트리거 및 동기점 프로시저를 포함한 WebSphere MQ 설치 간 상호통신에 관한 자세한 정보를 제공합니다.

66 페이지의 『UNIX, Linux, and Windows에서 채널 모니터링 및 제어』

DQM의 경우 리모트 큐 관리자에 대한 채널을 작성, 모니터, 제어해야 합니다. 명령, 프로그램, IBM WebSphere MQ Explorer, 채널 정의 파일, 동기화 정보 스토리지 영역을 사용하여 채널을 제어할 수 있습니다.

89 페이지의 『클라이언트와 서버 간의 연결 구성』

WebSphere MQ MQI 클라이언트와 서버 사이의 통신 링크를 구성하려면 통신 프로토콜을 결정하고 링크의 양 끝에서의 연결을 정의하고 리스너를 시작한 다음 채널을 정의하십시오.

Windows에서 TCP 연결 정의

대상의 주소를 지정하기 위해 송신측에서 채널을 구성하고 수신측에서 리스너 프로그램을 실행하여 TCP 연결을 정의합니다.

송신측

채널 정의의 연결 이름 필드에서 대상 시스템의 TCP 주소 또는 호스트 이름을 지정하십시오.

연결할 포트는 기본적으로 1414입니다. 포트 번호 1414는 iana(Internet Assigned Numbers Authority)에 의해 IBM WebSphere MQ에 지정됩니다.

기본값 이외의 포트 번호를 사용하려면 채널 오브젝트 정의의 연결 이름 필드에서 포트 번호를 지정하십시오.

```
DEFINE CHANNEL('channel name') CHLTYPE(SDR) +
    TRPTYPE(TCP) +
    CONNAME('OS2ROG3(1822)') +
    XMITQ('XMITQ name') +
    REPLACE
```

여기서 OS2ROG3은 리모트 큐 관리자의 DNS 이름이고 1822는 필요한 포트입니다. 이 포트는 수신측의 리스너가 청취하는 포트여야 합니다.

채널 오브젝트 정의 변경사항을 적용하려면 실행 채널을 중지했다가 재시작해야 합니다.

You can change the default port number by specifying it in the .ini file for IBM WebSphere MQ for Windows:

```
TCP:
Port=1822
```

참고: 사용할 TCP/IP 포트 번호를 선택하기 위해 IBM WebSphere MQ가 다음 순서에서 처음으로 발견한 포트 번호를 사용합니다.

1. 채널 정의나 명령행에 명시적으로 지정된 포트 번호. 이 번호를 사용하면 채널에 대해 기본 포트 번호를 대체할 수 있습니다.
2. .ini 파일의 TCP 스탠자에 지정된 포트 속성입니다. 이 번호를 사용하면 큐 관리자에 대해 기본 포트 번호를 대체할 수 있습니다.
3. 기본값 1414. 인터넷 할당 번호 관리 기관(IANA)이 인바운드 및 아웃바운드 연결 모두에 대해 IBM WebSphere MQ에 지정한 번호입니다.

qm.ini를 사용하여 설정하는 값에 대한 자세한 정보는 분산 큐잉에 대한 구성 파일 스탠자를 참조하십시오.

TCP에서 수신

수신 채널 프로그램을 시작하려면 리스너 프로그램이 시작되어 수신되는 네트워크 요청을 감지하고 관련 채널을 시작해야 합니다. IBM WebSphere MQ 리스너를 사용할 수 있습니다.

수신 채널 프로그램은 송신 채널의 시동 요청에 대한 응답으로 시작됩니다.

수신 채널 프로그램을 시작하려면 리스너 프로그램이 시작되어 수신되는 네트워크 요청을 감지하고 관련 채널을 시작해야 합니다. IBM WebSphere MQ 리스너를 사용할 수 있습니다.

IBM WebSphere MQ와 함께 제공되고 새 채널을 스레드로 시작하는 리스너를 실행하려면 `runmqtsr` 명령을 사용합니다.

`runmqtsr` 명령을 사용하는 기본 예:

```
runmqtsr -t tcp [-m QMNAME] [-p 1822]
```

대괄호는 선택적 매개변수를 나타냅니다. 즉 기본 큐 관리자의 경우에는 QMNAME이 필수가 아니고, 기본값(1414)을 사용하면 포트 번호가 필수가 아닙니다. 포트 번호는 65535를 초과해서는 안 됩니다.

참고: 사용할 TCP/IP 포트 번호를 선택하기 위해 IBM WebSphere MQ가 다음 순서에서 처음으로 발견한 포트 번호를 사용합니다.

1. 채널 정의나 명령행에 명시적으로 지정된 포트 번호. 이 번호를 사용하면 채널에 대해 기본 포트 번호를 대체할 수 있습니다.
2. .ini 파일의 TCP 스탠자에 지정된 포트 속성입니다. 이 번호를 사용하면 큐 관리자에 대해 기본 포트 번호를 대체할 수 있습니다.
3. 기본값 1414. 인터넷 할당 번호 관리 기관(IANA)이 인바운드 및 아웃바운드 연결 모두에 대해 IBM WebSphere MQ에 지정한 번호입니다.

성능을 극대화하려면 IBM WebSphere MQ 리스너를 신뢰 애플리케이션으로 실행하십시오(66 페이지의 『채널 및 리스너를 신뢰 애플리케이션으로 실행』의 설명 참조). 신뢰 애플리케이션에 대한 정보는 신뢰 애플리케이션에 대한 제한사항을 참조하십시오.

TCP/IP SO_KEEPALIVE 옵션 사용

Windows SO_KEEPALIVE 옵션을 사용하려면 레지스트리에 다음 항목을 추가해야 합니다.

```
TCP:
KeepAlive=yes
```

SO_KEEPALIVE 옵션에 대한 자세한 정보는 55 페이지의 『채널의 다른 쪽이 여전히 사용 가능한지 검사』의 내용을 참조하십시오.

Windows에서 Windows KeepAlive시간 옵션의 HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters 레지스트리 값은 연결이 확인되기 전에 경과되는 간격을 제어합니다. 기본값은 2시간입니다.

Windows에서 LU 6.2 연결 정의

SNA는 두 시스템 간에 LU 6.2 대화가 설정될 수 있도록 구성되어야 합니다.

SNA가 구성되면 다음과 같이 진행하십시오.

관련 정보는 다음 표를 참조하십시오.

표 9. 리모트 큐 관리자 플랫폼에 필요한 로컬 Windows 시스템 설정		
원격 플랫폼	TPNAME	TPPATH
코믹스 없는 z/OS 또는 MVS/ESA	리모트 큐 관리자에 관한 해당 부가 정보와 동일합니다.	-

표 9. 리모트 큐 관리자 플랫폼에 필요한 로컬 Windows 시스템 설정 (계속)		
원격 플랫폼	TPNAME	TPPATH
z/OS or MVS/ESA using 코믹스	CKRC(송신자) CKSV(요청자) CKRC(서버)	-
IBM i	IBM i 시스템의 라우팅 입력 항목에 있는 비교 값과 동일합니다.	-
UNIX and Linux 시스템	리모트 큐 관리자에 관한 해당 부가 정보와 동일합니다.	MQ_INSTALLATION_PATH/bin/amqcrs6a
Windows	Windows Run Listener 명령 또는 Windows에서 TpSetup을 사용하여 정의된 호출 가능한 트랜잭션 프로그램에서 지정한 대로.	MQ_INSTALLATION_PATH\bin\amqcrs6a

MQ_INSTALLATION_PATH는 WebSphere MQ가 설치되어 있는 상위 레벨 디렉토리를 표시합니다.

동일한 시스템에 둘 이상의 큐 관리자가 있다면, 반드시 채널 정의에서 TPname이 고유한지 확인하십시오.

TCP/IP를 통한 AnyNet® SNA 구성에 관한 최신 정보는 다음 온라인 IBM 문서를 참조하십시오. [AnyNet TCP/IP를 통한 SNA 및 SNA Node Operations](#).

관련 개념

78 페이지의 『LU 6.2의 송신측』

사용하는 LU 6.2 제품의 관리 애플리케이션에서 CPI-C측 오브젝트(기호 목적지)를 작성합니다. 채널 정의의 연결 이름 필드에 이 이름을 입력합니다. 또한 파트너에 대한 LU 6.2 링크도 작성합니다.

78 페이지의 『LU 6.2에서 수신』

수신 채널 프로그램은 송신 채널의 시동 요청에 대한 응답으로 시작됩니다.

LU 6.2의 송신측

사용하는 LU 6.2 제품의 관리 애플리케이션에서 CPI-C측 오브젝트(기호 목적지)를 작성합니다. 채널 정의의 연결 이름 필드에 이 이름을 입력합니다. 또한 파트너에 대한 LU 6.2 링크도 작성합니다.

CPI-C측 오브젝트에서 수신 시스템의 파트너 LU 이름, TP 이름, 모드 이름을 입력하십시오. 예를 들면, 다음과 같습니다.

```
Partner LU Name      OS2R0G2
Partner TP Name     recv
Mode Name           #INTER
```

LU 6.2에서 수신

수신 채널 프로그램은 송신 채널의 시동 요청에 대한 응답으로 시작됩니다.

수신 채널 프로그램을 시작하려면 리스너 프로그램이 수신 네트워크 요청을 감지하기 시작하고 연관된 채널을 시작해야 합니다. 대기할 TpName을 제공하여 RUNMQLSR 명령으로 이 리스너 프로그램을 시작합니다. 또는, Windows용 SNA 서버 아래에서 TpStart를 사용할 수 있습니다.

RUNMQLSR 명령 사용

리스너를 시작하는 명령 예는 다음과 같습니다.

```
RUNMQLSR -t LU62 -n RECV [-m QMNAME]
```

여기서 RECV는 다른(송신) 측에서 "원격 측에서 시작할 TpName"으로 지정되는 TpName입니다. 대괄호로 묶은 마지막 부분은 선택사항이며 기본 큐 관리자에 필요하지 않습니다.

한 시스템에서 큐 관리자가 두 개 이상 실행될 수 있습니다. 각 큐 관리자에게 다른 TpName을 할당한 후 각각에 대해 리스너 프로그램을 시작해야 합니다. 예를 들면, 다음과 같습니다.

```
RUNMQLSR -t LU62 -m QM1 -n TpName1
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

최상의 성능을 위해 신뢰할 수 있는 애플리케이션으로 채널 및 리스너 실행에 설명된 대로 WebSphere MQ 리스너를 신뢰할 수 있는 애플리케이션으로 실행하십시오. 신뢰 애플리케이션에 대한 정보는 [신뢰 애플리케이션에 대한 제한사항](#)을 참조하십시오.

다음 명령을 사용하여 비활성 상태인 큐 관리자에서 실행 중인 모든 WebSphere MQ 리스너를 중지시킬 수 있습니다.

```
ENDMQLSR [-m QMNAME]
```

에서 SNA 사용

TpSetup(SNA Server SDK에 있음)을 사용하여 amqcrs6a.exe를 실행하는 호출 가능한 TP를 정의하거나 다양한 레지스트리 값을 수동으로 설정할 수 있습니다. amqcrs6a.exe로 전달되어야 하는 매개변수는 다음과 같습니다.

```
-m QM -n TpName
```

여기서 QM은 큐 관리자 이름이고 TpName은 TP 이름입니다. 자세한 정보는 *Microsoft SNA Server APPC Programmers Guide* 또는 *Microsoft SNA Server CPI-C Programmers Guide*를 참조하십시오.

큐 관리자 이름을 지정하지 않으면 기본 큐 관리자로 가정됩니다.

Windows에서 NetBIOS 연결 정의

WebSphere MQ는 NetBIOS 연결을 다른 WebSphere MQ 제품에 설정할 때 세 유형의 NetBIOS 자원(세션, 명령 및 이름)을 사용합니다. 이러한 자원은 각각 NetBIOS 설치 중 선택적으로 또는 기본적으로 설정되는 한계가 있습니다.

각 실행 채널은 유형에 관계 없이 하나의 NetBIOS 세션과 하나의 NetBIOS 명령을 사용합니다. IBM NetBIOS 구현으로 다중 프로세스가 동일 로컬 NetBIOS 이름을 사용할 수 있습니다. 그러므로, WebSphere MQ에서 사용하도록 하나의 NetBIOS 이름만 사용 가능해야 합니다. 예를 들어, Novell의 NetBIOS 에뮬레이션과 같은 다른 벤더의 구현은 프로세스마다 다른 로컬 이름이 필요합니다. 사용 중인 NetBIOS 제품에 대한 문서에서 요구사항을 확인하십시오.

모든 경우, 각 유형의 충분한 자원이 이미 사용 가능한지 확인하거나 구성에 지정된 최대값을 늘리십시오. 값을 변경하려면 시스템을 다시 시작해야 합니다.

시스템 시동 중에, NetBIOS 디바이스 드라이버가 애플리케이션이 사용할 수 있는 세션, 명령, 이름의 수를 표시합니다. 이러한 자원은 동일한 시스템에서 실행 중인 NetBIOS 기반 애플리케이션이 사용할 수 있습니다. 그러므로, WebSphere MQ가 이 자원 획득을 요구하기 전에 다른 애플리케이션이 이 자원을 이용할 수 있습니다. LAN 네트워크 관리자는 사용자를 위해 이를 명백하게 설명할 수 있어야 합니다.

관련 개념

[80 페이지의 『IBM WebSphere MQ 로컬 NetBIOS 이름 정의』](#)

IBM WebSphere MQ 프로세스가 사용하는 로컬 NetBIOS 이름은 세 가지 방법으로 지정될 수 있습니다.

[80 페이지의 『큐 관리자 NetBIOS 세션, 명령 및 이름 한계 설정』](#)

NetBIOS 세션, 명령 및 이름에 대한 큐 관리자 한계는 두 가지 방법으로 지정할 수 있습니다.

[80 페이지의 『LAN 어댑터 번호 설정』](#)

NetBIOS에서 채널이 작동하려면 각 측의 어댑터 지원이 호환 가능해야 합니다. IBM WebSphere MQ qm.ini 파일의 NETBIOS 스탠자에 있는 AdapterNum 값을 사용하고 runmqsr 명령에 -a 매개변수를 지정하여 LAN 어댑터 (LANA) 번호의 선택사항을 제어할 수 있습니다.

[81 페이지의 『NetBIOS 연결 시작』](#)

연결을 시작하는 데 필요한 단계를 정의합니다.

[81 페이지의 『NetBIOS 연결을 위한 대상 리스너』](#)

NetBIOS 연결의 수신측에서 수행할 단계를 정의합니다.

IBM WebSphere MQ 로컬 NetBIOS 이름 정의
 IBM WebSphere MQ 프로세스가 사용하는 로컬 NetBIOS 이름은 세 가지 방법으로 지정될 수 있습니다.
 세 가지 방법의 우선순위는 다음과 같습니다.

1. RUNMQLS 명령의 -l 매개변수에 지정되는 값, 예를 들어 다음과 같습니다.

```
RUNMQLSR -t NETBIOS -l my_station
```

2. 명령으로 값이 설정되는 MQNAME 환경 변수:

```
SET MQNAME=my_station
```

MQNAME 값은 각 프로세스마다 설정할 수 있습니다. 또는 Windows 레지스트리에서 시스템 레벨에 설정할 수 있습니다.

고유한 이름이 필요한 NetBIOS 구현을 사용하는 경우에는 IBM WebSphere MQ 프로세스가 시작되는 각 창에서 SET MQNAME 명령을 실행해야 합니다. MQNAME 값은 임의로 정할 수 있지만 각 프로세스에 고유해야 합니다.

3. 큐 관리자 구성 파일 qm.ini의 NETBIOS 스탠자. 예를 들면, 다음과 같습니다.

```
NETBIOS:
    LocalName=my_station
```

참고:

1. 지원되는 NetBIOS 제품 구현의 다양성으로 인해 네트워크에서 각 NetBIOS 이름을 고유하게 만드는 것이 좋습니다. 그렇지 않으면 예상치 못한 결과가 발생할 수 있습니다. NetBIOS 채널 설정에 문제점이 있고 큐 관리자 오류 로그에 NetBIOS 리턴 코드 X'15'를 표시하는 오류 메시지가 있는 경우에는 NetBIOS 이름 사용을 검토하십시오.
2. Windows에서는 시스템 이름을 NetBIOS 이름으로 사용할 수 없습니다. Windows에서 이미 사용하고 있기 때문입니다.
3. 송신자 채널을 시작하려면 qm.ini 파일에서 MQNAME 환경 변수 또는 LocalName을 사용하여 NetBIOS 이름이 지정되어야 합니다.

큐 관리자 NetBIOS 세션, 명령 및 이름 한계 설정

NetBIOS 세션, 명령 및 이름에 대한 큐 관리자 한계는 두 가지 방법으로 지정할 수 있습니다.

우선순위에 따른 방법은 다음과 같습니다.

1. RUNMQLSR 명령에 지정된 값:

```
-s Sessions
-e Names
-o Commands
```

명령에 -m 피연산자가 지정되지 않은 경우에는 기본 큐 관리자에만 값이 적용됩니다.

2. 큐 관리자 구성 파일 qm.ini의 NETBIOS 스탠자. 예를 들면, 다음과 같습니다.

```
NETBIOS:
    NumSess=Qmgr_max_sess
    NumCmds=Qmgr_max_cmds
    NumNames=Qmgr_max_names
```

LAN 어댑터 번호 설정

NetBIOS에서 채널이 작동하려면 각 측의 어댑터 지원이 호환 가능해야 합니다. IBM WebSphere MQ qm.ini 파일의 NETBIOS 스탠자에 있는 AdapterNum 값을 사용하고 runmqslsr 명령에 -a 매개변수를 지정하여 LAN 어댑터 (LANA) 번호의 선택사항을 제어할 수 있습니다.

NetBIOS용 IBM WebSphere MQ연결에서 사용되는 기본 LAN 어댑터 번호는 0입니다. 다음과 같이 시스템에서 사용되는 번호를 확인하십시오.

Windows에서는 운영 체제를 통해 직접 LAN 어댑터 번호를 조회할 수 없습니다. 대신 Microsoft에서 사용 가능한 LANACFG.EXE 명령행 유틸리티를 사용합니다. 도구의 출력은 가상 LAN 어댑터 번호와 해당 적용 바인딩을 보여줍니다. LAN 어댑터 번호에 대한 추가 정보는 Microsoft Knowledge Base 기사 138037 *HOWTO: Use LANA Numbers in a 32-bit Environment*를 참조하십시오.

큐 관리자 구성 파일, qm.ini의 NETBIOS 스탠자에서 올바른 값을 지정하십시오.

```
NETBIOS:
AdapterNum=n
```

여기서 n은 이 시스템의 올바른 LAN 어댑터 번호입니다.

NetBIOS 연결 시작

연결을 시작하는 데 필요한 단계를 정의합니다.

연결을 시작하려면 수신측에서 다음 단계를 수행하십시오.

1. MQNAME 또는 LocalName 값을 사용하여 NetBIOS 스테이션 이름을 정의하십시오.
2. 시스템에서 사용하는 LAN 어댑터 번호를 확인하고 AdapterNum을 사용하여 올바른 파일을 지정하십시오.
3. 채널 정의의 ConnectionName 필드에서 대상 리스너 프로그램이 사용하는 NetBIOS 이름을 지정하십시오. Windows에서, NetBIOS 채널은 반드시 스레드로 실행되어야 합니다. 이 작업을 수행하려면 채널 정의에서 MCATYPE(THREAD)를 지정하십시오.

```
DEFINE CHANNEL (chname) CHLTYPE(SDR) +
  TRPTYPE(NETBIOS) +
  CONNAME(your_station) +
  XMITQ(xmitq) +
  MCATYPE(THREAD) +
  REPLACE
```

NetBIOS 연결을 위한 대상 리스너

NetBIOS 연결의 수신측에서 수행할 단계를 정의합니다.

수신측에서 다음 단계를 수행하십시오.

1. MQNAME 또는 LocalName 값을 사용하여 NetBIOS 스테이션 이름을 정의하십시오.
2. 시스템에서 사용하는 LAN 어댑터 번호를 확인하고 AdapterNum을 사용하여 올바른 파일을 지정하십시오.
3. 수신자 채널을 정의하십시오.

```
DEFINE CHANNEL (chname) CHLTYPE(RCVR) +
  TRPTYPE(NETBIOS) +
  REPLACE
```

4. WebSphere MQ 리스너 프로그램을 시작하여 스테이션을 설정하고 접속 가능하게 만드십시오. 예를 들면, 다음과 같습니다.

```
RUNMQLSR -t NETBIOS -l your_station [-m qmgr]
```

이 명령은 your_station을 접속 대기 중인 NetBIOS 스테이션으로 설정합니다. NetBIOS 스테이션 이름은 NetBIOS 네트워크 전체에서 고유해야 합니다.

최상의 성능을 위해 WebSphere MQ 리스너를 66 페이지의 『채널 및 리스너를 신뢰 애플리케이션으로 실행』에서 설명한 대로 트러스트 애플리케이션으로 실행하십시오. 신뢰 애플리케이션에 대한 정보는 신뢰 애플리케이션에 대한 제한사항을 참조하십시오.

다음 명령을 사용하여 비활성 상태인 큐 관리자에서 실행 중인 모든 WebSphere MQ 리스너를 중지시킬 수 있습니다.

```
ENDMQLSR [-m QMNAME]
```

큐 관리자 이름을 지정하지 않으면 기본 큐 관리자로 가정됩니다.

Windows에서 SPX 연결 정의

SPX 연결은 Windows XP 및 Windows 2003 Server를 실행 중인 클라이언트와 서버에만 적용됩니다.

송신측의 채널 정의는 대상의 주소를 지정합니다. 수신 종료 시 리스너 프로그램이 실행되어야 합니다.

관련 개념

82 페이지의 『SPX의 송신측』

대상 시스템이 원격일 경우, 채널 정의의 연결 이름 필드에서 대상 시스템의 SPX 주소를 지정하십시오.

82 페이지의 『SPX에서 수신』

수신 채널 프로그램은 송신 채널의 시동 요청에 대한 응답으로 시작됩니다.

84 페이지의 『IPX/SPX 매개변수』

대부분의 경우, IPX/SPX 매개변수의 기본 설정값이 사용자 요구에 맞을 것입니다. 그러나 사용자의 환경에서 WebSphere MQ에 이 매개변수를 사용하도록 일부 수정해야 경우도 있습니다.

SPX의 송신측

대상 시스템이 원격일 경우, 채널 정의의 연결 이름 필드에서 대상 시스템의 SPX 주소를 지정하십시오.

SPX 주소는 다음 형태로 지정됩니다.

```
network.node(socket)
```

설명:

network

원격 시스템이 상주하는 네트워크의 4바이트 네트워크 주소입니다.

node

6바이트의 노드 주소 즉, 원격 시스템에 있는 LAN 어댑터의 LAN 주소입니다.

socket

원격 시스템이 연결 대기할 2바이트 소켓 번호입니다.

로컬 및 원격 시스템이 동일 네트워크에 있을 경우에는 네트워크 주소를 지정할 필요가 없습니다. 원격측이 기본 소켓(5E86)에서 연결 대기할 경우에는 소켓을 지정할 필요가 없습니다.

완전하게 지정된 SPX 주소(MQSC 명령의 CONNAME 매개변수에 지정되어 있는)의 예는 다음과 같습니다.

```
CONNAME('00000001.08005A7161E5(5E87)')
```

기본적으로, 시스템이 모두 동일 네트워크에 있으며 이는 다음과 같습니다.

```
CONNAME(08005A7161E5)
```

큐 관리자 구성 파일(qm.ini)에서 소켓 번호를 지정하여 기본 소켓 번호를 변경할 수 있습니다.

```
SPX:  
Socket=5E87
```

qm.ini를 사용하여 설정한 값에 대한 자세한 정보는 [분산 큐잉의 구성 파일 스탠자](#)의 내용을 참조하십시오.

SPX에서 수신

수신 채널 프로그램은 송신 채널의 시동 요청에 대한 응답으로 시작됩니다.

수신 채널 프로그램을 시작하려면 리스너 프로그램이 시작되어 수신되는 네트워크 요청을 감지하고 관련 채널을 시작해야 합니다.

WebSphere MQ 리스너를 사용하십시오.

SPX 리스너 백로그 옵션 사용

SPX에서 수신 시, 최대 미해결 연결 요청 수가 설정됩니다. 이는 SPX 포트에서 리스너가 요청을 받아들이기를 기다리고 있는 요청의 백로그로 간주될 수 있습니다. 기본 리스너 백로그 값은 [83 페이지의 표 10](#)에 표시됩니다.

표 10. Windows의 디폴트 미해결 연결 요청	
플랫폼	기본 리스너 백로그 값
Windows 서버	5
Windows 워크스테이션	5

백로그가 [83 페이지의 표 10](#)에 있는 값에 도달하면, MQCONN 또는 MQCONNX를 사용하여 큐 관리자에 연결하려고 시도할 때 이유 코드 MQRC_Q_MGR_NOT_AVAILABLE이 수신됩니다. 이 경우 연결을 재시도할 수 있습니다.

그러나 이 오류를 피하기 위해 qm.ini 파일이나 Windows의 레지스트리에 다음 항목을 추가할 수 있습니다.

```
SPX:  
ListenerBacklog = n
```

이는 SPX 리스너에 대한 최대 미해결 요청 수([83 페이지의 표 10 참조](#))를 대체합니다.

참고: 일부 운영 체제에서는 기본값보다 더 큰 값을 지원합니다. 필요하다면, 연결 한계에 도달하지 않도록 이러한 큰 값을 사용할 수 있습니다.

백로그 옵션을 활성화한 상태로 리스너를 실행하려면 다음을 수행하십시오.

- RUNMQLSR -b 명령을 사용하십시오. 그렇지 않은 경우,
- BACKLOG 속성을 필수 값으로 설정하여 MQSC 명령 **DEFINE LISTENER**를 사용하십시오.

RUNMQLSR 명령에 대한 자세한 정보는 [runmqslsr](#)을 참조하십시오. **DEFINE LISTENER** 명령에 대한 정보는 [DEFINE LISTENER](#)를 참조하십시오.

WebSphere MQ 리스너 사용

새 채널을 스투드로서 시작하는 WebSphere MQ와 함께 제공된 리스너를 실행하려면 RUNMQLSR 명령을 사용하십시오. 예를 들면, 다음과 같습니다.

```
RUNMQLSR -t spx [-m QMNAME] [-x 5E87]
```

대괄호는 선택적 매개변수를 나타냅니다. 즉 QMNAME은 기본 큐 관리자의 경우 필수가 아니며, 기본값(5E86)을 사용하는 경우에는 소켓 번호가 필요하지 않습니다.

최상의 성능을 위해 WebSphere MQ 리스너를 [66 페이지의 『채널 및 리스너를 신뢰 애플리케이션으로 실행』](#)에서 설명한 대로 **트러스트 애플리케이션**으로 실행하십시오. **트러스트 애플리케이션**에 대한 자세한 정보는 [트러스트 애플리케이션의 제한사항](#)을 참조하십시오.

다음 명령을 사용하여 비활성 상태인 큐 관리자에서 실행 중인 모든 WebSphere MQ 리스너를 중지시킬 수 있습니다.

```
ENDMQLSR [-m QMNAME]
```

큐 관리자 이름을 지정하지 않으면 기본 큐 관리자로 가정됩니다.

IPX/SPX 매개변수

대부분의 경우, IPX/SPX 매개변수의 기본 설정값이 사용자 요구에 맞을 것입니다. 그러나 사용자의 환경에서 WebSphere MQ에 이 매개변수를 사용하도록 일부 수정해야 경우도 있습니다.

실제 매개변수 및 이를 변경하는 방법은 플랫폼 및 SPX 통신 지원 제공자에 따라 달라집니다. 예에 관한 절에서는 이러한 매개변수 중 일부, 특히 WebSphere MQ 채널 및 클라이언트 연결의 조작에 영향을 줄 수 있는 매개변수에 대해 설명합니다.

Windows 시스템

NWLink IPX 및 SPX 매개변수의 사용 및 설정에 대한 자세한 내용은 Microsoft 문서를 참조하십시오. IPX/SPX 매개변수는 레지스트리의 다음 경로에 있습니다.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkSPX\Parameters
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkIPX\Parameters
```

UNIX and Linux 시스템에서 통신 설정

DQM은 IBM WebSphere MQ의 리모트 큐잉 기능입니다. 이 기능은 시스템 운영자가 제어할 수 있는 통신 링크에 대한 인터페이스를 형성하는 큐 관리자용 채널 제어 프로그램을 제공합니다. 분산 큐잉 관리로 유지되는 채널 정의에서 이러한 연결을 사용합니다.

분산 큐잉 관리 채널이 시작되면 이 채널에서 채널 정의에 지정된 연결을 사용하려 합니다. 성공하려면 연결이 정의되고 사용 가능해야 합니다. 이 절에서는 이를 실행하는 방법을 설명합니다. 다음 절을 참조하는 것도 유용한 방법입니다.

- [구성 예 - AIX용 IBM WebSphere MQ](#)
- [구성 예 - HP-UX용 IBM WebSphere MQ](#)
- [구성 예 - Solaris용 IBM WebSphere MQ](#)
- [예제 구성 - Linux 에 대한 IBM WebSphere MQ](#)

Windows의 경우, [75 페이지](#)의 『Windows용 통신 설정』을 참조하십시오.

UNIX and Linux 시스템에서 WebSphere MQ용 통신의 다음 두 형식 중에서 선택할 수 있습니다.

- [84 페이지](#)의 『UNIX and Linux에서 TCP 연결 정의』
- [88 페이지](#)의 『UNIX and Linux에서 LU 6.2 연결 정의』

각 채널 정의마다 전송 프로토콜(전송 유형) 속성으로만 지정해야 합니다. 큐 관리자는 하나 이상의 프로토콜을 사용할 수 있습니다.

IBM WebSphere MQ Explorer MQI 클라이언트의 경우, 다른 전송 프로토콜을 사용하는 대체 채널이 있는 것이 유용할 수 있습니다. IBM WebSphere MQ Explorer MQI 클라이언트에 대한 자세한 정보는 [IBM WebSphere MQ MQI 클라이언트의 개요](#)를 참조하십시오.

관련 개념

[25 페이지](#)의 『분산 큐잉을 사용하여 애플리케이션 연결』

이 절에서는 큐 정의, 채널 정의, 트리거 및 동기점 프로시저를 포함한 WebSphere MQ 설치 간 상호통신에 관한 자세한 정보를 제공합니다.

[66 페이지](#)의 『UNIX, Linux, and Windows에서 채널 모니터링 및 제어』

DQM의 경우 리모트 큐 관리자에 대한 채널을 작성, 모니터, 제어해야 합니다. 명령, 프로그램, IBM WebSphere MQ Explorer, 채널 정의 파일, 동기화 정보 스토리지 영역을 사용하여 채널을 제어할 수 있습니다.

[89 페이지](#)의 『클라이언트와 서버 간의 연결 구성』

WebSphere MQ MQI 클라이언트와 서버 사이의 통신 링크를 구성하려면 통신 프로토콜을 결정하고 링크의 양 끝에서의 연결을 정의하고 리스너를 시작한 다음 채널을 정의하십시오.

UNIX and Linux에서 TCP 연결 정의

송신측의 채널 정의는 대상의 주소를 지정합니다. 리스너 또는 inet 디먼은 수신측에서의 연결용으로 구성됩니다.

송신측

채널 정의의 연결 이름 필드에서 대상 시스템의 TCP 주소 또는 호스트 이름을 지정하십시오. 연결할 포트는 기본적으로 1414입니다. Internet Assigned Numbers Authority에서 포트 번호 1414를 WebSphere MQ에 할당합니다.

기본값 이외의 포트 번호를 사용하려면 연결 이름 필드를 변경하십시오.

```
Connection Name REMHOST(1822)
```

여기서 REMHOST는 원격 시스템의 호스트 이름이고 1822는 필요한 포트 번호입니다. 이 포트는 수신측의 리스너가 청취하는 포트여야 합니다.

또는 큐 관리자 구성 파일(qm.ini)에서 포트 번호를 지정하여 변경할 수 있습니다.

```
TCP:  
Port=1822
```

qm.ini를 사용하여 설정하는 값에 대한 자세한 정보는 [분산 큐잉에 대한 구성 파일 스탠자를 참조하십시오](#).

TCP에서 수신

inet 디먼(inetd)인 TCP/IP 리스너 또는 WebSphere MQ 리스너를 사용할 수 있습니다.

일부 Linux 배포는 이제 inet 디먼 대신 확장 inet 디먼(xinetd)을 사용합니다. Linux 시스템에서 확장된 inet 디먼 사용법에 대한 정보는 [Linux에서 TCP 연결 설정을 참조하십시오](#).

관련 개념

[85 페이지의 『TCP/IP 리스너 사용』](#)

UNIX and Linux에서 채널을 시작하려면 /etc/services 파일과 inetd.conf 파일을 편집해야 합니다.

[86 페이지의 『TCP 리스너 백로그 옵션 사용』](#)

TCP에서는 서버와 클라이언트 사이에 3방향 데이터 교환이 발생하지 않는 한 연결이 완료되지 않은 것으로 처리됩니다. 이러한 연결을 미해결 연결 요청이라고 합니다. 이러한 미해결 연결 요청에 대해 최대값이 설정되며 이는 리스너가 요청을 승인할 때까지 TCP 포트에서 대기 중인 요청의 백로그로 간주될 수 있습니다.

[87 페이지의 『WebSphere MQ 리스너 사용』](#)

새 채널을 스레드로서 시작하는 WebSphere MQ와 함께 제공된 리스너를 실행하려면 runmqtsr 명령을 사용하십시오.

[88 페이지의 『TCP/IP SO_KEEPALIVE 옵션 사용』](#)

일부 UNIX and Linux 시스템에서는 연결을 계속 사용할 수 있는지 확인하기 전에 TCP가 대기하는 시간과 첫 번째 확인에 실패하는 경우 연결을 다시 시도하는 빈도를 정의할 수 있습니다. 이는 커널 조정 가능 매개변수이거나 명령행에서 입력할 수 있습니다.

TCP/IP 리스너 사용

UNIX and Linux에서 채널을 시작하려면 /etc/services 파일과 inetd.conf 파일을 편집해야 합니다.

다음 지시사항을 따르십시오.

1. /etc/services 파일을 편집하십시오.

참고: /etc/services 파일을 편집하려면 superuser 또는 루트로 로그인되어 있어야 합니다. 이를 변경할 수는 있지만 송신측에 지정된 포트 번호와 일치해야 합니다.

파일에 다음 행을 추가하십시오.

```
MQSeries 1414/tcp
```

여기서 1414는 WebSphere MQ에서 필요한 포트 번호입니다. 포트 번호는 65535를 초과해서는 안 됩니다.

2. 프로그램 amqcrsta를 호출하기 위해 inetd.conf 파일에 다음 행을 추가하십시오. 여기서 MQ_INSTALLATION_PATH는 WebSphere MQ가 설치된 상위 레벨 디렉토리를 나타냅니다.

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
[-m Queue_Man_Name]
```

inetd가 구성 파일을 다시 읽은 후 업데이트가 활성화됩니다. 이를 수행하려면 루트 사용자 ID에서 다음 명령을 실행하십시오.

- AIX의 경우,

```
refresh -s inetd
```

- HP-UX의 경우 mqm 사용자 ID를,

```
inetd -c
```

- Solaris 10 이상의 경우

```
inetconv
```

- 기타 UNIX and Linux 시스템(Solaris 9 포함)의 경우

```
kill -1 <process number>
```

inetd로 시작된 리스너 프로그램이 inetd에서 로케일을 상속하는 경우 MQMDE를 지원하지 않아(병합) 큐에 메시지가 데이터로 배치됩니다. MQMDE를 유지하려면 로케일을 올바르게 설정해야 합니다. inetd로 설정한 로케일이 WebSphere MQ 프로세스에서 사용하는 다른 로케일에 대해 선택된 것이 아닐 수도 있습니다. 로케일을 설정하려면 다음을 수행하십시오.

1. 다른 WebSphere MQ 프로세스에 사용되는 로케일로 로케일 환경 변수 LANG, LC_COLLATE, LC_CTYPE, LC_MONETARY, LC_NUMERIC, LC_TIME 및 LC_MESSAGES를 설정하는 셸 스크립트를 작성하십시오.
2. 동일한 셸 스크립트에서 리스너 프로그램을 호출하십시오.
3. 리스너 프로그램 대신 사용자의 셸 스크립트를 호출하도록 inetd.conf 파일을 수정하십시오.

서버에서 큐 관리자가 두 개 이상 실행될 수 있습니다. 각 큐 관리자마다 두 개 파일 각각에 행을 추가해야 합니다. 예를 들면, 다음과 같습니다.

```
MQSeries1    1414/tcp
MQSeries2    1822/tcp
```

```
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

여기서 MQ_INSTALLATION_PATH는 WebSphere MQ가 설치된 상위 레벨 디렉토리를 나타냅니다.

이를 통해 단일 TCP 포트에서 큐잉되는 미해결 연결 요청 수에 제한이 있는 경우 오류 메시지가 생성되지 않도록 할 수 있습니다. 미해결 연결 요청 수에 대한 정보는 [86 페이지의 『TCP 리스너 백로그 옵션 사용』](#)의 내용을 참조하십시오.

TCP 리스너 백로그 옵션 사용

TCP에서는 서버와 클라이언트 사이에 3방향 데이터 교환이 발생하지 않는 한 연결이 완료되지 않은 것으로 처리됩니다. 이러한 연결을 미해결 연결 요청이라고 합니다. 이러한 미해결 연결 요청에 대해 최대값이 설정되며 이는 리스너가 요청을 승인할 때까지 TCP 포트에서 대기 중인 요청의 백로그로 간주될 수 있습니다.

기본 리스너 백로그 값은 [87 페이지의 표 11](#)에 표시됩니다.

표 11. TCP/IP 포트에서 큐잉되는 최대 미해결 연결 요청 수	
서버 플랫폼	최대 연결 요청 수
AIX	100
HP-UX	20
Linux	100
IBM i	255
Solaris	100
Windows 서버	100
Windows 워크스테이션	100
z/OS	255

백로그가 87 페이지의 표 11에 나타난 값에 도달하면, TCP/IP 연결이 거부되고 채널이 시작할 수 없습니다. MCA 채널의 경우에는 채널이 RETRY 상태로 진행되고 나중에 연결을 다시 시도합니다. 그러나 이 오류를 방지하기 위해 qm.ini 파일에 다음 입력 항목을 추가할 수 있습니다.

```
TCP:
ListenerBacklog = n
```

이는 TCP/IP 요청에 대한 기본 최대 미해결 요청 수(87 페이지의 표 11 참조)를 대체합니다.

참고: 일부 운영 체제에서는 기본값보다 더 큰 값을 지원합니다. 필요한 경우 이 값은 연결 제한에 도달하는 것을 방지하기 위해 사용할 수 있습니다.

백로그 옵션을 활성화한 상태로 리스너를 실행하려면 다음을 수행하십시오.

- `runmqclsr -b` 명령을 사용하십시오. 또는
- BACKLOG 속성을 필수 값으로 설정하여 MQSC 명령 **DEFINE LISTENER**를 사용하십시오.

`runmqclsr` 명령에 대한 정보는 `runmqclsr`을 참조하십시오. `DEFINE LISTENER` 명령에 대한 정보는 `DEFINE LISTENER`를 참조하십시오.

WebSphere MQ 리스너 사용

새 채널을 스레드로서 시작하는 *WebSphere MQ*와 함께 제공된 리스너를 실행하려면 `runmqclsr` 명령을 사용하십시오.

예를 들면, 다음과 같습니다.

```
runmqclsr -t tcp [-m QMNAME] [-p 1822]
```

대괄호는 선택적 매개변수를 표시합니다. 기본 큐 관리자에는 QMNAME 가 필요하지 않으며 기본값 (1414) 을 사용하는 경우 포트 번호가 필요하지 않습니다. 포트 번호는 65535를 초과해서는 안 됩니다.

최상의 성능을 위해 *WebSphere MQ* 리스너를 66 페이지의 『채널 및 리스너를 신뢰 애플리케이션으로 실행』에서 설명한 대로 *트러스트 애플리케이션*으로 실행하십시오. 신뢰 애플리케이션에 대한 정보는 *신뢰 애플리케이션에 대한 제한사항*을 참조하십시오.

다음 명령을 사용하여 비활성 상태인 큐 관리자에서 실행 중인 모든 *WebSphere MQ* 리스너를 중지시킬 수 있습니다.

```
endmqclsr [-m QMNAME]
```

큐 관리자 이름을 지정하지 않으면 기본 큐 관리자로 가정됩니다.

TCP/IP SO_KEEPALIVE 옵션 사용

일부 UNIX and Linux 시스템에서는 연결을 계속 사용할 수 있는지 확인하기 전에 TCP가 대기하는 시간과 첫 번째 확인에 실패하는 경우 연결을 다시 시도하는 빈도를 정의할 수 있습니다. 이는 커널 조정 가능 매개변수이거나 명령행에서 입력할 수 있습니다.

SO_KEEPALIVE 옵션을 사용하려면(자세한 정보는 55 페이지의 『채널의 다른 쪽이 여전히 사용 가능한지 검사』 참조) 큐 관리자 구성 파일(qm.ini)에 다음 입력 항목을 추가해야 합니다.

```
TCP:
  KeepAlive=yes
```

자세한 정보는 UNIX and Linux 시스템 문서를 참조하십시오.

UNIX and Linux에서 LU 6.2 연결 정의

SNA는 두 시스템 간에 LU 6.2 대화가 설정될 수 있도록 구성되어야 합니다.

TCP/IP를 통한 SNA 구성에 대한 최신 정보는 다음 온라인 IBM 문서(통신 서버)를 참조하십시오.

SNA는 두 시스템 간에 LU 6.2 대화가 설정될 수 있도록 구성되어야 합니다.

해당 정보는 다중 플랫폼 APPC 구성 안내서와 다음 표를 참조하십시오.

표 12. 리모트 큐 관리자 플랫폼에 대한 로컬 UNIX and Linux 시스템의 설정		
원격 플랫폼	TPNAME	TPPATH
코믹스 없는 z/OS	리모트 큐 관리자에 대한 부가 정보에 있는 해당 TPName과 동일합니다.	-
코믹스 를 사용하는 z/OS	CKRC(송신자) CKSV(요청자) CKRC(서버)	-
IBM i	IBM i 시스템의 라우팅 입력 항목에 있는 비교 값과 동일합니다.	-
UNIX and Linux 시스템	리모트 큐 관리자에 대한 부가 정보에 있는 해당 TPName과 동일합니다.	MQ_INSTALLATION_PATH/bin/amqcrs6a
Windows	Windows Run Listener 명령 또는 Windows에서 TpSetup을 사용하여 정의된 호출 가능한 트랜잭션 프로그램에서 지정한 대로.	MQ_INSTALLATION_PATH\bin\amqcrs6a

MQ_INSTALLATION_PATH는 WebSphere MQ가 설치되어 있는 상위 레벨 디렉토리를 표시합니다.

동일한 시스템에 둘 이상의 큐 관리자가 있다면, 반드시 채널 정의에서 TPname이 고유한지 확인하십시오.

관련 개념

88 페이지의 『송신측』

UNIX and Linux 시스템에서 CPI-C측 오브젝트(기호 목적지)를 작성하고 채널 정의의 연결 이름 필드에 이 이름을 입력합니다. 또한 파트너에 대한 LU 6.2 링크도 작성합니다.

89 페이지의 『LU 6.2에서 수신』

UNIX and Linux 시스템에서 수신측의 대기 첨부, LU 6.2 논리 연결 프로파일, TPN 프로파일을 작성합니다.

송신측

UNIX and Linux 시스템에서 CPI-C측 오브젝트(기호 목적지)를 작성하고 채널 정의의 연결 이름 필드에 이 이름을 입력합니다. 또한 파트너에 대한 LU 6.2 링크도 작성합니다.

CPI-C측 오브젝트에서 수신 시스템의 파트너 LU 이름, 트랜잭션 프로그램 이름, 모드 이름을 입력하십시오. 예를 들면, 다음과 같습니다.

```
Partner LU Name      REMHOST
Remote TP Name      recv
```

```
Service Transaction Program no
Mode Name #INTER
```

HP-UX의 경우 APPCLLU 환경 변수를 사용해서 송신자가 사용해야 하는 로컬 LU를 이름 지정하십시오. Solaris의 경우, APPC_LOCAL_LU 환경 변수를 로컬 LU 이름에 설정하십시오.

WebSphere MQ가 SNA 세션 설정을 시도할 때 CPI-C가 지원되는 위치에서 SECURITY PROGRAM이 사용됩니다.

LU 6.2에서 수신

UNIX and Linux 시스템에서 수신측의 대기 첨부, LU 6.2 논리 연결 프로파일, TPN 프로파일을 작성합니다.

TPN 프로파일에서 실행 파일의 전체 경로와 트랜잭션 프로그램 이름을 입력하십시오.

```
Full path to TPN executable MQ_INSTALLATION_PATH/bin/amqcrs6a
Transaction Program name recv
User ID 0
```

MQ_INSTALLATION_PATH는 WebSphere MQ가 설치되어 있는 상위 레벨 디렉토리를 표시합니다.

사용자 ID를 설정할 수 있는 시스템의 경우 mqm 그룹의 구성원인 사용자를 지정하십시오. AIX, Solaris 및 HP-UX에서 APPCTPN(트랜잭션 이름) 및 APPCLLU(로컬 LU 이름) 환경 변수를 설정하십시오. (호출된 트랜잭션 프로그램용 구성 패널을 사용할 수 있습니다.)

기본 큐 관리자 이외의 큐 관리자를 사용해야 할 수 있습니다. 이러한 경우 다음 명령을 호출하는 명령 파일을 정의하십시오.

```
amqcrs6a -m Queue_Man_Name
```

그런 다음 명령 파일을 호출하십시오.

클라이언트와 서버 간의 연결 구성

WebSphere MQ MQI 클라이언트와 서버 사이의 통신 링크를 구성하려면 통신 프로토콜을 결정하고 링크의 양 끝에서의 연결을 정의하고 리스너를 시작한 다음 채널을 정의하십시오.

WebSphere MQ에서 오브젝트 사이의 논리 통신 링크를 채널이라고 합니다. WebSphere MQ MQI 클라이언트를 서버에 연결하는 데 사용되는 채널은 MQI 채널이라고 합니다. WebSphere MQ MQI 클라이언트의 WebSphere MQ 애플리케이션이 서버의 큐 관리자와 통신할 수 있도록 링크의 양 끝에 채널 정의를 설정하십시오. 이를 수행하는 방법에 대한 자세한 설명은 [사용자 정의 채널](#)을 참조하십시오.

MQI 채널을 결정하기 전에 다음을 수행해야 합니다.

1. 사용하려는 통신의 양식을 결정하십시오. [89 페이지의 『사용할 통신 유형』](#)의 내용을 참조하십시오.
2. 채널 양 끝에서 연결을 정의합니다.

연결을 정의하려면 다음을 수행해야 합니다.

- 연결을 구성합니다.
- 채널 정의에 필요한 매개변수의 값을 기록합니다.
- 리스너를 시작하여 WebSphere MQ MQI 클라이언트로부터 수신하는 네트워크 요청을 서버가 감지할 수 있도록 하십시오.

사용할 통신 유형

플랫폼에 따라 다른 전송 프로토콜을 지원합니다. 전송 프로토콜은 WebSphere MQ MQI 클라이언트와 서버 플랫폼의 조합에 따라 다르게 선택합니다.

클라이언트 및 서버 플랫폼에 따라 MQI 채널에 대해 최대 4개 유형의 전송 프로토콜이 있습니다.

- LU 6.2
- NetBIOS
- SPX

• TCP/IP

MQI 채널을 정의할 때 각 채널 정의는 전송 프로토콜(전송 유형) 속성을 지정해야 합니다. 서버는 한 개의 프로토콜로 제한되지 않으므로 채널 정의에 따라 다른 프로토콜을 지정할 수 있습니다. WebSphere MQ MQI 클라이언트의 경우 다른 전송 프로토콜을 사용하는 대체 MQI 채널을 가지는 것이 유용할 수 있습니다.

전송 프로토콜의 선택이 WebSphere MQ MQI 클라이언트 및 서버 플랫폼의 특정 조합으로 제한될 수 있습니다. 가능한 조합은 다음 테이블에 표시되어 있습니다.

표 13. 전송 프로토콜 - WebSphere MQ MQI 클라이언트 및 서버 플랫폼의 결합		
전송 프로토콜	WebSphere MQ MQI 클라이언트	WebSphere MQ 서버
TCP/IP	UNIX 시스템 Windows	UNIX 시스템 Windows z/OS
LU 6.2	UNIX 시스템 ¹ Windows	UNIX 시스템 ¹ Windows
NetBIOS	Windows	Windows
SPX	Windows	Windows
참고:		
1. Linux for Power Systems 제외		

여러 유형의 연결 설정에 대한 자세한 정보는 다음 링크를 참조하십시오.

- [76 페이지의 『Windows에서 TCP 연결 정의』](#)
- [84 페이지의 『UNIX and Linux에서 TCP 연결 정의』](#)
- [92 페이지의 『TCP/IP 연결 한계』](#)
- [77 페이지의 『Windows에서 LU 6.2 연결 정의』](#)
- [88 페이지의 『UNIX and Linux에서 LU 6.2 연결 정의』](#)
- [79 페이지의 『Windows에서 NetBIOS 연결 정의』](#)
- [82 페이지의 『Windows에서 SPX 연결 정의』](#)

관련 개념

[93 페이지의 『확장된 트랜잭션 클라이언트 구성』](#)

이 주제 콜렉션에서는 각 트랜잭션 관리자 범주에 대해 확장된 트랜잭션 기능을 구성하는 방법을 설명합니다.

[101 페이지의 『MQI 채널 정의』](#)

새 채널 정의를 작성하려면 같은 채널 이름과 호환 가능한 채널 유형을 사용하여 연결의 양 끝에 하나씩 두 개 채널 정의를 작성해야 합니다. 이 경우 채널 유형은 *server-connection* 및 *client-connection*입니다.

[102 페이지의 『다른 플랫폼에서 서버 연결 및 클라이언트 연결 정의 작성』](#)

적용할 컴퓨터에서 각 채널 정의를 작성할 수 있습니다. 클라이언트 컴퓨터에서 채널 정의를 작성하는 방법에는 제한이 있습니다.

[105 페이지의 『서버에서 서버 연결 및 클라이언트 연결 정의 작성』](#)

서버에서 두 정의 모두 작성한 다음 클라이언트가 클라이언트 연결 정의를 사용할 수 있도록 설정할 수 있습니다.

[110 페이지의 『MQI 채널에 대한 채널 엑시트 프로그램』](#)

UNIX, Linux 및 시스템 WebSphere MQ MQI 클라이언트 환경에는 세 가지 유형의 채널 엑시트가 사용 가능합니다.

[113 페이지의 『큐 공유 그룹에 클라이언트 연결』](#)

큐 공유 그룹의 멤버인 서버에서 클라이언트와 큐 관리자 간 MQI 채널을 작성하여 클라이언트를 큐 공유 그룹에 연결할 수 있습니다.

114 페이지의 『구성 파일을 사용하여 클라이언트 구성』

텍스트 파일의 속성을 사용하여 클라이언트를 구성하십시오. 이러한 속성은 환경 변수 또는 다른 플랫폼별 방식으로 대체될 수 있습니다.

관련 태스크

큐 관리자에 IBM MQ MQI 클라이언트 애플리케이션 연결

관련 참조

[DISPLAY CHLAUTH](#)

[SET CHLAUTH](#)

사용할 통신 유형

플랫폼에 따라 다른 통신 프로토콜을 지원합니다. 전송 프로토콜은 WebSphere MQ MQI 클라이언트와 서버 플랫폼의 조합에 따라 다르게 선택합니다.

플랫폼에 따라 MQI 채널을 위한 4개의 통신 유형이 있습니다.

- LU 6.2
- NetBIOS
- SPX
- TCP/IP

MQI 채널을 정의할 때 각 채널 정의는 전송 프로토콜(전송 유형) 속성을 지정해야 합니다. 서버는 한 개의 프로토콜로 제한되지 않으므로 채널 정의에 따라 다른 프로토콜을 지정할 수 있습니다. WebSphere MQ MQI 클라이언트의 경우 다른 전송 프로토콜을 사용하는 대체 MQI 채널을 가지는 것이 유용할 수 있습니다.

전송 프로토콜의 선택은 WebSphere MQ 클라이언트 및 서버 플랫폼의 특정 조합에 따라 달라집니다. 가능한 조합은 다음 테이블에 표시되어 있습니다.

표 14. 전송 프로토콜 - WebSphere MQ 클라이언트 및 서버 플랫폼의 결합		
전송 프로토콜	WebSphere MQ MQI 클라이언트	WebSphere MQ 서버
TCP/IP	UNIX 시스템 Windows	UNIX 시스템 Windows
LU 6.2	UNIX 시스템 ¹ Windows	UNIX 시스템 ¹ Windows
NetBIOS	Windows	Windows
SPX	Windows	Windows
참고:		
1. Linux (POWER® 플랫폼) 을 제외하고		

관련 개념

76 페이지의 『Windows에서 TCP 연결 정의』

대상의 주소를 지정하기 위해 송신측에서 채널을 구성하고 수신측에서 리스너 프로그램을 실행하여 TCP 연결을 정의합니다.

84 페이지의 『UNIX and Linux에서 TCP 연결 정의』

송신측의 채널 정의는 대상의 주소를 지정합니다. 리스너 또는 inet 디먼은 수신측에서의 연결용으로 구성됩니다.

77 페이지의 『Windows에서 LU 6.2 연결 정의』

SNA는 두 시스템 간에 LU 6.2 대화가 설정될 수 있도록 구성되어야 합니다.

88 페이지의 『UNIX and Linux에서 LU 6.2 연결 정의』

SNA는 두 시스템 간에 LU 6.2 대화가 설정될 수 있도록 구성되어야 합니다.

79 페이지의 『Windows에서 NetBIOS 연결 정의』

WebSphere MQ는 NetBIOS 연결을 다른 WebSphere MQ 제품에 설정할 때 세 유형의 NetBIOS 자원(세션, 명령 및 이름)을 사용합니다. 이러한 자원은 각각 NetBIOS 설치 중 선택적으로 또는 기본적으로 설정되는 한계가 있습니다.

82 페이지의 『Windows에서 SPX 연결 정의』

SPX 연결은 Windows XP 및 Windows 2003 Server를 실행 중인 클라이언트와 서버에만 적용됩니다.

관련 참조

92 페이지의 『TCP/IP 연결 한계』

단일 TCP/IP 포트에 큐 대기할 수 있는 미해결 연결 요청의 수는 플랫폼에 따라 다릅니다. 한계에 도달하면 오류가 발생합니다.

TCP/IP 연결 정의

클라이언트 측 채널 정의에서 TCP로 전송 유형을 지정합니다. 서버에서 리스너 프로그램을 시작하십시오.

채널 정의에서 전송 유형으로 TCP를 지정하여 클라이언트에서 TCP/IP 연결을 지정하십시오.

수신 채널 프로그램은 송신 채널의 시동 요청에 대한 응답으로 시작됩니다. 이를 수행하려면 리스너 프로그램이 시작되어 수신되는 네트워크 요청을 감지하고 연관된 채널을 시작해야 합니다. 리스너 프로그램을 시작하기 위한 프로시저는 서버 플랫폼에 따라 다릅니다.

클라이언트 및 서버 플랫폼에 대한 관련 주제를 참조하십시오.

TCP/IP 연결 한계

단일 TCP/IP 포트에 큐 대기할 수 있는 미해결 연결 요청의 수는 플랫폼에 따라 다릅니다. 한계에 도달하면 오류가 발생합니다.

이 연결 한계는 IBM WebSphere MQ 서버에 접속할 수 있는 최대 클라이언트 수와 같지 않습니다. 서버 시스템 자원에 의해 판별되는 레벨까지 서버에 추가 클라이언트를 연결할 수 있습니다. 연결 요청에 대한 백로그 값이 다음 테이블에 표시되어 있습니다.

표 15. TCP/IP 포트에서 큐잉되는 최대 미해결 연결 요청 수	
서버 플랫폼	최대 연결 요청 수
AIX	100
HP-UX	20
Linux	100
IBM	255임)
Solaris	100
Windows 서버	100
Windows 워크스테이션	100
z/OS	255임)

연결 한계에 도달하면 클라이언트는 MQCONN 호출에서 MQRC_HOST_NOT_AVAILABLE 리턴 코드를 수신하고 클라이언트 오류 로그의 AMQ9202 오류 (UNIX and Linux 시스템의 경우/var/mqm/errors/

AMQERR0n.LOG 또는 Windows의 IBM WebSphere MQ 클라이언트 설치의 오류 서브디렉토리에 있는 amqerr0n.log) 를 수신합니다. 클라이언트가 MQCONN 요청을 재시도하면 성공할 수도 있습니다.

수행할 수 있는 연결 요청의 수를 늘리고 이 제한으로 인해 오류가 생성되는 것을 피하기 위해 각각 다른 포트를 청취하는 다중 리스너를 보유하거나 둘 이상의 큐 관리자를 둘 수 있습니다.

NetBIOS 또는 SPX 연결 정의

NetBIOS 및 SPX 연결은 Windows 시스템에만 적용됩니다.

NetBIOS 연결은 Windows에서 실행되는 클라이언트 및 서버에만 적용됩니다. [NetBIOS 연결 정의](#)를 참조하십시오.

SPX 연결은 Windows XP 또는 Windows 2003 Server를 실행하는 클라이언트 및 서버에만 적용됩니다. [SPX 연결 정의](#)를 참조하십시오.

확장된 트랜잭션 클라이언트 구성

이 주제 콜렉션에서는 각 트랜잭션 관리자 범주에 대해 확장된 트랜잭션 기능을 구성하는 방법을 설명합니다.

각 플랫폼에서 확장된 트랜잭션 클라이언트는 다음 외부 트랜잭션 관리자에 대한 지원을 제공합니다.

XA 준수 트랜잭션 관리자

확장 트랜잭션 클라이언트는 XA 자원 관리자 인터페이스 CICS 및 Tuxedo와 같은 XA-호환 트랜잭션 관리자에 대한 XA 자원 관리자 인터페이스를 제공합니다.

Microsoft Transaction Server(Windows 시스템 전용)

Windows 시스템에서만 XA 자원 관리자 인터페이스가 MTS(Microsoft Transaction Server)를 지원합니다. 확장 트랜잭션 클라이언트와 함께 제공된 WebSphere MQ MTS 지원은 MTS와 XA 자원 관리자 인터페이스 사이의 브릿지를 제공합니다.

WebSphere Application Server

WebSphere MQ의 이전 버전은 WebSphere 애플리케이션 서버 버전 4 또는 버전 5를 지원하며 확장 트랜잭션 클라이언트 사용을 위해 사용자가 특정 구성 태스크를 수행하도록 요구합니다. WebSphere 애플리케이션 서버 버전 6 이상은 WebSphere MQ 메시징 제공자를 포함하므로 사용자가 확장 트랜잭션 클라이언트를 사용할 필요가 없습니다.

관련 개념

[93 페이지의 『XA 준수 트랜잭션 관리자 구성』](#)

먼저 WebSphere MQ 기본 클라이언트를 구성한 다음 이 주제에 있는 정보를 사용하여 확장 트랜잭션 기능을 구성하십시오.

[100 페이지의 『Server』](#)

MTS를 트랜잭션 관리자로 사용하기 전에 필요한 추가 구성은 없습니다. 그러나 몇 가지 참고해야 할 점이 있습니다.

XA 준수 트랜잭션 관리자 구성

먼저 WebSphere MQ 기본 클라이언트를 구성한 다음 이 주제에 있는 정보를 사용하여 확장 트랜잭션 기능을 구성하십시오.

참고: 이 절에서는 The Open Group에 의해 발행된 분산 트랜잭션 처리: XA 스펙에 나와 있는 XA 인터페이스에 대한 기본적 이해가 있는 것으로 가정합니다.

Extended Transactional Client를 구성하려면 먼저 [IBM WebSphere MQ 클라이언트 설치](#)에 설명된 대로 WebSphere MQ 기본 클라이언트를 구성해야 합니다. 그런 다음 이 절의 정보를 사용하여 CICS 및 Tuxedo와 같은 XA 준수 트랜잭션 관리자의 Extended Transactional Function을 구성할 수 있습니다.

트랜잭션 관리자는 큐 관리자에 연결되는 클라이언트 애플리케이션이 사용하는 것과 동일한 MQI 채널을 사용하는 자원 관리자로 큐 관리자와 통신합니다. 트랜잭션 관리자가 자원 관리자(xa_) 함수 호출을 발행하면 큐 관리자에 대한 호출을 전달하고 큐 관리자로부터 다시 출력을 수신하기 위해 MQI 채널이 사용됩니다.

트랜잭션 관리자가 큐 관리자를 자원 관리자로 열기 위해 xa_open 호출을 발행하여 MQI 채널을 시작하거나 클라이언트 애플리케이션이 MQCONN 또는 MQCONNx 호출을 발행하여 MQI 채널을 시작할 수 있습니다.

- 트랜잭션 관리자가 MQI 채널을 시작하고 같은 스레드에서 클라이언트 애플리케이션이 나중에 MQCONN 또는 MQCONNX를 호출하는 경우 MQCONN 또는 MQCONNX 호출이 성공적으로 완료되고 연결 핸들이 애플리케이션에 리턴됩니다. 애플리케이션은 MQRC_ALREADY_CONNECTED 이유 코드와 함께 MQCC_WARNING 완료 코드를 수신하지 않습니다.
- 클라이언트 애플리케이션이 MQI 채널을 시작하고 같은 스레드에서 트랜잭션 관리자가 나중에 xa_open을 호출하는 경우 xa_open 호출이 이 MQI 채널을 사용하는 큐 관리자에게 전달됩니다.

실패 후 복구 상황에서 실행 중인 클라이언트 애플리케이션이 없는 경우 트랜잭션 관리자는 전용 MQI 채널을 사용하여 실패 시 큐 관리자가 참여했던 불완전한 작업 단위를 복구할 수 있습니다.

확장된 트랜잭션 클라이언트를 XA 준수 트랜잭션 관리자와 함께 사용하는 경우 다음 조건을 참고하십시오.

- 단일 스레드 내에서 클라이언트 애플리케이션은 한 번에 한 큐 관리자에만 연결할 수 있습니다. 이 제한은 확장 트랜잭션 클라이언트를 사용할 때에만 적용되며 WebSphere MQ 기본 클라이언트를 사용 중인 클라이언트 애플리케이션은 단일 스레드 내에서 동시에 둘 이상의 큐 관리자에 연결될 수 있습니다.
- 클라이언트 애플리케이션의 각 스레드는 서로 다른 큐 관리자에 연결할 수 있습니다.
- 클라이언트 애플리케이션은 공유 연결 핸들을 사용할 수 없습니다.

확장된 트랜잭션 기능을 구성하려면 자원 관리자 역할을 하는 각 큐 관리자에 대한 다음 정보를 트랜잭션 관리자에게 제공해야 합니다.

- xa_open 문자열
- XA 스위치 구조에 대한 포인터

큐 관리자를 자원 관리자로 열기 위해 트랜잭션 관리자가 xa_open을 호출하면 호출 시 인수 xa_info로 xa_open 문자열이 확장된 트랜잭션 클라이언트에 전달됩니다. 확장된 트랜잭션 클라이언트는 다음과 같은 방식으로 xa_open 문자열의 정보를 사용합니다.

- 클라이언트 애플리케이션이 아직 MQI 채널을 시작하지 않은 경우 서버 큐 관리자에 대한 MQI 채널을 시작함
- 트랜잭션 관리자가 자원 관리자로 여는 큐 관리자와 클라이언트 애플리케이션이 연결하는 큐 관리자와 같은지 확인함
- 큐 관리자가 동적 등록을 사용하는 경우 트랜잭션 관리자의 ax_reg 및 ax_unreg 함수를 찾음

xa_open 문자열의 형식 및 확장된 트랜잭션 클라이언트가 xa_open 문자열의 정보를 사용하는 방법에 대한 자세한 정보는 95 페이지의 『[xa_open 문자열의 형식](#)』의 내용을 참조하십시오.

XA 스위치 구조는 트랜잭션 관리자가 확장된 트랜잭션 클라이언트가 제공한 xa_ 함수를 찾을 수 있게 하고 큐 관리자가 동적 등록을 사용하는지 여부를 지정합니다. 확장된 트랜잭션 클라이언트에 제공되는 XA 스위치 구조에 대한 정보는 98 페이지의 『[XA 스위치 구조](#)』의 내용을 참조하십시오.

특정 트랜잭션 관리자에 대해 확장된 트랜잭션 기능을 구성하는 방법에 대한 정보 및 확장된 트랜잭션 클라이언트에서 트랜잭션 관리자를 사용하는 것에 대한 그 외 정보는 다음 절을 참조하십시오.

- 99 페이지의 『[CICS를 위한 확장 트랜잭션 클라이언트 구성](#)』
- 100 페이지의 『[Tuxedo를 위한 확장된 트랜잭션 클라이언트 구성](#)』

관련 개념

96 페이지의 『[xa_open 문자열의 CHANNEL, TRPTYPE, CONNAME 및 QMNAME 매개변수](#)』

이 정보를 사용하여 확장된 트랜잭션 클라이언트가 연결할 큐 관리자를 판별하는 데 이러한 매개변수를 사용하는 방법에 대해 이해할 수 있습니다.

97 페이지의 『[xa_open에 대한 추가 오류 처리](#)』

xa_open 호출은 특정 환경에서 실패합니다.

관련 태스크

98 페이지의 『[SSL 채널에서 확장된 트랜잭션 클라이언트 사용](#)』

xa_open 문자열을 사용하여 SSL 채널을 설정할 수 없습니다. 클라이언트 채널 정의 테이블(ccdt)을 사용하려면 다음 지시사항을 따르십시오.

관련 참조

97 페이지의 『[TPM 및 AXLIB 매개변수](#)』

확장된 트랜잭션 클라이언트는 TPM 및 AXLIB 매개변수를 사용하여 트랜잭션 관리자의 ax_reg 및 ax_unreg 함수를 찾습니다. 이러한 함수는 큐 관리자가 동적 등록을 사용하는 경우에만 사용됩니다.

97 페이지의 『확장된 트랜잭션 처리에서 실패 후 복구』

실패 후 트랜잭션 관리자는 불완전한 작업 단위를 복구할 수 있어야 합니다. 이를 수행하려면 트랜잭션 관리자가 실패 시 불완전한 작업 단위에 참여했던 큐 관리자를 자원 관리자로 열 수 있어야 합니다.

xa_open 문자열의 형식

xa_open 문자열에는 정의된 매개변수 이름 및 값 쌍이 들어 있습니다.

xa_open 문자열은 다음과 같은 형식을 가집니다.

```
parm_name1=parm_value1,parm_name2=parm_value2, ...
```

여기서 *parm_name*은 매개변수의 이름이고 *parm_value*는 매개변수의 값입니다. 매개변수의 이름은 대소문자를 구분하지 않지만 별도로 지정하지 않는 한 매개변수의 값은 대소문자를 구분합니다. 매개변수는 순서와 관계없이 지정할 수 있습니다.

매개변수의 이름, 의미 및 올바른 값은 다음과 같습니다.

이름

의미 및 올바른 값

CHANNEL

MQI 채널의 이름입니다.

이 매개변수는 선택적 매개변수입니다. 이 매개변수가 제공되면 CONNAME 매개변수도 제공되어야 합니다.

TRPTYPE

MQI 채널의 통신 프로토콜입니다. 올바른 값은 다음과 같습니다.

LU62

SNA LU 6.2

NETBIOS

NetBIOS

SPX

IPX/SPX

TCP

TCP/IP

이 매개변수는 선택적 매개변수입니다. 이 매개변수가 생략되면 기본값 TCP로 가정합니다. 이 매개변수의 값은 대소문자를 구분하지 않습니다.

CONNAME

MQI 채널의 서버 측 큐 관리자의 네트워크 주소입니다. 이 매개변수의 올바른 값은 TRPTYPE 매개변수의 값에 따라 다릅니다.

LU62

CPI-C 부가 정보 입력 항목을 식별하는 기호 대상 이름입니다.

파트너 LU의 네트워크 규정 이름은 올바른 값이 아니며 파트너 LU 알리어스도 아닙니다. 이는 트랜잭션 프로그램(TP) 이름 및 모드 이름을 지정하기 위한 추가 매개변수가 없기 때문입니다.

NETBIOS

NetBIOS 이름입니다.

SPX

4바이트 네트워크 주소, 6 바이트 노드 주소 및 선택적 2바이트 소켓 번호입니다. 이러한 값은 16진 표기법으로 지정되어야 합니다. 마침표로 네트워크 및 노드 주소를 구분해야 하고 소켓 번호가 제공되는 경우 괄호로 묶어야 합니다. 예를 들면, 다음과 같습니다.

```
0a0b0c0d.804abcde23a1(5e86)
```

소켓 번호가 생략되는 경우 기본값 5e86으로 가정됩니다.

TCP

호스트 이름 또는 IP 주소이며 선택적으로 괄호로 묶은 포트 번호가 뒤에 올 수 있습니다. 포트 번호가 생략되는 경우 기본값 1414로 가정됩니다.

이 매개변수는 선택적 매개변수입니다. 이 매개변수가 제공되면 CHANNEL 매개변수도 제공되어야 합니다.

QMNAME

MQI 채널의 서버 측 큐 관리자의 이름입니다. 이 이름은 공백이거나 단일 별표(*)일 수 없으며 별표로 시작되는 이름일 수 없습니다. 이는 이 매개변수가 이름으로 특정 큐 관리자를 식별함을 의미합니다.

이 매개변수는 필수 매개변수입니다.

클라이언트 애플리케이션이 특정 큐 관리자에 연결되면 같은 큐 관리자에 의해 트랜잭션 복구가 처리되어야 합니다.

애플리케이션이 z/OS 큐 관리자에 연결되면 애플리케이션은 특정 큐 관리자의 이름 또는 QSG(Queue-Sharing Group)의 이름을 지정할 수 있습니다. 큐 관리자 이름 또는 QSG 이름을 사용하여 애플리케이션은 QMGR 복구 단위 속성 지정 또는 GROUP 복구 단위 속성 지정과의 트랜잭션에 참여할지 여부를 제어합니다. GROUP 복구 단위 속성 지정을 사용하여 트랜잭션 복구가 QSG의 모든 멤버에서 처리될 수 있습니다. GROUP 복구 단위를 사용하려면 **GROUPUR** 큐 관리자 속성을 사용으로 설정해야 합니다.

TPM

사용 중인 트랜잭션 관리자입니다. 올바른 값은 CICS 및 TUXEDO입니다.

확장된 트랜잭션 클라이언트는 이 매개변수 및 AXLIB 매개변수를 같은 목적으로 사용합니다. 이러한 매개변수에 대한 자세한 정보는 [TPM 및 AXLIB 매개변수](#)를 참조하십시오.

이 매개변수는 선택적 매개변수입니다. 이 매개변수의 값은 대소문자를 구분하지 않습니다.

AXLIB

트랜잭션 관리자의 ax_reg 및 ax_unreg 함수를 포함하는 라이브러리의 이름입니다.

이 매개변수는 선택적 매개변수입니다.

다음은 xa_open 문자열의 예입니다.

```
channel=MARS.SVR,trdtype=tcp,connname=MARS(1415),qmname=MARS,tpm=cics
```

xa_open 문자열의 CHANNEL, TRPTYPE, CONNAME 및 QMNAME 매개변수

이 정보를 사용하여 확장된 트랜잭션 클라이언트가 연결할 큐 관리자를 판별하는 데 이러한 매개변수를 사용하는 방법에 대해 이해할 수 있습니다.

CHANNEL 및 CONNAME 매개변수가 xa_open 문자열에 제공되면 확장된 트랜잭션 클라이언트가 이러한 매개변수 및 TRPTYPE 매개변수를 사용하여 서버 큐 관리자에 대한 MQI 채널을 시작합니다.

CHANNEL 및 CONNAME 매개변수가 xa_open 문자열에 제공되지 않으면 확장된 트랜잭션 클라이언트가 MQSERVER 환경 변수의 값을 사용하여 MQI 채널을 시작합니다. MQSERVER 환경 변수가 정의되지 않으면 확장된 트랜잭션 클라이언트가 QMNAME 매개변수로 식별되는 클라이언트 채널 정의의 입력 항목을 사용합니다.

이러한 각각의 경우에서 확장된 트랜잭션 클라이언트는 QMNAME 매개변수의 값이 MQI 채널의 서버 측 큐 관리자 이름인지 확인합니다. 그렇지 않은 경우 xa_open 호출이 실패하고 트랜잭션 관리자가 애플리케이션에 실패를 보고합니다.

애플리케이션 클라이언트가 V7.0.1 또는 나중에 z/OS 큐 관리자에 연결 중인 경우 QMNAME 매개변수에 QSG(queue-sharing group) 이름을 지정할 수 있습니다. 이 경우 애플리케이션 클라이언트는 GROUP 복구 단위 속성 지정과의 트랜잭션에 참여할 수 있습니다.

애플리케이션이 QMNAME 매개변수 필드에 QSG 이름을 사용하고 애플리케이션이 연결하는 큐 관리자에서 GROUPUR 특성이 사용 안함으로 설정되는 경우 xa_open 호출이 실패합니다.

애플리케이션이 V7.0.1 이전 버전 큐 관리자에 연결하는 경우 xa_open 호출은 성공하지만 트랜잭션이 QMGR 복구 단위 속성 지정을 가집니다. GROUP 복구 단위 속성 지정이 필요한 애플리케이션은 V7.0.1 이상의 큐 관리자에만 연결하도록 하십시오.

트랜잭션 관리자가 xa_open 호출을 발행하는 데 사용한 것과 같은 스레드에서 클라이언트 애플리케이션이 나중에 MQCONN 또는 MQCONNX를 호출하는 경우 이 애플리케이션은 xa_open 호출에 의해 시작된 MQI 채널에 대

한 연결 핸들을 수신합니다. 두 번째 MQI 채널이 시작되지 않습니다. 확장된 트랜잭션 클라이언트가 MQCONN 또는 MQCONNX 호출 시 QMgrName 매개변수의 값이 MQI 채널의 서버 측 큐 관리자의 이름인지 확인합니다. 그렇지 않은 경우 MQCONN 또는 MQCONNX 호출이 MQRC_ANOTHER_Q_MGR_CONNECTED 이유 코드로 실패합니다. QMgrName 매개변수의 값이 공백이거나 단일 별표(*)이거나 별표로 시작되는 경우 MQCONN 또는 MQCONNX 호출이 MQRC_Q_MGR_NAME_ERROR 이유 코드로 실패합니다.

트랜잭션 관리자가 같은 스레드에서 xa_open을 호출하기 전에 클라이언트 애플리케이션이 MQCONN 또는 MQCONNX를 호출하여 MQI 채널을 이미 시작한 경우 트랜잭션 관리자는 이 MQI 채널을 대신 사용합니다. 두 번째 MQI 채널이 시작되지 않습니다. 확장된 트랜잭션 클라이언트가 xa_open 문자열의 QMNAME 매개변수 값이 서버 큐 관리자의 이름인지 확인합니다. 그렇지 않은 경우 xa_open 호출이 실패합니다.

클라이언트 애플리케이션이 MQI 채널을 먼저 시작하는 경우 MQCONN 또는 MQCONNX 호출 시 QMgrName 매개변수의 값은 공백이거나 단일 별표(*)이거나 별표로 시작할 수 있습니다. 그러나 이러한 상황에서는 트랜잭션 관리자가 같은 스레드에서 나중에 xa_open을 호출할 때 자원 관리자로 열려고 하는 큐 관리자와 애플리케이션이 연결하는 큐 관리자가 같은지 확인해야 합니다. QMgrName 매개변수의 값이 이름에 따라 명시적으로 큐 관리자를 식별하는 경우 문제점이 더 적게 발생할 수 있습니다.

TPM 및 AXLIB 매개변수

확장된 트랜잭션 클라이언트는 TPM 및 AXLIB 매개변수를 사용하여 트랜잭션 관리자의 ax_reg 및 ax_unreg 함수를 찾습니다. 이러한 함수는 큐 관리자가 동적 등록을 사용하는 경우에만 사용됩니다.

TPM 매개변수가 xa_open 문자열에 제공되지만 AXLIB 매개변수는 제공되지 않는 경우 확장된 트랜잭션 클라이언트는 TPM 매개변수의 값을 기반으로 AXLIB 매개변수의 값을 가정합니다. AXLIB 매개변수의 가정 값에 대해서는 97 페이지의 표 16의 내용을 참조하십시오.

표 16. AXLIB 매개변수의 가정 값		
TPM의 값	플랫폼	AXLIB의 가정 값
CICS	AIX	/usr/lpp/encina/lib/libEncServer.a(EncServer_shr.o)
CICS	HP-UX	/opt/encina/lib/libEncServer.sl
CICS	Solaris	/opt/encina/lib/libEncServer.so
CICS	Windows 시스템	libEncServer
Tuxedo	AIX	/usr/lpp/tuxedo/lib/libtux.a(libtux.so.60)
Tuxedo	HP-UX	/opt/tuxedo/lib/libtux.sl
Tuxedo	Solaris	/opt/tuxedo/lib/libtux.so.60
Tuxedo	Windows 시스템	libtux

AXLIB 매개변수가 xa_open 문자열에 제공되는 경우 확장된 트랜잭션 클라이언트는 그 값을 사용하여 TPM 매개변수의 값을 기반으로 하여 가정된 값을 대체합니다. AXLIB 매개변수는 TPM 매개변수가 지정된 값을 갖지 않는 트랜잭션 관리자에 대해서도 사용할 수 있습니다.

xa_open에 대한 추가 오류 처리

xa_open 호출은 특정 환경에서 실패합니다.

이 절의 주제에서는 xa_open 호출이 실패하는 상황에 대해 설명합니다. 이 호출은 다음 상황 중 어느 하나가 발생하는 경우에도 실패합니다.

- xa_open 문자열에 오류가 있습니다.
- MQI 채널을 시작하기에는 정보가 충분하지 않습니다.
- MQI 채널을 시작하려고 시도하는 중에 문제점이 발생했습니다(예: 서버 큐 관리자가 실행되지 않음).

확장된 트랜잭션 처리에서 실패 후 복구

실패 후 트랜잭션 관리자는 불완전한 작업 단위를 복구할 수 있어야 합니다. 이를 수행하려면 트랜잭션 관리자가 실패 시 불완전한 작업 단위에 참여했던 큐 관리자를 자원 관리자로 열 수 있어야 합니다.

그러므로 구성 정보를 변경할 필요가 있는 경우, 변경사항을 작성하기 전에 완료되지 않은 모든 작업 단위를 해결해야 합니다. 또는 구성 변경이 트랜잭션 관리자가 열어야 하는 큐 관리자를 여는 기능에 영향을 주지 않는지 확인해야 합니다. 다음은 그러한 구성 변경의 예입니다.

- xa_open 문자열의 내용 변경
- MQSERVER 환경 변수의 값 변경
- 클라이언트 채널 정의 테이블(CCDT)의 입력 항목 변경
- 서버 연결 채널 정의 삭제

XA 스위치 구조

두 개의 XA 스위치 구조가 각 플랫폼의 확장된 트랜잭션 클라이언트에 제공됩니다.

이러한 스위치 구조는 다음과 같습니다.

MQRMIXASwitch

이 스위치 구조는 자원 관리자 역할을 하는 큐 관리자가 동적 등록을 사용하지 않는 경우에 트랜잭션 관리자가 사용합니다.

MQRMIXASwitchDynamic

이 스위치 구조는 자원 관리자 역할을 하는 큐 관리자가 동적 등록을 사용하는 경우에 트랜잭션 관리자가 사용합니다.

이러한 스위치 구조는 [98 페이지의 표 17](#)에 표시된 라이브러리에 있습니다.

표 17. XA 스위치 구조가 있는 WebSphere MQ 라이브러리	
플랫폼	XA 스위치 구조를 포함하는 라이브러리
AIX HP-UX Linux Solaris	MQ_INSTALLATION_PATH/lib/libmqcxa
Windows 시스템	MQ_INSTALLATION_PATH\bin\mqcxa.dll ¹
MQ_INSTALLATION_PATH는 WebSphere MQ가 설치되어 있는 상위 레벨 디렉토리를 표시합니다.	

각 스위치 구조에서 WebSphere MQ 자원 관리자의 이름은 MQSeries_XA_RMI이지만 다중 큐 관리자가 동일한 스위치 구조를 공유할 수 있습니다.

관련 개념

[98 페이지의 『동적 등록 및 확장된 트랜잭션 처리』](#)

동적 등록 사용은 트랜잭션 관리자가 발행하는 xa_ 함수 호출의 수를 줄일 수 있기 때문에 최적화된 양식입니다.

동적 등록 및 확장된 트랜잭션 처리

동적 등록 사용은 트랜잭션 관리자가 발행하는 xa_ 함수 호출의 수를 줄일 수 있기 때문에 최적화된 양식입니다.

큐 관리자가 동적 등록을 사용하지 않는 경우 트랜잭션 관리자는 모든 작업 단위에 큐 관리자를 관여시킵니다. 트랜잭션 관리자는 큐 관리자에게 작업 단위 내에서 업데이트되는 자원이 없더라도 xa_start, xa_end 및 xa_prepare를 호출하여 이를 수행합니다.

큐 관리자가 동적 등록을 사용하는 경우 트랜잭션 관리자는 큐 관리자가 작업 단위에 관여하지 않는 것으로 가정하여 시작하고 xa_start를 호출하지 않습니다. 그러면 큐 관리자는 해당 자원이 동기점 제어 내에서 업데이트되는 경우에만 작업 단위에 관여하게 됩니다. 이런 경우가 발생하면 확장 트랜잭션 클라이언트가 ax_reg를 호출하여 큐 관리자의 관여를 등록합니다.

SSL 채널에서 확장된 트랜잭션 클라이언트 사용

xa_open 문자열을 사용하여 SSL 채널을 설정할 수 없습니다. 클라이언트 채널 정의 테이블(ccdt)을 사용하려면 다음 지시사항을 따르십시오.

이 태스크 정보

xa_open xa_info 문자열의 제한된 크기로 인해 큐 관리자에 연결하는 xa_open 문자열 메소드를 사용하여 SSL 채널을 설정하는 데 필요한 모든 정보를 전달하는 것은 불가능합니다. 따라서 클라이언트 채널 정의 테이블을 사용해야 합니다. 또는 트랜잭션 관리자가 허용하는 경우 xa_open 호출을 발행하기 전에 MQCONN로 채널을 적성해야 합니다.

클라이언트 채널 정의 테이블을 사용하려면 다음 단계를 따르십시오.

프로시저

1. 필수 qmname(큐 관리자 이름) 매개변수만 포함하는 xa_open 문자열을 지정하십시오. 예를 들면, 다음과 같습니다. XA_Open_String=qmname=MYQM
2. 큐 관리자를 사용하여 필수 SSL 매개변수로 CLNTCONN(클라이언트 연결) 채널을 정의하십시오. 큐 관리자 이름을 CLNTCONN 정의의 QMNAME 속성에 포함시키십시오. 이는 xa_open 문자열의 qmname과 일치됩니다.
3. 클라이언트 채널 정의 테이블(CCDT) 또는 Windows의 활성 디렉토리에서 CLNTCONN 정의가 클라이언트 시스템에 사용 가능하게 하십시오.
4. CCDT를 사용 중인 경우 환경 변수 MQCHLLIB 및 MQCHLTAB를 사용하여 CLNTCONN 채널의 정의가 포함된 CCDT를 식별하십시오. 클라이언트 애플리케이션 및 트랜잭션 관리자 둘 다 사용하는 환경에서 이러한 변수를 설정하십시오.

결과

이는 SSLCIPH, CipherSpec 등의 올바르게 인증되어야 하는 SSL 속성이 있는 해당 큐 관리자에 대한 채널 정의를 트랜잭션 관리자에게 제공합니다.

CICS를 위한 확장 트랜잭션 클라이언트 구성

XAD 자원 정의를 CICS 영역에 추가하여 CICS에서 사용하도록 확장 트랜잭션 클라이언트를 구성합니다.

코믹스 자원 정의 온라인 (RDO) 명령 **cicsadd**을 사용하여 XAD 자원 정의를 추가하십시오. 이 XAD 자원 정의는 다음 정보를 지정합니다.

- xa_open 문자열
- 스위치 로드 파일의 완전한 경로 이름

하나의 스위치 로드 파일은 AIX, HP-UX, Solaris 및 윈도우 시스템 각각에서 코믹스 이 사용하도록 제공됩니다. 각 스위치 로드 파일은 동적 등록에 사용되는 XA 스위치 구조 MQRMIXASwitchDynamic에 대한 포인터를 리턴하는 기능을 포함합니다. 각 스위치 로드 파일의 완전한 경로 이름은 [99 페이지의 표 18](#)의 내용을 참조하십시오.

표 18. 스위치 로드 파일	
플랫폼	스위치 로드 파일
AIX HP-UX Linux Solaris	MQ_INSTALLATION_PATH/lib/amqczsc
Windows 시스템	MQ_INSTALLATION_PATH\bin\mqcc4swi.dll ¹

MQ_INSTALLATION_PATH는 WebSphere MQ가 설치되어 있는 상위 레벨 디렉토리를 표시합니다.

다음은 Windows 시스템의 XAD 자원 정의의 예입니다.

```
cicsadd -c xad -r REGION1 WMQXA \  
ResourceDescription="WebSphere MQ queue manager MARS" \  
XAOpen="channel=MARS.SVR,trptype=tcp,connname=MARS(1415),qmname=MARS,tpm=cics" \  
SwitchLoadFile="C:\Program Files\IBM\WebSphere MQ\bin\mqcc4swi.dll"
```

XAD 자원 정의를 코믹스 영역에 추가하는 방법에 자세한 정보는 코믹스 관리 참조 및 플랫폼의 코믹스 관리 안내서를 참조하십시오.

확장 트랜잭션 클라이언트에서 CICS를 사용할 때 다음 정보를 참고하십시오.

- WebSphere MQ에 대한 하나의 XAD 자원 정의만을 CICS 영역에 추가할 수 있습니다. 이는 하나의 큐 관리자만이 리전과 연관될 수 있으며 리전에서 실행되는 모든 애플리케이션 해당 큐 관리자에만 연결할 수 있음을 의미합니다. 다른 큐 관리자에 연결하는 애플리케이션 실행하려면 다른 리전에서 애플리케이션을 실행해야 합니다.
- 리전 내 각 애플리케이션 서버는 초기화 중 `xa_open`을 호출하고 리전과 연관된 큐 관리자에 대한 MQI 채널을 시작합니다. 이는 애플리케이션 서버가 시작되기 전에 큐 관리자를 시작해야 하고 그렇지 않은 경우 `xa_open` 호출이 실패함을 의미합니다. 나중에 애플리케이션 서버에서 처리되는 모든 WebSphere MQ MQI 클라이언트 애플리케이션은 동일한 MQI 채널을 사용합니다.
- MQI 채널이 시작되고 채널의 클라이언트 끝에 보안 엑시트가 없는 경우, 클라이언트 시스템에서 서버 연결 MCA로 흐르는 사용자 ID는 `cics`입니다. Under certain circumstances, the queue manager uses this user ID for authority checks when the server connection MCA subsequently attempts to access the queue manager resources on behalf of a client application. 이 사용자 ID가 권한 검사에 사용되는 경우 이 사용자 ID가 액세스해야 하는 모든 자원에 대한 액세스 권한을 가지고 있는지 확인해야 합니다.
큐 관리자가 권한 검사를 위해 이 사용자 ID를 사용하는 경우에 대한 정보는 [보안](#)을 참조하십시오.
- WebSphere MQ 클라이언트 시스템에서 사용하도록 제공된 코믹스 태스크 종료 엑시트가 **100 페이지의 표 19**에 나열됩니다. WebSphere MQ 서버 시스템에 대해 해당 엑시트를 구성하는 것과 동일한 방법으로 이러한 엑시트를 구성합니다. 그러므로 이에 대한 정보는 [CICS 사용자 엑시트 사용](#)을 참조하십시오.

표 19. CICS 태스크 종료 엑시트		
플랫폼	소스	라이브러리
AIX HP-UX Linux Solaris	amqzscgx.c	amqczscg
Windows 시스템	amqzscgn.c	mqcc1415.dll

Tuxedo를 위한 확장된 트랜잭션 클라이언트 구성

Tuxedo에서 사용하도록 XAD 자원 정의를 구성하려면 UBBCONFIG 파일 및 자원 관리자 테이블을 업데이트하십시오.

Tuxedo에서 사용할 XAD 자원 정의를 구성하려면 다음 조치를 수행하십시오.

- 애플리케이션의 Tuxedo UBBCONFIG 파일 GROUPS 섹션에서 OPENINFO 매개변수를 사용하여 `xa_open` 문자열을 지정하십시오.

이를 수행하는 방법에 대한 예는 Tuxedo 샘플 프로그램에서 사용하도록 제공된 샘플 UBBCONFIG 파일을 참조하십시오. AIX, HP-UX 및 Solaris에서 이 파일의 이름은 `ubbstxcx.cfg`이며 Windows 시스템에서 이 파일의 이름은 `ubbstxcn.cfg`입니다.

- Tuxedo 자원 관리자 테이블의 큐 관리자에 대한 항목에서:

- `udataobj/RM` (AIX, HP-UX 및 Solaris)
- `udataobj\rm` (윈도우 시스템)

XA 스위치 구조의 이름 및 이 구조를 포함하는 라이브러리의 완전한 경로 이름을 지정하십시오. 각 플랫폼에 대해 이를 수행하는 방법의 예는 [TUXEDO 샘플](#)을 참조하십시오. Tuxedo는 자원 관리자의 동적 등록을 지원하므로 `MQRMIASwitch` 또는 `MQRMIASwitchDynamic`을 사용할 수 있습니다.

Server

MTS를 트랜잭션 관리자 사용에 사용하기 전에 필요한 추가 구성은 없습니다. 그러나 몇 가지 참고해야 할 점이 있습니다.

확장된 트랜잭션 클라이언트에서 MTS를 사용하는 것에 대해 다음 정보를 참고하십시오.

- MTS 애플리케이션은 서버 큐 관리자에 연결할 때 항상 MQI 채널을 시작합니다. 트랜잭션 관리자 역할을 할 때 MTS는 같은 MQI 채널을 사용하여 큐 관리자와 통신합니다.
- 실패 후 MTS는 불완전한 작업 단위를 복구할 수 있어야 합니다. 이를 수행하려면 MTS가 실패 시 불완전한 작업 단위에 참여했던 큐 관리자와 통신할 수 있어야 합니다.

MTS 애플리케이션이 서버 큐 관리자에 연결하고 MQI 채널을 시작하는 경우 확장된 트랜잭션 클라이언트는 필요한 경우에 실패 후 채널이 재시작될 수 있도록 MQCONN 또는 MQCONNX 호출의 매개변수에서 충분한 정보를 추출합니다. 확장 트랜잭션 클라이언트는 이 정보를 MTS에 전달하고 MTS는 이 정보를 그 로그에 기록합니다.

MTS 애플리케이션이 MQCONN 호출을 발행하는 경우 이 정보는 단순히 큐 관리자의 이름입니다. MTS 애플리케이션이 MQCONNX 호출을 발행하고 채널 정의 구조인 MQCD를 제공하는 경우 이 정보에는 MQI 채널의 이름, 서버 큐 관리자의 네트워크 주소 및 채널의 통신 프로토콜이 포함됩니다.

복구 상황에서 MTS는 이 정보를 확장된 트랜잭션 클라이언트에 다시 전달하고 확장된 트랜잭션 클라이언트는 이 정보를 사용하여 MQI 채널을 재시작합니다.

구성 정보를 변경해야 하는 경우 변경을 작성하기 전에 불완전한 작업 단위가 모두 해결되었는지 확인하십시오. 또는 구성 변경이 확장된 트랜잭션 클라이언트가 MTS에서 기록된 정보를 사용하여 MQI 채널을 재시작하는 기능에 영향을 주지 않는지 확인하십시오. 다음은 그러한 구성 변경의 예입니다.

- MQSERVER 환경 변수의 값 변경
 - 클라이언트 채널 정의 테이블(CCDT)의 입력 항목 변경
 - 서버 연결 채널 정의 삭제
- 확장된 트랜잭션 클라이언트를 MTS와 함께 사용할 때 다음 조건을 참조하십시오.
 - 단일 스레드 내에서 클라이언트 애플리케이션은 한 번에 한 큐 관리자에만 연결할 수 있습니다.
 - 클라이언트 애플리케이션의 각 스레드는 서로 다른 큐 관리자에 연결할 수 있습니다.
 - 클라이언트 애플리케이션은 공유 연결 핸들을 사용할 수 없습니다.

MQI 채널 정의

새 채널 정의를 작성하려면 같은 채널 이름과 호환 가능한 채널 유형을 사용하여 연결의 양 끝에 하나씩 두 개 채널 정의를 작성해야 합니다. 이 경우 채널 유형은 *server-connection* 및 *client-connection*입니다.

사용자 정의 채널

서버가 자동으로 채널을 정의하지 않는 경우 채널 정의를 작성하여 WebSphere MQ MQI 클라이언트 시스템에서 WebSphere MQ 애플리케이션에 채널에 대한 액세스를 제공하는 두 가지 방법이 있습니다.

이러한 두 가지 방법은 다음에 자세히 설명되어 있습니다.

1. 하나의 채널 정의는 WebSphere MQ 클라이언트에서 작성하고 다른 하나는 서버에 작성하십시오.

이것은 모든 WebSphere MQ MQI 클라이언트 및 서버 플랫폼 조합에 적용됩니다. 시스템에서 시작하거나 설정을 테스트할 때 이 방법을 사용하십시오.

이 방법 사용에 대한 자세한 내용은 [102 페이지의 『다른 플랫폼에서 서버 연결 및 클라이언트 연결 정의 작성』](#)의 내용을 참조하십시오.

2. 서버 시스템에서 두 채널 정의를 모두 작성합니다.

다중 채널 및 WebSphere MQ MQI 클라이언트 시스템을 동시에 설정할 때 이 방법을 사용하십시오.

이 방법 사용에 대한 자세한 내용은 [105 페이지의 『서버에서 서버 연결 및 클라이언트 연결 정의 작성』](#)의 내용을 참조하십시오.

자동으로 정의되는 채널

WebSphere MQ products on platforms other than z/OS include a feature that can automatically create a channel definition on the server if one does not exist.

클라이언트로부터 인바운드 접속 요청이 수신될 때 해당 큐 관리자에서 적절한 서버 연결 정의를 찾을 수 없으면 WebSphere MQ는 자동으로 정의를 작성하여 큐 관리자에 추가합니다. 자동 정의는 기본 서버 연결 채널 정의 (SYSTEM.AUTO.SVRCONN)를 기반으로 합니다. CHAD 매개변수와 함께 ALTER QMGR 명령을 사용하여(또는 ChannelAutoDef 매개변수와 함께 PCF 명령 Change Queue Manager 사용) 큐 관리자 오브젝트를 업데이트함으로써 서버 연결 정의의 자동 정의를 사용할 수 있습니다.

채널 정의의 자동 작성에 대한 자세한 정보는 [수신자 및 서버 연결 채널의 자동 정의를 참조하십시오](#).

관련 개념

[102 페이지의 『자동으로 정의되는 채널』](#)

WebSphere MQ products on platforms other than z/OS include a feature that can automatically create a channel definition on the server if one does not exist.

[102 페이지의 『사용자 정의 채널』](#)

서버가 자동으로 채널을 정의하지 않는 경우 채널 정의를 작성하여 WebSphere MQ MQI 클라이언트 시스템에서 WebSphere MQ 애플리케이션에 채널에 대한 액세스를 제공하는 두 가지 방법이 있습니다.

[49 페이지의 『채널 제어 기능』](#)

채널 제어 기능은 채널을 정의, 모니터, 제어할 수 있는 기능을 제공합니다.

자동으로 정의되는 채널

WebSphere MQ products on platforms other than z/OS include a feature that can automatically create a channel definition on the server if one does not exist.

클라이언트로부터 인바운드 접속 요청이 수신될 때 해당 큐 관리자에서 적절한 서버 연결 정의를 찾을 수 없으면 WebSphere MQ는 자동으로 정의를 작성하여 큐 관리자에 추가합니다. 자동 정의는 기본 서버 연결 채널 정의 (SYSTEM.AUTO.SVRCONN)를 기반으로 합니다. CHAD 매개변수와 함께 ALTER QMGR 명령을 사용하여(또는 ChannelAutoDef 매개변수와 함께 PCF 명령 Change Queue Manager 사용) 큐 관리자 오브젝트를 업데이트함으로써 서버 연결 정의의 자동 정의를 사용할 수 있습니다.

사용자 정의 채널

서버가 자동으로 채널을 정의하지 않는 경우 채널 정의를 작성하여 WebSphere MQ MQI 클라이언트 시스템에서 WebSphere MQ 애플리케이션에 채널에 대한 액세스를 제공하는 두 가지 방법이 있습니다.

이러한 두 가지 방법은 다음에 자세히 설명되어 있습니다.

1. 하나의 채널 정의는 WebSphere MQ 클라이언트에서 작성하고 다른 하나는 서버에 작성하십시오.

이것은 모든 WebSphere MQ MQI 클라이언트 및 서버 플랫폼 조합에 적용됩니다. 시스템에서 시작하거나 설정을 테스트할 때 이 방법을 사용하십시오.

이 방법 사용에 대한 자세한 내용은 [102 페이지의 『다른 플랫폼에서 서버 연결 및 클라이언트 연결 정의 작성』](#)의 내용을 참조하십시오.

2. 서버 시스템에서 두 채널 정의를 모두 작성합니다.

다중 채널 및 WebSphere MQ MQI 클라이언트 시스템을 동시에 설정할 때 이 방법을 사용하십시오.

이 방법 사용에 대한 자세한 내용은 [105 페이지의 『서버에서 서버 연결 및 클라이언트 연결 정의 작성』](#)의 내용을 참조하십시오.

다른 플랫폼에서 서버 연결 및 클라이언트 연결 정의 작성

적용할 컴퓨터에서 각 채널 정의를 작성할 수 있습니다. 클라이언트 컴퓨터에서 채널 정의를 작성하는 방법에는 제한이 있습니다.

모든 플랫폼에서는 WebSphere MQ 스크립트(MQSC) 명령, PCF(programmable command format) 명령, 또는 IBM WebSphere MQ 탐색기를 사용하여 서버 시스템에서 서버 연결 채널을 정의할 수 있습니다.

WebSphere MQ가 WebSphere MQ MQI 클라이언트에서만 설치된 시스템에서 MQSC 명령이 사용 가능하지 않기 때문에 클라이언트 시스템에서 클라이언트 연결 채널을 정의하는 다른 방법을 사용해야 합니다.

관련 개념

[103 페이지의 『IBM WebSphere MQ MQI 클라이언트에서 클라이언트 연결 채널 작성』](#)

MQCONNX 호출 시 MQSERVER 또는 MQCNO 구조를 사용하여 클라이언트 워크스테이션에서 클라이언트 연결 채널을 정의할 수 있습니다.

관련 태스크

103 페이지의 『서버에서 서버 연결 채널 정의』

필요한 경우 MQSC를 시작한 다음 서버 연결 채널을 정의합니다.

서버에서 서버 연결 채널 정의

필요한 경우 MQSC를 시작한 다음 서버 연결 채널을 정의합니다.

프로시저

1. 옵션: 서버 플랫폼이 z/OS가 아닌 경우 먼저 큐 관리자를 작성하고 시작한 다음 MQSC 명령을 시작하십시오.
 - a) 예를 들어, QM1이라고 하는 큐 관리자를 작성하십시오.

```
critmqm QM1
```

- b) 큐 관리자를 시작하십시오.

```
strmqm QM1
```

- c) MQSC 명령을 시작하십시오.

```
runmqsc QM1
```

2. 선택한 이름 및 채널 유형 서버 연결을 사용하여 채널을 정의하십시오.

```
DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_1')
```

이 채널 정의는 서버에서 실행되는 큐 관리자와 연관됩니다.

3. 다음 명령을 사용하여 큐 관리자에 대한 인바운드 연결 액세스를 허용하십시오.

```
SET CHLAUTH(CHAN1) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- 여기서, SET CHLAUTH는 이전 단계에서 정의된 채널의 이름을 사용합니다.
- 여기서, 'IP address'는 클라이언트의 IP 주소입니다.
- 여기서, 'userid'는 대상 큐에 대한 액세스 제어를 위해 채널에 제공할 ID입니다. 이 필드는 대소문자를 구분합니다.

다양한 속성을 사용하여 인바운드 연결을 식별하도록 선택할 수 있습니다. 이 예는 IP 주소를 사용합니다. 대체 속성에는 클라이언트 사용자 ID 및 SSL 또는 TLS 주제 식별 이름 등이 있습니다. 자세한 정보는 [채널 인증 레코드](#)를 참조하십시오.

IBM WebSphere MQ MQI 클라이언트에서 클라이언트 연결 채널 작성

MQCONNX 호출 시 MQSERVER 또는 MQCNO 구조를 사용하여 클라이언트 워크스테이션에서 클라이언트 연결 채널을 정의할 수 있습니다.

MQSERVER 사용

MQSERVER 환경 변수를 사용하여 클라이언트 연결 채널의 단순 정의를 지정할 수 있습니다. 이 방법을 사용하여 적은 수의 채널 속성만 지정할 수 있다는 면에서 단순 정의입니다.

- Windows에서 다음과 같이 단순 채널 정의를 지정하십시오.

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

- UNIX and Linux 시스템에서 다음과 같이 단순 채널 정의를 지정하십시오.

```
export MQSERVER=ChannelName/TransportType/ConnectionName
```

설명:

- ChannelName은 서버에서 정의되는 이름과 같은 이름이어야 합니다. 슬래시는 포함시킬 수 없습니다.
- TransportType은 사용하는 IBM WebSphere MQ MQI 클라이언트 플랫폼에 따라 다음 값 중 하나일 수 있습니다.
 - LU62
 - TCP
 - NETBIOS
 - SPX

참고: UNIX and Linux 시스템에서 TransportType은 대소문자를 구분하고 대문자여야 합니다. TransportType이 인식되지 않는 경우 MQCONN 또는 MQCONNX 호출이 2058을 리턴합니다.

- ConnectionName은 통신 프로토콜(TransportType)에 정의된 서버의 이름입니다.

예를 들어, Windows의 경우는 다음과 같습니다.

```
SET MQSERVER=CHANNEL1/TCP/MCID66499
```

UNIX and Linux 시스템의 경우는 다음과 같습니다.

```
export MQSERVER=CHANNEL1/TCP/'MCID66499'
```

참고: TCP/IP 포트 번호를 변경하려면 134 페이지의 『MQSERVER』의 내용을 참조하십시오.

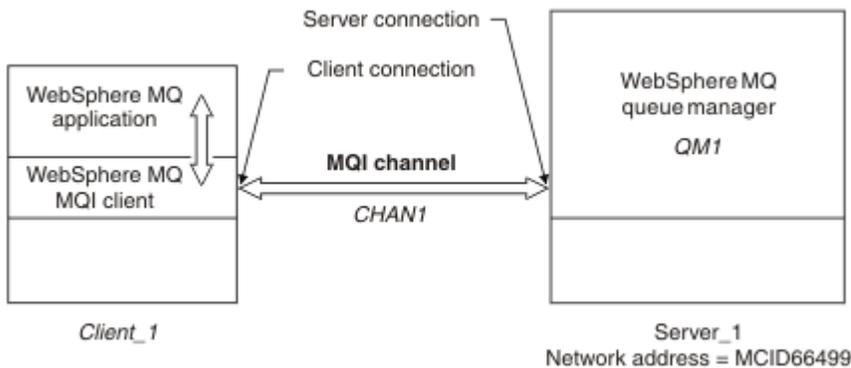


그림 17. 단순 채널 정의

단순 채널 정의의 몇 가지 다른 예는 다음과 같습니다.

- Windows의 경우:

```
SET MQSERVER=CHANNEL1/TCP/9.20.4.56  
SET MQSERVER=CHANNEL1/NETBIOS/BOX643
```

- UNIX and Linux 시스템의 경우:

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56'  
export MQSERVER=CHANNEL1/LU62/BOX99
```

여기서, BOX99는 LU 6.2 ConnectionName입니다.

IBM WebSphere MQ MQI 클라이언트에서 채널이 **MQCONNX**에 제공된 MQCNO 구조로부터 참조되는 MQCD 구조에서 대체되지 않으면 모든 **MQCONN** 또는 **MQCONNX** 요청은 사용자가 정의한 채널을 사용하려고 합니다.

참고: MQSERVER 환경 변수에 대한 자세한 정보는 134 페이지의 『MQSERVER』의 내용을 참조하십시오.

MQCONNX 호출 시 MQCNO 구조 사용

IBM WebSphere MQ MQI 클라이언트 응용프로그램은 MQCONNX 호출에서 연결 옵션 구조인 MQCNO를 사용하여 클라이언트 연결 채널의 정의를 포함하는 채널 정의 구조 MQCD를 참조할 수 있습니다.

이 방식으로 클라이언트 애플리케이션은 클라이언트 애플리케이션이 다중 서버 큐 관리자에 동시에 연결할 수 있도록 하는 채널의 **ChannelName**, **TransportType** 및 **ConnectionName** 속성을 런타임 시 지정할 수 있습니다.

MQSERVER 환경 변수를 사용하여 채널을 정의하는 경우 **ChannelName**, **TransportType** 및 **ConnectionName** 속성을 런타임 시 지정할 수 없음을 참고하십시오.

클라이언트 애플리케이션은 **MaxMsgLength** 및 **SecurityExit**와 같은 채널 속성을 지정할 수도 있습니다. 그러한 속성을 지정하면 클라이언트 애플리케이션이 속성에 기본값이 아닌 값을 지정할 수 있으며 채널 엑시트 프로그램을 MQI 채널의 클라이언트 측에서 호출할 수 있습니다.

채널이 SSL(Secure Sockets Layer) 또는 TLS(Transport Layer Security)를 사용하는 경우 클라이언트 애플리케이션은 MQCD 구조의 SSL 또는 TLS에 관련된 정보를 제공할 수도 있습니다. SSL 또는 TLS와 관련된 추가 정보는 SSL 또는 TLS 구성 옵션 구조인 MQSCO에서 제공될 수 있으며 이 구조는 MQCONNX 호출 시 MQCNO 구조라고도 합니다.

MQCNO, MQCD 및 MQSCO 구조에 대한 자세한 정보는 [MQCNO](#), [MQCD](#) 및 [MQSCO](#)를 참조하십시오.

참고: MQCONNX에 대한 샘플 프로그램은 **amqscnxc**라고 합니다. **amqssslc**라고 하는 다른 샘플 프로그램은 MQSCO 구조의 사용을 보여줍니다.

서버에서 서버 연결 및 클라이언트 연결 정의 작성

서버에서 두 정의 모두 작성한 다음 클라이언트가 클라이언트 연결 정의를 사용할 수 있도록 설정할 수 있습니다.

먼저 서버 연결 채널을 정의한 다음 클라이언트 연결 채널을 정의하십시오. 모든 플랫폼에서는 WebSphere MQ 스크립트(MQSC) 명령, PCF(programmable command format) 또는 IBM WebSphere MQ Explorer를 사용하여 서버 시스템에서 서버 연결 채널을 정의할 수 있습니다.

서버에서 작성된 클라이언트 연결 채널 정의를 클라이언트 채널 정의 테이블(CCDT)을 사용하여 클라이언트가 사용할 수 있도록 설정합니다.

관련 개념

[105 페이지의 『클라이언트 채널 정의 테이블』](#)

클라이언트 채널 정의 테이블(CCDT)은 클라이언트 애플리케이션이 큐 관리자에 연결하기 위해 사용하는 인증 정보 및 채널 정의를 판별합니다. z/OS 이외의 플랫폼에서는 CCDT가 자동으로 작성됩니다. 그러면 이 CCDT를 클라이언트 애플리케이션이 사용할 수 있도록 설정해야 합니다.

관련 태스크

[107 페이지의 『서버에서 서버 연결 채널 정의』](#)

큐 관리자에 대한 서버 연결 채널 정의를 작성하십시오.

[108 페이지의 『서버에서 클라이언트 연결 채널 정의』](#)

서버 연결 채널을 정의했으면 이제 해당 클라이언트 연결 채널을 정의합니다.

[109 페이지의 『클라이언트 연결 채널 정의 액세스』](#)

클라이언트 채널 정의 테이블(CCDT)을 복사하거나 공유하여 클라이언트 애플리케이션이 사용할 수 있도록 한 다음 그 위치를 지정하고 클라이언트 컴퓨터에서 이름을 지정합니다.

클라이언트 채널 정의 테이블

클라이언트 채널 정의 테이블(CCDT)은 클라이언트 애플리케이션이 큐 관리자에 연결하기 위해 사용하는 인증 정보 및 채널 정의를 판별합니다. z/OS 이외의 플랫폼에서는 CCDT가 자동으로 작성됩니다. 그러면 이 CCDT를 클라이언트 애플리케이션이 사용할 수 있도록 설정해야 합니다.

클라이언트 채널 정의 테이블(CCDT)의 용도는 클라이언트 애플리케이션이 큐 관리자에 연결하기 위해 사용하는 채널 정의를 판별하는 것입니다. 채널 정의는 연결에 적용되는 인증 정보도 지정합니다.

CCDT는 2진 파일입니다. CCDT는 큐 관리자에 의해 생성됩니다. 큐 관리자는 CCDT 파일을 읽지 않습니다.

z/OS 이외의 플랫폼에서 CCDT는 큐 관리자가 작성될 때 작성됩니다. **DEFINE CHANNEL** 명령을 사용하면 클라이언트 연결 채널이 테이블에 추가되고 **ALTER CHANNEL** 명령을 실행할 때 해당 정의가 변경됩니다.

CCDT를 사용하여 클라이언트에 SSL 인증서 폐기를 검사하기 위한 인증 정보를 제공할 수 있습니다. 인증 정보 오브젝트를 포함하는 이름 목록을 정의하고 큐 관리자 속성 **SSLCRLNameList**를 이름 목록의 이름으로 설정하십시오.

클라이언트 애플리케이션에서 CCDT를 사용하는 여러 가지 방법이 있습니다. CCDT를 클라이언트 컴퓨터에 복사할 수 있습니다. CCDT를 둘 이상의 클라이언트가 공유하는 위치에 복사할 수 있습니다. CCDT가 서버에 있는 상태에서 클라이언트가 공유 파일로서 CCDT에 액세스할 수 있게 설정할 수 있습니다.

파일을 복사하는 데 FTP를 사용하는 경우 bin 옵션을 사용하여 2진 모드를 설정하고 기본 ASCII 모드를 사용하지 마십시오. CCDT를 사용할 수 있도록 설정하기 위해 선택하는 메소드에 관계없이 채널을 권한 없이 변경하지 못하도록 그 위치를 보안 설정해야 합니다.

z/OS 이외의 플랫폼

큐 관리자를 작성할 때 AMQCLCHL.TAB라고 하는 기본 CCDT가 작성됩니다.

기본적으로 AMQCLCHL.TAB는 서버의 다음 디렉토리에 위치합니다.

- ▶ **UNIX** ▶ **Linux** UNIX and Linux 시스템의 경우:

```
/prefix/qmgrs/QUEUEMANAGERNAME/@ipcc
```

UNIX and Linux 시스템에서는 *QUEUEMANAGERNAME*이 참조하는 디렉토리 이름은 대소문자를 구분합니다. 이 디렉토리 이름은 큐 관리자 이름에 특수 문자가 있을 경우 큐 관리자 이름과 같지 않을 수 있습니다.

- ▶ **Windows** Windows의 경우:

```
MQ_INSTALLATION_PATH\data\qmgrs\QUEUEMANAGERNAME\@ipcc
```

*MQ_INSTALLATION_PATH*는 IBM WebSphere MQ가 설치된 상위 레벨 디렉토리를 나타냅니다.

그러나 큐 관리자 데이터에 대해 다른 디렉토리 사용을 선택했을 수 있습니다. **crtmqm** 명령을 사용할 때 **-md DataPath** 매개변수를 지정할 수 있습니다. 이 경우, AMQCLCHL.TAB은(는) 지정한 *DataPath*의 @ipcc 디렉토리에 있습니다.

CCDT에 대한 경로는 MQCHLLIB를 설정하여 변경할 수 있습니다. MQCHLLIB를 설정하지 않는 경우 동일한 서버에 여러 큐 관리자가 있으면 이러한 큐 관리자가 동일한 CCDT 위치를 공유함을 알아야 합니다.

CCDT는 큐 관리자가 작성될 때 작성됩니다. CCDT의 각 입력 항목은 특정 큐 관리자에 대한 클라이언트 연결을 나타냅니다. **DEFINE CHANNEL** 명령을 사용하여 클라이언트 연결 채널을 정의할 때 새 입력 항목이 추가되며 이 입력 항목은 **ALTER CHANNEL** 명령을 사용하여 클라이언트 연결 채널을 대체할 때 업데이트됩니다.

클라이언트에서 CCDT의 위치를 지정하는 방법

클라이언트 시스템에서는 다음 두 가지 방식으로 CCDT의 위치를 지정할 수 있습니다.

- 환경 변수 MQCHLLIB를 사용하여 테이블이 위치하는 디렉토리를 지정하고 MQCHLTAB를 사용하여 테이블의 파일 이름을 지정합니다.
- 클라이언트 구성 파일 사용. CHANNELS 스탠자에서 속성 ChannelDefinitionDirectory를 사용하여 테이블이 위치하는 디렉토리를 지정하고 ChannelDefinitionFile을 사용하여 파일 이름을 지정합니다.

위치가 클라이언트 구성 파일에 지정되고 동시에 환경 변수를 사용하여 지정된 경우, 환경 변수가 우선순위를 가집니다. 이 기능을 사용하여 클라이언트 구성 파일에 표준 위치를 지정하고 필요하면 환경 변수를 사용하여 이를 대체할 수 있습니다.

관련 참조

[132 페이지의 『MQCHLLIB』](#)

MQCHLLIB는 클라이언트 채널 정의 테이블(CCDT)이 들어 있는 파일에 대한 디렉토리 경로를 지정합니다. 파일은 서버에 작성되거나 WebSphere MQ MQI 클라이언트 워크스테이션에 복사할 수 있습니다.

관련 정보

폐기된 인증서에 대한 작업

클라이언트 채널 정의 테이블(CCDT) 마이그레이션

일반적으로 클라이언트 채널 정의 테이블의 내부 형식은 IBM WebSphere MQ의 한 릴리스 레벨에서 다음 릴리스 레벨로 변경할 수 있습니다. 결과적으로 IBM WebSphere MQ MQI 클라이언트는 클라이언트와 동일한 릴리스 레벨 또는 이전 릴리스 레벨에 있는 서버 큐 관리자에 의해 준비된 경우에만 클라이언트 채널 정의 테이블을 사용할 수 있습니다.

버전 7.1 IBM WebSphere MQ MQI 클라이언트는 버전 6.0 큐 관리자에 의해 준비된 클라이언트 채널 정의 테이블을 사용할 수 있습니다. 하지만 버전 6.0 클라이언트는 버전 7.1 큐 관리자에 의해 준비된 클라이언트 채널 정의 테이블을 사용할 수 없습니다.

Active Directory의 클라이언트 연결 채널

Active Directory를 지원하는 윈도우 시스템에서 IBM WebSphere MQ는 동적 클라이언트-서버 바인딩을 제공하기 위해 Active Directory에 클라이언트 연결 채널을 게시합니다.

클라이언트 연결 채널 오브젝트는 정의될 때 기본적으로 AMQCLCHL.TAB으로 부르는 클라이언트 연결 정의 파일에 기록됩니다. 클라이언트 연결 채널이 TCP/IP 프로토콜을 사용하는 경우 IBM WebSphere MQ 서버는 Active Directory에서 해당 프로토콜도 공개합니다. IBM WebSphere MQ 클라이언트가 서버에 연결할 방법을 판별할 때 다음 검색 순서를 사용하여 관련 클라이언트 채널 오브젝트 정의를 찾습니다.

1. MQCONNX MQCD 데이터 구조
2. MQSERVER 환경 변수
3. 클라이언트 채널 정의 파일
4. Active Directory

이 순서는 모든 현재 애플리케이션은 어떤 변경의 영향도 받지 않음을 의미합니다. Active Directory의 이들 입력 항목을 클라이언트 채널 정의 파일의 레코드로 생각할 수 있으며, IBM WebSphere MQ 클라이언트는 동일한 방법으로 이들을 처리합니다. Active Directory에서 클라이언트 연결 채널 정의를 공개하기 위한 지원을 구성하고 관리하려면 `setmqscp`에 설명된 대로 `setmqscp` 명령을 사용하십시오.

서버에서 서버 연결 채널 정의

큐 관리자에 대한 서버 연결 채널 정의를 작성하십시오.

프로시저

1. 서버 시스템에서 선택한 이름과 서버 연결 채널 유형으로 채널을 정의하십시오.
예를 들면, 다음과 같습니다.

```
DEFINE CHANNEL(CHAN2) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_2')
```

2. 다음 명령을 사용하여 큐 관리자에 대한 인바운드 연결 액세스를 허용하십시오.

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- 여기서, SET CHLAUTH는 이전 단계에서 정의된 채널의 이름을 사용합니다.
- 여기서, 'IP address' IP 주소는 클라이언트의 IP 주소입니다.
- 여기서, 'userid'는 대상 큐에 대한 액세스 제어를 위해 채널에 제공할 ID입니다. 이 필드는 대소문자를 구분합니다.

다양한 속성을 사용하여 인바운드 연결을 식별하도록 선택할 수 있습니다. 이 예는 IP 주소를 사용합니다. 대체 속성에는 클라이언트 사용자 ID 및 SSL 또는 TLS 주제 식별 이름 등이 있습니다. 자세한 정보는 [채널 인증 레코드](#)를 참조하십시오.

이 채널 정의는 서버에서 실행되는 큐 관리자와 연관됩니다.

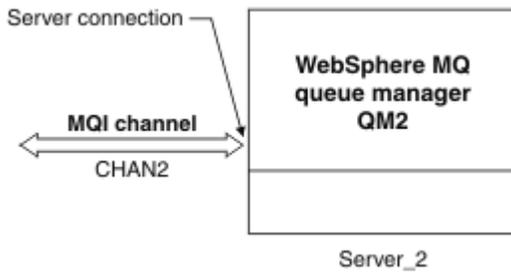


그림 18. 서버 연결 채널 정의

서버에서 클라이언트 연결 채널 정의

서버 연결 채널을 정의했으면 이제 해당 클라이언트 연결 채널을 정의합니다.

시작하기 전에

서버 연결 채널을 정의합니다.

프로시저

1. 서버 연결 채널과 이름은 같지만 채널 유형은 클라이언트 연결인 채널을 정의하십시오. 연결 이름 (CONNNAME)을 명시해야 합니다. TCP/IP의 경우 연결 이름은 서버 시스템의 네트워크 주소 또는 호스트 이름입니다. 클라이언트 환경에서 실행되는 IBM WebSphere MQ 애플리케이션이 연결할 큐 관리자 이름 (QMNAME)을 지정하는 것도 권장됩니다. 큐 관리자 이름을 변경하여 여러 큐 관리자에 연결하기 위한 일련의 채널을 정의할 수 있습니다.

```
DEFINE CHANNEL(CHAN2) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME(9.20.4.26) QMNAME(QM2) DESCR('Client-connection to Server_2')
```

2. 다음 명령을 사용하여 큐 관리자에 대한 인바운드 연결 액세스를 허용하십시오.

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP-address') MCAUSER('userid')
```

- 여기서, SET CHLAUTH는 이전 단계에서 정의된 채널의 이름을 사용합니다.
- 여기서, 'IP address'는 클라이언트의 IP 주소입니다.
- 여기서, 'userid'는 대상 큐에 대한 액세스 제어를 위해 채널에 제공할 ID입니다. 이 필드는 대소문자를 구분합니다.

다양한 속성을 사용하여 인바운드 연결을 식별하도록 선택할 수 있습니다. 이 예는 IP 주소를 사용합니다. 대체 속성에는 클라이언트 사용자 ID 및 SSL 또는 TLS 주제 식별 이름 등이 있습니다. 자세한 정보는 [채널 인증 레코드를 참조하십시오](#).

결과

z/OS 외의 플랫폼에서 이 채널 정의는 큐 관리자와 연관되며 클라이언트 채널 정의 테이블(CCDT)이라고 하는 파일에 저장됩니다. 클라이언트 채널 정의 테이블에는 둘 이상의 클라이언트 연결 채널 정의가 포함될 수 있습니다. 클라이언트 채널 정의 테이블에 대한 자세한 정보 및 클라이언트 연결 채널 정의가 z/OS에서 저장되는 방식에 대한 해당 정보는 105 페이지의 『클라이언트 채널 정의 테이블』의 내용을 참조하십시오.

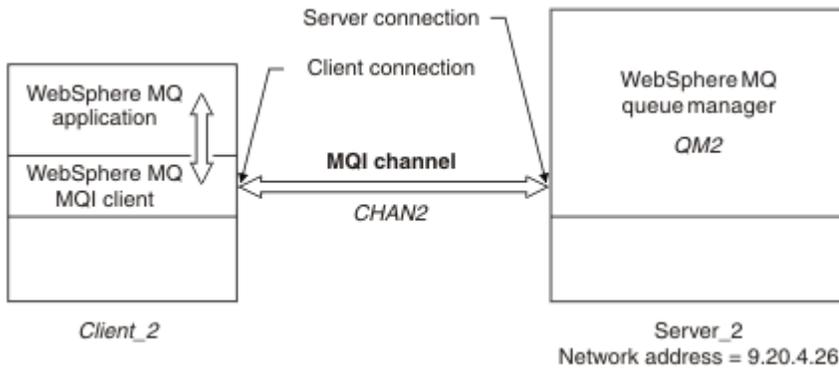


그림 19. 클라이언트 연결 채널 정의

클라이언트 연결 채널 정의 액세스

클라이언트 채널 정의 테이블(CCDT)을 복사하거나 공유하여 클라이언트 애플리케이션이 사용할 수 있도록 한 다음 그 위치를 지정하고 클라이언트 컴퓨터에서 이름을 지정합니다.

시작하기 전에

필요한 클라이언트 연결 채널을 정의했습니다.

z/OS의 경우 CCDT를 작성했습니다. 다른 플랫폼에서 CCDT가 자동으로 작성되고 업데이트됩니다.

이 태스크 정보

클라이언트 애플리케이션이 클라이언트 채널 정의 테이블(CCDT)을 사용하게 하려면 클라이언트 애플리케이션이 CCDT를 사용할 수 있도록 설정하고 그 위치 및 이름을 지정해야 합니다.

프로시저

1. 다음 세 가지 방법 중 하나로 클라이언트 애플리케이션이 CCDT를 사용할 수 있도록 설정하십시오.
 - a) 옵션: CCDT를 클라이언트 컴퓨터에 복사하십시오.
 - b) 옵션: CCDT를 둘 이상의 클라이언트가 공유하는 위치에 복사하십시오.
 - c) 옵션: 서버에 CCDT를 그대로 두고 클라이언트가 공유할 수 있도록 설정하십시오.

CCDT에 대해 선택하는 위치에 관계없이 채널을 권한 없이 변경하지 못하도록 그 위치를 보안 설정해야 합니다.
2. 클라이언트에서 다음 세 가지 방식 중 하나로 CCDT를 포함하는 파일의 위치 및 이름을 지정하십시오.
 - a) 옵션: 클라이언트 구성 파일의 CHANNELS 스탠자를 사용하십시오. 추가 정보는 [122 페이지의 『클라이언트 구성 파일의 CHANNELS 스탠자』](#)의 내용을 참조하십시오.
 - b) 옵션: 환경 변수 MQCHLLIB 및 MQCHLTAB를 사용하십시오.
예를 들어, 다음을 입력하여 환경 변수를 설정할 수 있습니다.
 - HP Integrity NonStop Server 및 UNIX and Linux 시스템의 경우:

```
export MQCHLLIB=MQ_INSTALLATION_PATH/qmgrs/QUEUEMANAGERNAME/@ipcc
export MQCHLTAB=AMQCLCHL.TAB
```

여기서 `MQ_INSTALLATION_PATH`는 WebSphere MQ가 설치되어 있는 상위 레벨 디렉토리를 표시합니다.

- c) 옵션: Windows의 경우에만 `setmqscp` 제어 명령을 사용하여 Active Directory에 클라이언트 연결 채널 정의를 발행하십시오.

MQSERVER 환경 변수가 설정된 경우 WebSphere MQ 클라이언트는 클라이언트 채널 정의 테이블의 어떤 정의보다도 MQSERVER에 의해 지정된 클라이언트 연결 채널 정의를 우선 사용합니다.

MQI 채널에 대한 채널 엑시트 프로그램

UNIX, Linux 및 시스템 WebSphere MQ MQI 클라이언트 환경에는 세 가지 유형의 채널 엑시트가 사용 가능합니다.

즉, 다음과 같습니다.

- 송신 엑시트
- 수신 엑시트
- 보안 엑시트

이러한 엑시트는 채널의 클라이언트 및 서버 측에서 사용 가능합니다. MQSERVER 환경 변수를 사용하는 경우에는 애플리케이션이 엑시트를 사용할 수 없습니다. 채널 엑시트는 [메시징 채널의 채널 엑시트 프로그램](#)에 설명되어 있습니다.

송신 및 수신 엑시트는 함께 작동합니다. 다음과 같이 엑시트를 사용하는 여러 가능한 방법이 있습니다.

- 메시지 분할 및 리어셈블링
- 메시지의 데이터 압축 및 압축 풀기(이 기능은 WebSphere MQ의 일부로 제공되지만 다른 압축 기술을 사용할 수도 있음)
- 사용자 데이터 암호화 및 복호화(이 기능은 WebSphere MQ의 일부로 제공되지만 다른 암호화 기술을 사용할 수도 있음)
- 송신 및 수신된 각 메시지 저널링

보안 엑시트를 사용하여 WebSphere MQ 클라이언트 및 서버가 올바르게 식별되는지 확인하고 액세스를 제어할 수 있습니다.

채널 인스턴스의 서버 연결 측 송신 또는 수신 엑시트가 연관된 연결에 대해 MQI 호출을 수행해야 하는 경우 해당 엑시트는 MQCXP Hconn 필드에 제공되는 연결 핸들을 사용합니다. 클라이언트 연결 송신 및 수신 엑시트가 MQI 호출을 수행할 수 없음을 알고 있어야 합니다.

관련 개념

[111 페이지의 『클라이언트 연결의 보안 엑시트』](#)

보안 엑시트 프로그램을 사용하여 채널의 다른 끝에 있는 파트너가 진짜인지 확인할 수 있습니다. 보안 엑시트가 클라이언트 연결에 적용될 때 특수 고려사항이 적용됩니다.

[사용자 엑시트, API 엑시트 및 WebSphere MQ 설치 가능 서비스](#)

관련 태스크

[큐 관리자 기능 확장](#)

관련 참조

[110 페이지의 『엑시트 경로』](#)

채널 엑시트의 위치에 대한 기본 경로는 클라이언트 구성 파일에서 정의됩니다. 채널 엑시트는 채널이 초기화될 때 로드됩니다.

[112 페이지의 『송신 또는 수신 엑시트 프로그램에서 API 호출 식별』](#)

클라이언트에 MQI 채널을 사용하는 경우 에이전트 버퍼의 바이트 10은 송신 또는 수신 엑시트가 호출될 때 사용 중인 API 호출을 식별합니다. 이는 사용자 데이터를 포함하고 암호화 또는 디지털 서명과 같은 처리를 요구할 수 있는 채널 플로우를 식별하는 데 유용합니다.

엑시트 경로

채널 엑시트의 위치에 대한 기본 경로는 클라이언트 구성 파일에서 정의됩니다. 채널 엑시트는 채널이 초기화될 때 로드됩니다.

UNIX, Linux 및 Windows 시스템에서는 WebSphere MQ MQI 클라이언트의 설치 중 클라이언트 구성 파일이 시스템에 추가됩니다. 클라이언트에서 채널 엑시트의 위치에 대한 기본 경로는 다음과 같은 스탠자를 사용하여 이 파일에서 정의됩니다.

```
ClientExitPath:  
  ExitsDefaultPath=string  
  ExitsDefaultPath64=string
```

여기서, *string*은 플랫폼에 적합한 형식의 파일 위치입니다.

채널이 초기화되면 MQCONN 또는 MQCONNX 호출 후에 클라이언트 구성 파일이 검색됩니다. ClientExitPath 스탠자를 읽고 채널 정의에서 지정되는 모든 채널 엑시트가 로드됩니다.

클라이언트 연결의 보안 엑시트

보안 엑시트 프로그램을 사용하여 채널의 다른 끝에 있는 파트너가 진짜인지 확인할 수 있습니다. 보안 엑시트가 클라이언트 연결에 적용될 때 특수 고려사항이 적용됩니다.

112 페이지의 그림 20은 WebSphere MQ OAM(Object Authority Manager)을 사용하여 사용자를 인증하는, 클라이언트 연결의 보안 엑시트 사용을 보여줍니다. SecurityParmsPtr 또는 SecurityParmsOffset이 클라이언트에서 MQCNO 구조로 설정되고 채널 양 끝에는 보안 엑시트가 있습니다. 정상 보안 메시지 교환이 종료되고 채널이 실행될 준비가 되고 나면 MQCXP SecurityParms 필드에서 액세스된 MQCSP 구조가 클라이언트의 보안 엑시트에 전달됩니다. 엑시트 유형은 MQXR_SEC_PARMS로 설정됩니다. 보안 엑시트는 사용자 ID 및 비밀번호를 그대로 두도록 선택하거나 이 둘 중 하나 또는 둘 다를 대체할 수 있습니다. 그러면 엑시트에서 리턴된 데이터가 채널의 서버 연결 측에 송신됩니다. MQCSP 구조가 채널의 서버 연결 측에서 다시 빌드되어 MQCXP SecurityParms 필드에서 액세스되는 서버 연결 보안 엑시트에 전달됩니다. 보안 엑시트는 이 데이터를 수신하고 처리합니다. 이 처리는 일반적으로 클라이언트 엑시트에서 큐 관리자 연결 권한 부여에 사용되는 사용자 ID 및 비밀번호 필드에 대해 수행된 변경을 되돌리는 것입니다. 결과로 생성되는 MQCSP 구조는 큐 관리자 시스템의 MQCNO 구조에서 SecurityParmsPtr을 사용하여 참조됩니다.

SecurityParmsPtr 또는 SecurityParmsOffset이 MQCNO 구조에서 설정되고 채널의 한쪽 끝에만 보안 엑시트가 있는 경우 이 보안 엑시트는 MQCSP 구조를 수신하고 처리합니다. 암호화와 같은 조치는 보완하는 조치를 수행할 엑시트가 없는 단일 사용자 엑시트에는 적합하지 않습니다.

SecurityParmsPtr 및 SecurityParmsOffset이 MQCNO 구조에서 설정되지 않고 채널의 한 측 또는 양 측에 보안 엑시트가 있는 경우 이 보안 엑시트가 호출됩니다. 이 중 한 보안 엑시트는 SecurityParmsPtr을 통해 처리되는 자체 MQCSP 구조를 리턴할 수 있으며 이 보안 엑시트가 종료될 때까지(MQXR_TERM의 ExitReason) 보안 엑시트가 다시 호출되지 않습니다. 이 엑시트 기록기는 해당 단계에서 MQCSP에 사용되는 메모리를 비울 수 있습니다.

서버 연결 채널 인스턴스가 둘 이상의 대화를 공유하는 경우에는 보안 엑시트에 대한 호출 패턴이 두 번째 및 후속 대화에서 제한됩니다.

첫 번째 대화에서 이 패턴은 채널 인스턴스가 대화를 공유하지 않는 경우와 같습니다. 두 번째 및 후속 대화에서 보안 엑시트는 MQXR_INIT, MQXR_INIT_SEC 또는 MQXR_SEC_MSG로 호출되지 않습니다. MQXR_SEC_PARMS로 호출됩니다.

공유 대화가 있는 채널 인스턴스에서 MQXR_TERM은 실행 중인 마지막 대화에 대해서만 호출됩니다.

각 대화에는 MQCD를 대체하기 위한 엑시트의 MQXR_SEC_PARMS 호출 기회가 있으며 채널의 서버 연결 측에서는 이 기능이 큐 관리자에 연결되기 전에 값(예: MCAUserIdentifier 또는 LongMCAUserIdPtr 값)을 변경하는데 유용할 수 있습니다.

Server-connection exit	Client-connection exit
	Invoked with MQXR_INIT Responds with MQXCC_OK
Invoked with MQXR_INIT Responds with MQXCC_OK	
	Invoked with MQXR_INIT_SEC Responds with MQXCC_OK
Invoked with MQXR_INIT_SEC Responds with MQXCC_OK	
	Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK
Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK	
Data transfer begins	
Invoked with MQXR_TERM Responds with MQXCC_OK	Invoked with MQXR_TERM Responds with MQXCC_OK

그림 20. 보안 매개변수를 사용한 클라이언트 연결 동의에 따른 클라이언트 연결 시작 교환

참고: WebSphere MQ v7.1 릴리스 이전에 구성된 보안 엑시트는 업데이트가 필요할 수 있습니다. 자세한 정보는 [채널 보안 엑시트 프로그램을 참조하십시오.](#)

송신 또는 수신 엑시트 프로그램에서 API 호출 식별

클라이언트에 MQI 채널을 사용하는 경우 에이전트 버퍼의 바이트 10은 송신 또는 수신 엑시트가 호출될 때 사용 중인 API 호출을 식별합니다. 이는 사용자 데이터를 포함하고 암호화 또는 디지털 서명과 같은 처리를 요구할 수 있는 채널 플로우를 식별하는 데 유용합니다.

다음 표는 API 호출을 처리 중일 때 채널 플로우의 바이트 10에 표시되는 데이터를 보여줍니다.

참고: 이러한 데이터가 이 바이트의 모든 값은 아닙니다. 다른 예약 값도 있습니다.

표 20. API 호출 식별		
API 호출	요청에 대한 바이트 10의 값	응답에 대한 바이트 10의 값
MQCONN 113 페이지의 『1』 , 113 페이지의 『2』	X'81'	X'91'
MQDISC 113 페이지의 『1』	X'82'	X'92'
MQOPEN 113 페이지의 『3』	X'83'	X'93'
MQCLOSE	X'84'	X'94'
MQGET 113 페이지의 『4』	X'85'	X'95'

표 20. API 호출 식별 (계속)

API 호출	요청에 대한 바이트 10의 값	응답에 대한 바이트 10의 값
MQPUT 113 페이지의 『4』	X'86'	X'96'
MQPUT1 요청 113 페이지의 『4』	X'87'	X'97'
MQSET 요청	X'88'	X'98'
MQINQ 요청	X'89'	X'99'
MQCMIT 요청	X'8A'	X'9A'
MQBACK 요청	X'8B'	X'9B'
MQSTAT 요청	X'8D'	X'9D'
MQSUB 요청	X'8E'	X'9E'
MQSUBRQ 요청	X'8F'	X'9F'
xa_start 요청	X'A1'	X'B1'
xa_end 요청	X'A2'	X'B2'
xa_open 요청	X'A3'	X'B3'
xa_close 요청	X'A4'	X'B4'
xa_prepare 요청	X'A5'	X'B5'
xa_commit 요청	X'A6'	X'B6'
xa_rollback 요청	X'A7'	X'B7'
xa_forget 요청	X'A8'	X'B8'
xa_recover 요청	X'A9'	X'B9'
xa_complete 요청	X'AA'	X'BA'

참고사항:

- 클라이언트와 서버 간 연결이 MQCONN을 사용하는 클라이언트 애플리케이션에 의해 시작됩니다. 따라서 이 명령의 경우에는 특히 여러 개의 다른 네트워크 플로우가 있습니다. 이는 네트워크 연결을 종료하는 MQDISC에도 동일하게 적용됩니다.
- MQCONNx는 클라이언트 서버 연결을 목적으로 MQCONN과 같은 방식으로 처리됩니다.
- 대형 분배 목록이 열리는 경우 모든 필수 데이터를 SVRCONN MCA에 전달하기 위해 MQOPEN 호출당 두 개 이상의 네트워크 플로우가 있을 수도 있습니다.
- 대형 메시지가 전송 세그먼트 크기를 초과할 수 있습니다. 이러한 상태가 발생하면 단일 호출 API 호출에서 다수의 네트워크 플로우가 발생할 수 있습니다.

큐 공유 그룹에 클라이언트 연결

큐 공유 그룹의 멤버인 서버에서 클라이언트와 큐 관리자 간 MQI 채널을 작성하여 클라이언트를 큐 공유 그룹에 연결할 수 있습니다.

큐 공유 그룹은 같은 공유 큐 세트에 액세스할 수 있는 큐 관리자 세트로 구성됩니다.

공유 큐에 있는 클라이언트는 큐 공유 그룹의 모든 멤버에 연결할 수 있습니다. 큐 공유 그룹 연결의 이점은 프론트 엔드 및 백엔드 가용성을 높이고 용량을 증가시킨다는 것입니다. 특정 큐 관리자 또는 일반 인터페이스에 연결할 수 있습니다.

큐 공유 그룹에 있는 큐 관리자에 직접 연결하면 메시지를 공유 대상 큐에 넣을 수 있는 이점으로 인해 백엔드 가용성을 증가시킬 수 있습니다.

큐 공유 그룹의 일반 인터페이스에 연결하면 그룹에 있는 큐 관리자 중 하나로 세션을 엽니다. 이는 클라이언트 큐 관리자가 그룹 내 모든 큐 관리자와 연결될 수 있기 때문에 프론트 엔드 가용성을 증가시킵니다. 큐 공유 그룹 내 특정 큐 관리자에게 연결하지 않으려는 경우 일반 인터페이스를 사용하여 그룹에 연결합니다.

일반 인터페이스는 WLM/DNS 그룹 이름 또는 VTAM® 일반 자원 이름 또는 큐 공유 그룹에 대한 다른 공용 인터페이스가 될 수 있습니다.

큐 공유 그룹의 일반 인터페이스에 연결하려면 그룹 내 임의의 큐 관리자가 액세스할 수 있는 채널 정의를 작성해야 합니다. 이를 수행하려면 그룹 내 각 큐 관리자에 대해 동일한 정의를 가져야 합니다.

다음과 같이 SVRCONN 채널을 정의하십시오.

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') QSGDISP(GROUP)
```

서버의 채널 정의는 공유 DB2® 저장소에 저장됩니다. 큐 공유 그룹 내 각 큐 관리자는 정의의 로컬 사본을 작성하여 MQCONN이나 MQCONNX 호출을 발행할 때 항상 올바른 서버 연결 채널에 연결하도록 합니다.

다음과 같이 CLNTCONN 채널을 정의하십시오.

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(WLM/DNS_groupname) QMNAME(QSG1) +
DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
```

큐 공유 그룹의 일반 인터페이스는 클라이언트 연결 채널의 CONNAME 필드에 저장되기 때문에 이제 그룹 내 어느 큐 관리자에게도 연결할 수 있으며 해당 그룹이 소유하는 공유 큐에 넣을 수 있습니다.

관련 개념

114 페이지의 『채널 정의 작성』

큐 공유 그룹의 일반 인터페이스에 연결하려면 그룹 내 임의의 큐 관리자가 액세스할 수 있는 채널 정의를 작성해야 합니다. 이를 수행하려면 그룹 내 각 큐 관리자에 대해 동일한 정의를 가져야 합니다.

채널 정의 작성

큐 공유 그룹의 일반 인터페이스에 연결하려면 그룹 내 임의의 큐 관리자가 액세스할 수 있는 채널 정의를 작성해야 합니다. 이를 수행하려면 그룹 내 각 큐 관리자에 대해 동일한 정의를 가져야 합니다.

다음과 같이 SVRCONN 채널을 정의하십시오.

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') QSGDISP(GROUP)
```

서버의 채널 정의는 공유 DB2 저장소에 저장됩니다. 큐 공유 그룹 내 각 큐 관리자는 정의의 로컬 사본을 작성하여 MQCONN이나 MQCONNX 호출을 발행할 때 항상 올바른 서버 연결 채널에 연결하도록 합니다.

다음과 같이 CLNTCONN 채널을 정의하십시오.

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(WLM/DNS_groupname) QMNAME(QSG1) +
DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
```

큐 공유 그룹의 일반 인터페이스는 클라이언트 연결 채널의 CONNAME 필드에 저장되기 때문에 이제 그룹 내 어느 큐 관리자에게도 연결할 수 있으며 해당 그룹이 소유하는 공유 큐에 넣을 수 있습니다.

구성 파일을 사용하여 클라이언트 구성

텍스트 파일의 속성을 사용하여 클라이언트를 구성하십시오. 이러한 속성은 환경 변수 또는 다른 플랫폼별 방식으로 대체될 수 있습니다.

UNIX and Linux 플랫폼에서 사용되는 큐 관리자 구성 파일 `qm.ini`와 유사한 텍스트 파일을 사용하여 IBM WebSphere MQ MQI client 를 구성합니다. 파일에는 다수의 스탠자가 포함되어 있으며, 각 스탠자에는 `attribute-name=value` 형식의 행 수가 포함되어 있습니다.

이 문서에서 이 파일은 *WebSphere MQ MQI* 클라이언트 구성 파일로 참조되며 이 파일의 이름은 일반적으로 `mqclient.ini`이지만 다른 이름을 부여하도록 선택할 수 있습니다. 이 파일의 구성 정보는 모든 플랫폼 및 MQI, IBM WebSphere MQ classes for Java, IBM WebSphere MQ classes for JMS, IBM WebSphere MQ classes for .NET 및 XMS를 사용하는 클라이언트에 적용됩니다.

속IBM WebSphere MQ MQI client 구성 파일의 속성은 대부분의 IBM WebSphere MQ 클라이언트에 적용할 수 있지만, 관리 .NET 및 XMS .NET 클라이언트나 IBM WebSphere MQ classes for Java 또는 IBM WebSphere MQ classes for JMS를 사용하는 클라이언트에서 읽을 수 없는 몇 가지 속성이 있습니다. 추가 정보는 [117 페이지](#)의 『각 속성을 읽을 수 있는 IBM WebSphere MQ 클라이언트』의 내용을 참조하십시오.

구성 기능은 한 큐 관리자에 대한 개별 연결에 특정한다기 보다는 클라이언트 애플리케이션이 임의의 큐 관리자에 대해 작성하는 모든 연결에 적용됩니다. 개별 큐 관리자 연결에 관련된 속성은 MQCD 구조를 사용하거나 CCDT(Client Channel Definition Table)를 사용하는 등의 프로그래밍 방식으로 구성할 수 있습니다.

IBM WebSphere MQ 7.0 이전 버전의 릴리스에서 지원되는 환경 변수는 계속해서 지원되고 이러한 환경 변수가 클라이언트 구성 파일의 동등한 값과 일치하는 위치에서 환경 변수는 클라이언트 구성 파일 값을 대체합니다.

IBM WebSphere MQ classes for JMS를 사용하는 클라이언트 애플리케이션의 경우 다음과 같은 방식으로 클라이언트 구성 파일을 대체할 수도 있습니다.

- JMS 구성 파일에 특성 설정
- JMS 구성 파일도 대체하는 Java 시스템 특성 설정

.NET 클라이언트의 경우 클라이언트 구성 파일 및 .NET 애플리케이션 구성 파일을 사용하는 동등한 환경 변수를 대체할 수도 있습니다.

클라이언트 구성 파일을 사용하여 다중 채널 연결을 설정할 수 없습니다.

Example client configuration file

```
## Module Name: mqclient.ini                *#
## Type       : WebSphere MQ MQI client configuration file      *#
## Function    : Define the configuration of a client           *#
##                                                    *#
##*****#
## Notes      :                                               *#
## 1) This file defines the configuration of a client          *#
##                                                    *#
##*****#

ClientExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

TCP:
  Library1=DLLName1
  KeepAlive = Yes
  ClntSndBuffSize=32768
  ClntRcvBuffSize=32768
  Connect_Timeout=0

MessageBuffer:
  MaximumSize=-1
  Updatepercentage=-1
  PurgeTime=0

LU62:
  TPName
  Library1=DLLName1
  Library2=DLLName2

PreConnect:
  Module=amqldapi
  Function=myFunc
  Data=ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
  Sequence=1
```

```
CHANNELS:  
DefRecon=YES  
ServerConnectionParms=SALES.SVRCONN/TCP/hostname.x.com(1414)
```

관련 참조

[116 페이지의 『클라이언트 구성 파일의 위치』](#)

IBM WebSphere MQ MQI 클라이언트 구성 파일은 여러 위치에 있을 수 있습니다.

[122 페이지의 『클라이언트 구성 파일의 CHANNELS 스탠자』](#)

CHANNELS 스탠자를 사용하여 클라이언트 채널에 대한 정보를 지정할 수 있습니다.

[124 페이지의 『클라이언트 구성 파일의 ClientExitPath 스탠자』](#)

ClientExitPath 스탠자를 사용하여 클라이언트에서 채널 엑시트의 기본 위치를 지정할 수 있습니다.

[125 페이지의 『클라이언트 구성 파일의 LU62, NETBIOS 및 SPX 스탠자』](#)

Windows 시스템에서만 이 스탠자를 사용하여 지정된 네트워크 프로토콜에 대한 구성 매개변수를 지정하십시오.

[125 페이지의 『클라이언트 구성 파일의 MessageBuffer 스탠자』](#)

MessageBuffer 스탠자를 사용하여 메시지 버퍼에 대한 정보를 지정할 수 있습니다.

[127 페이지의 『클라이언트 구성 파일의 SSL 스탠자』](#)

SSL 스탠자를 사용하여 SSL 또는 TLS의 사용에 대한 정보를 지정할 수 있습니다.

[129 페이지의 『클라이언트 구성 파일의 TCP 스탠자』](#)

TCP 스탠자를 사용하여 TCP 네트워크 프로토콜 구성 매개변수를 지정할 수 있습니다.

[130 페이지의 『WebSphere MQ 환경 변수 사용』](#)

이 절에서는 WebSphere MQ MQI 클라이언트 애플리케이션과 함께 사용할 수 있는 환경 변수에 대해 설명합니다.

[392 페이지의 『큐 관리자 구성 정보 변경』](#)

여기에 설명된 속성은 개별 큐 관리자의 구성을 수정합니다. WebSphere MQ의 모든 속성을 대체합니다.

클라이언트 구성 파일의 위치

IBM WebSphere MQ MQI 클라이언트 구성 파일은 여러 위치에 있을 수 있습니다.

클라이언트 애플리케이션은 다음 검색 경로를 사용하여 IBM WebSphere MQ MQI 클라이언트 구성 파일을 찾습니다.

1. 환경 변수 MQCLNTCF에 의해 지정되는 위치.

이 환경 변수의 형식은 전체 URL입니다. 이는 파일 이름이 반드시 `mqclient.ini`일 필요는 없음을 의미하고 네트워크 접속 파일 시스템에 파일 배치를 용이하게 합니다.

다음에 유의하십시오.

- C, .NET 및 XMS 클라이언트는 `file:` 프로토콜만을 지원합니다. URL 문자열이 `protocol:`로 시작되지 않은 경우 `file:` 프로토콜이 사용됩니다.
- 읽기 환경 변수를 지원하지 않는 Java 1.4.2 JREs를 허용하려면 MQCLNTCF 환경 변수를 MQCLNTCF Java 시스템 특성으로 대체할 수 있습니다.

2. 애플리케이션의 현재 작업 디렉토리에 있는 `mqclient.ini`라는 파일.

3. Windows, UNIX and Linux 시스템의 IBM WebSphere MQ 데이터 디렉토리에 있는 `mqclient.ini` 파일입니다.

다음에 유의하십시오.

- IBM WebSphere MQ 데이터 디렉토리는 특정 플랫폼(예: IBM i 및 z/OS)이나 클라이언트가 다른 제품과 함께 제공된 경우에는 존재하지 않습니다.
- UNIX and Linux 시스템에서 디렉토리는 `/var/mqm`입니다.
- Windows 플랫폼에서는 설치 중 `MQ_FILE_PATH` 환경 변수를 구성해서 데이터 디렉토리를 가리킵니다. 일반적으로 `C:\Program Files\IBM\WebSphere MQ`입니다.

- 읽기 환경 변수를 지원하지 않는 Java 1.4.2 JRE를 허용하려면 MQ_FILE_PATH 환경 변수를 MQ_FILE_PATH Java 시스템 특성으로 수동으로 대체할 수 있습니다.
4. 플랫폼에 해당하고 사용자가 액세스할 수 있는 표준 디렉토리의 mqclient.ini라는 파일:
- 모든 Java 클라이언트의 경우 이는 user.home Java 시스템 특성의 값입니다.
 - UNIX and Linux 플랫폼에서 C 클라이언트의 경우 이는 HOME 환경 변수의 값입니다.
 - Windows의 C 클라이언트의 경우 HOMEDRIVE 및 HOMEPATH 환경 변수의 연결된 값입니다.

참고: HP Integrity NonStop Server에 대한 IBM WebSphere MQ 클라이언트의 경우, mqclient.ini 파일은 OSS 파일 시스템에 있어야 합니다. 감시 애플리케이션은 mqclient.ini 파일을 IBM WebSphere MQ 데이터 디렉토리에 배치하거나 MQCLNTCF 환경 변수를 OSS 파일 시스템의 한 위치로 설정해야 합니다.

각 속성을 읽을 수 있는 IBM WebSphere MQ 클라이언트

IBM WebSphere MQ MQI client 구성 파일에 있는 대부분의 속성은 C 클라이언트 및 비관리 .NET 클라이언트에서 사용할 수 있습니다. 그러나 관리 .NET 및 XMS .NET 클라이언트나 IBM WebSphere MQ classes for Java 또는 IBM WebSphere MQ classes for JMS를 사용하는 클라이언트에서 읽을 수 없는 몇 가지 속성이 있습니다.

표 21. 클라이언트의 각 유형에 적용할 수 있는 속성						
mqclient.ini 스탠자 이름 및 속성	설명	C 및 비관리 .NET	Java	JMS	관리형.NET	관리형 XMS .NET
CHANNELS 스탠자						
<u>CCSID</u>	사용할 코드화 문자 세트 번호입니다.	예	아니오	아니오	예	예
<u>ChannelDefinitionDirectory</u>	클라이언트 채널 정의 테이블이 들어 있는 파일에 대한 디렉토리 경로입니다.	예	아니오	아니오	예	예
<u>ChannelDefinitionFile</u>	클라이언트 채널 정의 테이블이 들어 있는 파일의 이름입니다.	예	아니오	아니오	예	예
<u>ReconDelay</u>	자동으로 다시 연결할 수 있는 클라이언트 프로그램에 대해 다시 연결 지연을 구성하는 관리 옵션을 제공합니다.	예	아니오	예	예	예

표 21. 클라이언트의 각 유형에 적용할 수 있는 속성 (계속)

mqclient.ini 스탠자 이름 및 속성	설명	C 및 비관리 .NET	Java	JMS	관리형.NET	관리형 XMS .NET
DefRecon	클라이언트 프로그램이 자동으로 다시 연결되도록 설정하거나, 자동으로 다시 연결되도록 작성된 클라이언트 프로그램을 자동으로 다시 연결할 수 없도록 설정하는 관리 옵션을 제공합니다.	예	아니오	예	예	예
MQReconnectTimeout	클라이언트에 다시 연결되기까지의 제한시간(초)입니다.	예	아니오	아니오	예	아니오
ServerConnectionParms	IBM WebSphere MQ 서버의 위치와 사용할 통신 방법입니다.	예	아니오	아니오	예	예
Put1DefaultAlwaysSync	MQPMO_RESPONSE_AS_QDEF 옵션을 사용하여 MQPUT1 함수 호출의 작업을 제어합니다.	예	예	예	예	예
ClientExitPath 스탠자						
ExitsDefaultPath	클라이언트에 대한 32비트 채널 엑시트의 위치를 지정합니다.	예	아니오	아니오	예	예
ExitsDefaultPath64	클라이언트에 대해 64비트 채널 엑시트의 위치를 지정합니다.	예	아니오	아니오	예	예
JavaExitsClassPath	Java 엑시트가 실행될 때 클래스 경로에 추가될 값입니다.	아니오	예	예	아니오	아니오

표 21. 클라이언트의 각 유형에 적용할 수 있는 속성 (계속)

mqclient.ini 스탠자 이름 및 속성	설명	C 및 비관리 .NET	Java	JMS	관리형.NET	관리형 XMS .NET
MessageBuffer 스탠자						
<u>MaximumSize</u>	미리 읽기 버퍼의 크기로, 값은 1 - 999 999KB 범위입니다.	예	예	예	예	예
<u>PurgeTime</u>	미리 읽기 버퍼에 남겨진 메시지가 영구 제거되기 전까지의 간격(초)입니다.	예	예	예	예	예
<u>UpdatePercentage</u>	클라이언트 애플리케이션이 서버에 대한 새 요청을 작성하는 시간을 판별하기 위한 임계값 계산에서 사용되는 1 - 100 범위의 업데이트 백분율 값입니다.	예	예	예	예	예
SSL 스탠자						
<u>CDPCheckExtensions</u>	이 큐 관리자의 SSL 또는 TLS 채널이 CrlDistributionPoint 인증서 확장에 이름 지정된 CDP 서버를 검사하려고 시도할지 여부를 지정합니다.	예	아니오	아니오	아니오	아니오
<u>CertificateLabel</u>	채널 정의의 인증서 레이블입니다.	예	아니오	아니오	아니오	아니오
<u>CertificateValidationPolicy</u>	사용되는 인증서 유효성 검증의 유형을 판별합니다.	예	아니오	아니오	아니오	아니오

표 21. 클라이언트의 각 유형에 적용할 수 있는 속성 (계속)

mqclient.ini 스탠자 이름 및 속성	설명	C 및 비관리 .NET	Java	JMS	관리형.NET	관리형 XMS .NET
ClientRevocationChecks	클라이언트 연결 호출에 SSL/TLS 채널을 사용하는 경우 인증서 폐기 검사가 구성되는 방식을 판별합니다.	예	아니오	아니오	아니오	아니오
EncryptionPolicySuiteB	채널이 Suite-B 준수 암호화를 사용하는지 및 사용할 강도 레벨을 판별합니다.	예	아니오	아니오	아니오	아니오
OCSPAuthentication	OCSP가 사용 가능하고 OCSP 폐기 검사가 인증서 폐기 상태를 판별할 수 없는 경우에 IBM WebSphere MQ의 작동을 정의합니다.	예	아니오	아니오	아니오	아니오
OCSPCheckExtensions	IBM WebSphere MQ가 AuthorityInfo Access 인증서 확장자에 대해 작동하는지 여부를 제어합니다.	예	아니오	아니오	아니오	아니오
SSLCryptoHardware	PKCS #11 암호화 하드웨어가 시스템에 표시되도록 구성하기 위해 필요한 매개변수 문자열을 설정합니다.	예	아니오	아니오	아니오	아니오

표 21. 클라이언트의 각 유형에 적용할 수 있는 속성 (계속)

mqclient.ini 스탠자 이름 및 속성	설명	C 및 비관리 .NET	Java	JMS	관리형.NET	관리형 XMS .NET
SSLFipsRequired	IBM WebSphere MQ에서 암호화가 수행되는 경우 FIPS 인증 알고리즘만 사용할 것인지를 여부를 지정합니다.	예	아니오	아니오	아니오	아니오
SSLHTTPProxyName	문자열은 OCSP 검사를 위해 GSKit에서 사용될 HTTP 프록시 서버의 호스트 이름 또는 네트워크 주소입니다.	예	아니오	아니오	아니오	아니오
SSLKeyRepository	사용자의 디지털 인증서를 보유하는 키 저장소의 위치이며 스템 형식입니다.	예	아니오	아니오	아니오	아니오
SSLKeyResetCount	비밀 키를 재협상하기 전에 SSL 또는 TLS 채널에서 송신 및 수신된 암호화되지 않은 바이트의 수입니다.	예	아니오	아니오	아니오	아니오
TCP 스탠자						
ClntRcvBufferSize	클라이언트 연결 서버 연결 채널의 클라이언트 측에서 사용되는 TCP/IP 수신 버퍼의 크기 (바이트)입니다.	예	예	예	예	예

표 21. 클라이언트의 각 유형에 적용할 수 있는 속성 (계속)

mqclient.ini 스탠자 이름 및 속성	설명	C 및 비관리 .NET	Java	JMS	관리형.NET	관리형 XMS .NET
<u>ClntSndBufferSize</u>	클라이언트 연결 서버 연결 채널의 클라이언트 측에서 사용되는 TCP/IP 송신 버퍼의 크기 (바이트)입니다.	예	예	예	예	예
<u>Connect Timeout</u>	소켓 연결 시도가 제한시간 초과하기 전의 시간(초)입니다.	예	예	예	아니오	아니오
<u>IPAddressVersion</u>	채널 연결에 사용할 IP 프로토콜을 지정합니다.	예	아니오	아니오	예	예
<u>KeepAlive</u>	KeepAlive 기능을 설정하거나 해제합니다.	예	예	예	예	예
 <u>Windows Library1</u>	(Windows만 해당) TCP/IP 소켓 DLL의 이름입니다.	예	아니오	아니오	아니오	아니오

HP Integrity NonStop Server의 경우 TMF 및 TmfGateway 스탠자를 사용하여 TMF/Gateway와 통신할 수 있습니다.

클라이언트 구성 파일의 CHANNELS 스탠자

CHANNELS 스탠자를 사용하여 클라이언트 채널에 대한 정보를 지정할 수 있습니다.

다음 속성이 CHANNELS 스탠자에 포함될 수 있습니다.

CCSID=number

사용할 코드화 문자 세트 번호입니다.

CCSID 번호는 MQCCSID 환경 매개변수와 같습니다.

ChannelDefinitionDirectory=경로

클라이언트 채널 정의 테이블이 들어 있는 파일에 대한 디렉토리 경로입니다.

Windows 시스템에서 기본값은 IBM WebSphere MQ 설치 디렉토리(보통 C:\Program Files\IBM\WebSphere MQ)입니다. UNIX and Linux 시스템에서 기본값은 /var/mqm입니다.

ChannelDefinitionDirectory 경로는 MQCHLLIB 환경 매개변수와 같습니다.

ChannelDefinitionFile=filename|_AMQCLCHL.TAB

클라이언트 채널 정의 테이블이 들어 있는 파일의 이름입니다.

클라이언트 채널 정의 테이블은 MQCHLTAB 환경 매개변수와 같습니다.

ReconDelay=(지연 [, rand])(지연 [, rand]) ...

ReconDelay 속성은 클라이언트 프로그램에 대해 자동으로 다시 연결할 수 있는 다시 연결 지연을 구성하는 관리 옵션을 제공합니다. 다음은 구성 예입니다.

```
ReconDelay=(1000,200)(2000,200)(4000,1000)
```

표시된 예는 1초에 최대 200밀리초의 임의 간격을 더한 초기 지연을 정의합니다. 다음 지연은 2초에 최대 200밀리초의 임의 간격을 더합니다. 모든 후속 지연은 4초에 최대 1000밀리초의 임의 간격을 더합니다.

DefRecon=NO|YES|QMGR|DISABLED

DefRecon 속성은 클라이언트 프로그램이 자동으로 다시 연결할 수 있도록 하거나, 자동으로 다시 연결하도록 작성된 클라이언트 프로그램의 자동으로 다시 연결을 사용할 수 없도록 하는 관리 옵션을 제공합니다. 프로그램이 다시 연결과 호환되지 않는 옵션(예: MQPMO_LOGICAL_ORDER)을 사용하는 경우 위의 옵션 중 후자를 설정하는 것으로 선택할 수 있습니다.

DefRecon 옵션의 해석은 MQCNO_RECONNECT_* 값이 클라이언트 프로그램에도 설정되어 있는지 여부와 설정된 값에 따라 다릅니다.

클라이언트 프로그램이 MQCONN을 사용하여 연결하거나 MQCONNX를 사용하여 MQCNO_RECONNECT_AS_DEF 옵션을 설정하는 경우 DefRecon에 의해 설정된 다시 연결 값이 적용됩니다. 프로그램에서 또는 DefRecon 옵션에 의해 다시 연결 값이 설정되지 않은 경우 이 클라이언트 프로그램은 자동으로 다시 연결되지 않습니다.

자동 클라이언트 재연결은 Java의 IBM WebSphere MQ 클래스에서 지원되지 않습니다.

NO

MQCONNX로 대체되지 않는 한, 클라이언트는 자동으로 다시 연결되지 않습니다.

YES

MQCONNX로 대체되지 않으면 클라이언트는 자동으로 다시 연결됩니다.

QMGR

MQCONNX로 대체되지 않으면 클라이언트는 자동으로 다시 연결되지만 동일한 큐 관리자에만 다시 연결됩니다. QMGR 옵션은 MQCNO_RECONNECT_Q_MGR와 동일한 효과를 갖습니다.

DISABLED

MQCONNX MQI 호출을 사용하여 클라이언트 프로그램이 요청한 경우에도 다시 연결을 사용할 수 없습니다.

자동 클라이언트 다시 연결은 다음 두 가지 값에 따라 달라집니다.

- 애플리케이션에서 설정된 다시 연결 옵션
- mqclient.ini 파일의 DefRecon 값

표 22. 자동으로 다시 연결은 애플리케이션 및 mqclient.ini 파일에서 설정된 값에 따라 달라집니다.

mqclient.ini의 DefRecon 값	애플리케이션에 설정된 재연결 옵션			
	MQCNO_RECONNECT	MQCNO_RECONNECT_Q_MGR	MQCNO_RECONNECT_AS_DEF	MQCNO_RECONNECT_DISABLED
NO	YES	큐 관리자	아니오	아니오
YES	YES	큐 관리자	YES	아니오
QMGR	YES	큐 관리자	큐 관리자	아니오
DISABLED	아니오	아니오	아니오	아니오

MQReconnectTimeout

클라이언트에 다시 연결되기까지의 제한시간(초)입니다. 기본 값은 1800초(30분)입니다.

IBM WebSphere MQ classes for XMS .NET 클라이언트는 특성 `XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT`을 사용하여 다시 연결 제한시간을 지정할 수 있습니다. 이 특성의 기본값은 1800초(30분)입니다.

ServerConnectionParms

ServerConnectionParms는 MQSERVER 환경 매개변수와 동등하며 IBM WebSphere MQ 서버의 위치 및 사용할 통신 방법을 지정합니다. ServerConnectionParms 속성은 단순 채널만을 정의합니다. 이 속성을 사용하여 SSL 채널 또는 채널 엑시트가 있는 채널을 정의할 수 없습니다. 이는 `ChannelName / TransportType / ConnectionName` 형식의 문자열이고 `ConnectionName`은 완전한 네트워크 이름이어야 합니다. `ChannelName`에 슬래시("/") 문자를 포함할 수 없습니다. 이 문자는 채널 이름, 전송 유형 및 연결 이름을 분리하는 데 사용됩니다.

ServerConnectionParms가 클라이언트 채널을 정의하는 데 사용될 때 최대 메시지 길이 100MB가 사용됩니다. 따라서 채널에 적용되는 최대 메시지 크기는 서버에서 SVRCONN 채널에서 지정되는 값입니다.

단일 클라이언트 채널 연결만 수행할 수 있음을 참고하십시오. 예를 들어, 다음 두 가지 입력 항목이 있는 경우:

```
ServerConnectionParms=R1.SVRCONN/TCP/localhost(1963)
ServerConnectionParms=R2.SVRCONN/TCP/localhost(1863)
```

두 번째 입력 항목만 사용됩니다.

`ConnectionName`을 명시된 전송 유형에 대한 심표로 구분된 이름 목록으로 지정하십시오. 일반적으로 한 개의 이름만 필요합니다. 동일한 특성과의 다중 연결을 구성하기 위해 다중 `hostnames`를 제공할 수 있습니다. 이러한 연결은 연결이 성공적으로 설정될 때까지 연결 목록에 지정되는 순서대로 시도됩니다. 연결에 실패하면 클라이언트가 다시 처리를 시작합니다. 연결 목록은 큐 관리자 그룹에 대해 다시 연결 가능한 클라이언트에 대한 연결을 구성하기 위한 대안입니다.

Put1DefaultAlwaysSync=NO|YES

MQPMO_RESPONSE_AS_Q_DEF 옵션을 사용하여 MQPUT1 함수 호출의 작동을 제어합니다.

아니오

MQPUT1을 MQPMO_SYNCPOINT를 사용하여 설정하면 MQPMO_ASYNC_RESPONSE처럼 작동합니다. 마찬가지로 MQPUT1을 MQPMO_NO_SYNCPOINT를 사용하여 설정하면 MQPMO_SYNC_RESPONSE처럼 작동합니다. 이 값은 기본값입니다.

YES

MQPUT1 은 MQPMO_SYNCPOINT 또는 MQPMO_NO_SYNCPOINT 가 설정되었는지 여부에 관계없이 MQPMO_SYNC_RESPONSE 가 설정된 것처럼 작동합니다.

클라이언트 구성 파일의 ClientExitPath 스탠자

ClientExitPath 스탠자를 사용하여 클라이언트에서 채널 엑시트의 기본 위치를 지정할 수 있습니다.

다음 속성이 ClientExitPath 스탠자에 포함될 수 있습니다.

ExitsDefaultPath=string

클라이언트에 대한 32비트 채널 엑시트의 위치를 지정합니다.

ExitsDefaultPath64=string

클라이언트에 대해 64비트 채널 엑시트의 위치를 지정합니다.

JavaExitsClassPath=string

Java 엑시트가 실행될 때 CLASSPATH에 추가할 값. 이 값은 다른 언어로 된 엑시트에서는 무시됩니다.

JMS 구성 파일에서 JavaExitsClassPath 이름에는 표준 `com.ibm.mq.cfg`가 제공됩니다. 접두부가 지정되고 이 전체 이름은 Websphere MQ V7.0 시스템 특성에서도 사용됩니다. 버전 6.0에서는 버전 6.0 Readme에 문서화된 시스템 특성 `com.ibm.mq.exitClasspath`를 사용하여 이 속성이 지정되었습니다.

`com.ibm.mq.exitClasspath`는 더 이상 사용되지 않습니다. JavaExitsClassPath와 `exitClasspath`가 모두 있으면 JavaExitsClassPath를 사용합니다. `exitClasspath` 사용만 있는 경우에는 Websphere MQ V7.0에서도 여전히 사용됩니다.

클라이언트 구성 파일의 LU62, NETBIOS 및 SPX 스탠자

Windows 시스템에서만 이 스탠자를 사용하여 지정된 네트워크 프로토콜에 대한 구성 매개변수를 지정하십시오.

LU62

LU62 스탠자를 사용하여 SNA LU 6.2 프로토콜 구성 매개변수를 지정할 수 있습니다. 다음 속성이 이 스탠자에 포함될 수 있습니다.

Library1=DLLName| WCPIC32

APPC DLL의 이름입니다.

Library2=DLLName| WCPIC32

코드가 두 개의 개별 라이브러리에 저장된 경우에 사용되는 Library1과 동일합니다.

TPName

원격 사이트에서 시작할 TP 이름입니다.

NETBIOS

NETBIOS 스탠자를 사용하여 NetBIOS 프로토콜 구성 매개변수를 지정하십시오. 다음 속성이 이 스탠자에 포함될 수 있습니다.

AdapterNum=number| 0

LAN 어댑터의 번호입니다.

Library1=DLLName| NETAPI32

NetBIOS DLL의 이름입니다.

LocalName=name

LAN에서 이 컴퓨터가 표시되는 이름입니다.

이는 MQNAME 환경 매개변수와 같습니다.

NumCmds=number| 1

할당할 명령 수입니다.

NumSess=number| 1

할당할 세션 수입니다.

SPX

SPX 스탠자를 사용하여 SPX 프로토콜 구성 매개변수를 지정하십시오. 다음 속성이 이 스탠자에 포함될 수 있습니다.

BoardNum=number| 0

LAN 어댑터 번호입니다.

KeepAlive=YES|NO

KeepAlive 기능을 켜거나 끕니다.

KeepAlive=YES로 설정하면 연결의 다른 편 끝이 여전히 사용 가능한지 SPX가 주기적으로 점검합니다. 그렇지 않은 경우 채널이 닫힙니다.

Library1=DLLName| WSOCK32.DLL

SPX DLL의 이름입니다.

Library2=DLLName| WSOCK32.DLL

Library1과 동일하며 코드가 두 개의 별도 라이브러리에 저장된 경우에 사용됩니다.

Socket=number| 5E86

16진 표기법의 SPX 소켓 번호입니다.

클라이언트 구성 파일의 MessageBuffer 스탠자

MessageBuffer 스탠자를 사용하여 메시지 버퍼에 대한 정보를 지정할 수 있습니다.

다음 속성이 MessageBuffer 스탠자에 포함될 수 있습니다.

MaximumSize=integer| 1

미리 읽기 버퍼의 크기로, 값은 1 - 999 999KB 범위입니다.

다음 특수 값이 존재합니다.

-1

클라이언트가 적절한 값을 판별합니다.

0

클라이언트에 대해 미리 읽기 가 사용 불가능합니다.

PurgeTime=integer| 600

미리 읽기 버퍼에 남겨진 메시지가 영구 제거되기 전까지의 간격(초)입니다.

클라이언트 애플리케이션이 MsgId 또는 CorrelId에 따라 메시지를 선택하는 경우 미리 읽기 버퍼가 이전에 요청된 MsgId 또는 CorrelId를 사용하여 클라이언트에게 송신된 메시지를 포함할 수 있습니다. 이러한 메시지는 적절한 MsgId 또는 CorrelId로 MQGET이 발행될 때까지 미리 읽기 버퍼에 남아 있습니다. PurgeTime을 설정해서 미리 읽기 버퍼로부터 메시지를 제거할 수 있습니다. 영구 제거 간격보다 긴 시간 동안 미리 읽기 버퍼에 남아 있는 모든 메시지는 자동으로 영구 제거됩니다. 이러한 메시지는 큐 관리자의 큐에서 이미 제거되었으므로 찾아보지 않으면 손실됩니다.

유효한 범위는 1 - 999 999초이거나 영구 제거가 발생하지 않음을 의미하는 특수 값 0입니다.

UpdatePercentage=integer| -1

클라이언트 애플리케이션이 서버에 대한 새 요청을 작성하는 시간을 판별하기 위한 임계값 계산에서 사용되는 1 - 100 범위의 업데이트 백분율 값입니다. 특수 값 -1은 클라이언트가 적절한 값을 판별함을 나타냅니다.

클라이언트는 클라이언트 애플리케이션이 이용한 데이터 양을 표시하는 요청을 서버에 주기적으로 송신합니다. MQGET 호출을 통해 클라이언트가 검색한 바이트 수 n 이 임계값 T 를 초과하면 요청이 전송됩니다. 서버로 새 요청이 전송될 때마다 n 이 0으로 재설정됩니다.

임계값 T 는 다음과 같이 계산됩니다.

$$T = Upper - Lower$$

상한값은 미리 읽기 버퍼 크기와 같으며 *MaximumSize* 속성에 의해 KB 단위로 지정됩니다. 그 기본값은 100KB입니다.

하한값은 상한값보다 낮으며 *UpdatePercentage* 속성에 의해 지정됩니다. 이 속성은 1 - 100 범위의 숫자이고 기본값은 20입니다. 하한값은 다음과 같이 계산됩니다.

$$Lower = Upper \times UpdatePercentage / 100$$

예 1:

MaximumSize 및 *UpdatePercentage* 속성은 해당 속성의 기본값 100KB 및 20KB를 사용합니다.

클라이언트는 메시지를 검색하기 위해 MQGET을 호출하고 이를 반복적으로 수행합니다. 이는 MQGET이 n 바이트를 이용할 때까지 계속됩니다.

다음 계산 사용

$$T = Upper - Lower$$

T 는 $(100 - 20) = 80$ KB입니다.

따라서 MQGET 호출이 큐에서 80KB를 제거하면 클라이언트가 자동으로 새 요청을 작성합니다.

예제 2:

MaximumSize 속성은 그 기본값 100KB를 사용하고 *UpdatePercentage*에 대해서는 값 40이 선택됩니다.

클라이언트는 메시지를 검색하기 위해 MQGET을 호출하고 이를 반복적으로 수행합니다. 이는 MQGET이 n 바이트를 이용할 때까지 계속됩니다.

다음 계산 사용

T = Upper - Lower

T는 (100 - 40) = 60KB입니다.

따라서 MQGET 호출이 큐에서 60KB를 제거하면 클라이언트가 자동으로 새 요청을 작성합니다. 이는 기본값 사용 시 예 1보다 빠릅니다.

따라서 임계값 T로 큰 값을 선택할수록 클라이언트에서 서버로 요청이 송신되는 빈도가 감소하게 됩니다. 반대로 임계값 T로 작은 값을 선택할수록 클라이언트에서 서버로 송신되는 요청의 빈도가 증가하게 됩니다.

그러나 임계값 T를 선택하는 것은 미리 읽기 버퍼가 비어 있는 경우가 증가할 수 있어서 미리 읽기의 성능이 떨어지는 것을 의미할 수 있습니다. 이러한 상황이 발생하면 데이터가 서버에서 도착할 때까지 대기하면서 MQGET 호출을 일시정지해야 할 수 있습니다.

클라이언트 구성 파일의 SSL 스탠자

SSL 스탠자를 사용하여 SSL 또는 TLS의 사용에 대한 정보를 지정할 수 있습니다.

다음 속성이 SSL 스탠자에 포함될 수 있습니다.

CDPCheckExtensions=YES|NO

CDPCheckExtensions는 이 큐 관리자의 SSL 또는 TLS 채널이 CrLDistributionPoint 인증서 확장에서 이름 지정되는 CDP 서버를 검사하려고 시도할지 여부를 지정합니다.

이 속성에는 다음 값이 사용될 수 있습니다.

- YES: SSL 또는 TLS 채널이 디지털 인증서가 취소되었는지 여부를 판별하기 위해 CDP 서버를 검사하려고 시도합니다.
- NO: SSL 또는 TLS 채널이 CDP 서버를 검사하려고 시도하지 않습니다. 이 값은 기본값입니다.

CertificateLabel = string

채널 정의의 인증서 레이블입니다.

이 속성은 C 및 비관리 .NET 클라이언트에서 읽을 수 있습니다.

CertificateValPolicy=string

사용되는 인증서 유효성 검증의 유형을 판별합니다.

ANY

기본 보안 소켓 라이브러리가 지원하는 모든 인증서 유효성 검증 정책을 사용합니다. 이 설정이 기본 설정입니다.

RFC5280

RFC 5280 표준을 준수하는 인증서 유효성 검증만 사용합니다.

ClientRevocationChecks = REQUIRED | OPTIONAL | DISABLED

클라이언트 연결 호출에 SSL/TLS 채널을 사용하는 경우 인증서 폐기 검사가 구성되는 방식을 판별합니다.

[OCSPAuthentication](#)도 참조하십시오.

이 속성은 C 및 비관리 .NET 클라이언트에서 읽을 수 있습니다.

이 속성에는 다음 값이 사용될 수 있습니다.

REQUIRED(기본값)

CCDT에서 인증서 폐기 구성을 로드하고 구성된 대로 폐기 검사를 수행하려고 시도합니다. CCDT 파일을 열 수 없거나 인증서를 유효성 검증할 수 없는 경우(예를 들어, OCSP 또는 CRL 서버가 사용 불가능함으로 인해) MQCONN 호출이 실패합니다. CCDT에 폐기 구성이 없는 경우 폐기 검사가 수행되지 않지만 이로 인해 채널이 실패하지는 않습니다.

Windows Windows 시스템에서 CRL 폐기 검사를 위해 Active Directory를 사용할 수도 있습니다. OCSP 폐기 검사에는 Active Directory를 사용할 수 없습니다.

선택사항

REQUIRED의 경우 인증서 폐기 구성을 로드할 수 없는 경우에도 채널이 실패하지 않습니다.

DISABLED

CCDT에서 인증서 폐기 구성을 로드하기 위한 시도를 하지 않고 인증서 폐기 검사도 수행되지 않습니다.

참고: MQCONN 호출이 아닌 MQCONNX를 사용하는 경우 MQSCO를 통해 인증 정보 레코드(MQAIR)를 제공하도록 선택할 수 있습니다. 따라서 CCDT 파일을 열 수 없는 경우에 MQCONNX의 기본 작동은 실패하지 않지만 사용자가 선택하지 않아도 사용자가 MQAIR을 제공하는 것으로 가정합니다.

EncryptionPolicySuiteB=string

채널이 Suite-B 준수 암호화를 사용하는지 및 사용할 강도 레벨을 판별합니다. 가능한 값은 다음과 같습니다.

NONE

Suite-B 준수 암호화는 사용되지 않습니다. 이 설정이 기본 설정입니다.

128_BIT,192_BIT

보안 강도를 128비트 및 192비트 레벨로 설정합니다.

128_BIT

보안 강도를 128비트 레벨로 설정합니다.

192_BIT

보안 강도를 192비트 레벨로 설정합니다.

OCSPAuthentication=OPTIONAL| REQUIRED|WARN

OCSP가 사용되고 OCSP 폐기 검사로 인증서 폐기 상태를 판별할 수 없을 때 WebSphere MQ의 작동을 정의합니다. 가능한 값은 다음 세 가지입니다.

선택사항

OCSP 검사로 판별할 수 없는 폐기 상태인 모든 인증서가 승인되고 경고 또는 오류 메시지가 생성되지 않습니다. SSL 또는 TLS 연결은 폐기 검사가 수행되지 않은 것처럼 계속됩니다.

REQUIRED

OCSP 검사에서는 검사한 모든 SSL 또는 TLS 인증서에 대해 명확한 폐기 결과가 나와야 합니다. 확인할 수 없는 폐기 상태인 SSL 또는 TLS 인증은 오류 메시지와 함께 거부됩니다. 큐 관리자 SSL 이벤트 메시지가 사용 가능한 경우 ReasonQualifier가 MQRQ_SSL_HANDSHAKE_ERROR인 MQRQ_CHANNEL_SSL_ERROR 메시지가 생성됩니다. 연결이 닫혔습니다.

이 값이 기본값입니다.

WARN

OCSP 폐기 검사에서 SSL 또는 TLS 인증서의 폐기 상태를 판별할 수 없는 경우 큐 관리자 오류 로그에 경고가 보고됩니다. 큐 관리자 SSL 이벤트 메시지가 사용 가능한 경우 ReasonQualifier가 MQRQ_SSL_UNKNOWN_REVOCATION인 MQRQ_CHANNEL_SSL_WARNING 메시지가 생성됩니다. 연결을 계속할 수 있습니다.

OCSPCheckExtensions=YES|NO

WebSphere MQ가 AuthorityInfoAccess 인증 확장에서 수행하는지 여부를 제어합니다. 값이 NO로 설정되면 WebSphere MQ는 AuthorityInfoAccess 인증 확장을 무시하고 OCSP 보안 검사를 시도하지 않습니다. 기본값은 YES입니다.

SSLCryptoHardware=string

PKCS #11 암호화 하드웨어가 시스템에 표시되도록 구성하기 위해 필요한 매개변수 문자열을 설정합니다.

다음 형식으로 문자열을 지정하십시오. *GSK_PKCS11=driver path and filename; token label; token password; symmetric cipher setting;*

예를 들면, 다음과 같습니다. *GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so; tokenlabel; passwd; SYMMETRIC_CIPHER_ON*

드라이버 경로는 PKCS #11 카드에 대한 지원을 제공하는 공유 라이브러리의 절대 경로입니다. 드라이버 파일 이름은 공유 라이브러리의 이름입니다. PKCS #11 드라이버 경로 및 파일 이름에 필요한 값의 예는 */usr/lib/pkcs11/PKCS11_API.so*입니다. GSKit을 통해 대칭 암호 조작에 액세스하려면 대칭 암호 설정 매개변수를 지정하십시오. 이 매개변수의 값은 다음 중 하나입니다.

SYMMETRIC_CIPHER_OFF

대칭 암호 조작에 액세스하지 않습니다. 이 설정이 기본 설정입니다.

SYMMETRIC_CIPHER_ON

대칭 암호 조작에 액세스합니다.

문자열의 최대 길이는 256자입니다. 기본값은 공백입니다. 올바른 형식이 아닌 문자열을 지정하는 경우 오류가 생성됩니다.

SSLFipsRequired=YES|_NO

암호화가 WebSphere MQ에서 수행되는 경우 FIPS 인증 알고리즘만 사용할지 여부를 지정합니다. 암호화 하드웨어가 구성된 경우에 사용되는 암호화 모듈은 하드웨어 제품에 제공된 모듈입니다. 이러한 모듈은 사용 중인 하드웨어 제품에 따라 특정 레벨의 FIPS 인증이거나 이 인증이 아닐 수 있습니다.

SSLHTTPProxyName=string

문자열은 OCSP 검사를 위해 GSKit에서 사용될 HTTP 프록시 서버의 호스트 이름 또는 네트워크 주소입니다. 이 주소 뒤에 선택적 포트 번호를 괄호로 묶어 표시할 수 있습니다. 포트 번호가 지정되지 않으면 기본 HTTP 포트 80이 사용됩니다. HP-UX PA-RISC 및 Sun Solaris SPARC 플랫폼과 AIX의 32비트 클라이언트에서 네트워크 주소는 IPv4 주소만 가능합니다. 다른 플랫폼에서는 IPv4 또는 IPv6 주소를 사용할 수 있습니다.

예를 들어 방화벽이 OCSP 응답자의 URL에 대한 액세스를 막는 경우 이 속성이 필요할 수 있습니다.

SSLKeyRepository=pathname

사용자의 디지털 인증서를 보유하는 키 저장소의 위치이며 스템 형식입니다. 즉, 전체 경로와 확장자가 없는 파일 이름을 포함합니다.

SSLKeyResetCount=integer|_0

비밀 키를 재협상하기 전에 SSL 또는 TLS 채널에서 송신 및 수신된 암호화되지 않은 바이트의 수입니다.

값의 범위는 0 - 999999999여야 합니다.

기본값은 0이고 보안 키가 재협상되지 않음을 의미합니다.

값 1 - 32768을 지정하는 경우 SSL 또는 TLS 채널은 비밀 키 재설정 수 32768(32Kb)을 사용합니다. 이는 보안 키 재설정 값이 작으면 발생할 수 있는 과도한 키 재설정 수를 피하기 위한 것입니다.

클라이언트 구성 파일의 TCP 스탠자

TCP 스탠자를 사용하여 TCP 네트워크 프로토콜 구성 매개변수를 지정할 수 있습니다.

다음 속성이 TCP 스탠자에 포함될 수 있습니다.

Clntrcvbuffsize=number|_32768

클라이언트 연결 서버 연결 채널의 클라이언트 측에서 사용되는 TCP/IP 수신 버퍼의 크기(바이트)입니다. 0 값은 WebSphere MQ에서 버퍼 크기를 고정하는 것과는 반대로 운영 체제가 버퍼 크기를 관리함을 표시합니다.

Clntrsndbuffsize=number|_32768

클라이언트 연결 서버 연결 채널의 클라이언트 측에서 사용되는 TCP/IP 송신 버퍼의 크기(바이트)입니다. 0 값은 WebSphere MQ에서 버퍼 크기를 고정하는 것과는 반대로 운영 체제가 버퍼 크기를 관리함을 표시합니다.

Connect_Timeout=number

소켓에 대한 연결 시도의 제한시간이 초과되는 시간(초 단위). 0이 아닌 클라이언트 채널 가중치로 채널이 구성된 경우(이 때에는 기본값이 5임)가 아니면 기본값은 0입니다.

IPAddressVersion=MQIPADDR_IPV4|MQIPADDR_IPV6

채널 연결에 사용할 IP 프로토콜을 지정합니다.

가능한 문자열 값은 MQIPADDR_IPV4 또는 MQIPADDR_IPV6입니다. 이러한 값은 ALTER QMGR IPADDRV의 IPV4 및 IPV6 과 동일한 의미를 갖습니다.

KeepAlive=YES|NO

KeepAlive 기능을 켜거나 끕니다. KeepAlive=YES로 설정하면 TCP/IP가 연결의 다른 측이 여전히 사용 가능한지 주기적으로 검사합니다. 그렇지 않은 경우 채널이 닫힙니다.

Windows Library1=DLLName|_WSOCK32

(Windows 전용) TCP/IP 소켓 DLL의 이름

TMF 및 TMF/게이트웨이 스탠자

IBM WebSphere MQ 제공 TMF/게이트웨이는 Pathway 환경에서 실행됩니다. TMF 및 TMF/Gateway 스탠자를 사용하여 TMF/Gateway와 통신하기 위해 HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트에 필요한 구성 매개변수를 지정하십시오.

TMF를 사용하려는 경우에는 통신하고 있는 각 큐 관리자에 대해 TMF 스탠자와 TmfGateway 스탠자를 정의해야 합니다. 모든 값은 구성에서 도출됩니다.

TMF 스탠자

PathMon=name

TMF/게이트웨이에 대한 서버 클래스를 정의하는, 정의된 Pathmon 프로세스의 이름.

TmfGateway 스탠자

다음 속성이 이 스탠자에 포함될 수 있습니다.

QManager=name

대기열 관리자의 이름입니다.

Server=name

해당 큐 관리자에 대해 구성된 TMF/게이트웨이의 서버 클래스 이름.

예

서로 다른 서버에 있는 두 개의 서로 다른 큐 관리자에 대한, 두 개의 TmfGateway 스탠자로 정의된 TMF 스탠자의 예는 다음과 같습니다.

```
TMF:
  PathMon=$PSD1P

TmfGateway:
  QManager=MQ5B
  Server=MQ-MQ5B

TmfGateway:
  QManager=MQ5C
  Server=MQ-MQ5C
```

WebSphere MQ 환경 변수 사용

이 절에서는 WebSphere MQ MQI 클라이언트 애플리케이션과 함께 사용할 수 있는 환경 변수에 대해 설명합니다.

다음과 같은 방식으로 환경 변수를 사용할 수 있습니다.

- 시스템 프로파일에서 값을 설정하여 영구적 변경 수행
- 명령행에서 명령을 발행하여 이 세션에 대해서만 변경 수행
- 하나 이상의 변수에 실행 중인 애플리케이션에 따른 특정 값을 지정하도록 애플리케이션이 사용하는 명령 스크립트 파일에 명령 추가

WebSphere MQ는 설정하지 않은 변수에 대해 기본 값을 사용합니다.

별도로 명시되지 않는 한 명령은 모든 WebSphere MQ MQI 클라이언트 플랫폼에서 사용 가능합니다.

각 환경 변수에 대해 플랫폼에 맞는 명령을 사용하여 현재 설정을 표시하거나 변수의 값을 재설정하십시오.

예를 들면, 다음과 같습니다.

환경 변수의 값 설정 또는 재설정		
결과	명령	
	Windows	UNIX and Linux 시스템
변수 제거	SET MQSERVER=	unset MQSERVER
현재 설정 표시	SET MQSERVER	echo \$MQSERVER
세션에 대해 모든 환경 변수를 표시하십시오.	set	set

개별 변수에 대한 정보는 다음 하위 주제를 참조하십시오.

관련 개념

[114 페이지의 『구성 파일을 사용하여 클라이언트 구성』](#)

텍스트 파일의 속성을 사용하여 클라이언트를 구성하십시오. 이러한 속성은 환경 변수 또는 다른 플랫폼별 방식으로 대체될 수 있습니다.

관련 참조

[환경 변수](#)

MQCCSID

MQCCSID는 사용할 코드화된 문자 세트 번호를 지정하고 서버 구성에 사용된 CCSID 값을 대체합니다.

자세한 정보는 [클라이언트 또는 서버 코드화 문자 세트 ID\(CCSID\)](#)를 참조하십시오.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows의 경우:

```
SET MQCCSID=number
```

- UNIX and Linux 시스템의 경우:

```
export MQCCSID=number
```

MQCERTVPOL

MQCERTVPOL은 사용되는 인증서 유효성 검증 정책을 지정합니다.

WebSphere MQ에서 인증서 유효성 검증 정책에 대한 자세한 정보는 [WebSphere MQ에서 인증서 유효성 검증 정책을 참조하십시오](#).

이 환경 변수는 클라이언트 ini 파일의 SSL 스탠자에서 *CertificateValPolicy* 설정을 대체합니다. 이 변수는 다음 두 값 중 하나로 설정할 수 있습니다.

ANY

기본 보안 소켓 라이브러리가 지원하는 모든 인증서 유효성 검증 정책을 사용합니다.

RFC5280

RFC 5280 표준을 준수하는 인증서 유효성 검증만 사용합니다.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows의 경우:

```
SET MQCERTVPOL=value
```

- UNIX and Linux 시스템의 경우:

```
export MQCERTVPOL=value
```

MQCHLLIB

MQCHLLIB는 클라이언트 채널 정의 테이블(CCDT)이 들어 있는 파일에 대한 디렉토리 경로를 지정합니다. 파일은 서버에 작성되거나 WebSphere MQ MQI 클라이언트 워크스테이션에 복사할 수 있습니다.

MQCHLLIB를 설정하지 않으면 클라이언트에 대한 경로는 기본값 설정됩니다.

- ▶ **Windows** Windows의 경우: `MQ_INSTALLATION_PATH`
- ▶ **UNIX** ▶ **Linux** UNIX and Linux 시스템의 경우: `/var/mqm/`

`crtmqm` 및 `strmqm` 명령의 경우 이 경로는 두 경로 세트 중 하나로 기본값 설정됩니다. `datapath`를 설정하면 기본 경로는 첫 번째 세트 중 하나입니다. `datapath`를 설정하지 않으면 기본 경로는 두 번째 세트 하나입니다.

- ▶ **Windows** Windows의 경우: `datapath\@ipcc`
- ▶ **UNIX** ▶ **Linux** Linux의 경우: `datapath/@ipcc`

또는

- ▶ **Windows** 의: `MQ_INSTALLATION_PATH\data\qmgrs\qmgrname\@ipcc`
- ▶ **UNIX** ▶ **Linux** UNIX and Linux 시스템의 경우: `/prefix/qmgrs/qmgrname/@ipcc`

설명:

- `MQ_INSTALLATION_PATH`는 IBM WebSphere MQ가 설치된 상위 레벨 디렉토리를 나타냅니다.
- 존재하는 경우 `datapath`는 큐 관리자 스탠자에 정의된 DataPath 값입니다.
- `prefix`는 큐 관리자 스탠자에 정의된 접두부 값입니다. Prefix is typically `/var/mqm` on 유닉스 and Linux platforms.
- `qmgrname`은 큐 관리자 스탠자에 정의된 Directory 속성 값입니다. 이 값은 실제 큐 관리자 이름과 다를 수 있습니다. 이 값은 특수 문자를 대체하기 위해 변경되었을 수 있습니다.
- 큐 관리자 스탠자는 유닉스, Linux의 `mq5.ini` 파일 및 윈도우의 레지스트리에 정의되어 있습니다.

참고사항:

1. 설정된 경우 MQCHLLIB는 CCDT를 찾을 때 사용되는 경로를 대체합니다.
2. MQCHLLIB와 같은 환경 변수는 프로세스 또는 작업으로 범위가 설정되거나 플랫폼별 방식에서는 시스템 전체로 범위가 설정될 수 있습니다.
3. 서버에서 MQCHLLIB를 시스템 전체에 설정하는 경우 서버의 모든 큐 관리자에 대해 동일한 CCDT 파일 경로를 설정합니다. MQCHLLIB 환경 변수를 설정하지 않는 경우 이 경로는 각 큐 관리자마다 다릅니다. 이 환경 변수가 `crtmqm` 또는 `strmqm` 명령에서 설정되면 큐 관리자가 MQCHLLIB의 값을 읽습니다.
4. 서버에서 다중 큐 관리자를 작성하는 경우 다음과 같은 이유로 해당 큐 관리자를 구분해야 합니다. MQCHLLIB를 시스템 전체에 설정하면 각 큐 관리자는 동일한 CCDT 파일을 업데이트합니다. 이 파일은 서버에 있는 모든 큐 관리자의 클라이언트 연결 정의를 포함합니다. 다중 큐 관리자에서 동일한 정의가 있을 경우 (예: SYSTEM.DEF.CLNTCONN) 이 파일은 최근 정의를 포함합니다. 큐 관리자를 작성할 때 MQCHLLIB가 설정되면 SYSTEM.DEF.CLNTCONN가 CCDT에서 업데이트됩니다. 갱신은 다른 큐 관리자가 작성한 SYSTEM.DEF.CLNTCONN를 겹쳐씁니다. 이전 정의를 수정한 경우 수정사항이 유실됩니다. 이러한 이유로 서버에서 MQCHLLIB를 시스템 전체 환경 변수로 설정하는 것에 대한 대안을 찾는 것을 고려해야 합니다.
5. 클라이언트 정의에서 MQSC 및 PCF NOREPLACE 옵션은 CCDT 파일의 콘텐츠를 검사하지 않습니다. 이전에 작성한 것과 같은 이름이지만 이 큐 관리자가 아닌 다른 큐 관리자에 의해 작성된 클라이언트 연결 채널 정의는 NOREPLACE 옵션에 관계없이 대체됩니다. 같은 큐 관리자가 이전에 작성한 정의인 경우 이 정의는 대체되지 않습니다.
6. `rcrmqobj -t clchltab` 명령은 CCDT 파일을 삭제하고 다시 작성합니다. 명령이 실행 중인 큐 관리자에 작성된 클라이언트-연결 정의만을 사용하여 파일이 재작성됩니다.
7. CCDT를 업데이트하는 다른 명령은 같은 채널 이름을 가지는 클라이언트 연결 채널만 수정합니다. 파일의 다른 클라이언트 연결 채널은 대체되지 않습니다.

8. MQCHLLIB에 대한 경로는 인용 부호가 필요하지 않습니다.

예:

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- **Windows** Windows의 경우,

```
SET MQCHLLIB=pathname
```

예를 들면, 다음과 같습니다.

```
SET MQCHLLIB=C:\wmqtest
```

- **UNIX Linux** UNIX and Linux 시스템의 경우:

```
export MQCHLLIB=pathname
```

MQCHLTAB

MQCHLTAB는 클라이언트 채널 정의 테이블(CCDT)이 들어 있는 파일의 이름을 지정합니다. 기본 파일 이름은 AMQCLCHL.TAB입니다.

서버에서 클라이언트 채널 정의 테이블의 위치에 대한 정보는 [105 페이지의 『클라이언트 채널 정의 테이블』](#)의 내용을 참조하십시오.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows의 경우:

```
SET MQCHLTAB=filename
```

- UNIX and Linux 시스템의 경우:

```
export MQCHLTAB=filename
```

예를 들면, 다음과 같습니다.

```
SET MQCHLTAB=ccdf1.tab
```

클라이언트에서와 같은 방식으로 서버에서 MQCHLTAB 환경 변수는 클라이언트 채널 정의 테이블의 이름을 지정합니다.

MQIPADDRV

MQIPADDRV는 채널 연결에 사용할 IP를 지정합니다. 포함될 수 있는 문자열 값은 "MQIPADDR_IPV4" 또는 "MQIPADDR_IPV6"입니다. 이 값의 의미는 ALTER QMGR IPADDRV의 IPV4 및 IPV6과 동일합니다. 이 변수에 값이 설정되지 않으면 "MQIPADDR_IPV4"가 사용됩니다.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows의 경우:

```
SET MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

- UNIX and Linux 시스템의 경우:

```
export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

MQNAME

MQNAME은 WebSphere MQ 프로세스가 사용할 수 있는 로컬 NetBIOS 이름을 지정합니다.

클라이언트 및 서버에서 우선순위 규칙 및 전체 설명은 79 페이지의 『Windows에서 NetBIOS 연결 정의』의 내용을 참조하십시오.

이 변수를 설정하려면 이 명령을 사용하십시오.

```
SET MQNAME=Your_env_Name
```

예를 들면, 다음과 같습니다.

```
SET MQNAME=CLIENT1
```

WebSphere MQ MQI 클라이언트에서 다중 WebSphere MQ 애플리케이션을 동시에 실행할 경우, 일부 플랫폼의 NetBIOS에서 각 애플리케이션마다 다른 이름(MQNAME으로 설정된)을 요구합니다.

MQSERVER

MQSERVER 환경 변수는 최소 채널을 정의하는 데 사용됩니다. MQSERVER는 WebSphere MQ 서버의 위치와 사용할 통신 방법을 지정합니다.

채널 엑시트가 있는 채널 또는 SSL 채널을 정의하는 데 MQSERVER를 사용할 수 없습니다. SSL 채널을 정의하는 방법에 대한 자세한 내용은 [SSL로 채널 보호](#)를 참조하십시오.

*ConnectionName*은 완전한 네트워크 이름이어야 합니다. 슬래시 문자는 채널 이름, 전송 유형 및 연결 이름을 구분하는 데 사용되기 때문에 *ChannelName*은 슬래시(/) 문자를 포함할 수 없습니다. MQSERVER 환경 변수가 클라이언트 채널을 정의하는 데 사용될 때 최대 메시지 길이(MAXMSGL) 100MB가 사용됩니다. 따라서 채널에 적용되는 최대 메시지 크기는 서버에서 SVRCONN 채널에 지정되는 값입니다.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows의 경우:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

- UNIX and Linux 시스템의 경우:

```
export MQSERVER='ChannelName/TransportType/ConnectionName'
```

*TransportType*은 IBM WebSphere MQ 클라이언트 플랫폼에 따라 다음 값 중 하나일 수 있습니다.

- LU62
- TCP
- NETBIOS
- SPX

*ConnectionName*은 심표로 구분된 연결 이름 목록일 수 있습니다. 목록에 있는 연결 이름은 클라이언트 연결 테이블의 다중 연결과 비슷한 방식으로 사용됩니다. *ConnectionName* 목록은 큐 관리자 그룹에 대한 대안으로 사용되어 시도할 클라이언트에 대해 다중 연결을 지정할 수 있습니다. 다중 인스턴스 큐 관리자를 구성하는 경우 서로 다른 큐 관리자 인스턴스를 지정하기 위해 *ConnectionName* 목록을 사용할 수 있습니다.

TCP/IP 기본 포트

기본적으로 TCP/IP의 경우, WebSphere MQ는 채널이 포트 1414에 연결되는 것으로 가정합니다.

이 값을 다음과 같이 변경할 수 있습니다.

- 대괄호 안의 포트 번호를 *ConnectionName*의 마지막 부분에 추가:

- Windows의 경우:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)
```

- UNIX and Linux 시스템의 경우:

```
export MQSERVER='ChannelName/TransportType/ConnectionName(PortNumber)'
```

- 포트 번호를 프로토콜 이름에 추가하여 mqclient.ini 파일 변경. 예를 들면, 다음과 같습니다.

```
TCP:
port=2001
```

- 85 페이지의 『TCP/IP 리스너 사용』에 설명된 대로 서비스 파일에 WebSphere MQ 를 추가합니다.

SPX 기본 소켓

기본적으로 SPX의 경우, WebSphere MQ는 채널이 소켓 5E86에 연결되는 것으로 가정합니다.

이 값을 다음과 같이 변경할 수 있습니다.

- 대괄호 안의 소켓 번호를 ConnectionName의 마지막 부분에 추가:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(SocketNumber)
```

SPX 연결의 경우, ConnectionName 및 소켓을 network.node(socket) 양식으로 지정하십시오. WebSphere MQ 클라이언트 및 서버가 동일한 네트워크에 있으면 네트워크를 지정하지 않아도 됩니다. 기본 소켓을 사용하는 경우 소켓을 지정하지 않아도 됩니다.

- 포트 번호를 프로토콜 이름에 추가하여 qm.ini 파일 변경. 예를 들면, 다음과 같습니다.

```
SPX:
socket=5E87
```

MQSERVER 사용

MQSERVER 환경 변수를 사용하여 WebSphere MQ MQI 클라이언트 시스템과 서버 시스템 사이의 채널을 정의할 경우, 애플리케이션은 이 채널만을 사용할 수 있으며 클라이언트 채널 정의 테이블(CCDT)을 참조하지 않습니다.

이 경우 서버 시스템에서 실행 중인 리스너 프로그램이 애플리케이션을 연결할 큐 관리자를 판별합니다. 이 큐 관리자는 리스너 프로그램이 연결되는 것과 같은 큐 관리자입니다.

MQCONN 또는 MQCONNX 요청이 리스너가 연결된 큐 관리자 외의 큐 관리자를 지정하거나 MQSERVER 매개변수 TransportType이 인식되지 않는 경우 MQCONN 또는 MQCONNX 요청이 리턴 코드 MQRC_Q_MGR_NAME_ERROR로 실패합니다.

UNIX and Linux 시스템에서 MQSERVER를 다음 예 중 하나에서와 같이 정의할 수 있습니다.

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56(2002)'  
export MQSERVER=CHANNEL1/LU62/BOX99
```

모든 MQCONN 또는 MQCONNX 요청은 MQCD 구조가 MQCONNX에 제공된 MQCNO 구조에서 참조되지 않는 한, 정의한 채널을 사용하려고 시도합니다. 이 경우, MQCD 구조로 지정된 채널은 MQSERVER 환경 변수에 의해 지정된 것보다 우선합니다.

MQSERVER 환경 변수는 MQCHLLIB 및 MQCHLTAB에 의해 지정되는 클라이언트 채널 정의보다 우선합니다.

MQSERVER 취소

MQSERVER를 취소하고 MQCHLLIB 및 MQCHLTAB가 지정하는 클라이언트 채널 정의 테이블로 돌아가려면 다음을 입력하십시오.

- Windows의 경우:

```
SET MQSERVER=
```

- UNIX and Linux 시스템의 경우:

```
unset MQSERVER
```

MQSSLCRYP

MQSSLCRYP는 시스템에 표시되는 암호화 하드웨어 구성을 가능하게 하는 매개변수 문자열을 보유합니다. 허용되는 값은 ALTER QMGR 명령의 SSLCRYP 매개변수에 대한 값과 동일합니다.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows 시스템의 경우:

```
SET MQSSLCRYP=string
```

- UNIX and Linux 시스템의 경우:

```
export MQSSLCRYP=string
```

관련 참조

[ALTER QMGR 명령 SSLCRYP 매개변수](#)

MQSSLFIPS

MQSSLFIPS는 WebSphere MQ에서 암호화가 수행되는 경우에 FIPS 인증 알고리즘만 사용되는지 여부를 지정합니다. 해당 값은 ALTER QMGR 명령의 SSLFIPS 매개변수에 대한 값과 동일합니다.

FIPS 인증 알고리즘의 사용은 암호화 하드웨어의 영향을 받습니다. [MQI 클라이언트에서 런타임 시 FIPS 인증 CipherSpec만 사용하도록 지정](#)을 참조하십시오.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows 시스템의 경우:

```
SET MQSSLFIPS=YES|NO
```

- UNIX and Linux 시스템의 경우:

```
export MQSSLFIPS=YES|NO
```

기본값은 NO입니다.

MQSSLKEYR

MQSSLKEYR은 사용자에게 속하는 디지털 인증서가 보관된 키 저장소의 위치를 스템 형식으로 지정합니다. 스템 형식은 전체 경로와 확장자 없는 파일 이름을 포함함을 의미합니다. 자세한 내용은 ALTER QMGR 명령의 SSLKEYR 매개변수를 참조하십시오.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows 시스템의 경우:

```
SET MQSSLKEYR=pathname
```

- UNIX and Linux 시스템의 경우:

```
export MQSSLKEYR=pathname
```

기본값은 없습니다.

MQSSLPROXY

MQSSLPROXY는 OSCP 검사를 위해 GSKit에서 사용하는 HTTP 프록시 서버의 호스트 이름 또는 포트 번호를 지정합니다.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows 시스템의 경우:

```
SET MQSSLPROXY=string
```

- UNIX and Linux 시스템의 경우:

```
export MQSSLPROXY="string"
```

문자열은 OSCP 검사를 위해 GSKit에서 사용될 HTTP 프록시 서버의 호스트 이름 또는 네트워크 주소입니다. 이 주소 뒤에 선택적 포트 번호를 괄호로 묶어 표시할 수 있습니다. 포트 번호가 지정되지 않으면 기본 HTTP 포트 80이 사용됩니다.

예를 들어, UNIX and Linux 시스템에서는 다음 명령 중 하나를 사용할 수 있습니다.

- ```
export MQSSLPROXY="proxy.example.com(80)"
```

- ```
export MQSSLPROXY="127.0.0.1"
```

MQSSLRESET

MQSSLRESET는 비밀 키가 재협상되기 전에 SSL 또는 TLS 채널에서 송신 및 수신된 암호화되지 않은 바이트 수를 나타냅니다.

비밀 키 재협상에 대한 자세한 정보는 [SSL 및 TLS 비밀 키 재설정을 참조하십시오](#).

이 변수는 0 - 999 999 999 범위의 정수로 설정될 수 있습니다. 기본값은 0이며 보안 키가 재협상되지 않음을 의미합니다. SSL 또는 TLS 비밀 키 재설정 계수를 1바이트 - 32KB 범위로 지정하면 SSL 또는 TLS 채널이 32KB의 비밀 키 재설정 수를 사용합니다. 이 비밀 키 재설정 수는 SSL 또는 TLS 비밀 키 재설정 값이 작으면 발생하게 되는 과도한 키 재설정 수를 피하기 위한 것입니다.

이 변수를 설정하려면 다음 명령 중 하나를 사용하십시오.

- Windows 시스템:

```
SET MQSSLRESET=integer
```

- UNIX and Linux 시스템:

```
export MQSSLRESET=integer
```

큐된 발행/구독 제어

큐 대기된 발행/구독의 상태를 시작, 중지 또는 표시할 수 있습니다. 스트림을 추가하거나 제거할 수 있고 브로커 계층에서 큐 관리자를 추가하고 삭제할 수도 있습니다.

큐된 발행/구독 제어에 대한 자세한 정보는 다음 하위 주제를 참조하십시오.

큐된 발행/구독 메시지 속성 설정

큐 관리자 속성을 사용하여 일부 발행/구독 메시지 속성의 작동을 제어합니다. `qm.ini` 파일의 *Broker* 스탠자에서 제어하는 다른 속성입니다.

이 태스크 정보

다음 발행/구독 속성을 설정할 수 있습니다. 자세한 내용은 [큐 관리자 매개변수](#)를 참조하십시오.

표 23. 발행/구독 구성 매개변수	
설명	MQSC 매개변수 이름
명령 메시지 재시도 수	PSRTYCNT
전달할 수 없는 명령 입력 메시지 제거	PSNPMSG
전달할 수 없는 명령 응답 메시지 다음의 작동	PSNPRES
동기점에서 명령 메시지 처리	PSSYNCPT

브로커 스탠자는 다음 구성 설정을 관리하는 데 사용됩니다.

- `PersistentPublishRetry=yes | force`

`Yes`를 지정한 다음 큐된 발행/구독 인터페이스를 통한 지속 메시지의 발행이 실패하고 부정적인 응답이 요청되지 않은 경우 발행 조작이 재시도됩니다.

부정적 응답 메시지를 요청한 경우 부정적 응답이 송신되고 추가 재시도는 발생하지 않습니다.

`Force`를 지정한 다음 큐된 발행/구독 인터페이스를 통한 지속 메시지 발행이 실패하는 경우 성공적으로 처리될 때까지 발행 조작이 재시도됩니다. 부정적인 응답이 송신되지 않습니다.

- `NonPersistentPublishRetry=yes | force`

`Yes`를 지정한 다음 큐된 발행/구독 인터페이스를 통한 비지속 메시지의 발행이 실패하고 부정적인 응답이 요청되지 않은 경우 발행 조작이 재시도됩니다.

부정적 응답 메시지를 요청한 경우 부정적 응답이 송신되고 추가 재시도는 발생하지 않습니다.

`Force`를 지정한 다음 큐된 발행/구독 인터페이스를 통한 비지속 메시지의 발행이 실패하는 경우 성공적으로 처리될 때까지 발행 조작이 재시도됩니다. 부정적인 응답이 송신되지 않습니다.

참고: 비지속 메시지에 대해 이 기능을 사용하려는 경우 `NonPersistentPublishRetry` 값을 설정하는 것 외에도 큐 관리자 속성 `PSSYNCPT`가 `Yes`로 설정되는지 확인해야 합니다.

이를 수행하면 `STREAM` 큐의 `MQGET`이 이제 동기점 아래에서 발생하므로 비지속적 발행물 처리 성능에 영향을 줄 수도 있습니다.

- `PublishBatchSize=number`

브로커는 일반적으로 동기점 내 발행 메시지를 처리합니다. 각 발행물을 개별적으로 커미트하는 것은 비효율적일 수 있고 일부 환경에서 브로커는 단일 작업 단위로 여러 발행 메시지를 처리할 수 있습니다. 이 매개변수는 단일 작업 단위로 처리할 수 있는 최대 발행 메시지 수를 지정합니다.

`PublishBatchSize`의 기본값은 5입니다.

- `PublishBatchInterval=number`

브로커는 일반적으로 동기점 내 발행 메시지를 처리합니다. 각 발행물을 개별적으로 커미트하는 것은 비효율적일 수 있고 일부 환경에서 브로커는 단일 작업 단위로 여러 발행 메시지를 처리할 수 있습니다. 이 매개변수는 배치 내 첫 번째 메시지와 같은 배치에 포함된 후속 발행물 간 최대 시간(밀리초 단위)을 지정합니다.

배치 간격이 0이면 메시지가 즉시 사용 가능하다는 전제 하에 최대 `PublishBatchSize`개의 메시지를 처리할 수 있음을 나타내는 것입니다.

`PublishBatchInterval`의 기본값은 0입니다.

프로시저

WebSphere MQ Explorer, 프로그램 가능 명령 또는 `runmqsc` 명령을 사용하여 발행/구독의 작동을 제어하는 큐 관리자 속성을 대체하십시오.

예

```
ALTER QMGR PSNPRES(SAFE)
```

큐된 발행/구독 시작

시작하기 전에

PSMODE에 대한 설명을 읽고 발행/구독의 다음 세 가지 모드에 대해 이해하십시오.

- COMPAT
- 사용 안함
- 사용 가능

참고: 업그레이드된 큐 관리자에 대해 작업하는 경우 Version 6.0에서 마이그레이션했으면 **strmqbrk**를 사용하여 Version 6.0 Pub/Sub 브로커 상태를 마이그레이션해야 합니다. 이는 z/OS에는 적용되지 않습니다.

이 태스크 정보

큐된 발행/구독 인터페이스(브로커라고도 함) 또는 발행/구독 엔진(버전 7 발행/구독이라고도 함) 또는 이 둘 다를 시작하려면 QMGR PSMODE 속성을 설정하십시오. 큐된 발행/구독을 시작하려면 PSMODE를 ENABLED로 설정해야 합니다. 기본값은 ENABLED입니다.

프로시저

인터페이스가 아직 사용되도록 설정되지 않은 경우 WebSphere MQ Explorer나 **runmqsc** 명령을 사용하여 큐된 발행/구독 인터페이스를 사용하도록 하십시오.

예

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

다음에 수행할 작업

WebSphere MQ는 큐된 발행/구독 명령과 발행/구독 MQI(Message Queue Interface) 호출만 처리합니다.

큐된 발행/구독 중지

시작하기 전에

큐잉된 발행/구독이 더 이상 사용되지 않습니다.

PSMODE에 대한 설명을 읽고 발행/구독의 다음 세 가지 모드에 대해 이해하십시오.

- COMPAT
- DISABLED
- ENABLED

이 태스크 정보

큐된 발행/구독 인터페이스(브로커라고도 함) 또는 발행/구독 엔진(버전 7 발행/구독이라고도 함)을 중지하거나 이 둘 다를 중지하려면 QMGR PSMODE 속성을 설정하십시오. 큐된 발행/구독을 중지하려면 PSMODE를 COMPAT로 설정해야 합니다. 발행/구독 엔진 전체를 중지하려면 PSMODE를 DISABLED로 설정하십시오.

프로시저

WebSphere MQ Explorer 또는 **runmqsc** 명령을 사용하여 큐잉된 발행/구독 인터페이스를 사용 불가능하도록 설정하십시오.

예

```
ALTER QMGR PSMODE(COMPAT)
```

스트림 추가

Version 6.0 큐 관리자에서 마이그레이션된 스트림과 함께 있도록 수동으로 스트림을 추가할 수 있습니다.

시작하기 전에

스트림 및 토픽 토픽을 읽어 발행/구독 스트림이 작동하는 방식을 학습하십시오.

이 태스크 정보

PCF 명령 **runmqsc** 또는 IBM WebSphere MQ Explorer를 사용하여 다음 단계를 수행하십시오.

참고: 순서에 관계없이 단계 1과 단계 2를 수행할 수 있습니다. 단계 1 및 단계 2를 둘 다 완료한 후에는 단계 3만 수행하십시오.

프로시저

1. Version 6.0 스트림과 같은 이름을 사용하여 로컬 큐를 정의하십시오.
2. Version 6.0 스트림과 같은 이름을 사용하여 로컬 토픽을 정의하십시오.
3. 이름 목록에 큐 이름을 추가하십시오. `SYSTEM.QPUBSUB.QUEUE.NAMELIST`
4. 발행/구독 계층에서 Version 7.1 이상의 모든 큐 관리자에 대해 반복하십시오.

'Sport' 추가

스트림 'Sport'를 공유하는 예에서 Version 6.0 및 Version 7.1 큐 관리자는 동일한 발행/구독 계층에서 작동합니다. Version 6.0 큐 관리자는 'Sport'라는 스트림을 공유합니다. 이 예에서는 버전 6 스트림 'Sport'과 공유되는 토픽 문자열 'Sport'를 사용하여 'Sport'라는 Version 7.1 큐 관리자에서 큐 및 토픽을 작성하는 방법을 보여줍니다.

토픽 문자열 'Soccer/Results'를 사용하여 토픽 'Sport'를 발행하는 Version 7.1 발행 애플리케이션이 결과 토픽 문자열 'Sport/Soccer/Results'를 작성합니다. Version 7.1 큐 관리자에서 토픽 문자열 'Soccer/Results'를 사용하여 토픽 'Sport'에 대한 구독자는 발행물을 수신합니다.

Version 6.0 큐 관리자에서 토픽 문자열 'Soccer/Results'가 있는 스트림 'Sport'에 대한 구독자는 발행물을 수신합니다.

```
runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
  1 : define qlocal('Sport')
AMQ8006: WebSphere MQ queue created.
define topic('Sport') topicstr('Sport')
  2 : define topic('Sport') topicstr('Sport')
AMQ8690: WebSphere MQ topic created.
alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
  3 : alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport',
'SYSTEM.BROKER.DEFAULT.STREAM', 'SYSTEM.BROKER.ADMIN.STREAM')
AMQ8551: WebSphere MQ namelist changed.
```

참고: 이름 목록 오브젝트에 있는 기존 이름 외에도 추가 중인 새 이름을 **alter namelist** 명령에 제공해야 합니다.

다음에 수행할 작업

스트림에 대한 정보는 계층에 있는 다른 브로커에게 전달됩니다.

브로커가 Version 6.0이면, Version 6.0 브로커처럼 관리하십시오. 즉, 스트림 큐를 수동으로 작성하거나 필요한 경우 브로커가 스트림 큐를 동적으로 작성하게 하는 것 중에 선택할 수 있습니다. 큐는 모델 큐 정의 `SYSTEM.BROKER.MODEL.STREAM`(를) 기반으로 합니다.

브로커가 Version 7.1인 경우 계층에 있는 각 Version 7.1 큐 관리자를 수동으로 구성해야 합니다.

스트림 삭제

IBM WebSphere MQ Version 7.1 이상 큐 관리자에서 스트림을 삭제할 수 있습니다.

시작하기 전에

큐잉된 발행/구독은 IBM WebSphere MQ Version 7.1에서는 더 이상 사용되지 않습니다.

스트림을 삭제하기 전에 스트림에 대해 남아 있는 구독이 없는지 확인하고 스트림을 사용하는 모든 애플리케이션을 일시정지해야 합니다. 발행물이 계속해서 삭제된 스트림으로 플로우되는 경우 시스템을 깨끗하게 작동되는 상태로 복원하려면 많은 관리 노력이 필요합니다.

이 태스크 정보

연결된 Version 6.0 큐 관리자에서 스트림 삭제에 대한 지시사항은 [스트림 삭제 \(v6.0 문서의ps11870 .htm\)](#)를 참조하십시오.

프로시저

1. 이 스트림을 호스팅하는 모든 연결된 브로커를 찾으십시오.
2. 모든 브로커에서 스트림에 대한 모든 구독을 취소하십시오.
3. 이름 목록 SYSTEM.QPUBSUB.QUEUE.NAMELIST에서 큐(스트림과 같은 이름을 가진 큐)를 제거하십시오.
4. 스트림과 이름이 같은 큐에서 모든 메시지를 삭제하거나 영구 제거하십시오.
5. 스트림과 이름이 같은 큐를 삭제하십시오.
6. 연관된 토픽 오브젝트를 삭제하십시오.

다음에 수행할 작업

1. 스트림을 호스트하는 다른 모든 연결된 Version 7.1 이상 큐 관리자에서 단계 3 - 5를 반복하십시오.
2. 연결된 다른 모든 Version 6.0 이전 큐 관리자에서 스트림을 제거하십시오.

구독 지점 추가

migmbbrk에 의해 IBM WebSphere MQ Event Broker 또는 IBM WebSphere MQ Message Broker에서 마이그레이션되지 않은 구독 지점을 추가하는 방법입니다. 새 구독 지점을 사용하여 IBM WebSphere MQ Event Broker 또는 IBM WebSphere MQ Message Broker에서 마이그레이션한 기존의 큐잉된 발행/구독 애플리케이션을 확장하십시오.

시작하기 전에

1. IBM WebSphere MQ Event Broker 및 IBM WebSphere MQ Message Broker Version 6.0에서 IBM WebSphere MQ Version 7.1로의 마이그레이션을 완료하십시오.
2. SYSTEM.QPUBSUB.SUBPOINT.NAMELIST에서 구독 지점이 이미 정의되지 않았는지 검사하십시오.
3. 구독 지점과 이름이 같은 토픽 오브젝트 또는 토픽 문자열이 있는지 검사하십시오.

이 태스크 정보

기존 IBM WebSphere MQ Event Broker 애플리케이션은 구독 지점을 사용합니다. 새 IBM WebSphere MQ Version 7.1 애플리케이션은 subscription 지점을 사용하지 않지만, subscription 지점 마이그레이션 메커니즘을 사용하여 수행하는 기존 애플리케이션과 상호 운영할 수 있습니다.

구독 지점이 마이그레이션 시점에서 사용 중이 아니었던 경우 구독 지점은 **migmbbrk**에 의해 마이그레이션되지 않았을 수 있습니다.

IBM WebSphere MQ Event Broker에서 마이그레이션된 기존 큐잉된 발행/구독 프로그램에 구독 지점을 추가할 수 있습니다.

등록 지점은 IBM WebSphere MQ Version 6.0 또는 이전 버전에서 마이그레이션된 MQRFH1 헤더를 사용하는 큐된 발행/구독 프로그램에서 작동하지 않습니다.

IBM WebSphere MQ Version 7.1을 위해 작성된 통합 발행/구독 애플리케이션을 사용하기 위해 구독 지점을 추가할 필요는 없습니다.

프로시저

1. 구독 지점의 이름을 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST에 추가하십시오.
 - z/OS에서 **NLTYPE**은 기본값 NONE입니다.
 - 같은 발행/구독 토플로지에 연결되어 있는 모든 큐 관리자에서 이 단계를 반복하십시오.
2. 구독 지점의 이름과 일치하는 토플 문자열을 포함하는 토플 오브젝트를 구독 지점의 이름을 부여하여 추가하십시오.
 - 구독 지점이 클러스터에 있는 경우 토플 오브젝트를 클러스터 토플 호스트의 클러스터 토플 오브젝트로 추가하십시오.
 - 구독 지점의 이름과 같은 토플 문자열을 포함하는 토플 오브젝트가 있을 경우 기존 토플 오브젝트를 사용하십시오. 기존 토플을 재사용하는 구독 지점의 결과에 대해 이해해야 합니다. 기존 토플이 기존 애플리케이션의 일부인 경우 두 개의 동일하게 이름 지정된 토플 간 충돌을 해결해야 합니다.
 - 구독 지점과 같은 이름, 다른 토플 문자열을 갖는 토플 오브젝트가 있을 경우 다른 이름을 갖는 토플을 작성하십시오.
3. **Topic** 속성 WILDCARD를 값 BLOCK으로 설정하십시오.
 - # 또는 *는 구독 지점에 대한 와일드카드 구독을 차단합니다([와일드카드 및 서브스크립션 포인트 참조](#)).
4. 토플 오브젝트에서 필요한 모든 속성을 설정하십시오.

예

이 예는 두 개의 구독 지점인 USD와 GBP를 추가하는 **runmqsc** 명령 파일을 보여줍니다.

```
DEFINE TOPIC(USD) TOPICSTR(USD)
DEFINE TOPIC(GBP) TOPICSTR(GBP) WILDCARD(BLOCK)
ALTER NL(SYSTEM.QPUBSUB.SUBPOINT.NAMELIST) NAMES(SYSTEM.BROKER.DEFAULT.SUBPOINT, USD, GBP)
```

참고:

1. **ALTER** 명령을 사용하여 추가되는 구독 지점의 목록에 기본 구독 지점을 포함시키십시오. **ALTER**는 이름 목록에서 기존 이름을 삭제합니다.
2. 이름 목록을 대체하기 전에 토플을 정의하십시오. 큐 관리자는 큐 관리자가 시작할 때 및 이름 목록이 대체될 때에만 이름 목록을 검사합니다.

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](#)

[WebSphere 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.

큐 관리자 클러스터 구성

클러스터가 작업하는 방법, 클러스터 구성을 설계하는 방법을 찾고 단순한 클러스터를 설정하는 방법의 예를 알려면 이 토픽의 링크를 사용하십시오.

시작하기 전에

클러스터링 개념 소개는 다음 주제를 참조하십시오.

- [클러스터 작동 방법](#)
- [146 페이지의 『클러스터링과 분산 큐잉의 비교』](#)
- [148 페이지의 『클러스터의 컴포넌트』](#)

큐 관리자 클러스터를 설계할 때 몇 가지를 결정해야 합니다. 먼저 클러스터에서 클러스터 정보의 전체 저장소를 보유할 큐 관리자를 결정해야 합니다. 작성한 큐 관리자가 클러스터에서 작업할 수 있습니다. 이 용도로 임의의 수의 큐 관리자를 선택할 수 있지만 이상적인 수는 2입니다. 전체 저장소를 보유할 큐 관리자 선택에 대한 정보는 [160 페이지의 『전체 저장소를 보유할 클러스터 큐 관리자 선택 방법』](#)의 내용을 참조하십시오.

클러스터 설계에 대한 자세한 정보는 다음 주제를 참조하십시오.

- [161 페이지의 『클러스터 조직화』](#)
- [162 페이지의 『클러스터 이름 지정 규칙』](#)
- [163 페이지의 『중첩 클러스터』](#)

예

가능한 최소 클러스터는 두 개의 큐 관리자만 포함합니다. 이 경우에는 두 큐 관리자가 모두 전체 저장소를 보유합니다. 클러스터를 설정하기 위해 몇 개의 정의만 필요하며 각 큐 관리자의 자용도가 아직은 높습니다.

[145 페이지의 그림 21](#)에서는 두 개의 큐 관리자가 QM1 및 QM2인 DEMOCLSTR 클러스터를 보여줍니다.

DEMOCLSTR

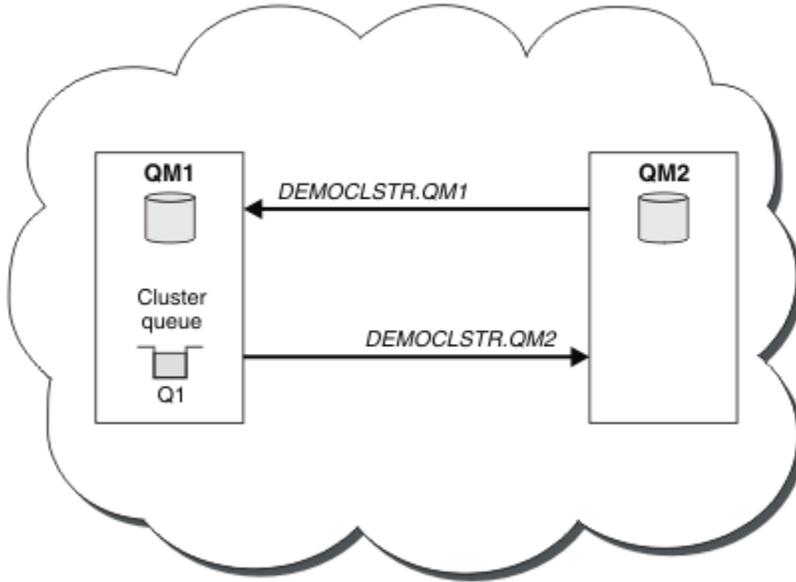


그림 21. 두 큐 관리자의 작은 클러스터

- 큐 관리자는 LONDON 및 NEWYORK 같은 긴 이름을 갖고 있습니다. 동일한 이름이 고급 및 워크로드 밸런싱 태스크에서 사용됩니다. z/OS에 대해 IBM WebSphere MQ에서는 큐 관리자 이름이 4자로 제한됩니다.
- 큐 관리자의 이름은 각 큐 관리자가 별도의 시스템에 위치함을 암시합니다. 모든 큐 관리자를 동일한 시스템에 갖고 이들 태스크를 수행할 수도 있습니다.
- 태스크는 **MQSC** 명령을 사용하여 시스템 관리자가 입력하는 대로 IBM WebSphere MQ 스크립트 명령을 사용합니다. 더 쉬운 IBM WebSphere MQ 탐색기 사용을 포함하여 명령을 입력하는 다른 방법이 있습니다. WebSphere MQ 스크립트 명령 사용의 요점은 IBM WebSphere MQ 명령을 태스크에서 사용하는 내용을 설명합니다.

비슷한 클러스터 예 설정에 대한 지시사항은 168 페이지의 『새 클러스터 설정』의 내용을 참조하십시오.

다음에 수행할 작업

클러스터 구성 및 작업에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 165 페이지의 『클러스터에서 통신 설정』
- 167 페이지의 『IBM WebSphere MQ 클러스터 관리』
- 227 페이지의 『메시지를 클러스터로(부터) 라우팅』
- 240 페이지의 『워크로드 관리에 클러스터 사용』

클러스터를 구성하는 데 도움이 되는 자세한 정보는 164 페이지의 『클러스터링 팁』의 내용을 참조하십시오.

관련 개념

[클러스터](#)

액세스 제어 및 다중 클러스터 전송 큐

애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드는 클러스터 큐에 대해 원격으로 검사, SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사 또는 클러스터 큐나 클러스터 큐 관리자의 로컬 프로파일에 대해 검사입니다.

IBM WebSphere MQ는 사용자가 리모트 큐에 메시지를 넣을 권한이 있는지에 대한 로컬 또는 로컬 및 리모트 검사의 선택사항을 제공합니다. 일반 IBM WebSphere MQ 애플리케이션은 로컬 검사만을 사용하며 로컬 큐 관리자에 대한 액세스 검사를 신뢰하는 리모트 큐 관리자에 의존합니다. 원격 검사를 사용하지 않으면 원격 메시지 채

널 프로세스 권한이 있는 대상 큐에 메시지를 넣습니다. 원격 검사를 사용하려면 수신 채널의 넣기 권한을 컨텍스트 보안에 설정해야 합니다.

애플리케이션이 여는 큐에 대한 로컬 검사가 수행됩니다. 분산 큐에서, 애플리케이션은 항상 리모트 큐 정의를 열며 액세스 검사는 리모트 큐 정의에 대해 수행됩니다. 전체 라우팅 헤더에 메시지를 넣는 경우에는 전송 큐에 대해 검사합니다. 애플리케이션이 로컬 큐 관리자에 있지 않은 클러스터 큐를 여는 경우 검사할 로컬 오브젝트가 없습니다. 클러스터 전송 큐, SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대한 액세스 제어 검사가 수행됩니다. 다중 클러스터 전송 큐에서라도 리모트 클러스터 큐에 대한 Version 7.5에서 로컬 액세스 제어 검사는 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 수행됩니다.

로컬 또는 원격 검사의 선택은 두 가지 극단 상황의 선택입니다. 원격 검사는 세분화되어 있습니다. 모든 사용자가 클러스터 큐에 넣으려면 클러스터의 모든 큐 관리자에 대한 액세스 제어 프로파일을 보유해야 합니다. 로컬 검사는 대략적입니다. 모든 사용자가 연결된 큐 관리자의 클러스터 전송 큐에 대한 액세스 제어 프로파일만 있으면 됩니다. 이 프로파일로 큐 관리자는 클러스터의 어느 큐 관리자에 있는 클러스터 큐에나 메시지를 넣을 수 있습니다.

Version 7.1 이후로는 관리자가 클러스터 큐에 대한 액세스 제어를 다른 방식으로 설정할 수 있습니다.

setmqaut 명령을 사용하여 클러스터의 큐 관리자에 클러스터 큐에 대한 보안 프로파일을 작성할 수 있습니다. 큐 이름만 지정한 채 리모트 클러스터 큐를 로컬로 여는 경우 프로파일이 적용됩니다. 리모트 큐 관리자에 대한 프로파일을 설정할 수도 있습니다. 그렇게 되면 큐 관리자가 완전한 이름을 제공하여 클러스터 큐를 여는 사용자의 프로파일을 검사할 수 있습니다.

새 프로파일은 큐 관리자 스탠자, **ClusterQueueAccessControl**을 RQMName으로 변경하는 경우에만 작용합니다. 기본값은 Xmitq입니다. 기존 애플리케이션이 클러스터 큐를 사용하는 모든 클러스터 큐에 대한 프로파일을 작성해야 합니다. 프로파일을 작성하지 않고 스탠자를 RQMName으로 변경할 경우 애플리케이션이 실패할 수 있습니다.

팁: Version 7.1에서 검사하는 클러스터 큐 액세스에 작성된 변경은 리모트 큐잉에 적용되지 않습니다. 액세스 검사는 여전히 로컬 정의에 대해 수행됩니다. 변경사항은 동일한 접근법에 따라 클러스터 큐와 클러스터 토픽에 대한 액세스 검사를 구성할 수 있음을 의미합니다.

관련 개념

258 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

클러스터링과 분산 큐잉의 비교

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

클러스터를 사용하지 않는 경우 큐 관리자는 독립적이며 분산 큐잉을 사용하여 통신합니다. 한 큐 관리자가 다른 큐 관리자에 메시지를 보내야 하는 경우 다음을 정의해야 합니다.

- 전송 큐
- 리모트 큐 관리자로 향하는 채널

147 페이지의 그림 22에서는 분산 큐잉에 필요한 컴포넌트를 보여줍니다.

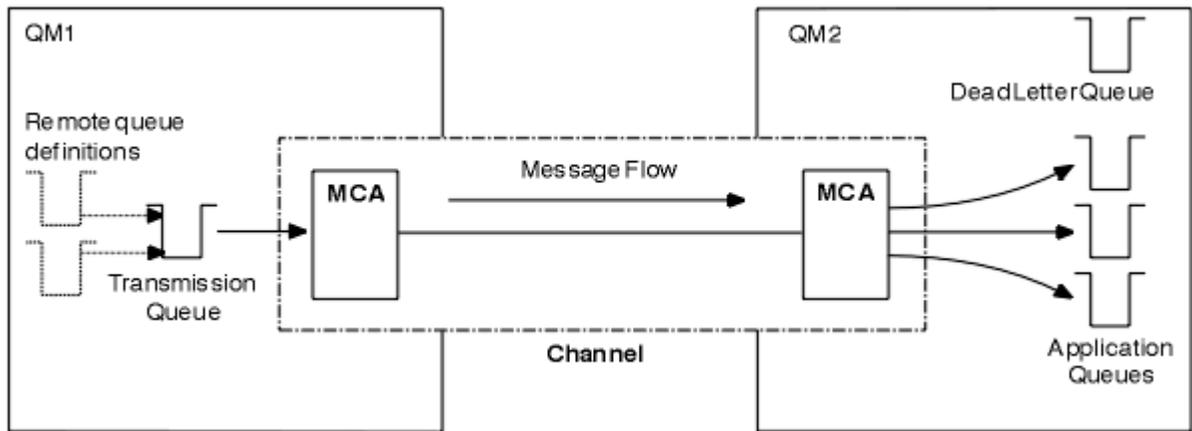


그림 22. 분산 큐잉

클러스터의 큐 관리자를 그룹화할 경우 클러스터의 다른 모든 큐 관리자가 큐 관리자의 큐를 사용할 수 있습니다. 어느 큐 관리자나 명확한 정의 없이도 동일한 클러스터의 다른 큐 관리자에 메시지를 보낼 수 있습니다. 각 목적지에 대한 채널 정의, 리모트 큐 정의 또는 전송 큐는 제공하지 않습니다. 클러스터의 모든 큐 관리자마다 클러스터의 다른 큐 관리자에 메시지를 전송할 수 있는 단일 전송 큐가 있습니다. 클러스터의 각 큐 관리자는 다음 항목만 정의하면 됩니다.

- 메시지를 수신할 하나의 클러스터 수신자 채널
- 자신을 소개하고 클러스터에 대해 배우는 하나의 클러스터 송신자 채널

클러스터 대 분산 큐잉 설정을 위한 정의

각각 2개의 큐가 포함된 4개의 큐 관리자를 보여주는 147 페이지의 [그림 23](#)를 살펴보십시오. 분산 큐잉을 사용하여 이러한 큐 관리자를 연결하기 위해 필요한 정의의 수를 고려하십시오. 클러스터로서 동일한 네트워크를 설정하는 데 필요한 정의의 수를 비교하십시오.

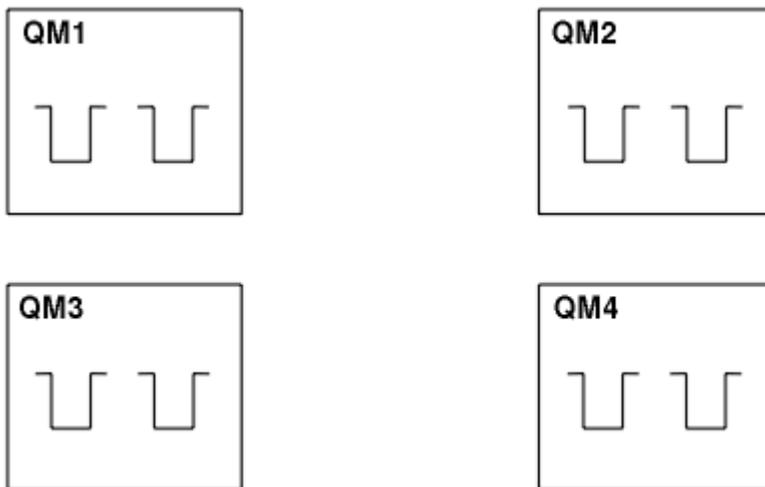


그림 23. 네 개의 큐 관리자 네트워크

분산 큐잉을 사용하여 네트워크를 설정하기 위한 정의

분산 큐잉을 사용하여 147 페이지의 [그림 22](#)에 표시된 네트워크를 설정하려면 다음과 같은 정의가 필요할 수 있습니다.

표 24. 분산 큐잉에 대한 정의		
설명	큐 관리자당 개수	총 수
다른 모든 큐 관리자에 메시지를 보낼 채널에 대한 송신자 채널 정의	3	12
다른 모든 큐 관리자로부터 메시지를 받을 채널에 대한 수신자 채널 정의	3	12
다른 모든 큐 관리자로부터 향하는 전송 큐에 대한 전송 큐 정의	3	12
각 로컬 큐에 대한 로컬 큐 정의	2	8
이 큐 관리자가 메시지를 넣을 각 리모트 큐에 대한 리모트 큐 정의	6	24

일반 수신자 채널 정의를 사용하여 이 정의의 수를 줄일 수 있습니다. 최대 정의 수는 각 큐 관리자당 17개를 초과할 수 없으며 이 네트워크에서는 총 68입니다.

클러스터를 사용하여 네트워크를 설정하기 위한 정의

클러스터를 사용하여 [147 페이지의 그림 22](#)에 표시된 네트워크를 설정하려면 다음과 같은 정의가 필요할 수 있습니다.

표 25. 클러스터링에 대한 정의		
설명	큐 관리자당 개수	총 수
저장소 큐 관리자에 메시지를 보낼 채널에 대한 클러스터 송신자 채널 정의	1	4
클러스터의 다른 큐 관리자로부터 메시지를 받을 채널에 대한 클러스터 수신자 채널 정의	1	4
각 로컬 큐에 대한 로컬 큐 정의	2	8

큐 관리자의 이 클러스터(전체 저장소 2개 포함)를 설정하려면 큐 관리자당 4개의 정의가 필요하며 총 16개의 정의입니다. 두 개 큐 관리자의 경우에는 클러스터의 전체 저장소 큐 관리자로 지정하기 위해 큐 관리자 정의를 대체해야 합니다.

하나의 CLUSSDR 및 하나의 CLUSRCVR 채널 정의만 필요합니다. 클러스터가 정의될 때 다른 저장소 큐 관리자를 방해하지 않고 큐 관리자(저장소 큐 관리자 제외)를 추가하거나 제거할 수 있습니다.

클러스터를 사용하면 여러 큐 관리자를 포함한 네트워크를 설정하는 데 필요한 정의 수가 감소합니다.

필요한 정의 수가 더 적으므로 다음과 같은 오류가 발생할 위험도 낮습니다.

- 오브젝트 이름이 항상 일치합니다(예를 들어, 송신자-수신자 쌍의 채널 이름).
- 채널 정의에 지정된 전송 큐 이름이 항상 올바른 전송 큐 정의 또는 전송 큐 정의에 지정된 전송 큐 이름과 일치합니다.
- QREMOTE 정의가 항상 리모트 큐 관리자의 올바른 큐를 가리킵니다.

클러스터가 일단 설정되면 다른 큐 관리자에서 시스템 관리 작업을 수행할 필요 없이 클러스터의 한 큐 관리자에서 다른 큐 관리자로 클러스터 큐를 이동시킬 수 있습니다. 채널, 리모트 큐 또는 전송 큐 정의를 삭제하거나 수정하는 것을 잊어버릴 가능성이 없습니다. 기존 네트워크를 방해하지 않고 새 큐 관리자를 클러스터에 추가할 수 있습니다.

클러스터의 컴포넌트

클러스터는 큐 관리자, 클러스터 저장소, 클러스터 채널 및 클러스터 큐로 구성됩니다.

각 클러스터 컴포넌트에 대한 정보는 다음 하위 주제를 참조하십시오.

관련 개념

클러스터

146 페이지의 『클러스터링과 분산 큐잉의 비교』

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

167 페이지의 『IBM WebSphere MQ 클러스터 관리』

IBM WebSphere MQ 클러스터를 작성, 확장 및 유지보수할 수 있습니다.

관련 태스크

144 페이지의 『큐 관리자 클러스터 구성』

클러스터가 작업하는 방법, 클러스터 구성을 설계하는 방법을 찾고 단순한 클러스터를 설정하는 방법의 예를 알려면 이 토픽의 링크를 사용하십시오.

168 페이지의 『새 클러스터 설정』

다음 지시사항에 따라 클러스터 예 설정합니다. 개별 지시사항에서는 TCP/IP, LU 6.2 및 단일 전송 큐 또는 다중 전송 큐에서의 클러스터 설정을 설명합니다. 한 큐 관리자에서 다른 큐 관리자로 메시지를 송신하여 클러스터가 작동하는지 테스트합니다.

클러스터 저장소

저장소는 클러스터의 멤버인 큐 관리자에 대한 정보의 컬렉션입니다.

저장소 정보에는 큐 관리자 이름, 해당 위치, 해당 채널, 호스팅하고 있는 큐 및 기타 정보가 포함됩니다. 이 정보는 SYSTEM.CLUSTER.REPOSITORY.QUEUE라고 하는 큐에 메시지의 양식으로 저장됩니다. 큐는 기본 오브젝트 중 하나입니다. z/OS용 WebSphere MQ를 제외하고 WebSphere MQ 큐 관리자를 작성할 때 큐가 정의됩니다.

일반적으로, 클러스터에서 두 개의 큐 관리자가 전체 저장소를 보유하고 있습니다. 나머지 큐 관리자 모두 부분 저장소를 보유하고 있습니다.

전체 저장소와 부분 저장소

클러스터에서 모든 큐 관리자에 대한 전체 정보 세트를 호스팅하는 큐 관리자에 전체 저장소가 있습니다. 클러스터의 다른 큐 관리자에는 전체 저장소에 있는 정보의 서브세트를 포함하는 부분 저장소가 있습니다.

부분 저장소에는 큐 관리자가 메시지를 교환해야 하는 큐 관리자에 대한 정보만 포함됩니다. 큐 관리자는 필요한 정보에 대한 업데이트를 요청합니다. 그러면 필요한 경우, 전체 저장소 큐 관리자가 큐 관리자에 새 정보를 보낼 수 있습니다. 많은 시간 동안, 부분 저장소에는 큐 관리자가 클러스터 내에서 수행하는 데 필요한 모든 정보가 포함됩니다. 큐 관리자에서 일부 추가 정보가 필요한 경우, 큐 관리자는 전체 저장소를 조회하여 자신의 부분 저장소를 업데이트합니다. 큐 관리자는 SYSTEM.CLUSTER.COMMAND.QUEUE라고 하는 큐를 사용하여 저장소에 대한 업데이트를 요청하고 수신합니다. 이 큐는 기본 오브젝트 중 하나입니다.

클러스터 큐 관리자

클러스터 큐 관리자는 클러스터의 멤버인 큐 관리자입니다.

큐 관리자는 둘 이상의 클러스터의 멤버가 될 수 있습니다. 각 클러스터 큐 관리자의 이름은 멤버인 모든 클러스터에서 고유해야 합니다.

클러스터 큐 관리자는 큐를 호스팅할 수 있습니다. 이 큐는 클러스터에서 다른 큐 관리자에 광고합니다. 클러스터 큐 관리자가 큐를 호스트하거나 통지할 필요는 없습니다. 클러스터 큐 관리자는 클러스터에 메시지를 제공하고 통지된 큐가 아닌 클러스터에 명시적으로 지시되는 응답만 수신할 수 있습니다.

z/OS용 WebSphere MQ에서 클러스터 큐 관리자는 큐 공유 그룹의 구성원이 될 수 있습니다. 이러한 경우, 클러스터 큐 관리자는 샘플 큐 공유 그룹의 다른 큐 관리자와 해당되는 큐 정의를 공유합니다.

클러스터 큐 관리자는 자율적입니다. 클러스터 큐 관리자는 정의하는 채널 및 큐에 대해 전체 제어를 가지고 있습니다. 해당되는 정의는 다른 큐 관리자(동일한 큐 공유 그룹에 있는 큐 관리자가 아닌 다른)에서 수행할 수 없습니다. 저장소 큐 관리자는 클러스터의 다른 큐 관리자에서 정의를 제어하지 않습니다. 필요할 때 사용하기 위해 모든 전체 정의의 세트를 보유하고 있습니다. 클러스터는 큐 관리자의 연합입니다.

클러스터 큐 관리자에서 정의를 작성하거나 변경한 후, 정보는 전체 저장소 큐 관리자로 보내집니다. 클러스터의 다른 저장소는 나중에 업데이트됩니다.

전체 저장소 큐 관리자

전체 저장소 큐 관리자는 클러스터의 자원에 대한 전체 표시를 보유하는 클러스터 큐 관리자입니다. 사용 가능성을 보증하기 위해, 각 클러스터에서 두 개 이상의 전체 저장소 큐 관리자를 설정하십시오. 전체 저장소 큐 관리자는 클러스터의 다른 큐 관리자가 보낸 정보를 수신하고 해당되는 저장소를 업데이트합니다. 둘 다 클러스터에 대한 새 정보로 최신 상태를 유지하도록 하기 위해 서로 메시지를 보냅니다.

큐 관리자 및 저장소

모든 클러스터에는 클러스터에 있는 큐 관리자, 큐 및 채널에 대한 정보의 전체 저장소를 보유하는 최소 하나의 (두 개가 선호됨) 큐 관리자가 있습니다. 이 저장소는 또한 클러스터의 다른 큐 관리자로부터의 정보에 대한 업데이트 요청을 포함합니다.

다른 큐 관리자는 각각 통신해야 하는 큐 관리자와 큐 서브세트에 대한 정보를 포함하는 부분 저장소를 보유합니다. 큐 관리자는 먼저 다른 큐 또는 큐 관리자에 액세스해야 하는 경우 조회를 작성하여 부분 저장소를 빌드합니다. 큐 관리자는 해당 큐 또는 큐 관리자에 관한 새 정보를 알리도록 요청합니다.

각 큐 관리자는 SYSTEM.CLUSTER.REPOSITORY.QUEUE라고 하는 큐에 메시지의 저장소 정보를 저장합니다. 큐 관리자는 SYSTEM.CLUSTER.COMMAND.QUEUE라고 하는 큐에서 메시지의 저장소 정보를 교환합니다.

클러스터에 조인하는 각 큐 관리자는 클러스터 송신자 CLUSSDR 채널을 저장소 중 하나에 대해 정의합니다. 클러스터의 다른 큐 관리자가 전체 저장소를 보유한다고 즉시 학습합니다. 그때부터, 큐 관리자는 저장소에서 정보를 요청할 수 있습니다. 큐 관리자가 정보를 선택된 저장소로 보낼 때, 하나의 다른 저장소(있는 경우)로도 정보를 보냅니다.

전체 저장소는 호스팅하는 큐 관리자가 링크되는 큐 관리자 중 하나에서 새 정보를 수신할 때 업데이트됩니다. 새 정보는 또한 저장소 큐 관리자가 서비스 외부에 있는 경우 지연되는 위험을 줄이기 위해 다른 저장소로 보냅니다. 모든 정보는 두 번 보내므로, 저장소는 중복을 버려야 합니다. 정보의 각 항목은 저장소가 중복을 식별하기 위해 사용하는 순서 번호를 전달합니다. 모든 저장소는 메시지를 교환하여 서로 맞춥니다.

클러스터 큐

클러스터 큐는 클러스터 큐 관리자에 의해 호스팅되며 클러스터의 다른 큐 관리자가 사용할 수 있는 큐입니다.

클러스터 큐를 큐가 호스트되는 클러스터 큐 관리자의 로컬 큐로 정의하십시오. 큐가 속한 클러스터의 이름을 지정하십시오. 다음 예는 CLUSTER 옵션과 함께 클러스터 큐를 정의하는 **runmqsc** 명령을 보여줍니다.

```
DEFINE QLOCAL(Q1) CLUSTER(SALES)
```

클러스터 큐 정의는 클러스터의 다른 큐 관리자에 통지됩니다. 다른 큐 관리자는 해당하는 리모트 큐 정의 없이도 클러스터 큐에 메시지를 넣을 수 있습니다. 클러스터 큐는 클러스터 이름 목록을 사용하여 둘 이상의 클러스터에 통지될 수 있습니다.

큐가 통지되면 클러스터의 큐 관리자가 해당 큐에 메시지를 넣을 수 있습니다. 메시지를 넣으려면 큐 관리자가 전체 저장소에서 큐가 호스팅되고 있는 위치를 찾아야 합니다. 그런 다음 메시지에 몇 가지 라우팅 정보를 추가하고 클러스터 전송 큐에 메시지를 넣습니다.

클러스터 큐는 IBM WebSphere MQ for z/OS에서 큐 공유 그룹의 멤버가 공유하는 큐일 수 있습니다.

바인딩

둘 이상의 큐 관리자가 같은 클러스터 큐의 인스턴스를 호스팅하는 클러스터를 작성할 수 있습니다. 순서에 있는 모든 메시지가 큐의 동일한 인스턴스로 송신되는지 확인하십시오. MQOPEN 호출 시 MQOO_BIND_ON_OPEN 옵션을 사용하여 일련의 메시지를 특정 큐에 바인딩할 수 있습니다.

클러스터 전송 큐

z/OS을 제외하고, 큐 관리자는 다중 전송 큐에 대해 클러스터에 있는 다른 큐 관리자를 위해 메시지를 저장할 수 있습니다. 서로 다른 두 가지 방법으로 다중 클러스터 전송 큐에서 메시지를 저장하도록 큐 관리자를 구성할 수 있습니다. 큐 관리자 속성 DEFCLXQ를 CHANNEL로 설정하면 클러스터 송신자 채널마다 다른 클러스터 전송 큐가 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE에서 자동으로 작성됩니다. 하나 이상의 클러스터 송신자 채

널과 일치하도록 CLCHNAME 전송 큐 옵션을 설정할 경우 큐 관리자가 일치하는 채널에 대한 메시지를 해당 전송 큐에 저장할 수 있습니다.



주의: 제품의 이전 버전에서 업그레이드된 큐 관리자와 함께 전송 SYSTEM.CLUSTER.TRANSMIT.QUEUES 을 사용하는 경우, SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 에 공유/공유하기 옵션이 공유로 설정되어 있는지 확인하십시오.

다른 큐 관리자에 있는 클러스터 큐에 대한 메시지는 송신되기 전에 클러스터 전송 큐에 배치됩니다. 클러스터 송신자 채널은 클러스터 전송 큐에서 다른 큐 관리자의 클러스터 수신자 채널로 메시지를 전송합니다. 기본적으로 한 시스템에서 정의한 클러스터 전송 큐는 다른 클러스터 큐 관리자로 전송할 모든 메시지를 보유하고 있습니다. 이 큐를 SYSTEM.CLUSTER.TRANSMIT.QUEUE라고 합니다. 클러스터에 속한 큐 관리자는 이 클러스터 전송 큐에 있는 메시지를 같은 클러스터에 있는 다른 큐 관리자에게 송신할 수 있습니다.

단일 SYSTEM.CLUSTER.TRANSMIT.QUEUE 큐에 대한 정의는 z/OS를 제외한 모든 큐 관리자에서 기본적으로 작성됩니다.

z/OS가 아닌 다른 플랫폼에서는, 다중 전송 큐를 사용하여 메시지를 클러스터된 다른 큐 관리자로 전송하도록 큐 관리자를 구성할 수 있습니다. 수동으로 추가 클러스터 전송 큐를 정의하거나 큐 관리자가 이 큐를 자동으로 작성하게 할 수 있습니다.

큐 관리자가 자동으로 큐를 작성하게 하려면 큐 관리자 속성 DEFCLXQ를 SCTQ에서 CHANNEL로 변경하십시오. 결과적으로 이 큐 관리자는 작성되는 각 클러스터 송신자 채널에 대해 개별 클러스터 전송 큐를 작성합니다. 전송 큐는 모델 큐 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE로부터 영구적 동적 큐로 작성됩니다. 각 영구적 동적 큐의 이름은 SYSTEM.CLUSTER.TRANSMIT.ChannelName입니다. 각 영구적 동적 클러스터 전송 큐와 연관된 클러스터 송신자 채널의 이름은 로컬 전송 큐 속성 CLCHNAME에서 설정됩니다. 리모트 클러스터된 큐 관리자에 대한 메시지는 SYSTEM.CLUSTER.TRANSMIT.QUEUE가 아니라 연관된 클러스터 송신자 채널의 영구적 동적 클러스터 전송 큐에 배치됩니다.

클러스터 전송 큐를 수동으로 작성하려면 USAGE 속성이 XMITQ로 설정되어 있고 CLCHNAME 속성은 하나 이상의 클러스터-송신자 채널로 해석되는 일반 채널 이름으로 설정되어 있는 로컬 큐를 작성하십시오 (ClusterChannelName 참조). 클러스터 전송 큐를 수동으로 작성하는 경우 이 전송 큐를 단일 클러스터 송신자 채널과 연관시킬지 또는 다중 클러스터 송신자 채널과 연관시킬지 선택할 수 있습니다. CLCHNAME 속성은 일반 이름이며 이는 이름에 여러 개의 와일드카드 문자 "*"를 배치할 수 있음을 의미합니다.

큐 관리자를 전체 저장소에 연결하기 위해 수동으로 작성하는 초기 클러스터 송신자 채널을 제외한 클러스터 송신자 채널은 자동으로 작성됩니다. 이러한 클러스터 송신자 채널은 클러스터 큐 관리자에 전송할 메시지가 있는 경우에 자동으로 작성됩니다. 목적지 큐 관리자의 해당 특정 클러스터에 대한 클러스터 메시지를 수신하는 클러스터 수신자 채널의 이름과 같은 이름으로 작성됩니다.

클러스터 수신자 채널에 대한 이름 지정 규칙을 따르는 경우 여러 종류의 클러스터 메시지를 서로 다른 전송 큐로 필터링하는 CLCHNAME에 일반 값을 정의할 수 있습니다. 예를 들어, ClusterName.QmgrName의 클러스터-수신자 채널에 대한 이름 지정 규칙을 따르면, 일반 이름 ClusterName.*는 다른 클러스터의 메시지를 다른 전송 큐로 필터링합니다. 전송 큐를 수동으로 정의하고 각 전송 큐에서 CLCHNAME을 ClusterName.*로 설정해야 합니다.

클러스터 송신자 채널에 대한 클러스터 전송 큐의 연관을 변경하면 즉시 적용되지 않습니다. 클러스터 송신자 채널이 서비스를 제공하고 있는 현재 연관된 전송 큐가 클러스터 송신자 채널에서 전송 중인 메시지를 포함할 수 있습니다. 현재 연관된 전송 큐에 클러스터 송신자 채널이 처리 중인 메시지가 없는 경우에만 큐 관리자가 다른 전송 큐에 대한 클러스터 송신자 채널의 연관을 변경할 수 있습니다. 이러한 상황은 클러스터 송신자 채널이 처리할 메시지가 전송 큐에 남아 있지 않은 경우 또는 메시지 처리가 일시중단되어 클러스터 송신자 채널에 "인플라이트" 메시지가 없는 경우에 발생할 수 있습니다. 이러한 상황이 발생하면 클러스터 송신자 채널의 미처리된 메시지가 새로 연관된 전송 큐에 전송되고 클러스터 송신자 채널의 연관이 변경됩니다.

클러스터 전송 큐로 해석되는 리모트 큐 정의를 작성할 수 있습니다. 이 정의에서 큐 관리자 QMX는 로컬 큐 관리자와 같은 클러스터에 있으며 전송 큐 QMX가 없습니다.

```
DEFINE QREMOTE(A) RNAME(B) RQMNAME(QMX)
```

큐 이름 해석 중에 클러스터 전송 큐는 기본 전송 큐보다 우선합니다. A에 배치되는 메시지는 클러스터 전송 큐에 저장된 후 QMX의 리모트 큐 B로 송신됩니다.

큐 관리자는 클러스터에 속하지 않는 다른 큐 관리자와 통신할 수도 있습니다. 분산 큐잉 환경에서와 동일한 방식으로 다른 큐 관리자에 대한 채널 및 전송 큐를 정의해야 합니다.

참고: 애플리케이션은 클러스터 전송 큐로 해석되는 큐에 기록되어야 하고 클러스터 전송 큐에 직접적으로 기록되어서는 안 됩니다.

리모트 큐의 자동 정의

클러스터에 있는 큐 관리자는 클러스터의 리모트 큐에 대한 리모트 큐 정의가 필요하지 않습니다. 클러스터 큐 관리자는 전체 저장소에서 리모트 큐의 위치를 찾습니다. 메시지에 라우팅 정보를 추가하고 이 메시지를 클러스터 전송 큐에 넣습니다. WebSphere MQ 는 메시지를 전송할 수 있도록 리모트 큐 정의와 동등한 정의를 자동으로 작성합니다.

자동으로 작성된 리모트 큐 정의는 대체하거나 삭제할 수 없습니다. 그러나 `DISPLAY QUEUE runmqsc` 명령을 `CLUSINFO` 속성과 함께 사용하여 리모트 큐 관리자의 클러스터 큐를 포함한 모든 클러스터 큐 및 큐 관리자의 모든 로컬 큐를 볼 수 있습니다. 예를 들면, 다음과 같습니다.

```
DISPLAY QUEUE(*) CLUSINFO
```

관련 참조

[ClusterChannelName\(MQCHAR20\)](#)

클러스터 채널

클러스터의 큐 관리자에 클러스터 수신자 및 클러스터 송신자 채널을 정의해야 합니다. 특별한 고려사항이 전체 저장소에 적용됩니다.

클러스터 내에서, 클러스터 수신자 채널 정의 및 클러스터 송신자 채널 정의가 필요한 특수 유형의 채널에 있는 클러스터 큐 관리자에 메시지를 분배합니다.

클러스터-송신자 채널: CLUSSDR

클러스터의 각 큐 관리자에서 전체 저장소에 대한 클러스터 송신자 채널을 수동으로 정의하십시오. 클러스터 송신자 정의를 사용하면 큐 관리자가 클러스터와 처음 접속할 수 있습니다. 이 정의는 큐 관리자가 클러스터 정보를 송신하도록 우선적으로 선택하는 전체 저장소 큐 관리자의 이름을 지정합니다. 클러스터 송신자 채널은 큐 관리자의 상태에 대한 변경사항을 저장소에 알리는 데 사용됩니다. 예를 들어, 큐가 추가되거나 제거되는 경우입니다. 메시지 전송에도 사용됩니다.

전체 저장소 큐 관리자 자체에는 서로 가리키는 클러스터-송신자 채널이 있습니다. 이를 사용하여 서로 클러스터 상태 변경사항을 통신할 수 있습니다.

CLUSSDR 채널 정의가 가리키는 전체 저장소가 어느 것인지 중요하지 않습니다. 초기 접속이 이루어지면 필요한 경우 추가 클러스터 큐 관리자 오브젝트가 자동으로 정의됩니다. 큐 관리자는 클러스터 정보를 모든 전체 저장소로 송신하고 메시지를 모든 큐 관리자로 송신합니다.

전체 저장소 큐 관리자에서의 CLUSSDR 정의는 특별합니다. 전체 저장소에 의해 교환되는 모든 업데이트는 이 채널에서 독점적으로 플로우됩니다. 관리자는 전체 저장소의 네트워크를 명시적으로 제어합니다. 관리자는 모든 전체 저장소 큐 관리자에서 클러스터의 다른 모든 전체 저장소 큐 관리자로서의 CLUSSDR 채널을 정의해야 합니다. 관리자는 전체 저장소 큐 관리자에서 CLUSSDR 정의를 수동으로 작성하고 자동 정의되게 하지 않습니다.

클러스터 송신자 채널은 반드시 전체 저장소에 부분 저장소를 연결하거나 두 개의 전체 저장소를 함께 연결하는 경우 정의해야 합니다. 부분 저장소의 주소를 지정하는 CLUSSDR 채널을 수동으로 구성하거나 클러스터에 큐 관리자가 없는 경우 오류 메시지(예: AMQ9427 및 AMQ9428)가 발행됩니다.

이것이 때로는 일시적인 상황(예를 들면 전체 저장소의 위치를 수정할 경우)으로서 불가피할 수도 있지만, 올바른 정의는 이러한 오류의 발생을 막기 위해 가능한 한 빨리 삭제해야 합니다.

클러스터-수신자 채널: CLUSRCVR

클러스터 수신자 채널 정의는 클러스터 큐 관리자가 클러스터의 다른 큐 관리자에서 메시지를 수신할 수 있는 채널의 끝을 정의합니다.

클러스터-수신자 채널은 또한 로컬 저장소를 대상으로 하는 클러스터 정보에 대한 정보를 전달합니다. 큐 관리자는 클러스터 수신자 채널을 정의하여 메시지를 수신할 수 있는 다른 클러스터 큐 관리자를 표시합니다. 클러스터 큐 관리자마다 최소 하나의 클러스터-수신자 채널이 필요합니다.

CLUSRCVR 정의를 사용하면 다른 큐 관리자가 해당 CLUSSDR 채널 정의를 자동 정의할 수 있습니다.

관련 개념

153 페이지의 『클러스터 채널의 자동 정의』

큐 관리자가 메시지를 원격 대상으로 송신하려면 클러스터 송신자 채널의 정의가 필요합니다. 초기 CLUSSDR 및 CLUSRCVR 정의를 작성하여 클러스터에 큐 관리자를 도입한 후, WebSphere MQ가 필요한 경우 자동으로 클러스터 송신자 채널 정의를 작성합니다. 자동 정의 클러스터 송신자 채널을 수정할 수는 없습니다. 채널 자동 정의 엑시트를 사용하여 해당 작동을 수정할 수 있습니다.

클러스터 채널의 자동 정의

큐 관리자가 메시지를 원격 대상으로 송신하려면 클러스터 송신자 채널의 정의가 필요합니다. 초기 CLUSSDR 및 CLUSRCVR 정의를 작성하여 클러스터에 큐 관리자를 도입한 후, WebSphere MQ가 필요한 경우 자동으로 클러스터 송신자 채널 정의를 작성합니다. 자동 정의 클러스터 송신자 채널을 수정할 수는 없습니다. 채널 자동 정의 엑시트를 사용하여 해당 작동을 수정할 수 있습니다.

채널의 클러스터 송신자 끝과 클러스터 수신자 끝을 모두 정의한 경우, 채널이 시작됩니다. 자동 정의된 채널은 더 이상 필요하지 않을 때까지 활성 상태이며 정상적인 연결 해제 간격 규칙을 사용하여 종료됩니다.

자동 정의된 클러스터-송신기 채널은 수신 큐 관리자에서 대응하는 클러스터-수신기 채널 정의의 속성을 사용합니다. 수동으로 정의된 클러스터 송신자 채널이 있는 경우에도, 속성은 해당 클러스터 수신자 정의와 일치하도록 자동으로 수정됩니다. 예를 들어, CONNAME 매개변수에서 포트 번호를 지정하지 않고 CLUSRCVR을 정의하고 포트 번호를 지정하는 CLUSSDR을 수동으로 정의한다고 가정합니다. 자동 정의된 CLUSSDR이 수동으로 정의된 CLUSSDR을 대체하는 경우 포트 번호(CLUSRCVR에서 가져옴)가 공백이 됩니다. 기본 포트 번호를 사용하고 채널이 실패합니다.

자동 정의 클러스터 송신자 정의를 수정할 수 없습니다.

DISPLAY CHANNEL **runmqsc** 명령을 사용해서는 자동으로 정의된 채널을 볼 수 없습니다. 자동 정의 채널을 확인하려면 다음 명령을 사용하십시오.

```
DISPLAY CLUSQMGR(qMgrName)
```

작성한 CLUSRCVR 채널 정의에 해당하는 자동 정의된 CLUSSDR 채널의 상태를 표시하려면 다음 명령을 사용하십시오.

```
DISPLAY CHSTATUS(channelName)
```

클러스터 송신자 채널 또는 클러스터 수신자 채널을 사용자 정의하여 사용자 엑시트 프로그램을 작성하려는 경우, WebSphere MQ 채널 자동 정의 엑시트를 사용할 수 있습니다. 클러스터 환경에서 채널 자동 정의 엑시트를 사용하여 다음을 수행할 수 있습니다.

- 통신 정의(SNA LU6.2 이름) 조정
- 다른 엑시트(예: 보안 엑시트) 추가 또는 제거
- 채널 엑시트의 이름을 변경하십시오. CLUSSDR 채널 엑시트 이름이 CLUSRCVR 채널 정의에서 자동 생성되므로 CLUSSDR 채널 엑시트의 이름을 변경해야 합니다. 자동 생성된 이름이 올바르지 않을 수 있으며 채널의 두 끝이 서로 다른 플랫폼에 있는 경우에 특히 올바르지 않습니다. 엑시트 이름의 형식은 플랫폼에 따라 다릅니다. 예를 들어, 는 SCYEXIT('drive:\path\library(secexit)')입니다.

z/OS가 아닌 다른 플랫폼의 엑시트 이름은 일반적인 양식 *path/library(function)*입니다. *function*이 있는 경우 그 중 최대 8자가 사용됩니다. 없는 경우에는, 8자로 잘린 *library*가 사용됩니다. 예를 들면 다음과 같습니다.

- /var/mqm/exits/myExit.so(MsgExit) 은(는) MSGEXIT(으)로 변환됩니다.
- /var/mqm/exits/myExit은(는) MYEXIT(으)로 변환됩니다.
- /var/mqm/exits/myExit.so(ExitLongName) 은(는) EXITLONG(으)로 변환됩니다.

아웃바운드(TCP) 채널이 특정 IP 주소, 포트 또는 포트 범위를 사용할 수 있도록 하려면 채널 속성 LOCLADDR를 사용하십시오. 둘 이상의 네트워크 카드가 있고 아웃바운드 통신에 특정 네트워크 카드를 사용하는 채널이 필요한 경우, LOCLADDR이 유용합니다.

CLUSSDR 채널에서 가상 IP 주소를 지정하려면 수동으로 정의된 CLUSSDR에서 LOCLADDR의 IP 주소를 사용하십시오. 포트 범위를 지정하려면 CLUSRCVR의 포트 범위를 사용하십시오.

클러스터가 아웃바운드 통신 채널을 가져와 특정 IP 주소에 바인드하는 데 LOCLADDR를 사용해야 하는 경우, LOCLADDR 값을 자동으로 정의된 CLUSSDR 채널에 강제로 지정하도록 채널 자동 정의 엑시트를 작성해야 하고 수동으로 정의된 CLUSSDR 채널에서 이를 지정해야 합니다.

모든 큐 관리자가 같은 서버에 있지 않은 경우 CLUSRCVR 채널의 LOCLADDR 필드에 IP 주소를 넣지 마십시오. LOCLADDR IP 주소는 CLUSRCVR 채널을 사용하여 연결된 모든 큐 관리자의 자동 정의된 CLUSSDR 채널로 전파됩니다.

클러스터의 모든 큐 관리자가 모든 아웃바운드 통신에 특정 포트 또는 포트 범위를 사용하게 하려는 경우, CLUSRCVR 채널의 LOCLADDR에 포트 번호 또는 포트 범위를 넣으십시오.

distributed 분산 플랫폼에서는 로컬 주소가 정의되지 않은 모든 송신자 채널에 사용되는 기본 로컬 주소 값을 설정할 수 있습니다. 큐 관리자를 시작하기 전에 MQ_LCLADDR 환경 변수를 설정하여 기본값이 정의됩니다. 값 형식은 MQSC 속성 LOCLADDR과 일치합니다.

자동 정의된 클러스터 송신자 채널 정의는 실제 채널 오브젝트가 아닙니다. z/OS가 아닌 플랫폼에서는, OAM(object authority manager)이 해당 정의가 있는지 인식하지 못합니다. 자동 정의된 클러스터 송신자 채널에서 시작, 중지, ping, 재설정 또는 해석 명령을 실행하려는 경우, OAM이 사용자에게 클러스터의 클러스터 수신자 채널에서 동일한 조치를 수행할 수 있는 권한이 있는지를 확인합니다.

클러스터가 PROPCTL을 사용하여 WebSphere MQ 버전 7 큐 관리자에서 이전 레벨의 WebSphere MQ에 있는 큐 관리자로 전달되는 메시지에서 RFH2와 같은 애플리케이션 헤더를 제거해야 하는 경우, PROPCTL을 NONE 값으로 강제 설정하는 채널 자동 정의 엑시트를 작성해야 합니다. 클러스터 송신자 채널에 해당 클러스터 수신자 채널을 기반으로 한 정의가 있으므로 엑시트가 필수입니다. 이전 레벨의 클러스터 수신자 채널에 PROPCTL 속성이 없으므로 속성이 자동 클러스터 송신자 채널에 의해 COMPAT으로 설정됩니다. 속성은 수동 클러스터 송신자 채널에 설정된 내용에 관계없이 COMPAT으로 설정됩니다.

관련 참조

[로컬 주소\(LOCLADDR\)](#)

기본 클러스터 오브젝트

WebSphere MQ 클러스터 사용 시 기본 클러스터 오브젝트를 작성합니다. 해당 오브젝트는 큐 관리자를 정의할 때 자동으로 작성되는 기본 오브젝트의 세트에 포함됩니다.

다른 채널 정의와 동일한 방법으로, MQSC 또는 PCF 명령을 실행하여 기본 채널 정의를 변경할 수 있습니다.

SYSTEM.CLUSTER.HISTORY.QUEUE를 제외하고, 기본 큐 정의는 변경하지 마십시오.

SYSTEM.CLUSTER.COMMAND.QUEUE

클러스터의 각 큐 관리자는 메시지를 전체 저장소로 전송하는 데 사용되는 SYSTEM.CLUSTER.COMMAND.QUEUE라고 하는 로컬 큐입니다. 메시지에는 큐 관리자에 대한 새 정보나 변경된 정보, 또는 다른 큐 관리자에 관한 정보에 대한 요청이 포함됩니다. SYSTEM.CLUSTER.COMMAND.QUEUE는 보통 비어 있습니다.

SYSTEM.CLUSTER.HISTORY.QUEUE

클러스터의 각 큐 관리자에는 SYSTEM.CLUSTER.HISTORY.QUEUE라고 하는 로컬 큐가 있습니다. SYSTEM.CLUSTER.HISTORY.QUEUE는 서비스 목적을 위해 클러스터 상태 정보의 실행 기록을 저장하는 데 사용됩니다.

기본 오브젝트 설정에서 SYSTEM.CLUSTER.HISTORY.QUEUE 는 PUT(ENABLED) 으로 설정됩니다. 히스토리 콜렉션을 억제하려면 설정을 PUT(DISABLED) 로 변경하십시오.

SYSTEM.CLUSTER.REPOSITORY.QUEUE

클러스터의 각 큐 관리자에는 SYSTEM.CLUSTER.REPOSITORY.QUEUE라고 하는 로컬 큐가 있습니다. 이 큐는 모든 전체 저장소 정보를 저장하는 데 사용됩니다. 이 큐는 보통 비어있지 않습니다.

SYSTEM.CLUSTER.TRANSMIT.QUEUE

각 큐 관리자에는 SYSTEM.CLUSTER.TRANSMIT.QUEUE라고 하는 로컬 큐에 대한 정의가 있습니다.

SYSTEM.CLUSTER.TRANSMIT.QUEUE는 클러스터 내에 있는 모든 큐 및 큐 관리자에 대한 모든 메시지의 기본 전송 큐입니다. 큐 관리자 속성 DEFQMITQ) 을 변경하여 각 클러스터 송신자 채널에 대한 기본 전송 큐를 SYSTEM.CLUSTER.TRANSMIT.ChannelName로 변경할 수 있습니다.

SYSTEM.CLUSTER.TRANSMIT.QUEUE를 삭제할 수 없습니다. 사용되는 기본 전송 큐가

SYSTEM.CLUSTER.TRANSMIT.QUEUE 또는 SYSTEM.CLUSTER.TRANSMIT.ChannelName인지에 대한 권한 검사를 정의하는 데도 사용됩니다.

SYSTEM.DEF.CLUSRCVR

각 클러스터에는 SYSTEM.DEF.CLUSRCVR이라고 하는 기본 CLUSRCVR 채널 정의가 있습니다.

SYSTEM.DEF.CLUSRCVR은 클러스터의 큐 관리자에서 클러스터 수신자 채널을 작성할 때 지정하지 않는 속성에 대한 기본값을 제공하기 위해 사용됩니다.

SYSTEM.DEF.CLUSSDR

각 클러스터에는 SYSTEM.DEF.CLUSSDR이라고 하는 기본 CLUSSDR 채널 정의가 있습니다.

SYSTEM.DEF.CLUSSDR은 클러스터의 큐 관리자에서 클러스터 송신자 채널을 작성할 때 지정하지 않는 속성에 대한 기본값을 제공하기 위해 사용됩니다.

클러스터 전송 큐 및 클러스터-송신자 채널

클러스터된 큐 관리자 간 메시지는 클러스터 전송 큐에 저장되고 클러스터 송신자 채널에서 전달됩니다.

클러스터-송신자 채널을 표시하면 전송 큐와 연관되어 있는 것이 보입니다. 임의의 시점에서 클러스터 송신자 채널은 한 개의 전송 큐와 연관됩니다. 채널의 구성을 변경하는 경우 다음에 시작할 때 다른 전송 큐로 전환할 수도 있습니다. 다음 MQSC 명령을 실행하여 클러스터 송신자 채널과 연관되는 전송 큐를 표시하십시오.

```
DISPLAY CHSTATUS(*) WHERE(CHLTYPE EQ CLUSSDR)
```

```
AMQ8417: Display Channel Status details.
CHANNEL(TO.QM2)                CHLTYPE(CLUSSDR)
CONNAME(9.146.163.190(1416))    CURRENT
QMNAME(QM2)                     STATUS(STOPPED)
SUBSTATE( )                      XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

중지된 클러스터 송신자 채널의 저장된 채널 상태에 표시되는 전송 큐는 채널이 다시 시작되면 변경될 수 있습니다. 156 페이지의 『클러스터 송신자 채널에 따라 기본 전송 큐 선택』 describes the process of selecting a default transmission queue; 156 페이지의 『클러스터 송신자 채널에 따라 수동으로 정의된 전송 큐 선택』 describes the process of selecting a manually defined transmission queue.

클러스터 송신자 채널이 시작되면 전송 큐와의 연관을 다시 검사합니다. 전송 큐의 구성 또는 큐 관리자 기본값이 변경되면 채널을 다른 전송 큐와 다시 연관시킬 수 있습니다. 구성을 변경함에 따라 채널이 다른 전송 큐로 재시작되면 새로 연관된 전송 큐에 대한 메시지 전송 프로세스가 발생합니다. 157 페이지의 『클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식』에서는 한 전송 큐에서 다른 전송 큐로 클러스터 송신자 채널을 전송하는 프로세스를 설명합니다.

클러스터 송신자 채널의 작동은 송신자 채널 및 서버 채널에 대해 다릅니다. 채널 속성 XMITQ이 대체될 때까지 같은 전송 큐와 연관 상태로 남습니다. 송신자 또는 서버 채널에서 전송 큐 속성을 대체하고 재시작하는 경우 이전 전송 큐에서 새 전송 큐로 메시지가 전송되지 않습니다.

클러스터 송신자 채널과 송신자 또는 서버 채널 간 또 다른 차이점은 다중 클러스터 송신자 채널은 클러스터 전송 큐를 열 수 있지만 한 송신자 또는 서버 채널만 정상 전송 큐를 열 수 있다는 것입니다. Version 7.5 클러스터 연결이 단일 클러스터 전송 큐 (SYSTEM.CLUSTER.TRANSMIT.QUEUE) 를 공유할 때까지. Version 7.5 이후로, 전송 큐를 공유하지 않는 클러스터-송신자 채널 옵션이 있습니다. 독점성은 강제 실행되지 않습니다. 이는 구성의 결과입니다. 다른 애플리케이션 사이를 플로우하는 메시지와 전송 큐나 채널을 공유하지 않도록 클러스터에서 메시지가 이동하는 경로를 구성할 수 있습니다. 261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』 및 187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』의 내용을 참조하십시오.

클러스터 송신자 채널에 따라 기본 전송 큐 선택

클러스터 전송 큐는 SYSTEM.CLUSTER.TRANSMIT으로 시작되는 이름의 시스템 기본 큐이거나 수동으로 정의된 큐입니다. 클러스터 송신자 채널은 기본 클러스터 전송 큐 메커니즘 또는 수동 구성 방식 중 하나로 클러스터 전송 큐와 연관됩니다.

기본 클러스터 전송 큐는 큐 관리자 속성 **DEFCLXQ**로 설정됩니다. 그 값은 SCTQ 또는 CHANNEL입니다. 새로운 큐 관리자 및 마이그레이션된 큐 관리자는 SCTQ로 설정됩니다. 이 값을 CHANNEL로 대체할 수 있습니다.

SCTQ가 설정되는 경우 기본 클러스터 전송 큐는 SYSTEM.CLUSTER.TRANSMIT.QUEUE입니다. 모든 클러스터 송신자 채널은 이 큐를 열 수 있습니다. 큐를 여는 클러스터 송신자 채널은 수동으로 정의된 클러스터 전송 큐와 연관되지 않은 채널입니다.

CHANNEL이 설정되면 큐 관리자가 모든 클러스터 송신자 채널에 대해 개별적으로 영구적 동적 전송 큐를 작성할 수 있습니다. 각 큐의 이름은 SYSTEM.CLUSTER.TRANSMIT.ChannelName이며 모델 큐, SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE로부터 작성됩니다. 수동으로 정의된 클러스터 전송 큐와 연관되지 않은 각 클러스터 송신자 채널은 영구적 동적 클러스터 전송 큐와 연관됩니다. 이 큐는 이 클러스터 송신자 채널이 서비스를 제공하는 클러스터 목적지에 대한 별도의 클러스터 전송 큐가 필요하고 큐가 존재하지 않는 경우에 큐 관리자가 작성합니다.

일부 클러스터 목적지는 수동으로 정의된 전송 큐와 연관된 클러스터 송신자 채널이 서비스를 제공할 수 있고 다른 목적지는 기본 큐 또는 여러 큐가 서비스를 제공할 수 있습니다. 클러스터 송신자 채널과 전송 큐와의 연관에서 수동으로 정의된 전송 큐는 항상 기본 전송 큐보다 우선합니다.

클러스터 전송 큐의 우선순위는 156 페이지의 그림 24에 설명됩니다. 수동으로 정의된 클러스터 전송 큐와 연관되지 않은 유일한 클러스터 송신자 채널은 CS.QM1입니다. 전송 큐의 **CLCHNAME** 속성에 있는 어떤 채널 이름도 CS.QM1과 일치하지 않기 때문에 이 채널은 수동으로 정의된 전송 큐와 연관되지 않습니다.

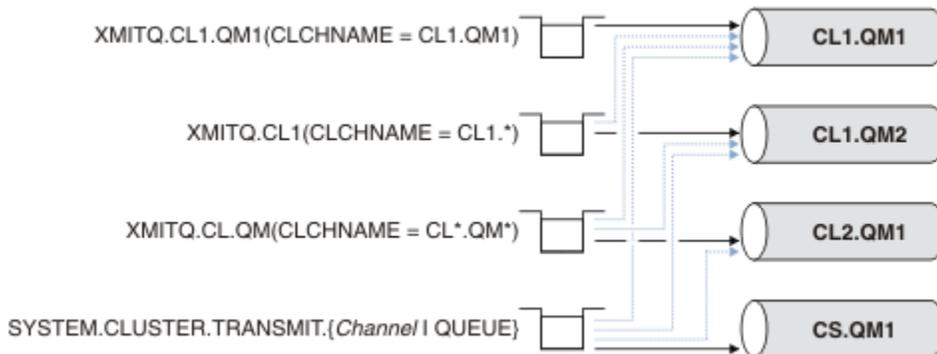


그림 24. 전송 큐 / 클러스터 송신자 채널 우선순위

클러스터 송신자 채널에 따라 수동으로 정의된 전송 큐 선택

수동으로 정의된 큐에는 **USAGE** 속성이 XMITQ로 설정되고 클러스터 채널 이름 속성 **CLCHNAME**이 특정 또는 일반 채널 이름으로 설정된 전송 큐가 있습니다.

CLCHNAME 큐 속성에 있는 이름이 클러스터 송신자 채널 이름과 일치하는 경우 이 채널은 큐와 연관됩니다. 이름에 와일드카드가 포함되지 않은 경우에는 정확히 일치하는 이름이고 이름에 와일드카드가 포함된 경우에는 가장 일치하는 이름입니다.

다중 전송 큐에서 **CLCHNAME** 정의가 같은 클러스터 송신자 채널과 일치하는 경우 정의가 중첩된다고 합니다. 모호함을 해결하기 위해 일치 간 우선순위가 있습니다. 정확한 일치는 항상 우선순위를 갖습니다. 156 페이지의 그림 24에서는 전송 큐와 클러스터 송신자 채널 간 연관을 보여줍니다. 검은색 화살표는 실제 연관을 표시하고 회색 화살표는 잠재적 연관을 표시합니다. 156 페이지의 그림 24에서 전송 큐의 우선순위는 다음과 같습니다.

XMITQ.CL1.QM1

전송 큐 XMITQ.CL1.QM1의 **CLCHNAME** 속성은 CL1.QM1으로 설정됩니다. **CLCHNAME** 속성 CL1.QM1의 정의에는 와일드카드가 없으며 와일드카드와 일치하는 다른 전송 큐에서 정의된 다른 **CLCHNAME** 속성보다 우선합니다. 큐 관리자는 XMITQ.CL1.QM1 전송 큐에서 CL1.QM1 클러스터 송신자 채널에 의해 전송될 클

러스터 메시지를 저장합니다. 유일한 예외는 다중 전송 큐에서 그 **CLCHNAME** 속성이 CL1.QM1으로 설정되어 있는 경우입니다. 이러한 경우에 큐 관리자는 그러한 큐 중 임의의 큐에 CL1.QM1 클러스터 송신자 채널에 대한 메시지를 저장합니다. 채널이 시작될 때 임의로 큐를 선택합니다. 채널이 다시 시작될 때 다른 큐를 선택할 수도 있습니다.

XMITQ.CL1

전송 큐 XMITQ.CL1의 **CLCHNAME** 속성은 CL1.*으로 설정됩니다. **CLCHNAME** 속성 CL1.*의 정의에는 후미 문자 와일드카드가 있으며 이 정의는 CL1.으로 시작하는 클러스터 송신자 채널의 이름과 일치합니다. 큐 관리자는 큐 XMITQ.CL1.QM1과 같이 보다 구체적인 일치에 있는 전송 큐가 없는 경우 전송 큐 XMITQ.CL1에서 이름이 CL1.으로 시작하는 클러스터 송신자 채널에 의해 전송될 클러스터 메시지를 저장합니다. 한 개의 후미 문자 와일드카드가 있는 정의는 와일드카드가 없는 정의에 비해 덜 구체적이고 와일드카드가 여러 개 있거나 와일드카드 뒤에 추가 후미 문자가 있는 정의에 비해 더 구체적입니다.

XMITQ.CL.QM

XMITQ.CL.QM은 그 **CLCHNAME** 속성이 CL*.QM*로 설정된 전송 큐의 이름입니다. CL*.QM*의 정의에는 두 개의 와일드카드가 있으며 이 정의는 CL.로 시작하고 QM을 포함하거나 QM으로 끝나는 클러스터 송신자 채널의 이름과 일치합니다. 이 일치하는 한 개의 와일드카드 일치보다 덜 구체적입니다.

SYSTEM.CLUSTER.TRANSMIT.channelName|QUEUE

큐 관리자가 사용할 클러스터 송신자 채널의 이름과 일치하는 **CLCHNAME** 속성을 가진 전송 큐가 없는 경우 큐 관리자는 기본 클러스터 전송 큐를 사용합니다. 기본 클러스터 전송 큐는 특정 클러스터-송신자 채널 SYSTEM.CLUSTER.TRANSMIT.channelName에 대해 큐 관리자가 작성한 시스템 클러스터 전송 큐 또는 단일 시스템 클러스터 전송 큐 SYSTEM.CLUSTER.TRANSMIT.QUEUE입니다. 어떤 큐를 기본 큐로 할 것인지는 큐 관리자 **DEFXMITQ** 속성이 설정에 따라 다릅니다.

팁: 중첩되는 정의에 대한 분명한 필요성이 없다면 이해하기 어려운 복잡한 구성으로 이어질 수 있으므로 중첩되는 정의는 피하십시오.

클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식

클러스터 송신자 채널과 클러스터 전송 큐의 연관을 변경하려면 전송 큐의 **CLCHNAME** 매개변수 또는 큐 관리자 매개변수 **DEFCLXQ**를 언제든지 변경하십시오. 변경 직후에는 변화가 없습니다. 채널이 시작되는 경우에만 변경이 발생합니다. 채널이 시작되면 같은 전송 큐에서 계속 메시지를 전달하는지 여부를 검사합니다. 세 가지 유형의 변경이 클러스터 송신자 채널과 전송 큐와의 연관을 대체합니다.

1. 클러스터 송신자 채널이 현재 연관되어 있는 전송 큐의 **CLCHNAME** 매개변수를 덜 구체적으로 또는 공백으로 재정의하거나 채널이 중지될 때 클러스터 전송 큐를 삭제합니다.

일부 다른 클러스터 전송 큐가 이제 클러스터 이름에 대해 더 나은 일치일 수 있습니다. 또는 클러스터 송신자 채널의 이름과 일치하는 다른 전송 큐가 없는 경우 연관은 기본 전송 큐로 되돌아가야 합니다.

2. 다른 클러스터 전송 큐의 **CLCHNAME** 매개변수를 재정의하거나 클러스터 전송 큐를 추가합니다.

다른 전송 큐의 **CLCHNAME** 매개변수가 이제 클러스터 송신자 채널에 대해 클러스터 송신자 채널이 현재 연관된 전송 큐에 비해 더 나은 일치일 수 있습니다. 클러스터 송신자 채널이 현재 기본 클러스터 전송 큐와 연관되어 있는 경우 이 채널은 수동으로 정의된 클러스터 전송 큐와 연관이 될 수 있습니다.

3. 클러스터 송신자 채널이 현재 기본 클러스터 전송 큐와 연관되어 있는 경우 **DEFCLXQ** 큐 관리자 매개변수를 변경합니다.

클러스터 송신자 채널의 연관이 변경되는 경우 채널이 시작되면 이 연관이 새 전송 큐에 대한 연관으로 전환됩니다. 전환 중에 메시지가 손실되지 않아야 합니다. 메시지는 채널이 리모트 큐 관리자로 메시지를 전송하는 순서대로 새 전송 큐에 전송됩니다.

알아두기: 클러스터의 메시지를 전달할 때와 마찬가지로, 순서대로 전달되어야 하는 메시지가 순서대로 전달되도록 메시지를 그룹으로 묶어야 합니다. 드물지만 클러스터 내에서 메시지 순서가 뒤바뀔 수 있습니다.

전환 프로세스는 다음 트랜잭션 단계를 거칩니다. 전환 프로세스가 방해되는 경우 채널이 재시작될 때 현재 트랜잭션 단계가 재개됩니다.

단계 1 - 원래 전송 큐의 메시지 처리

클러스터 송신자 채널이 다른 클러스터 송신자 채널과 공유할 수도 있는 새 전송 큐와 연관됩니다. 클러스터 송신자 채널의 메시지는 계속해서 원래 전송 큐에 배치됩니다. 과도기적 전환 프로세스는 메시지를 원래 전

송 큐에서 새 전송 큐로 전송합니다. 클러스터 송신자 채널은 새 전송 큐에서 클러스터 수신자 채널로 메시지를 전달합니다. 채널 상태가 클러스터 송신자 채널이 여전히 이전 전송 큐와 연관되어 있음을 표시합니다.

전환 프로세스가 계속해서 새로 도착한 메시지도 전송합니다. 이 단계는 전환 프로세스에 의해 전달될 남아 있는 메시지의 수가 0에 도달할 때까지 계속됩니다. 메시지 수가 0에 도달하면 프로시저가 단계 2로 이동합니다.

단계 1 중에 채널의 디스크 활동이 늘어납니다. 지속 메시지가 첫 번째 전송 큐에서 두 번째 전송 큐로 커밋됩니다. 이 디스크 활동은 일반적인 메시지 전송의 일부로 전송 큐에서 메시지가 배치되고 제거될 때 커밋되는 메시지에 추가됩니다. 전환이 최대한 빨리 이루어질 수 있도록 전환 프로세스 중 메시지가 도착하지 않는 것이 이상적입니다. 메시지가 도착하면 전환 프로세스에 의해 처리됩니다.

단계 2 - 새 전송 큐에서 메시지 처리

클러스터 송신자 채널에 대한 원래 전송 큐에 남아 있는 메시지가 없게 되는 즉시 새 메시지가 새 전송 큐에 바로 배치됩니다. 채널 상태는 클러스터 송신자 채널이 새 전송 큐와 연관되어 있음을 표시합니다. 큐 관리자 오류 로그에 "AMQ7341 채널 ChannelName에 대한 전송 큐는 QueueName입니다." 메시지가 작성됩니다.

다중 클러스터 전송 큐 및 클러스터 전송 큐 속성

단일 클러스터 전송 큐 또는 다중 큐에 메시지를 저장하는 여러 큐 관리자에게 클러스터 메시지를 전달하도록 선택할 수 있습니다. 큐 하나로는 클러스터 전송 큐 속성 세트 하나를 세트 및 조회로 설정하거나 다중 큐로는 다중 세트를 설정할 수 있습니다. 일부 속성에서는 다중 세트가 있을 경우 유리합니다. 예를 들어, 큐 용량을 조회하면 모든 채널에서가 아닌 한 채널 또한 한 채널 세트가 전달할 대기 중 메시지의 수를 알려줍니다. 다른 속성에서는 다중 세트가 있는 것이 불리합니다. 예를 들면, 모든 클러스터 전송 큐에 대해 같은 액세스 권한을 구성하는 것을 원치 않는 경우입니다. 이러한 이유로 액세스 권한은 항상 특정 클러스터 전송 큐의 프로파일에 대해서가 아닌 SYSTEM.CLUSTER.TRANSMIT.QUEUE의 프로파일에 대해 검사됩니다. 보다 일반적인 보안 검사를 적용하려면 145 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』의 내용을 참조하십시오.

다중 클러스터 송신자 채널 및 다중 전송 큐

큐 관리자는 클러스터 송신자 채널에서 메시지를 전달하기 전에 메시지를 클러스터 전송 큐에 저장합니다. 큐 관리자는 메시지의 목적지에 연결되는 클러스터 송신자 채널을 선택합니다. 모두 같은 목적지에 연결되는 클러스터 송신자 채널을 선택할 수 있습니다. 목적지는 같은 물리적 큐일 수 있으며 다중 클러스터 송신자 채널에 의해 단일 큐 관리자에 연결됩니다. 목적지는 같은 클러스터 내 다른 큐 관리자에서 호스팅되며 같은 큐 이름을 가지는 다수의 물리적 큐일 수도 있습니다. 목적지에 연결된 클러스터-송신자 채널 선택사항이 있는 위치에서 워크로드 밸런싱 알고리즘이 하나를 선택합니다. 이 선택은 다수의 요인에 따라 달라집니다. 클러스터 워크로드 관리 알고리즘을 참조하십시오.

159 페이지의 그림 25에서 CL1.QM1, CL1.QM2 및 CS.QM1이 같은 목적지로 이어질 수 있는 모든 채널입니다. 예를 들어, Q1을 QM1 및 QM2의 CL1에서 정의하면 CL1.QM1 및 CL1.QM2 모두가 다른 두 큐 관리자에 있는 동일 목적지, Q1으로의 라우트를 제공합니다. 채널 CS.QM1도 CL1에 있는 경우 이 채널도 Q1에 대한 메시지가 선택할 수 있는 채널입니다. CS.QM1의 클러스터 멤버십은 클러스터 이름 목록에서 정의될 수 있으며, 채널 이름이 그 구성에 클러스터 이름을 포함하지 않는 이유입니다. 워크로드 밸런싱 매개변수 및 송신 애플리케이션에 따라 Q1에 대한 메시지가 전송 큐 XMITQ.CL1.QM1, XMITQ.CL1 및 SYSTEM.CLUSTER.TRANSMIT.CS.QM1 각각에 배치될 수 있습니다.

메시지 트래픽을 분리하여 동일 목적지에 대한 메시지가 다른 목적지에 대한 메시지와 함께 큐 또는 채널을 공유하지 않으려면, 처음에 다른 클러스터-송신자 채널로 트래픽을 나누는 방법, 그런 다음 특정 채널에 대한 메시지를 다른 전송 큐로 분리하는 방법을 고려해야 합니다. 같은 큐 관리자의 같은 클러스터에 있는 클러스터 큐는 일반적으로 같은 클러스터 채널을 공유합니다. 다중 클러스터 전송 큐를 단독으로 정의하는 것은 클러스터 메시지 트래픽을 여러 큐로 분리하는 데 충분하지 않습니다. 여러 목적지 큐에 대한 메시지를 여러 채널로 분리하지 않으면 메시지가 같은 클러스터 전송 큐를 공유합니다.

메시지가 선택하는 채널의 직접적인 분리 방법은 다중 클러스터를 작성하는 것입니다. 각 클러스터에 있는 큐 관리자에서 한 개의 클러스터 큐만 정의하십시오. 그런 다음 각 클러스터/큐 관리자 조합에 대해 다른 클러스터 송신자 채널을 정의하면 각 클러스터 큐에 대한 메시지가 다른 클러스터 큐에 대한 메시지와 클러스터 채널을 공유하지 않습니다. 클러스터 채널에 대해 별도의 전송 큐를 정의하면 송신 큐 관리자는 각 전송 큐에 한 개의 클러스터 큐에 대한 메시지만 저장합니다. 예를 들어, 두 개의 클러스터가 자원을 공유하지 않도록 하려는 경우 이러한

클러스터 자원을 같은 큐 관리자에 있는 다른 클러스터에 배치하거나 같은 클러스터에 있는 다른 큐 관리자에 배치할 수 있습니다.

클러스터 전송 큐의 선택은 워크로드 밸런싱 알고리즘에 영향을 주지 않습니다. 워크로드 밸런싱 알고리즘은 메시지를 전달할 클러스터 송신자 채널을 선택합니다. 이 알고리즘은 메시지를 해당 채널이 서비스를 제공하는 전송 큐에 배치합니다. 채널이 중지되는 경우와 같이 채널을 다시 선택하기 위해 워크로드 밸런싱 알고리즘이 호출되는 경우 이 알고리즘은 메시지를 전달할 다른 채널을 선택할 수 있습니다. 다른 채널을 선택하고 새 채널이 다른 클러스터 전송 큐의 메시지를 전달하는 경우 이 워크로드 밸런싱 알고리즘은 메시지를 다른 전송 큐로 전송합니다.

159 페이지의 그림 25에서 두 개의 클러스터 송신자 채널인 CS.QM1 및 CS.QM2가 기본 시스템 전송 큐와 연관됩니다. 워크로드 밸런싱 알고리즘이 메시지를 SYSTEM.CLUSTER.TRANSMIT.QUEUE 또는 다른 클러스터 전송 큐에 저장하면 메시지를 전달할 클러스터 송신자 채널의 이름이 메시지의 상관 ID에 저장됩니다. 각 채널은 채널 이름과 상관 ID가 일치하는 메시지만 전달합니다.

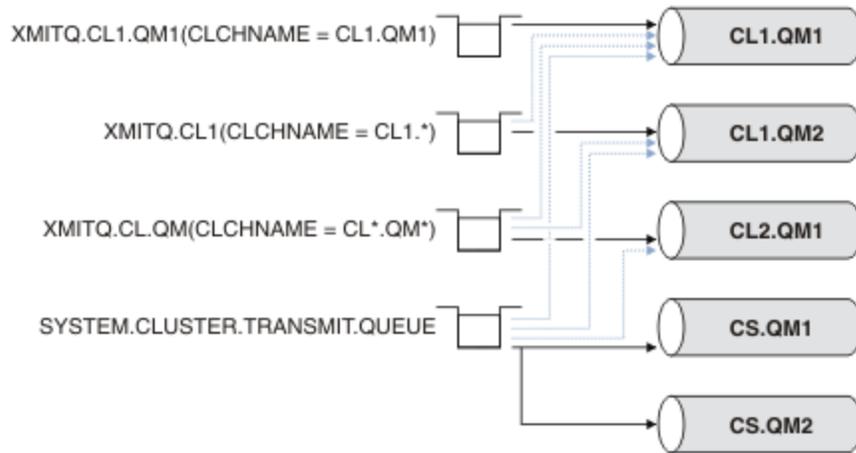


그림 25. 다중 클러스터 송신자 채널

CS.QM1이 중지되면 해당 클러스터 송신자 채널의 전송 큐에 있는 메시지를 조사합니다. 다른 채널이 전달할 수 있는 메시지는 워크로드 밸런싱 알고리즘에 의해 다시 처리됩니다. 해당 상관 ID는 대체 클러스터 송신자 채널 이름으로 재설정됩니다. 대체 클러스터 송신자 채널이 CS.QM2인 경우 메시지는 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 남습니다. 대체 채널이 CL1.QM1인 경우 워크로드 밸런싱 알고리즘이 메시지를 XMITQ.CL1.QM1에 전송합니다. 클러스터 송신자 채널이 재시작되면 새 메시지 및 다른 클러스터 송신자 채널에 대해 플래그 지정되지 않은 메시지가 다시 이 채널에서 전송됩니다.

실행 중인 시스템에서 전송 큐와 클러스터 송신자 채널 간 연관을 변경할 수도 있습니다. 전송 큐에서 CLCHNAME 매개변수를 변경하거나 DEFCLXQ 큐 관리자 매개변수를 변경할 수 있습니다. 변경의 영향을 받는 채널이 재시작되면 이 채널은 전송 큐 전환 프로세스를 시작합니다. 157 페이지의 『클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식』의 내용을 참조하십시오.

채널이 재시작되면 전송 큐를 전환하는 프로세스가 시작됩니다. 채널이 중지되면 워크로드 재밸런싱 프로세스가 시작됩니다. 두 프로세스가 병렬로 실행될 수 있습니다.

단순한 예로는 클러스터 송신자 채널이 중지되어도 재밸런싱 프로세스가 큐의 메시지를 전송할 클러스터 송신자 채널을 변경하지 않는 경우입니다. 이 경우는 메시지를 올바른 목적지로 전달할 수 있는 다른 클러스터 송신자 채널이 없는 경우입니다. 메시지를 해당 목적지로 전송할 대체 클러스터 송신자 채널이 없는 경우 메시지는 클러스터 송신자 채널이 중지된 후 같은 클러스터 송신자 채널에 대해 플래그 지정된 상태로 남습니다. 채널이 시작될 때 전환이 보류 중이면 전환 프로세스는 같은 클러스터 송신자 채널이 메시지를 처리하는 다른 전송 큐로 메시지를 이동합니다.

이보다 복잡한 예로는 둘 이상의 클러스터 송신자 채널이 일부 메시지를 동일한 목적지로 처리할 수 있는 경우입니다. 클러스터 송신자 채널을 중지하고 재시작하여 전송 큐 전환을 트리거합니다. 대부분의 경우 채널을 재시작하는 시점에는 이미 워크로드 밸런싱 알고리즘이 원래 전송 큐에서 다른 클러스터 송신자 채널이 서비스를 제공하는 다른 전송 큐로 메시지를 이동한 상태입니다. 다른 클러스터 송신자 채널이 전달할 수 없는 메시지만 새 전송 큐로 전송되기 위해 남습니다. 일부 경우에 채널을 빠르게 재시작하면 워크로드 밸런싱 알고리즘에 의해 전송

될 수 있는 일부 메시지가 남습니다. 이 경우 남아 있는 일부 메시지는 워크로드 밸런싱 프로세스에 의해 전환되고 일부는 전송 큐 전환 프로세스에 의해 전환됩니다.

관련 개념

258 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

367 페이지의 『로그의 크기 계산』

큐 관리자가 필요로 하는 로그의 크기를 계산합니다.

관련 태스크

261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』

클러스터 전송 큐의 선택에 대한 지침입니다. 하나의 공용 기본 큐, 별도의 기본 큐, 또는 수동으로 정의된 큐를 구성할 수 있습니다. 다중 클러스터 전송 큐 구성은 z/OS 이외의 플랫폼에 적용됩니다.

197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』

태스크의 지시사항에 따라 게이트웨이 큐 관리자가 포함된 중첩 클러스터를 구성합니다. 한 애플리케이션에 대한 메시지를 클러스터 내 다른 애플리케이션에 대한 메시지로부터 격리하는 다음 예에 대한 시작점으로 이 클러스터를 사용합니다.

179 페이지의 『클러스터에 큐 관리자 추가: 개별 전송 큐』

다음 지시사항에 따라 사용자가 작성한 클러스터에 큐 관리자를 추가합니다. 클러스터 큐 및 토픽에 대한 메시지는 다중 클러스터 전송 큐를 사용하여 전송됩니다.

184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 추가 클러스터 전송 큐를 사용하여 클러스터의 단일 큐 관리자로 메시지 트래픽을 분리합니다.

187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 솔루션은 추가 클러스터를 사용하여 메시지를 특정 클러스터 큐로 분리합니다.

전체 저장소를 보유할 클러스터 큐 관리자 선택 방법

각 클러스터에서 전체 저장소를 보유할 최소 하나, 가급적이면 두 개의 큐 관리자를 선택해야 합니다. 가장 예외적 상황을 제외하면 두 개의 전체 저장소가 충분합니다. 가능하면 강력하고 영구적으로 연결된 플랫폼에서 호스트되고, 동시 가동 중단이 없으며, 지리적으로 중앙에 위치한 큐 관리자를 선택하십시오. 또한 전체 저장소 호스트로 사용할 시스템을 다른 태스크에 사용하지 않고 전용 시스템으로 사용할 것을 고려하십시오.

전체 저장소는 클러스터 상태의 전체 그림을 유지보수하는 큐 관리자입니다. 이 정보를 공유하기 위해 각 전체 저장소는 CLUSSDR 채널(및 해당 CLUSRCVR 정의)을 통해 클러스터의 다른 모든 전체 저장소에 연결됩니다. 이 채널을 수동으로 정의해야 합니다.



그림 26. 연결된 두 개의 전체 저장소.

클러스터의 다른 모든 큐 관리자는 부분 저장소에서 클러스터 상태에 대해 현재 알고 있는 사항을 유지합니다. 이 큐 관리자는 사용 가능한 두 개의 전체 저장소를 사용하여 자신에 대한 정보를 발행하고 다른 큐 관리자에 대한 정보를 요청합니다. 선택된 전체 저장소가 사용 불가능하면 다른 전체 저장소가 사용됩니다. 선택된 전체 저장소가 다시 사용 가능하게 되면 보조를 맞추기 위해 다른 큐 관리자로부터 새로운 최신 정보와 변경된 정보를 수집합니다. 모든 전체 저장소가 서비스 불가능 상태가 되면 다른 큐 관리자는 부분 저장소의 정보를 사용합니다. 하지

만 보유한 정보만을 사용하도록 제한되며 새 정보 및 업데이트에 대한 요청은 처리 불가능합니다. 전체 저장소가 네트워크에 다시 연결되면 메시지가 교환되어 모든 저장소(전체 및 부분 모두)는 최신 상태가 됩니다.

전체 저장소 할당을 계획할 때에는 다음을 고려하십시오.

- 전체 저장소를 보유하도록 선택한 큐 관리자는 신뢰할 수 있으며 관리되어야 합니다. 강력하고 영구적으로 연결된 플랫폼에서 호스트되는 큐 관리자를 선택하십시오.
- 전체 저장소를 호스트하는 시스템의 계획된 가동 중단을 고려하고 동시 가동 중단이 없는지 확인하십시오.
- 네트워크 성능을 고려하십시오. 지리적으로 중앙에 위치하거나 클러스터의 다른 큐 관리자와 동일한 시스템을 공유하는 큐 관리자를 선택하십시오.
- 큐 관리자가 둘 이상 클러스터의 멤버인지 여부를 고려하십시오. 동일한 큐 관리자를 사용하여 여러 클러스터에 대한 전체 저장소를 호스트하는 것이 관리상 편리할 수 있습니다. 단 이는 큐 관리자가 얼마나 바쁠지 예상한 바와 이 장점이 밸런스를 이룰 때입니다.
- 전체 저장소만을 포함한 일부 시스템을 다른 태스크에 사용하지 않고 전용 시스템으로 사용할 것을 고려하십시오. 그러면 이 시스템은 큐 관리자 구성에 대한 유지보수만을 수행하며 기타 비즈니스 애플리케이션의 유지보수를 위해 서비스에서 제거되지 않습니다. 또한 저장소를 유지보수하는 태스크가 시스템 자원을 두고 애플리케이션과 경쟁하지 않습니다. 이는 특히, 전체 저장소에 클러스터 상태를 유지보수하는 워크로드가 훨씬 많은 대형 클러스터(즉, 수천 개 이상의 큐 관리자를 포함한 클러스터)에 유익할 수 있습니다.

셋 이상의 전체 저장소도 가능하지만 거의 권장되지 않습니다. 오브젝트 정의(즉, 큐, 토픽, 채널)는 사용 가능한 모든 전체 저장소로 플로우되지만 요청은 부분 저장소에서 최대 두 개의 전체 저장소로만 플로우됩니다. 이는 셋 이상의 전체 저장소가 정의되고 두 개의 전체 저장소가 사용 불가능하게 되면 일부 부분 저장소가 예상하는 업데이트를 수신하지 못할 수 있음을 의미합니다. MQ 클러스터: 왜 전체 저장소는 두 개뿐입니까?를 참조하십시오.

셋 이상의 전체 저장소를 정의하는 것이 유용할 수 있는 유일한 상황은 기존 전체 저장소를 새 하드웨어나 새 큐 관리자로 마이그레이션할 때입니다. 이 경우에는 대체 전체 저장소를 도입하고 이전 전체 저장소를 제거하기 전에 이 저장소가 완전히 채워졌는지 확인해야 합니다. 전체 저장소를 추가할 때마다 CLUSSDR 채널로 다른 모든 전체 저장소에 이 전체 저장소를 직접 연결해야 함을 기억하십시오.

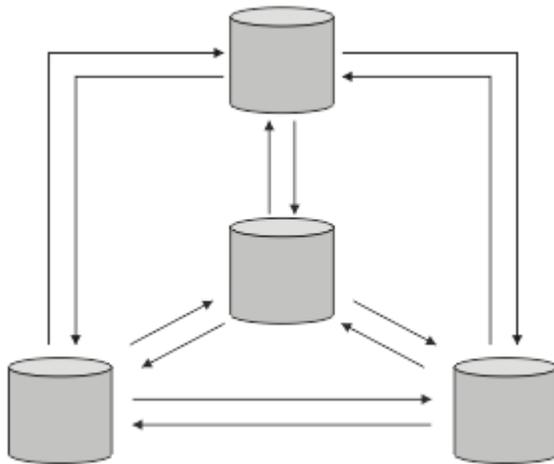


그림 27. 연결된 셋 이상의 전체 저장소

관련 정보

[MQ 클러스터: 왜 전체 저장소는 두 개뿐입니까?](#)

[MQ 클러스터의 가능한 크기](#)

클러스터 조직화

전체 저장소로 링크할 큐 관리자를 선택하십시오. 성능 영향, 큐 관리자 버전, 다수의 CLUSSDR 채널을 원하는지 여부를 고려하십시오.

전체 저장소를 보유할 큐 관리자를 선택했다면 전체 저장소로 링크할 큐 관리자를 결정해야 합니다. CLUSSDR 채널 정의는 클러스터의 다른 전체 저장소를 알아내는 전체 저장소로 큐 관리자를 링크합니다. 이 때부터 큐 관리자는 두 개의 전체 저장소로 메시지를 보냅니다. 항상 CLUSSDR 채널 정의가 있는 전체 저장소를 먼저 사용하려 시

도합니다. 큐 관리자를 두 전체 저장소 중 하나로 링크하도록 선택할 수 있습니다. 선택 시 구성의 토폴로지와 큐 관리자의 물리적 또는 지리적 위치를 고려하십시오.

모든 클러스터 정보가 두 개의 전체 저장소에 송신되기 때문에 두 번째 CLUSSDR 채널 정의를 작성하려는 경우가 있을 수 있습니다. 광범위한 영역에 많은 전체 저장소가 분산된 클러스터에 두 번째 CLUSSDR 채널을 정의할 수 있습니다. 그런 다음 정보가 송신되는 두 개의 전체 저장소를 제어할 수 있습니다.

클러스터 이름 지정 규칙

큐 관리자가 속한 클러스터를 식별하는 이름 지정 규칙을 사용하여 동일한 클러스터의 큐 관리자 이름을 지정할 것을 고려하십시오. 채널 이름에 유사한 이름 지정 규칙을 사용하고 채널 특성을 설명하도록 확장하십시오.

MQ 클러스터 이름 지정 시 우수 사례

클러스터 이름은 최대 48자까지 가능하지만 상대적으로 짧은 클러스터 이름은 다른 오브젝트에 이름 지정 규칙을 적용할 때 유용합니다. 162 페이지의 『클러스터 채널 이름 선택 시 우수 사례』의 내용을 참조하십시오.

클러스터 이름을 선택할 때 일반적으로 '컨텐츠'가 아닌 클러스터의 '목적' (수명이 길 수 있음) 을 표시하는 것이 유용합니다. 예를 들어, 'QM1_QM2_QM3_CLUS' 대신 'B2BPROD' 또는 'ACTTEST' 입니다.

클러스터 큐 관리자 이름 선택 시 우수 사례

처음부터 새 클러스터 및 해당 멤버를 작성하는 경우 해당 클러스터 사용량을 반영하는 큐 관리자에 대한 이름 지정 규칙을 고려하십시오. 모든 큐 관리자의 이름이 서로 달라야 합니다. 그러나 논리 그룹을 식별하고 기억하는데 도움이 되도록 클러스터의 큐 관리자에 유사한 이름 세트를 제공할 수 있습니다 (예: 'ACTTQM1, ACTTQM2').

상대적으로 짧은 큐 관리자 이름 (예: 8자미만) 은 다음 절에 설명된 규칙을 사용하거나 채널 이름에 대해 유사한 규칙을 사용하도록 선택하는 경우에 도움이 됩니다.

클러스터 채널 이름 선택 시 우수 사례

큐 관리자 및 클러스터는 최대 48자의 이름을 가질 수 있고 채널 이름은 20자로 제한되므로 프로젝트를 통해 중간에 이름 지정 규칙을 변경하지 않도록 오브젝트의 이름을 처음 지정할 때 주의하십시오 (이전 절 참조).

채널을 정의할 때 클러스터에 있는 임의의 큐 관리자에서 자동으로 작성된 클러스터 송신자 채널은 클러스터의 수신 큐 관리자에 구성된 해당 클러스터 수신자 채널에서 해당 이름을 사용하므로 이러한 채널은 고유해야 하며 클러스터의 리모트 큐 관리자에서 의미가 있어야 합니다.

한 가지 일반적인 방법은 클러스터 이름이 앞에 오는 큐 관리자 이름을 사용하는 것입니다. 예를 들어, 클러스터 이름이 CLUSTER1이고 큐 관리자가 QM1, QM2이면 클러스터 수신자 채널은 CLUSTER1.QM1, CLUSTER1.QM2 입니다.

채널에 다른 우선순위가 있거나 다른 프로토콜을 사용하는 경우 이 규칙을 확장할 수 있습니다. 예를 들면, 다음과 같습니다.

- CLUSTER1.QM1.S1
- CLUSTER1.QM1.N3
- CLUSTER1.QM1.T4

이 예에서 S1 는 첫 번째 SNA 채널일 수 있고, N3 는 네트워크 우선순위가 3인 NetBIOS 채널일 수 있으며, T4 는 IPV4 네트워크를 사용하는 TCP/IP일 수 있습니다.

공유 채널 정의 이름 지정

단일 채널 정의를 여러 클러스터에서 공유할 수 있습니다. 이 경우 여기에 제안된 이름 지정 규칙을 수정해야 합니다. 그러나 채널 정의 관리 에 설명된 대로 일반적으로 어떤 경우에도 각 클러스터에 대해 개별 채널을 정의하는 것이 좋습니다.

이전 채널 이름 지정 규칙

클러스터 환경 외부에서는 'FROMQM.TO.TARGETQM' 이름 지정 규칙을 사용하는 것이 일반적이므로 기존 클러스터에서 유사한 규칙 (예: CLUSTER.TO.TARGET) 을 사용했음을 알 수 있습니다. 이는 채널 이름 내에

서 '유용한' 정보를 전달하기 위해 사용 가능한 문자를 더 줄이기 때문에 새 클러스터 이름 지정 스킴의 일부로 권장되지 않습니다.

중첩 클러스터

중첩 클러스터는 추가 관리 기능을 제공합니다. 이름 목록을 사용하여 중첩 클러스터를 관리하는 데 필요한 명령의 수를 줄이십시오.

중첩되는 클러스터를 작성할 수 있습니다. 중첩 클러스터를 정의할 수 있는 많은 이유가 있습니다. 예를 들어, 다음과 같습니다.

- 서로 다른 조직이 각자 고유의 관리를 할 수 있도록 하기 위해
- 독립 애플리케이션을 개별적으로 관리할 수 있도록 하기 위해
- 서비스 클래스를 작성하기 위해

163 페이지의 그림 28에서 큐 관리자 STF2는 두 클러스터 모두의 멤버입니다. 한 큐 관리자가 둘 이상의 클러스터의 멤버이면 이름 목록을 사용하여 필요한 정의의 수를 줄일 수 있습니다. 이름 목록은 예를 들어, 클러스터 이름 등의 이름 목록을 포함합니다. 클러스터의 이름을 지정하는 이름 목록을 작성할 수 있습니다. STF2에 대한 ALTER QMGR 명령에 이름 목록을 지정하여 이 큐 관리자가 두 클러스터 모두의 전체 저장소 큐 관리자가 되게 하십시오.

네트워크에 둘 이상의 클러스터가 있으면 각각 다른 이름을 지정해야 합니다. 동일한 이름의 두 클러스터가 병합되는 경우 다시 클러스터를 분리할 수 없습니다. 클러스터 및 채널에 서로 다른 이름을 지정하는 것도 좋은 방법입니다. DISPLAY 명령의 출력을 볼 때 이들이 보다 쉽게 구별됩니다. 큐 관리자가 제대로 작동하려면 클러스터 내에서 큐 관리자 이름이 고유해야 합니다.

서비스 클래스 정의

각 직원 구성원과 각 학생별로 하나의 큐 관리자가 있는 대학교를 상상해 보십시오. 직원 구성원 간의 메시지는 우선순위가 높은 고대역폭의 채널로 이동합니다. 학생 간의 메시지는 보다 저렴하고 느린 채널에서 이동합니다. 일반적인 분산 큐잉 기술을 사용하여 이 네트워크를 설정할 수 있습니다. WebSphere MQ는 목적지 큐 이름 및 큐 관리자 이름을 확인하여 사용할 채널을 선택합니다.

직원과 학생을 명확하게 구분하기 위해 163 페이지의 그림 28에 표시된 대로 큐 관리자를 두 개의 클러스터로 그룹화할 수 있습니다. WebSphere MQ는 Staff 클러스터에 정의된 채널을 통해서만 해당 클러스터의 meetings 큐로 메시지를 이동합니다. Students 클러스터의 gossip 큐 메시지는 이 클러스터에 정의된 채널을 통해 이동하고 적절한 서비스 클래스를 수신합니다.

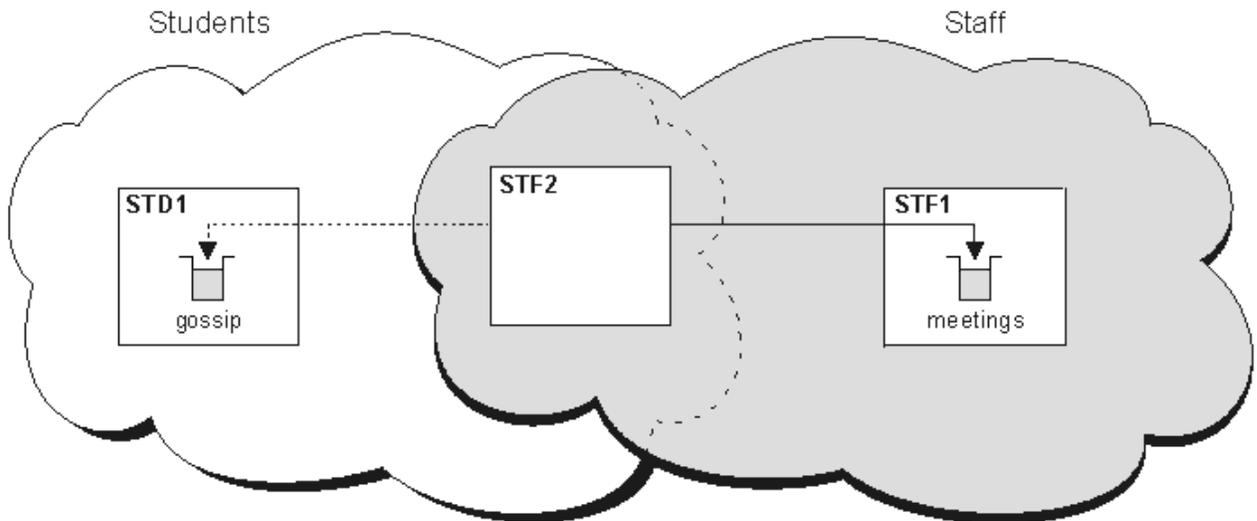


그림 28. 서비스 클래스

클러스터링 팁

클러스터링을 사용하기 전에 시스템 또는 애플리케이션을 약간 수정해야 할 수 있습니다. 분산 큐잉 작동의 차이점 및 유사성이 모두 존재합니다.

- WebSphere MQ Explorer는 버전 6.0 이전 버전에서 z/OS용 WebSphere MQ 큐 관리자를 직접 관리할 수 없습니다.
- 이러한 큐 관리자가 클러스터 큐에 액세스하도록 클러스터 외부의 큐 관리자에 수동 구성 정의를 추가해야 합니다.
- 동일한 이름의 두 클러스터를 병합하는 경우 이 클러스터를 다시 분리할 수 없습니다. 따라서 모든 클러스터에 고유 이름을 지정하는 것이 좋습니다.
- 메시지가 큐 관리자에 도달하지만 수신할 큐가 없으면 데드-레터 큐에 메시지를 넣습니다. 데드-레터 큐가 없으면 채널이 실패하고 다시 시도합니다. 데드-레터 큐의 사용은 분산 큐잉의 경우와 동일합니다.
- 지속 메시지의 무결성이 유지됩니다. 클러스터 사용으로 인해 메시지가 복제 또는 손실되지 않습니다.
- 클러스터를 사용하면 시스템 관리가 감소합니다. 클러스터는 분산 큐잉을 사용하여 고려할 수 있는 것보다 더 많은 큐 관리자가 있는 보다 큰 네트워크에 쉽게 연결합니다. 클러스터의 모든 큐 관리자 간에 통신을 사용하려 시도하는 경우에는 과도한 네트워크 자원을 소모할 위험이 있습니다.
- 트리 구조에서 큐 관리자를 나타내는 WebSphere MQ Explorer를 사용하는 경우, 대형 클러스터 보기를 사용하기 어려울 수 있습니다.
- WebSphere MQ Explorer는 z/OS용 WebSphere MQ 버전 6 이상에서 저장소 큐 관리자를 사용하여 클러스터를 관리할 수 있습니다. 개별 시스템에 추가 저장소를 지정할 필요가 없습니다. z/OS에 있는 WebSphere MQ의 이전 버전에서는, WebSphere MQ Explorer가 저장소 큐 관리자를 사용하여 클러스터를 관리할 수 없습니다. WebSphere MQ Explorer가 관리할 수 있는 시스템에서 추가 저장소를 지정해야 합니다.
- 분배 목록의 용도는 단일 MQPUT 명령을 사용하여 여러 목적지에 동일한 메시지를 보내기 위해서입니다. Distribution lists are supported on WebSphere MQ for AIX, IBM i, HP-UX, Solaris, Linux, and 윈도우. 분배 목록을 큐 관리자 클러스터와 함께 사용할 수 있습니다. 클러스터에서 모든 메시지는 MQPUT 시에 확장됩니다. 네트워크 트래픽의 관점에서 보면 비클러스터링 환경에서처럼 장점이 그리 많지 않습니다. 분배 목록의 장점은 수많은 채널과 전송 큐를 수동으로 정의할 필요가 없다는 점입니다.
- 클러스터를 사용하여 워크로드 밸런싱을 조절하려는 경우 애플리케이션을 검사하십시오. 특정 큐 관리자로 또는 특정 순서로 메시지를 처리해야 하는지 확인하십시오. 이러한 애플리케이션은 메시지 연관관계가 있습니다. 복잡한 클러스터에서 애플리케이션을 사용하려면 먼저 수정해야 할 수 있습니다.
- 특정 목적지로 메시지를 강제로 보내기 위해 MQOPEN에 MQOO_BIND_ON_OPEN 옵션 사용을 선택할 수 있습니다. 목적지 큐 관리자가 사용 불가능하면 큐 관리자가 다시 사용 가능하게 될 때까지 메시지가 전달되지 않습니다. 복제 위험으로 인해 다른 큐 관리자로 메시지가 라우팅되지 않습니다.
- 큐 관리자가 클러스터 저장소를 호스팅할 경우에는 호스트 이름 또는 IP 주소를 알고 있어야 합니다. 클러스터를 조인하는 기타 큐 관리자에 대한 CLUSSDR 정의를 작성할 때 CONNAME 매개변수에 이 정보를 지정해야 합니다. DHCP를 사용하면 시스템을 재시작할 때마다 DHCP가 새 IP 주소를 할당할 수 있어서 IP 주소가 변경되기 쉽습니다. 따라서 CLUSSDR 정의에 IP 주소를 지정하면 안 됩니다. 모든 CLUSSDR 정의가 IP 주소가 아닌 호스트 이름을 지정해도 정의는 여전히 신뢰할 수 없습니다. DHCP는 호스트의 DNS 디렉토리 입력 항목을 반드시 새 주소로 업데이트하지는 않습니다. DHCP를 사용하는 시스템의 전체 저장소로 큐 관리자를 지정해야 하는 경우에는 DNS 디렉토리를 최신 상태로 유지함이 보장되는 소프트웨어를 설치하십시오.
- Do not use generic names, for example VTAM generic resources or Dynamic Domain Name Server (DDNS) generic names as the connection names for your channels. 그러할 경우 채널이 예상과 다른 큐 관리자에 연결할 수 있습니다.
- 로컬 클러스터 큐에서만 메시지를 가져올 수 있지만 메시지 넣기는 클러스터의 어느 큐에나 가능합니다. MQGET 명령을 사용하기 위해 큐를 여는 경우 큐 관리자는 로컬 큐를 엽니다.
- 간단한 WebSphere MQ 클러스터를 설정하는 경우 애플리케이션 중 하나를 대체하지 않아도 됩니다. 애플리케이션이 MQOPEN 호출에 대상 큐의 이름을 지정할 수 있으며 큐 관리자의 위치에 대해 알 필요가 없습니다. 워크로드 관리를 위한 클러스터를 설정하면 애플리케이션을 검토하고 필요에 따라 수정해야 합니다.
- DISPLAY CHSTATUS 및 DISPLAY QSTATUS **runmqsc** 명령을 사용하여 채널 또는 큐에 대한 현재 모니터링 및 상태 데이터를 볼 수 있습니다. 모니터링 정보를 사용하여 시스템의 성능 및 상태를 파악할 수 있습니다.

모니터링은 큐 관리자, 큐, 채널 속성으로 제어됩니다. 자동 정의된 클러스터 송신자 채널의 모니터링은 MONACLS 큐 관리자 속성으로 가능합니다.

관련 개념

클러스터

클러스터 작동 방법

146 페이지의 『클러스터링과 분산 큐잉의 비교』

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

148 페이지의 『클러스터의 컴포넌트』

클러스터는 큐 관리자, 클러스터 저장소, 클러스터 채널 및 클러스터 큐로 구성됩니다.

167 페이지의 『IBM WebSphere MQ 클러스터 관리』

IBM WebSphere MQ 클러스터를 작성, 확장 및 유지보수할 수 있습니다.

관련 태스크

144 페이지의 『큐 관리자 클러스터 구성』

클러스터가 작업하는 방법, 클러스터 구성을 설계하는 방법을 찾고 단순한 클러스터를 설정하는 방법의 예를 알려면 이 토픽의 링크를 사용하십시오.

168 페이지의 『새 클러스터 설정』

다음 지시사항에 따라 클러스터 예를 설정합니다. 개별 지시사항에서는 TCP/IP, LU 6.2 및 단일 전송 큐 또는 다중 전송 큐에서의 클러스터 설정을 설명합니다. 한 큐 관리자에서 다른 큐 관리자로 메시지를 송신하여 클러스터가 작동하는지 테스트합니다.

클러스터에서 통신 설정

전달할 메시지가 있을 때 통신 채널을 시작하기 위한 채널 시작기가 필요합니다. 채널 리스너는 메시지를 수신하기 위해 채널의 다른 한 측이 시작할 때까지 대기합니다.

시작하기 전에

클러스터 내 큐 관리자 간 통신을 설정하려면 지원되는 통신 프로토콜 중 하나를 사용하여 링크를 구성하십시오. 지원되는 프로토콜은 모든 플랫폼에서 TCP 또는 LU 6.2이고 Windows 시스템에서는 NetBIOS 또는 SPX입니다. 분산 큐잉에서와 마찬가지로 이 구성의 일부로 채널 시작기 및 채널 리스너가 필요합니다.

이 태스크 정보

모든 클러스터 큐 관리자는 시스템 정의 시작 큐인 SYSTEM.CHANNEL.INITQ를 모니터링하기 위한 채널 시작기가 필요합니다. SYSTEM.CHANNEL.INITQ는 클러스터 전송 큐를 포함한 모든 전송 큐에 대한 시작 큐입니다.

각 큐 관리자에는 채널 리스너가 있어야 합니다. 채널 리스너 프로그램은 수신되는 네트워크 요청을 대기하고 필요할 때 적절한 수신자 채널을 시작합니다. 채널 리스너의 구현은 플랫폼별로 다르지만 일부 공통 기능이 있습니다. 모든 WebSphere MQ 플랫폼에서 START LISTENER 명령을 사용하여 리스너를 시작할 수 있습니다. On WebSphere MQ for IBM i, 윈도우, UNIX and Linux systems, you can start the listener automatically at the same time as the queue manager. 리스너를 자동으로 시작하려면 LISTENER 오브젝트의 CONTROL 속성을 QMGR 또는 STARTONLY로 설정하십시오.

프로시저

1. 채널 시작기를 시작하십시오.



IBM WebSphere MQ for 윈도우, UNIX and Linux systems

큐 관리자를 시작할 때 큐 관리자 속성 SCHINIT가 QMGR로 설정되면 채널 시작기가 자동으로 시작됩니다. 그렇지 않으면 **runmqsc START CHINIT** 명령 또는 **runmqchi** 제어 명령을 사용하여 시작할 수 있습니다.

2. 채널 리스너를 시작하십시오.

Windows

IBM WebSphere MQ (의 경우

WebSphere MQ에서 제공하는 채널 리스너 프로그램 또는 운영 체제에서 제공하는 기능을 사용하십시오.

WebSphere MQ 채널 리스너를 시작하려면 RUNMQLSR 명령을 사용하십시오. 예를 들면, 다음과 같습니다.

```
RUNMQLSR -t tcp -p 1414 -m QM1
```

UNIX

Linux

UNIX and Linux 시스템의 IBM WebSphere MQ

WebSphere MQ에서 제공하는 채널 리스너 프로그램 또는 운영 체제에서 제공하는 기능(예: TCP 통신을 위한 **inetd**)을 사용하십시오.

WebSphere MQ 채널 리스너를 시작하려면 **runmqclsr** 명령을 사용하십시오. 예를 들면, 다음과 같습니다.

```
runmqclsr -t tcp -p 1414 -m QM1
```

inetd를 사용하여 채널을 시작하려면 다음 두 파일을 구성하십시오.

- /etc/services 파일을 편집하십시오. superuser 또는 루트로 로그인해야 합니다. 다음 행이 파일에 없는 경우 이 행을 다음과 같이 추가하십시오.

```
MQSeries      1414/tcp      # Websphere MQ channel listener
```

여기서, 1414는 IBM WebSphere MQ에서 요구되는 포트 번호입니다. 포트 번호를 변경할 수 있지만 포트 번호는 송신 측에서 지정된 포트 번호와 일치해야 합니다.

- /etc/inetd.conf 파일을 편집하십시오. 해당 파일에 다음 행이 없는 경우에는 다음과 같이 추가하십시오.

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrista amqcrista  
-m queue.manager.name
```

여기서 **MQ_INSTALLATION_PATH**는 WebSphere MQ가 설치된 상위 레벨 디렉토리로 바꿉니다.

업데이트는 **inetd**가 구성 파일을 다시 읽은 후에 활성화 됩니다. 루트 사용자 ID에서 다음 명령을 실행하십시오.

AIX의 경우,

```
refresh -s inetd
```

HP-UX의 경우:

```
inetd -c
```

Solaris 또는 Linux:

- 다음 명령을 사용하여 **inetd**의 프로세스 ID를 찾으십시오.

```
ps -ef | grep inetd
```

- 다음과 같이 적절한 명령을 실행하십시오.

- Solaris 9 및 Linux의 경우:

```
kill -1 inetd processid
```

- Solaris 10 이상 버전:

큐 관리자 저장소가 정보를 보유하는 기간

큐 관리자 저장소는 정보를 30일 동안 보유합니다. 자동 프로세스가 효율적으로 사용 중인 정보를 새로 고칩니다.

큐 관리자가 자체적인 일부 정보를 전송하는 경우 전체 및 부분 저장소 큐 관리자가 정보를 30일 동안 저장합니다. 정보는 예를 들어 큐 관리자가 새 큐 작성을 광고하는 경우에 전송됩니다. 이 정보가 만기되지 않도록 하기 위해 큐 관리자는 27일 후에 자체에 대한 모든 정보를 자동으로 다시 전송합니다. 부분 저장소가 30일의 지속 기간 동안 정보 파트에 대한 새 요청을 전송하면 만기 시간이 원래 30일로 유지됩니다.

정보가 만기되면 저장소에서 즉시 제거되지는 않습니다. 대신, 60일의 유예 기간 동안 보유됩니다. 유예 기간 중에 업데이트가 수신되지 않으면 정보가 제거됩니다. 유예 기간을 사용하면 큐 관리자가 만기 날짜에 임시로 서비스되지 않을 수도 있습니다. 큐 관리자가 90일 이상 클러스터에 연결되지 않으면 더 이상 클러스터의 파트가 아닙니다. 그렇지만 네트워크에 다시 연결되는 경우 다시 클러스터의 파트가 됩니다. 전체 저장소는 다른 큐 관리자의 새 요청을 충족시키기 위해 만기된 정보를 사용하지 않습니다.

유사하게 큐 관리자가 전체 저장소에서의 최신 정보에 대한 요청을 전송하면 요청이 30일 동안 계속됩니다. 27일 후에 IBM WebSphere MQ는 요청을 검사합니다. 27일 동안 참조된 경우에는 자동으로 새로 고쳐집니다. 그렇지 않은 경우, 만기된 상태가 되고 다시 필요하게 될 때 큐 관리자가 이를 새로 고칩니다. 요청 만기를 통해 휴면 중인 큐 관리자의 정보 요청이 증가되는 것을 방지합니다.

참고: 대형 클러스터인 경우, 동시에 자체적인 모든 정보를 다수의 큐 관리자가 자동으로 재전송하는 경우 문제가 될 수 있습니다. 278 페이지의 『[대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음](#)』의 내용을 참조하십시오.

관련 개념

278 페이지의 『[클러스터링: REFRESH CLUSTER 사용 우수 사례](#)』

REFRESH CLUSTER 명령을 사용하여 로컬에 보유된 클러스터에 대한 모든 정보를 제거하고 클러스터의 전체 저장소에서 이 정보를 다시 빌드합니다. 예외 상황을 제외하고는 이 명령을 사용해서는 안 됩니다. 사용해야 하는 경우 사용 방법에 대한 특수 고려사항이 있습니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

IBM WebSphere MQ 클러스터 관리

IBM WebSphere MQ 클러스터를 작성, 확장 및 유지보수할 수 있습니다.

IBM WebSphere MQ 클러스터 관리 방법에 대한 자세한 내용은 다음 하위 주제를 참조하십시오.

관련 개념

[클러스터](#)

[클러스터 작동 방법](#)

146 페이지의 『[클러스터링과 분산 큐잉의 비교](#)』

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

148 페이지의 『[클러스터의 컴포넌트](#)』

클러스터는 큐 관리자, 클러스터 저장소, 클러스터 채널 및 클러스터 큐로 구성됩니다.

관련 태스크

144 페이지의 『[큐 관리자 클러스터 구성](#)』

클러스터가 작업하는 방법, 클러스터 구성을 설계하는 방법을 찾고 단순한 클러스터를 설정하는 방법의 예를 알려면 이 토픽의 링크를 사용하십시오.

168 페이지의 『[새 클러스터 설정](#)』

다음 지시사항에 따라 클러스터 예를 설정합니다. 개별 지시사항에서는 TCP/IP, LU 6.2 및 단일 전송 큐 또는 다중 전송 큐에서의 클러스터 설정을 설명합니다. 한 큐 관리자에서 다른 큐 관리자로 메시지를 송신하여 클러스터가 작동하는지 테스트합니다.

새 클러스터 설정

다음 지시사항에 따라 클러스터 예를 설정합니다. 개별 지시사항에서는 TCP/IP, LU 6.2 및 단일 전송 큐 또는 다중 전송 큐에서의 클러스터 설정을 설명합니다. 한 큐 관리자에서 다른 큐 관리자로 메시지를 송신하여 클러스터가 작동하는지 테스트합니다.

시작하기 전에

- 다음 지시사항을 따르는 대신 IBM WebSphere MQ Explorer에서 제공하는 마법사 중 하나를 사용하여 이 태스크에서 작성되는 클러스터와 같은 클러스터를 작성할 수 있습니다. 큐 관리자 클러스터 폴더를 마우스 오른쪽 단추로 누른 다음 **새로 작성 > 큐 관리자 클러스터**를 누르고 마법사에 제공된 지시사항을 따르십시오.
- 클러스터 설정을 위해 수행한 단계에 대한 이해를 돕는 백그라운드 정보는 [150 페이지의 『클러스터 큐』](#), 채널 및 리스너를 참조하십시오.

이 태스크 정보

체인 저장소에 대한 새 IBM WebSphere MQ 네트워크를 설정합니다. 이 저장소에는 런던 및 뉴욕에 하나씩 두 개의 분기가 있습니다. 각 저장소의 데이터 및 애플리케이션은 개별 큐 관리자를 실행하는 시스템에 의해 호스팅됩니다. 이 두 큐 관리자를 LONDON 및 NEWYORK이라고 합니다. 인벤토리 애플리케이션은 큐 관리자 NEWYORK에 연결된 뉴욕의 시스템에서 실행됩니다. 이 애플리케이션은 INVENTQ 큐에 메시지가 도착하면 구동되며 NEWYORK에 의해 호스팅됩니다. 두 큐 관리자 LONDON 및 NEWYORK은 INVENTORY라는 클러스터에서 링크되어 둘 다 INVENTQ에 메시지를 넣을 수 있습니다.

168 페이지의 그림 29에서는 이 클러스터가 표시되는 모양을 보여줍니다.

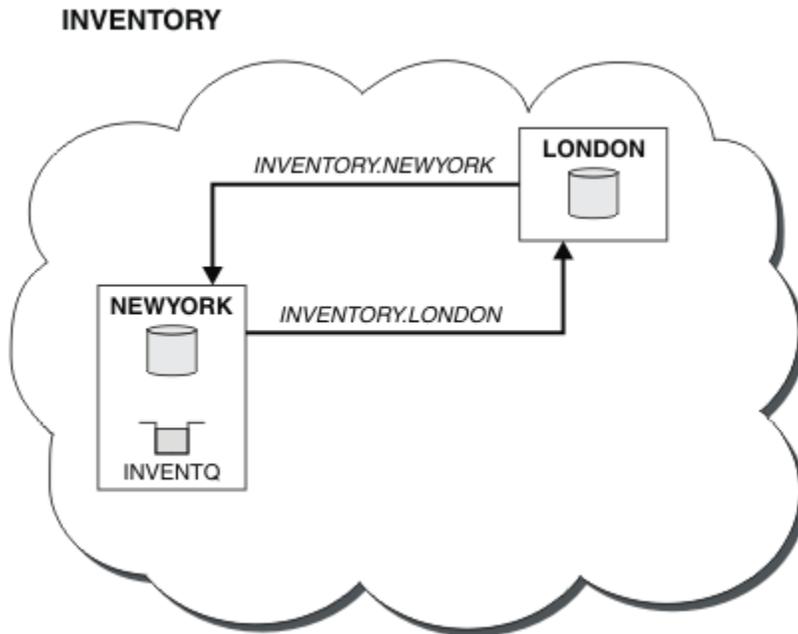


그림 29. 두 개의 큐 관리자가 포함된 INVENTORY 클러스터

z/OS에 없는 클러스터에서 서로 다른 클러스터 전송 큐를 사용하여 클러스터의 다른 큐 관리자에 메시지를 송신하도록 각 큐 관리자를 구성할 수 있습니다.

클러스터를 설정하기 위한 지시사항은 전송 프로토콜, 전송 큐 수 또는 플랫폼에 따라 약간씩 달라집니다. 세 가지 조합을 선택할 수 있습니다. 확인 프로시저는 모든 조합에 대해 동일합니다.

프로시저

- [169 페이지의 『TCP/IP를 사용하여 큐 관리자당 단일 전송 큐가 있는 클러스터 설정』](#)

- [171 페이지의 『큐 관리자당 다중 전송 큐를 사용하여 TCP/IP에서 클러스터 설정』](#)
- [174 페이지의 『z/OS 에서 LU 6.2 를 사용하여 클러스터 설정』](#)
- [176 페이지의 『클러스터 확인』](#)

결과

168 페이지의 그림 29에서는 이 태스크를 통해 설정되는 INVENTORY 클러스터를 보여줍니다.

INVENTORY는 분명히 작은 클러스터입니다. 그러나 개념 검증용으로 유용합니다. 이 클러스터를 이해하는 데 중요한 것은 이 클러스터가 향후 개선을 위해 제공하는 범위입니다.

관련 개념

[클러스터](#)

[클러스터 작동 방법](#)

[146 페이지의 『클러스터링과 분산 큐잉의 비교』](#)

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

[148 페이지의 『클러스터의 컴포넌트』](#)

클러스터는 큐 관리자, 클러스터 저장소, 클러스터 채널 및 클러스터 큐로 구성됩니다.

[167 페이지의 『IBM WebSphere MQ 클러스터 관리』](#)

IBM WebSphere MQ 클러스터를 작성, 확장 및 유지보수할 수 있습니다.

관련 태스크

[144 페이지의 『큐 관리자 클러스터 구성』](#)

클러스터가 작업하는 방법, 클러스터 구성을 설계하는 방법을 찾고 단순한 클러스터를 설정하는 방법의 예를 알려면 이 토픽의 링크를 사용하십시오.

TCP/IP를 사용하여 큐 관리자당 단일 전송 큐가 있는 클러스터 설정

시작하기 전에

- AIX, HP-UX, IBM i, Linux, Solaris, and Windows에서는 큐 관리자 속성 **DEFCLXQ**를 기본값인 SCTQ로 두어야 합니다.

이 태스크 정보

다음 단계에서는 전송 프로토콜 TCP/IP를 사용하여 AIX, HP-UX, IBM i, Linux, Solaris, and Windows에서 클러스터를 설정합니다.

프로시저

1. 클러스터의 조직 및 해당 이름을 결정하십시오.

두 큐 관리자 LONDON 및 NEWYORK을 클러스터에 링크하기로 결정했습니다. 두 개의 큐 관리자만 있는 클러스터는 분산 큐잉을 사용하는 네트워크에 대해서는 약간의 이익만 제공하지만 분산 큐잉을 시작하기 좋은 방법이며 향후 확장을 위한 범위를 제공합니다. 저장소의 새 분기를 여는 경우 클러스터에 새 큐 관리자를 쉽게 추가할 수 있습니다. 새 큐 관리자를 추가해도 기존 네트워크에 지장을 주지 않습니다. [177 페이지의 『클러스터에 큐 관리자 추가』](#)의 내용을 참조하십시오.

잠시 동안 실행 중인 애플리케이션만 인벤토리 애플리케이션입니다. 클러스터 이름은 INVENTORY입니다.

2. 전체 저장소를 보유할 큐 관리자를 결정하십시오.

모든 클러스터에서 전체 저장소를 보유할 최소 하나 이상의(가급적이면 둘) 큐 관리자를 지정해야 합니다. 이 예에는 둘 다 전체 저장소를 보유하는 두 큐 관리자인 LONDON 및 NEWYORK만 있습니다.

- a. 순서에 관계없이 나머지 단계를 수행할 수 있습니다.
- b. 단계에 따라 진행하는 경우 경고 메시지가 큐 관리자 로그에 기록될 수 있습니다. 이러한 메시지는 아직 추가지 않는 정의가 누락된 결과입니다.

Examples of the responses to the commands are shown in a box like this after each step in this task. These examples show the responses returned by WebSphere MQ for AIX. The responses vary on other platforms.

- c. 다음 단계를 진행하기 전에 큐 관리자가 시작되었는지 확인하십시오.
3. 저장소 정의를 추가하려면 큐 관리자 정의를 대체하십시오.

전체 저장소를 보유하는 각 큐 관리자에서 ALTER QMGR 명령을 사용하고 REPOS 속성을 지정하여 로컬 큐 관리자 정의를 대체하십시오.

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: Websphere MQ queue manager changed.
```

예를 들어, 다음을 입력하는 경우

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON이 전체 저장소로 변경됩니다.

4. 리스너를 정의하십시오.

클러스터 내 모든 큐 관리자에 대해 다른 큐 관리자의 네트워크 요청을 승인하는 리스너를 정의하십시오. LONDON 큐 관리자에서 다음 명령을 실행하십시오.

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

참고: 리스너를 정의할 때 CONNAME 필드에서 IP 주소를 사용 중이고 포트 번호가 기본 포트(1414)가 아닌 경우 포트 번호를 지정해야 합니다. 예를 들면, 다음과 같습니다.

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(1415)
```

CONTROL 속성은 큐 관리자가 시작 및 중지될 때 리스너가 시작 및 중지되도록 합니다.

리스너는 정의될 때 시작되지 않으므로 처음에는 다음 MQSC 명령을 사용하여 수동으로 시작해야 합니다.

```
START LISTENER(LONDON_LS)
```

각각에 대해 리스너 이름을 변경하여 클러스터에 있는 다른 모든 큐 관리자에 대해 유사한 명령을 발행하십시오.

리스너에 표시된 대로 이러한 리스너를 정의하는 몇 가지 방법이 있습니다.

5. LONDON 큐 관리자에 대해 CLUSRCVR 채널을 정의하십시오.

클러스터에 있는 모든 큐 관리자에서 큐 관리자가 메시지를 수신할 수 있는 클러스터 수신자 채널을 정의하십시오. CLUSRCVR은 큐 관리자의 연결 이름을 정의합니다. 연결 이름은 다른 큐 관리자가 참조할 수 있는 저장소에 저장됩니다. CLUSTER 키워드는 클러스터 내 다른 큐 관리자로부터 메시지를 수신하는 큐 관리자의 가용성을 표시합니다.

이 예에서 채널 이름은 INVENTORY.LONDON이고 연결 이름 (CONNAME)은 큐 관리자가 상주하는 시스템의 네트워크 주소로, LONDON.CHSTORE.COM입니다. 네트워크 주소는 IPv4 점분리 십진수 양식의 영숫자 DNS 호스트 이름 또는 IP 주소로 입력할 수 있습니다. 예를 들어, 192.0.2.0 또는 IPv6 16진 형식의 2001:DB8:0204:acff:fe97:2c34:fde0:3485입니다. 포트 번호는 지정되지 않으므로 기본 포트(1414)가 사용됩니다.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: Websphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

6. NEWYORK 큐 관리자에 대해 CLUSRCVR 채널을 정의하십시오.

채널 리스너가 기본 포트(일반적으로 1414)를 사용하고 있고 z/OS에서 클러스터가 큐 관리자를 포함하지 않는 경우 CONNAME을 생략할 수 있습니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

7. LONDON 큐 관리자에서 CLUSSDR 채널을 정의하십시오.

클러스터의 모든 큐 관리자에서 하나의 클러스터 송신자 채널을 정의하십시오. 큐 관리자는 클러스터 송신자 채널의 전체 저장소 큐 관리자 중 하나로 메시지를 송신합니다. 이 경우 두 개의 큐 관리자만 있으며 둘 다 전체 저장소를 보유하고 있습니다. 이들 큐 관리자에는 각각 다른 큐 관리자에 정의된 CLUSRCVR 채널을 가리키는 CLUSSDR 정의가 있습니다. CLUSSDR 정의에서 지정되는 채널 이름은 해당 CLUSRCVR 정의의 채널 이름과 일치해야 합니다. 큐 관리자에 동일한 클러스터에 있는 클러스터 수신자 채널 및 클러스터 송신자 채널에 대한 정의가 있으면 클러스터 송신자 채널이 시작됩니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: Websphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

8. NEWYORK 큐 관리자에서 CLUSSDR 채널을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. 클러스터 큐 INVENTQ를 정의하십시오.

CLUSTER 키워드를 지정하여 NEWYORK 큐 관리자에서 INVENTQ 큐를 정의하십시오.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: Websphere MQ queue created.
```

CLUSTER 키워드를 사용하면 큐를 클러스터에 알릴 수 있습니다. 큐를 정의하면 바로 클러스터의 다른 큐 관리자가 이 큐를 사용할 수 있습니다. 해당 큐 관리자는 리모트 큐 정의를 작성하지 않고도 큐에 메시지를 송신할 수 있습니다.

모든 정의가 완료되었습니다. 모든 플랫폼의 각 큐 관리자에서 리스너 프로그램을 시작하십시오. 리스너 프로그램은 수신되는 네트워크 요청을 대기하고 필요한 경우 클러스터 수신자 채널을 시작합니다.

큐 관리자당 다중 전송 큐를 사용하여 TCP/IP에서 클러스터 설정

이 태스크 정보

다음 단계에서는 전송 프로토콜 TCP/IP를 사용하여 AIX, HP-UX, IBM i, Linux, Solaris, and Windows에서 클러스터를 설정합니다. 저장소 큐 관리자는 다른 클러스터 전송 큐를 사용하여 클러스터 내에서 서로 및 다른 큐 관리자에게 메시지를 송신하도록 구성됩니다. 클러스터에 서로 다른 전송 큐를 사용하는 큐 관리자를 추가할 경우

179 페이지의 『클러스터에 큐 관리자 추가: 개별 전송 큐』 태스크를 수행하십시오. z/OS에서는 별도의 클러스터 전송 큐를 사용하도록 큐 관리자를 설정할 수 없습니다.

프로시저

1. 클러스터의 조직 및 해당 이름을 결정하십시오.

두 큐 관리자 LONDON 및 NEWYORK을 클러스터에 링크하기로 결정했습니다. 두 개의 큐 관리자만 있는 클러스터는 분산 큐잉을 사용하는 네트워크에 대해서는 약간의 이익만 제공하지만 분산 큐잉을 시작하기 좋은 방법이며 향후 확장을 위한 범위를 제공합니다. 저장소의 새 분기를 여는 경우 클러스터에 새 큐 관리자를 쉽게 추가할 수 있습니다. 새 큐 관리자를 추가해도 기존 네트워크에 지장을 주지 않습니다. [177 페이지의 『클러스터에 큐 관리자 추가』](#)의 내용을 참조하십시오.

잠시 동안 실행 중인 애플리케이션만 인벤토리 애플리케이션입니다. 클러스터 이름은 INVENTORY입니다.

2. 전체 저장소를 보유할 큐 관리자를 결정하십시오.

모든 클러스터에서 전체 저장소를 보유할 최소 하나 이상의(가급적이면 둘) 큐 관리자를 지정해야 합니다. 이 예에는 둘 다 전체 저장소를 보유하는 두 큐 관리자인 LONDON 및 NEWYORK만 있습니다.

- 순서에 관계없이 나머지 단계를 수행할 수 있습니다.
- 단계에 따라 진행하는 경우 경고 메시지가 큐 관리자 로그에 기록될 수 있습니다. 이러한 메시지는 아직 추가하지 않는 정의가 누락된 결과입니다.

Examples of the responses to the commands are shown in a box like this after each step in this task. These examples show the responses returned by WebSphere MQ for AIX. The responses vary on other platforms.

- 다음 단계를 진행하기 전에 큐 관리자가 시작되었는지 확인하십시오.

3. 저장소 정의를 추가하려면 큐 관리자 정의를 대체하십시오.

전체 저장소를 보유하는 각 큐 관리자에서 ALTER QMGR 명령을 사용하고 REPOS 속성을 지정하여 로컬 큐 관리자 정의를 대체하십시오.

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: Websphere MQ queue manager changed.
```

예를 들어, 다음을 입력하는 경우

- runmqsc LONDON
- ALTER QMGR REPOS(INVENTORY)

LONDON이 전체 저장소로 변경됩니다.

4. 각 목적지에 대해 별도의 클러스터 전송 큐를 작성하려면 큐 관리자 정의를 대체하십시오.

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

클러스터에 추가하는 각 큐 관리자에 대해 개별 전송 큐를 사용할지 여부를 결정하십시오. [177 페이지의 『클러스터에 큐 관리자 추가』](#) 및 [179 페이지의 『클러스터에 큐 관리자 추가: 개별 전송 큐』](#) 주제를 참조하십시오.

5. 리스너를 정의하십시오.

클러스터 내 모든 큐 관리자에 대해 다른 큐 관리자의 네트워크 요청을 승인하는 리스너를 정의하십시오. LONDON 큐 관리자에서 다음 명령을 실행하십시오.

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

참고: 리스너를 정의할 때 CONNAME 필드에서 IP 주소를 사용 중이고 포트 번호가 기본 포트(1414)가 아닌 경우 포트 번호를 지정해야 합니다. 예를 들면, 다음과 같습니다.

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(1415)
```

CONTROL 속성은 큐 관리자가 시작 및 중지될 때 리스너가 시작 및 중지되도록 합니다.

리스너는 정의될 때 시작되지 않으므로 처음에는 다음 MQSC 명령을 사용하여 수동으로 시작해야 합니다.

```
START LISTENER(LONDON_LS)
```

각각에 대해 리스너 이름을 변경하여 클러스터에 있는 다른 모든 큐 관리자에 대해 유사한 명령을 발행하십시오.

리스너에 표시된 대로 이러한 리스너를 정의하는 몇 가지 방법이 있습니다.

6. LONDON 큐 관리자에 대해 CLUSRCVR 채널을 정의하십시오.

클러스터에 있는 모든 큐 관리자에서 큐 관리자가 메시지를 수신할 수 있는 클러스터 수신자 채널을 정의하십시오. CLUSRCVR은 큐 관리자의 연결 이름을 정의합니다. 연결 이름은 다른 큐 관리자가 참조할 수 있는 저장소에 저장됩니다. CLUSTER 키워드는 클러스터 내 다른 큐 관리자로부터 메시지를 수신하는 큐 관리자의 가용성을 표시합니다.

이 예에서 채널 이름은 INVENTORY.LONDON이고 연결 이름 (CONNAME)은 큐 관리자가 상주하는 시스템의 네트워크 주소로, LONDON.CHSTORE.COM입니다. 네트워크 주소는 IPv4 점분리 십진수 양식의 영숫자 DNS 호스트 이름 또는 IP 주소로 입력할 수 있습니다. 예를 들어, 192.0.2.0 또는 IPv6 16진 형식의 2001:DB8:0204:acff:fe97:2c34:fde0:3485입니다. 포트 번호는 지정되지 않으므로 기본 포트 (1414)가 사용됩니다.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: Websphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

7. NEWYORK 큐 관리자에 대해 CLUSRCVR 채널을 정의하십시오.

채널 리스너가 기본 포트(일반적으로 1414)를 사용하고 있고 z/OS에서 클러스터가 큐 관리자를 포함하지 않는 경우 CONNAME을 생략할 수 있습니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

8. LONDON 큐 관리자에서 CLUSSDR 채널을 정의하십시오.

클러스터의 모든 큐 관리자에서 하나의 클러스터 송신자 채널을 정의하십시오. 큐 관리자는 클러스터 송신자 채널의 전체 저장소 큐 관리자 중 하나로 메시지를 송신합니다. 이 경우 두 개의 큐 관리자만 있으며 둘 다 전체 저장소를 보유하고 있습니다. 이들 큐 관리자에는 각각 다른 큐 관리자에 정의된 CLUSRCVR 채널을 가리키는 CLUSSDR 정의가 있습니다. CLUSSDR 정의에서 지정되는 채널 이름은 해당 CLUSRCVR 정의의 채널 이름과 일치해야 합니다. 큐 관리자에 동일한 클러스터에 있는 클러스터 수신자 채널 및 클러스터 송신자 채널에 대한 정의가 있으면 클러스터 송신자 채널이 시작됩니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: Websphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

9. NEWYORK 큐 관리자에서 CLUSSDR 채널을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

10. 클러스터 큐 INVENTQ를 정의하십시오.

CLUSTER 키워드를 지정하여 NEWYORK 큐 관리자에서 INVENTQ 큐를 정의하십시오.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: Websphere MQ queue created.
```

CLUSTER 키워드를 사용하면 큐를 클러스터에 알릴 수 있습니다. 큐를 정의하면 바로 클러스터의 다른 큐 관리자가 이 큐를 사용할 수 있습니다. 해당 큐 관리자는 리모트 큐 정의를 작성하지 않고도 큐에 메시지를 송신할 수 있습니다.

모든 정의가 완료되었습니다. 모든 플랫폼의 각 큐 관리자에서 리스너 프로그램을 시작하십시오. 리스너 프로그램은 수신되는 네트워크 요청을 대기하고 필요한 경우 클러스터 수신자 채널을 시작합니다.

z/OS 에서 LU 6.2 를 사용하여 클러스터 설정

프로시저

1. 클러스터의 조직 및 해당 이름을 결정하십시오.

두 큐 관리자 LONDON 및 NEWYORK을 클러스터에 링크하기로 결정했습니다. 두 개의 큐 관리자만 있는 클러스터는 분산 큐잉을 사용하는 네트워크에 대해서는 약간의 이익만 제공하지만 분산 큐잉을 시작하기 좋은 방법이며 향후 확장을 위한 범위를 제공합니다. 저장소의 새 분기를 여는 경우 클러스터에 새 큐 관리자를 쉽게 추가할 수 있습니다. 새 큐 관리자를 추가해도 기존 네트워크에 지장을 주지 않습니다. 177 페이지의 『클러스터에 큐 관리자 추가』의 내용을 참조하십시오.

잠시 동안 실행 중인 애플리케이션만 인벤토리 애플리케이션입니다. 클러스터 이름은 INVENTORY입니다.

2. 전체 저장소를 보유할 큐 관리자를 결정하십시오.

모든 클러스터에서 전체 저장소를 보유할 최소 하나 이상의(가급적이면 둘) 큐 관리자를 지정해야 합니다. 이 예에는 둘 다 전체 저장소를 보유하는 두 큐 관리자인 LONDON 및 NEWYORK만 있습니다.

a. 순서에 관계없이 나머지 단계를 수행할 수 있습니다.

b. 단계에 따라 진행하는 중에 z/OS 시스템 콘솔에 경고 메시지가 기록될 수 있습니다. 이러한 메시지는 아직 추가하지 않는 정의가 누락된 결과입니다.

c. 다음 단계를 진행하기 전에 큐 관리자가 시작되었는지 확인하십시오.

3. 저장소 정의를 추가하려면 큐 관리자 정의를 대체하십시오.

전체 저장소를 보유하는 각 큐 관리자에서 ALTER QMGR 명령을 사용하고 REPOS 속성을 지정하여 로컬 큐 관리자 정의를 대체하십시오.

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: Websphere MQ queue manager changed.
```

예를 들어, 다음을 입력하는 경우

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON이 전체 저장소로 변경됩니다.

4. 리스너를 정의하십시오.

리스너는 정의될 때 시작되지 않으므로 처음에는 다음 MQSC 명령을 사용하여 수동으로 시작해야 합니다.

```
START LISTENER(LONDON_LS)
```

각각에 대해 리스너 이름을 변경하여 클러스터에 있는 다른 모든 큐 관리자에 대해 유사한 명령을 발행하십시오.

5. LONDON 큐 관리자에 대해 CLUSRCVR 채널을 정의하십시오.

클러스터에 있는 모든 큐 관리자에서 큐 관리자가 메시지를 수신할 수 있는 클러스터 수신자 채널을 정의하십시오. CLUSRCVR은 큐 관리자의 연결 이름을 정의합니다. 연결 이름은 다른 큐 관리자가 참조할 수 있는 저장소에 저장됩니다. CLUSTER 키워드는 클러스터 내 다른 큐 관리자로부터 메시지를 수신하는 큐 관리자의 가용성을 표시합니다.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
AMQ8014: Websphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

6. NEWYORK 큐 관리자에 대해 CLUSRCVR 채널을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager NEWYORK')
```

7. LONDON 큐 관리자에서 CLUSSDR 채널을 정의하십시오.

클러스터의 모든 큐 관리자에서 하나의 클러스터 송신자 채널을 정의하십시오. 큐 관리자는 클러스터 송신자 채널의 전체 저장소 큐 관리자 중 하나로 메시지를 송신합니다. 이 경우 두 개의 큐 관리자만 있으며 둘 다 전체 저장소를 보유하고 있습니다. 이들 큐 관리자에는 각각 다른 큐 관리자에 정의된 CLUSRCVR 채널을 가리키는 CLUSSDR 정의가 있습니다. CLUSSDR 정의에서 지정되는 채널 이름은 해당 CLUSRCVR 정의의 채널 이름과 일치해야 합니다. 큐 관리자에 동일한 클러스터에 있는 클러스터 수신자 채널 및 클러스터 송신자 채널에 대한 정의가 있으면 클러스터 송신자 채널이 시작됩니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(CPIC) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: Websphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

8. NEWYORK 큐 관리자에서 CLUSSDR 채널을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. 클러스터 큐 INVENTQ를 정의하십시오.

CLUSTER 키워드를 지정하여 NEWYORK 큐 관리자에서 INVENTQ 큐를 정의하십시오.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: Websphere MQ queue created.
```

CLUSTER 키워드를 사용하면 큐를 클러스터에 알릴 수 있습니다. 큐를 정의하면 바로 클러스터의 다른 큐 관리자가 이 큐를 사용할 수 있습니다. 해당 큐 관리자는 리모트 큐 정의를 작성하지 않고도 큐에 메시지를 송신할 수 있습니다.

모든 정의가 완료되었습니다. 모든 플랫폼의 각 큐 관리자에서 리스너 프로그램을 시작하십시오. 리스너 프로그램은 수신되는 네트워크 요청을 대기하고 필요한 경우 클러스터 수신자 채널을 시작합니다.

클러스터 확인

이 태스크 정보

다음 방법 중 하나 이상을 사용하여 클러스터를 확인할 수 있습니다.

1. 클러스터 및 채널 속성을 표시하는 관리 명령을 실행합니다.
2. 샘플 프로그램을 실행하여 클러스터 큐에서 메시지를 송신 및 수신합니다.
3. 요청 메시지를 클러스터에 송신하고 비클러스터 응답 큐에 응답 메시지로 응답하는 자체 프로그램을 작성합니다.

프로시저

DISPLAY **runmqsc** 명령을 발행하여 클러스터를 확인하십시오.

표시되는 응답은 다음 단계의 응답과 유사해야 합니다.

1. NEWYORK 큐 관리자에서 **DISPLAY CLUSQMGR** 명령을 실행하십시오.

```
dis clusqmgr(*)
```

```
1 : dis clusqmgr(*)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(NEWYORK)          CLUSTER(INVENTORY)
CHANNEL(INVENTORY.NEWYORK)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(LONDON)          CLUSTER(INVENTORY)
CHANNEL(INVENTORY.LONDON)
```

2. NEWYORK 큐 관리자에서 **DISPLAY CHANNEL STATUS** 명령을 실행하십시오.

```
dis chstatus(*)
```

```

1 : dis chstatus(*)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.NEWYORK) XMITQ( )
CONNNAME(192.0.2.0) CURRENT
CHLTYPE(CLUSRCVR) STATUS(RUNNING)
RQMNAME(LONDON)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.LONDON) XMITQ(SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON)
CONNNAME(192.0.2.1) CURRENT
CHLTYPE(CLUSSDR) STATUS(RUNNING)
RQMNAME(LONDON)

```

amqsput을 사용하여 두 큐 관리자 사이에서 메시지를 송신하십시오.

3. LONDON에서 **amqsput INVENTQ LONDON** 명령을 실행하십시오.

몇 가지 메시지를 입력하고 공백 행을 입력하십시오.

4. NEWYORK에서 **amqsget INVENTQ NEWYORK** 명령을 실행하십시오.

이제 LONDON에 입력한 메시지가 표시됩니다. 15초 후에 프로그램이 종료됩니다.

자체 프로그램을 사용하여 두 큐 관리자 사이에서 메시지를 송신하십시오.

다음 단계에서 LONDON은 메시지를 NEWYORK의 INVENTQ에 넣고 해당 큐 LONDON_reply에서 응답을 수신합니다.

5. LONDON에서 메시지를 클러스터 큐에 넣으십시오.

- a) LONDON_reply이라는 로컬 큐를 정의하십시오.
- b) MQOPEN 옵션을 MQOO_OUTPUT로 설정하십시오.
- c) MQOPEN 호출을 실행하여 INVENTQ 큐를 여십시오.
- d) 메시지 설명자의 ReplyToQ 이름을 LONDON_reply로 설정하십시오.
- e) 메시지를 넣으려면 MQPUT 호출을 실행하십시오.
- f) 메시지를 커미트하십시오.

6. NEWYORK의 클러스터 큐에서 메시지를 수신하고 응답을 응답 큐에 넣으십시오.

- a) MQOPEN 옵션을 MQOO_BROWSE로 설정하십시오.
- b) MQOPEN 호출을 실행하여 INVENTQ 큐를 여십시오.
- c) MQGET 호출을 실행하여 INVENTQ에서 메시지를 가져오십시오.
- d) 메시지 설명자에서 ReplyToQ 이름을 검색합니다.
- e) 오브젝트 설명자의 ObjectName 필드에 ReplyToQ 이름을 입력하십시오.
- f) MQOPEN 옵션을 MQOO_OUTPUT로 설정하십시오.
- g) MQOPEN 호출을 실행하여 큐 관리자 LONDON에서 LONDON_reply 을 여십시오.
- h) MQPUT 호출을 실행하여 LONDON_reply에 메시지를 넣으십시오.

7. LONDON에서 응답을 수신하십시오.

- a) MQOPEN 옵션을 MQOO_BROWSE로 설정하십시오.
- b) MQOPEN 호출을 실행하여 LONDON_reply 큐를 여십시오.
- c) MQGET 호출을 실행하여 LONDON_reply에서 메시지를 가져오십시오.

클러스터에 큐 관리자 추가

다음 지시사항에 따라 사용자가 작성한 클러스터에 큐 관리자를 추가합니다. 클러스터 큐 및 토픽에 대한 메시지는 단일 클러스터 전송 큐 SYSTEM.CLUSTER.TRANSMIT.QUEUE를 사용하여 전송됩니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 작업을 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- INVENTORY 클러스터가 168 페이지의 『새 클러스터 설정』에서 설명된 대로 설정됩니다. 이 클러스터는 둘 다 전체 저장소를 보유하는 두 개의 큐 관리자 LONDON 및 NEWYORK을 포함합니다.
- 큐 관리자 PARIS는 기본 설치에 속합니다. 그렇지 않은 경우 `setmqenv` 명령을 실행하여 PARIS가 속한 설치에 대한 명령 환경을 설정해야 합니다.
- TCP 연결성은 세 시스템 사이에 모두 존재하며 큐 관리자는 큐 관리자의 제어 하에 시작하는 TCP 리스너로 구성됩니다.

이 태스크 정보

1. 체인 저장소의 새 분기가 Paris에서 설정되고 PARIS라고 하는 큐 관리자를 클러스터에 추가하려고 합니다.
2. 큐 관리자 PARIS는 메시지를 INVENTQ 큐에 넣어 New York의 시스템에서 실행되는 애플리케이션에 인벤토리 업데이트를 송신합니다.

다음 단계에 따라 큐 관리자를 클러스터에 추가하십시오.

프로시저

1. PARIS가 처음 참조할 전체 저장소를 결정하십시오.

클러스터 내 모든 큐 관리자는 전체 저장소 중 하나를 참조해야 합니다. 큐 관리자는 전체 저장소에서 클러스터에 대한 정보를 수집하고 자체 부분 저장소를 빌드합니다. 저장소 중 하나를 전체 저장소로 선택하십시오. 새 큐 관리자는 클러스터에 추가되는 즉시 다른 저장소에 대해서도 학습합니다. 큐 관리자 변경에 대한 정보는 두 저장소에 직접 송신됩니다. 이 예에서는 순전히 지리적 이유로 PARIS를 큐 관리자 LONDON에 링크합니다.

참고: 큐 관리자 PARIS가 시작된 후에는 순서에 관계없이 나머지 단계를 수행하십시오.

2. 큐 관리자 PARIS에서 CLUSRCVR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 수신할 수 있는 클러스터 수신자 채널을 정의해야 합니다. PARIS에서 다음을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

클러스터 수신자 채널은 클러스터 INVENTORY에 있는 다른 큐 관리자로부터 메시지를 수신할 큐 관리자의 가용성을 알립니다. 클러스터 수신자 채널 INVENTORY.PARIS에 대한 송신 측의 다른 큐 관리자에서 정의를 작성하지 않아도 됩니다. 기타 정의는 필요한 경우 자동으로 작성됩니다.

3. 큐 관리자 PARIS에서 CLUSSDR 채널을 정의하십시오.

클러스터에 있는 모든 큐 관리자는 초기 전체 저장소로 메시지를 송신할 수 있는 클러스터 송신자 채널 하나를 정의해야 합니다.

PARIS에서, 네트워크 주소가 LONDON.CHSTORE.COM인 큐 관리자에 대한 채널 INVENTORY.LONDON의 정의를 다음과 같이 작성하십시오.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. 옵션: 이 큐 관리자가 클러스터에 다시 결합할 경우 일부 추가 단계를 수행하십시오.

- a) 이전에 같은 클러스터에서 제거된 큐 관리자를 클러스터에 추가하는 경우, 이제 클러스터 멤버로 표시하는지 확인하십시오. 그렇지 않은 경우 다음 추가 단계를 완료하십시오.

- i) 추가할 큐 관리자에서 **REFRESH CLUSTER** 명령을 발행하십시오. 이 단계는 클러스터 채널을 중지하고 로컬 클러스터 캐시에 클러스터의 나머지 부분에서 최신임이 보장되는 새로운 순서 번호 세트를 제공합니다.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

참고: 대형 클러스터의 경우, **REFRESH CLUSTER** 명령을 사용하면 진행 중에 클러스터에 혼란을 줄 수 있으며, 클러스터 오브젝트가 모든 관심 있는 큐 관리자에 자동으로 상태 업데이트를 보낸 이후 27일 간격으로 다시 수행됩니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오.

- ii) CLUSSDR 채널을 다시 시작하십시오(예: **START CHANNEL** 명령 사용).

- iii) CLUSRCVR 채널을 재시작하십시오.

- b) 클러스터가 발행/구독 클러스터이고 재결합 큐 관리자에게 구독물이 있는 경우 다음 명령을 실행하여 클러스터 전체에서 프록시 구독이 올바르게 동기화되도록 합니다.

```
REFRESH QMGR TYPE(PROXYSUB)
```

결과

다음 그림은 이 작업을 통해 설정되는 클러스터를 보여줍니다.

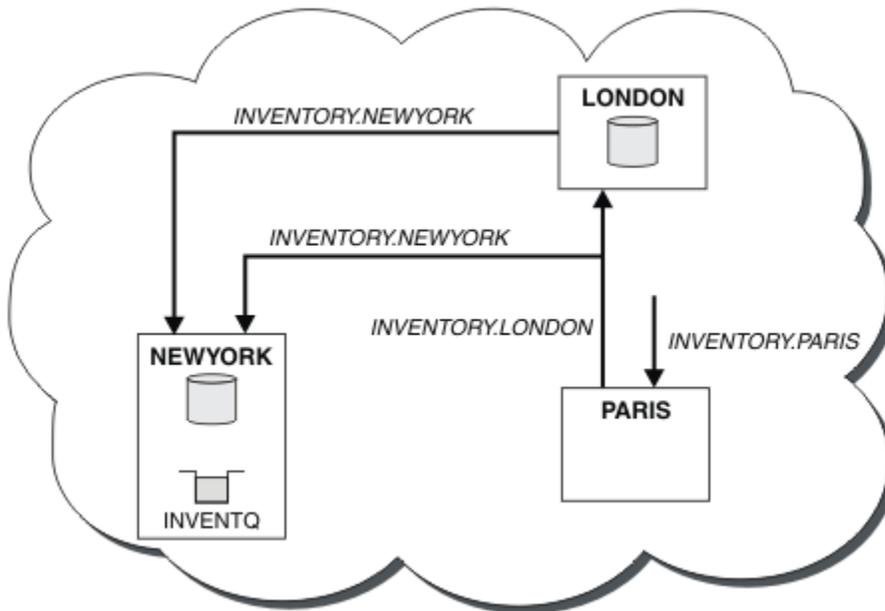


그림 30. 세 개의 큐 관리자가 있는 *INVENTORY* 클러스터

CLUSRCVR 정의 및 CLUSSDR 정의 두 개만 작성하여 큐 관리자 PARIS를 클러스터에 추가했습니다.

이제 PARIS 큐 관리자는 LONDON의 전체 저장소에서 INVENTQ 큐가 큐 관리자 NEWYORK에 의해 호스트됨을 학습합니다. Paris의 시스템에 의해 호스팅되는 애플리케이션이 INVENTQ에 메시지를 넣으려고 시도하면 PARIS가 클러스터 수신자 채널 INVENTORY.NEWYORK에 연결하기 위한 클러스터 송신자 채널을 자동으로 정의합니다. 이 애플리케이션은 해당 큐 관리자 이름이 대상 큐 관리자로 지정되고 응답 대상 큐가 제공되는 경우에 응답을 수신할 수 있습니다.

클러스터에 큐 관리자 추가: 개별 전송 큐

다음 지시사항에 따라 사용자가 작성한 클러스터에 큐 관리자를 추가합니다. 클러스터 큐 및 토픽에 대한 메시지는 다중 클러스터 전송 큐를 사용하여 전송됩니다.

시작하기 전에

- 큐 관리자는 z/OS가 아닌 다른 플랫폼에서 정의됩니다.
- 큐 관리자는 클러스터의 멤버가 아닙니다.
- 클러스터가 존재하고 이 큐 관리자가 직접 연결할 수 있는 전체 저장소가 있으며 이 저장소가 사용 가능합니다. 클러스터를 작성하기 위한 단계는 [168 페이지의 『새 클러스터 설정』](#)의 내용을 참조하십시오.

이 태스크 정보

이 태스크는 [177 페이지의 『클러스터에 큐 관리자 추가』](#)에 대한 대안으로, 클러스터 메시지를 단일 전송 큐에 배치하는 클러스터에 큐 관리자를 추가합니다.

이 태스크에서는 각 클러스터 송신자 채널에 대해 개별 클러스터 전송 큐를 자동으로 작성하는 클러스터에 큐 관리자를 추가합니다.

큐의 정의 수를 작게 유지하기 위해 기본값은 단일 전송 큐를 사용하는 것입니다. 여러 큐 관리자와 여러 클러스터를 목적으로 하는 트래픽을 모니터하려는 경우 개별 전송 큐를 사용하는 것이 유리합니다. 또한 다른 목적으로의 트래픽을 분리하여 격리 또는 성능 목표를 달성할 수도 있습니다.

프로시저

1. 기본 클러스터 채널 전송 큐 유형을 대체하십시오.

큐 관리자 PARIS를 다음과 같이 대체하십시오.

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

큐 관리자는 큐 관리자에게 메시지를 송신하기 위한 클러스터 송신자 채널을 작성할 때마다 클러스터 전송 큐를 작성합니다. 이 전송 큐는 이 클러스터 송신자 채널에서만 사용됩니다. 전송 큐는 영구적이고 동적입니다. 이름이 SYSTEM.CLUSTER.TRANSMIT.ChannelName인 모델 큐 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE에서 작성됩니다.



주의: 제품의 이전 버전에서 업그레이드된 큐 관리자와 함께 전용 SYSTEM.CLUSTER.TRANSMIT.QUEUES을 사용하는 경우, SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE에 [공유/공유하기](#) 옵션이 [공유](#)로 설정되어 있는지 확인하십시오.

2. PARIS가 처음 참조할 전체 저장소를 결정하십시오.

클러스터 내 모든 큐 관리자는 전체 저장소 중 하나를 참조해야 합니다. 큐 관리자는 전체 저장소에서 클러스터에 대한 정보를 수집하고 자체 부분 저장소를 빌드합니다. 저장소 중 하나를 전체 저장소로 선택하십시오. 새 큐 관리자는 클러스터에 추가되는 즉시 다른 저장소에 대해서도 학습합니다. 큐 관리자 변경에 대한 정보는 두 저장소에 직접 송신됩니다. 이 예에서는 순전히 지리적 이유로 PARIS를 큐 관리자 LONDON에 링크합니다.

참고: 큐 관리자 PARIS가 시작된 후에는 순서에 관계없이 나머지 단계를 수행하십시오.

3. 큐 관리자 PARIS에서 CLUSRCVR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 수신할 수 있는 클러스터 수신자 채널을 정의해야 합니다. PARIS에서 다음을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

클러스터 수신자 채널은 클러스터 INVENTORY에 있는 다른 큐 관리자로부터 메시지를 수신할 큐 관리자의 가용성을 알립니다. 클러스터 수신자 채널 INVENTORY.PARIS에 대한 송신 측의 다른 큐 관리자에서 정의를 작성하지 않아도 됩니다. 기타 정의는 필요한 경우 자동으로 작성됩니다.

4. 큐 관리자 PARIS에서 CLUSSDR 채널을 정의하십시오.

클러스터에 있는 모든 큐 관리자는 초기 전체 저장소로 메시지를 송신할 수 있는 클러스터 송신자 채널 하나를 정의해야 합니다.

PARIS에서, 네트워크 주소가 LONDON.CHSTORE.COM인 큐 관리자에 대한 채널 INVENTORY.LONDON의 정의를 다음과 같이 작성하십시오.

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

큐 관리자는 모델 큐 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE로부터 영구적 동적 클러스터 전송 큐 SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON을 자동으로 작성합니다. 큐 관리자는 전송 큐의 CLCHNAME 속성을 INVENTORY.LONDON으로 설정합니다.

결과

다음 그림은 이 태스크를 통해 설정되는 클러스터를 보여줍니다.

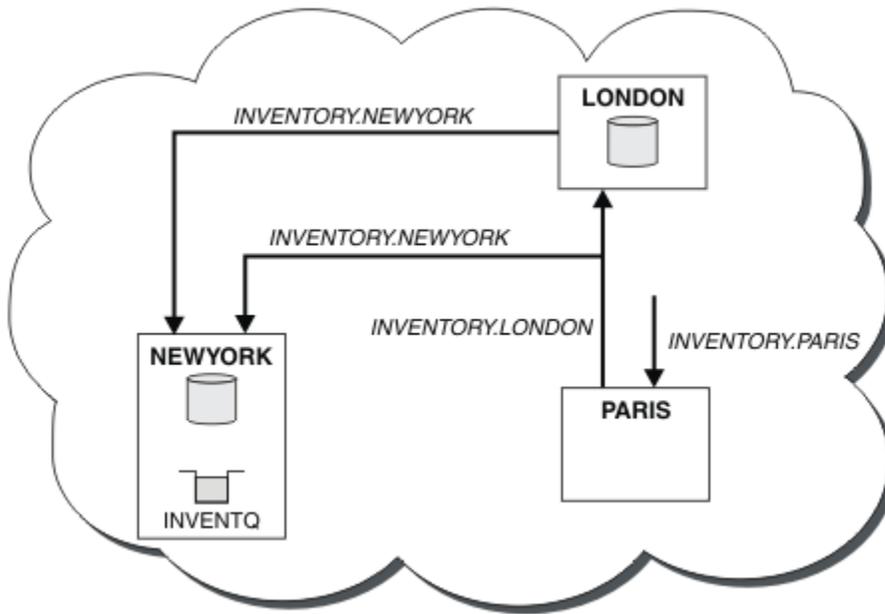


그림 31. 세 개의 큐 관리자가 있는 INVENTORY 클러스터

CLUSRCVR 정의 및 CLUSSDR 정의 두 개만 작성하여 큐 관리자 PARIS를 클러스터에 추가했습니다.

이제 PARIS 큐 관리자는 LONDON의 전체 저장소에서 INVENTQ 큐가 큐 관리자 NEWYORK에 의해 호스트됨을 학습합니다. Paris의 시스템에 의해 호스팅되는 애플리케이션이 INVENTQ에 메시지를 넣으려고 시도하면 PARIS가 클러스터 수신자 채널 INVENTORY.NEWYORK에 연결하기 위한 클러스터 송신자 채널을 자동으로 정의합니다. 이 애플리케이션은 해당 큐 관리자 이름이 대상 큐 관리자로 지정되고 응답 대상 큐가 제공되는 경우에 응답을 수신할 수 있습니다.

게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 클러스터된 큐 리모트 정의 및 개별 송신자 채널과 전송 큐를 사용합니다.

시작하기 전에

해당 태스크의 단계를 따라 197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』의 198 페이지의 그림 37에 표시된 겹쳐진 클러스터를 구성하십시오.

이 태스크 정보

솔루션은 분산 큐잉을 사용하여 게이트웨이 큐 관리자에서 다른 메시지 트래픽으로부터 Server App 애플리케이션에 대한 메시지를 분리합니다. QM1에서 클러스터된 리모트 큐 정의를 정의하여 메시지를 다른 전송 큐와 다른 채널로 우회시켜야 합니다. 리모트 큐 정의는 QM3의 Q1에 대한 메시지만 저장하는 특정 전송 큐에 대한 참조를 포함해야 합니다. 182 페이지의 그림 32에서 클러스터 큐 알리어스 Q1A는 리모트 큐 정의 Q1R에 의해 보완되고 전송 큐 및 송신자 채널이 추가됩니다.

이 솔루션에서 응답 메시지는 공통 SYSTEM.CLUSTER.TRANSMIT.QUEUE를 사용하여 리턴됩니다.

이 솔루션의 장점은 같은 클러스터에 있는 같은 큐 관리자에서 여러 목적지 큐에 대한 트래픽을 쉽게 분리할 수 있다는 것입니다. 이 솔루션의 단점은 서로 다른 큐 관리자에서 Q1의 여러 사본 사이에 클러스터 워크로드 밸런싱을 사용할 수 없다는 것입니다. 이 단점을 극복하려면 184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』의 내용을 참조하십시오. 또한 한 전송 큐에서 다른 전송 큐로의 전환도 관리해야 합니다.

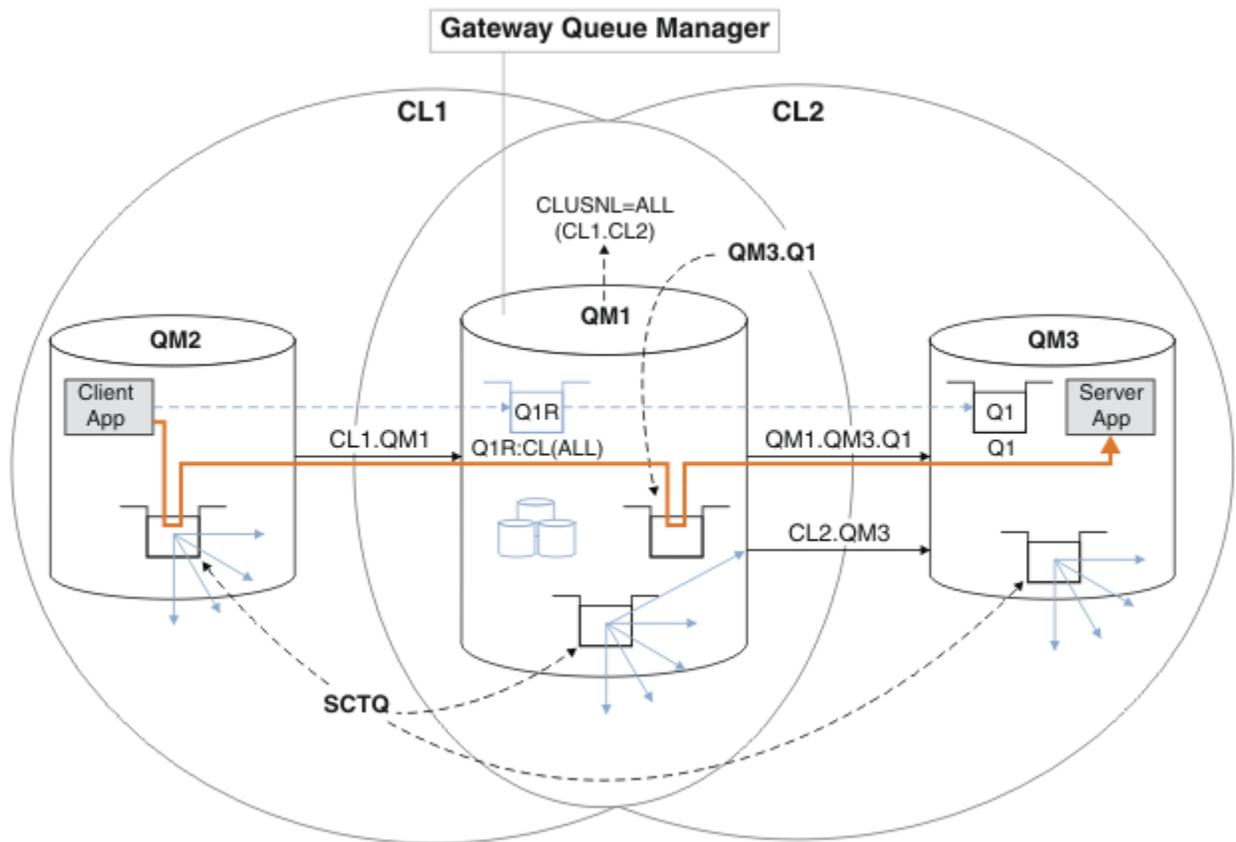


그림 32. 리모트 큐 정의를 사용하여 허브 및 스포크 클러스터 아키텍처에 배치되는 클러스터 서버 애플리케이션

프로시저

1. Q1에 대한 메시지 트래픽을 게이트웨이 큐 관리자로부터 분리하기 위한 채널을 작성하십시오.

a) 게이트웨이 큐 관리자 QM1에서 대상 큐 관리자 QM3에 대한 송신자 채널을 작성하십시오.

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(SDR) CONNAME(QM3HostName(1413)) XMITQ(QM3.Q1) REPLACE
```

b) 대상 큐 관리자 QM3에서 수신자 채널을 작성하십시오.

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(RCVR) REPLACE
```

2. 게이트웨이 큐 관리자에서 Q1에 대한 메시지 트래픽용 전송 큐를 작성하십시오.

```
DEFINE QLOCAL(QM3.Q1) USAGE(XMITQ) REPLACE
START CHANNEL(QM1.QM3.Q1)
```

전송 큐와 연관된 채널을 시작할 때 이 전송 큐를 채널과 연관시키십시오. 전송 큐가 채널과 연관되는 즉시 채널이 자동으로 시작됩니다.

3. 게이트웨이 큐 관리자에서 클러스터된 리모트 큐 정의를 사용하여 Q1에 대한 클러스터된 큐 알리어스 정의를 보완하십시오.

```
DEFINE QREMOTE CLUSNL(ALL) RNAME(Q1) RQMNAME(QM3) XMITQ(QM3.Q1) REPLACE
```

다음에 수행할 작업

게이트웨이 큐 관리자 QM1에서 클러스터된 큐 리모트 정의 Q1R을 통해 QM2에서 QM3의 Q1으로 메시지를 송신하여 구성을 테스트하십시오.

1. 메시지를 넣으려면 QM2에서 샘플 프로그램 **amqspu**t를 실행하십시오.

```
C:\IBM\MQ>amqspu Q1R QM2
Sample AMQSPUT0 start
target queue is Q1R
Sample request message from QM2 to Q1 using Q1R
```

```
Sample AMQSPUT0 end
```

2. QM3의 Q1에서 메시지를 가져오려면 샘플 프로그램 **amqsget**를 실행하십시오.

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1R>
no more messages
Sample AMQSGET0 end
```

관련 개념

[145 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』](#)

애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드는 클러스터 큐에 대해 원격으로 검사, SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사 또는 클러스터 큐나 클러스터 큐 관리자의 로컬 프로파일에 대해 검사입니다.

[258 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』](#)

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

관련 태스크

[179 페이지의 『클러스터에 큐 관리자 추가: 개별 전송 큐』](#)

다음 지시사항에 따라 사용자가 작성한 클러스터에 큐 관리자를 추가합니다. 클러스터 큐 및 토픽에 대한 메시지는 다중 클러스터 전송 큐를 사용하여 전송됩니다.

[197 페이지의 『게이트웨이 큐 관리자 두 개의 중첩 클러스터 작성』](#)

태스크의 지시사항에 따라 게이트웨이 큐 관리자가 포함된 중첩 클러스터를 구성합니다. 한 애플리케이션에 대한 메시지를 클러스터 내 다른 애플리케이션에 대한 메시지로부터 격리하는 다음 예에 대한 시작점으로 이 클러스터를 사용합니다.

[181 페이지의 『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』](#)

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 클러스터된 큐 리모트 정의 및 개별 송신자 채널과 전송 큐를 사용합니다.

[202 페이지의 『메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경』](#)

클러스터된 큐 또는 전송 큐의 토픽에 대해 큐 관리자가 메시지를 저장하는 기본 방식을 변경할 수 있습니다. 기본값을 변경하여 게이트웨이 큐 관리자에서 클러스터 메시지를 격리하는 방식을 제공합니다.

184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 추가 클러스터 전송 큐를 사용하여 클러스터의 단일 큐 관리자로 메시지 트래픽을 분리합니다.

187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 솔루션은 추가 클러스터를 사용하여 메시지를 특정 클러스터 큐로 분리합니다.

261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』

클러스터 전송 큐의 선택에 대한 지침입니다. 하나의 공용 기본 큐, 별도의 기본 큐, 또는 수동으로 정의된 큐를 구성할 수 있습니다. 다중 클러스터 전송 큐 구성은 z/OS 이외의 플랫폼에 적용됩니다.

게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 추가 클러스터 전송 큐를 사용하여 클러스터의 단일 큐 관리자로 메시지 트래픽을 분리합니다.

시작하기 전에

1. 게이트웨이 큐 관리자는 Version 7.5 이상 및 z/OS 이외의 플랫폼에 있어야 합니다.
2. 해당 태스크의 단계를 따라 197 페이지의 『게이트웨이 큐 관리자 두 개의 중첩 클러스터 작성』의 198 페이지의 그림 37에 표시된 겹쳐진 클러스터를 구성하십시오.

이 태스크 정보

게이트웨이 큐 관리자 QM1에서 전송 큐를 추가하고 그 큐 속성 CLCHNAME을 설정하십시오. CLCHNAME을 QM3의 클러스터-수신자 채널의 이름으로 설정하십시오. 185 페이지의 그림 33의 내용을 참조하십시오.

이 솔루션에는 181 페이지의 『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』에 설명되어 있는 솔루션을 넘어서는 수 많은 장점이 있습니다.

- 적은 수의 추가 정의가 필요합니다.
- 같은 클러스터 CL2의 여러 다른 큐 관리자에서 대상 큐 Q1의 여러 사본 간 워크로드 밸런싱을 지원합니다.
- 채널이 재시작되면 게이트웨이 큐 관리자가 메시지를 손실하지 않고 새 구성으로 자동 전환됩니다.
- 게이트웨이 큐 관리자가 메시지를 수신한 순서와 같은 순서로 계속해서 메시지를 전달합니다. 이는 스위치가 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 있는 QM3의 큐 Q1에 대한 메시지와 함께 발생하는 경우에도 그렇습니다.

185 페이지의 그림 33의 클러스터 메시지 트래픽을 격리하기 위한 구성은 181 페이지의 『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』에서 리모트 큐를 사용하는 구성만큼 트래픽 격리 능력이 뛰어나지는 않습니다. CL2의 큐 관리자 QM3이 (가) 여러 개의 다른 클러스터 큐 및 서버 애플리케이션을 호스팅하는 경우, 이러한 모든 큐는 QM1를 QM3에 연결하는 클러스터 채널 CL2.QM3을 공유합니다. The additional flows are illustrated in 185 페이지의 그림 33 by the gray arrow representing potential cluster message traffic from the SYSTEM.CLUSTER.TRANSMIT.QUEUE to the cluster-sender channel CL2.QM3.

해결책은 큐 관리자를 특정 클러스터의 한 클러스터 큐를 호스팅하는 것으로만 제한하는 것입니다. 큐 관리자가 이미 다수의 클러스터 큐를 호스팅 중이고 이 제한을 충족하도록 하려면, 다른 큐 관리자를 작성하거나 다른 클러스터를 작성해야 합니다. 187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』의 내용을 참조하십시오.

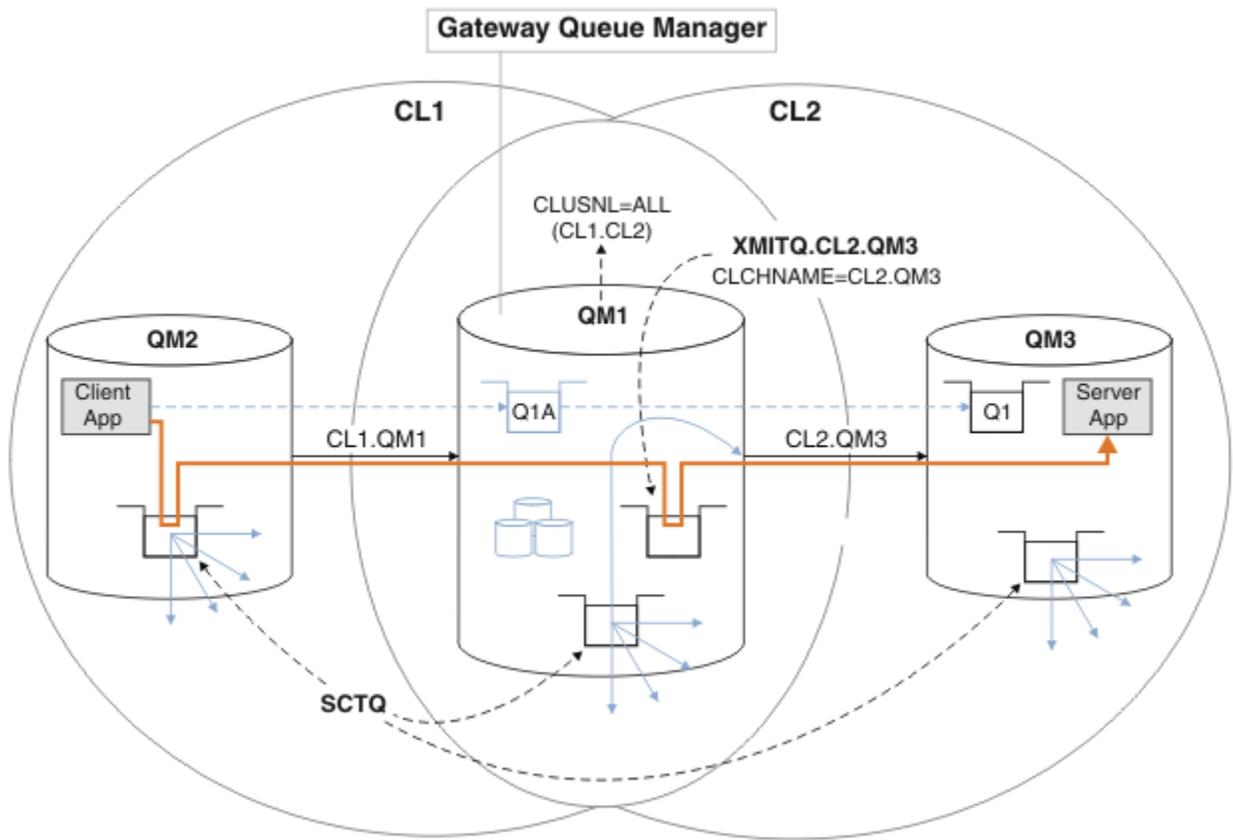


그림 33. 추가 클러스터 전송 큐를 사용하여 허브 및 스포크 아키텍처에 배치되는 클라이언트 서버 애플리케이션

프로시저

1. 게이트웨이 큐 관리자 QM1에서 클러스터 송신자 채널 CL2.QM3에 대한 추가 클러스터 전송 큐를 작성하십시오.

```
*... on QM1
DEFINE QLOCAL(XMITQ.CL2.QM3) USAGE(XMITQ) CLCHNAME(CL2.QM3)
```

2. 전송 큐 XMITQ.CL2.QM3 사용으로 전환하십시오.

- a) 클러스터 송신자 채널 CL2.QM3를 중지하십시오.

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

응답은 명령이 승인됨입니다.

AMQ8019: Stop WebSphere MQ channel accepted.

- b) 채널 CL2.QM3가 중지되는지 확인하십시오.

채널이 중지되지 않으면 **FORCE** 옵션을 사용하여 **STOP CHANNEL** 명령을 다시 실행할 수 있습니다. **FORCE** 옵션 설정의 예는 채널이 중지되지 않고 채널을 동기화하기 위해 다른 큐 관리자를 재시작할 수 없는 경우입니다.

```
*... On QM1
start
```

응답은 채널 상태의 요약입니다.

AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3) CHLTYPE(CLUSSDR)

```

CONNAME (127.0.0.1(1413))    CURRENT
RQMNAME (QM3)              STATUS (STOPPED)
SUBSTATE (MQGET)           XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)

```

c) 채널 CL2.QM3를 시작하십시오.

```

*... On QM1
START CHANNEL (CL2.QM3)

```

응답은 명령이 승인됨입니다.

```
AMQ8018: Start WebSphere MQ channel accepted.
```

d) 채널이 시작되었는지 확인하십시오.

```

*... On QM1
DISPLAY CHSTATUS (CL2.QM3)

```

응답은 채널 상태의 요약입니다.

```
AMQ8417: Display Channel Status details.
```

```

CHANNEL (CL2.QM3)          CHLTYPE (CLUSSDR)
CONNAME (127.0.0.1(1413)) CURRENT
RQMNAME (QM3)             STATUS (RUNNING)
SUBSTATE (MQGET)         XMITQ (XMITQ.CL2.QM3)

```

e) 전송 큐가 전환되었는지 확인하십시오.

"AMQ7341 메시지에 대한 게이트웨이 큐 관리자 오류 로그를 모니터링하십시오. 채널 CL2.QM3용 전송 큐는 XMITQ.CL2.QM3"입니다.

다음에 수행할 작업

큐 알리아스 정의 Q1A를 사용하여 QM2의 메시지를 QM3의 Q1로 전송하여 개별 전송 큐를 테스트하십시오.

1. 메시지를 넣으려면 QM2에서 샘플 프로그램 **amqspout**를 실행하십시오.

```

C:\IBM\MQ>amqspout Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

```

```
Sample AMQSPUT0 end
```

2. QM3의 Q1에서 메시지를 가져오려면 샘플 프로그램 **amqsget**를 실행하십시오.

```

C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end

```

관련 개념

145 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』

애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드는 클러스터 큐에 대해 원격으로 검사, SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사 또는 클러스터 큐나 클러스터 큐 관리자의 로컬 프로파일에 대해 검사입니다.

258 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

155 페이지의 『클러스터 전송 큐 및 클러스터-송신자 채널』

클러스터된 큐 관리자 간 메시지는 클러스터 전송 큐에 저장되고 클러스터 송신자 채널에서 전달됩니다.

관련 태스크

179 페이지의 『클러스터에 큐 관리자 추가: 개별 전송 큐』

다음 지시사항에 따라 사용자가 작성한 클러스터에 큐 관리자를 추가합니다. 클러스터 큐 및 토픽에 대한 메시지는 다중 클러스터 전송 큐를 사용하여 전송됩니다.

197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』

태스크의 지시사항에 따라 게이트웨이 큐 관리자가 포함된 중첩 클러스터를 구성합니다. 한 애플리케이션에 대한 메시지를 클러스터 내 다른 애플리케이션에 대한 메시지로부터 격리하는 다음 예에 대한 시작점으로 이 클러스터를 사용합니다.

181 페이지의 『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 클러스터된 큐 리모트 정의 및 개별 송신자 채널과 전송 큐를 사용합니다.

202 페이지의 『메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경』

클러스터된 큐 또는 전송 큐의 토픽에 대해 큐 관리자가 메시지를 저장하는 기본 방식을 변경할 수 있습니다. 기본값을 변경하여 게이트웨이 큐 관리자에서 클러스터 메시지를 격리하는 방식을 제공합니다.

184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 추가 클러스터 전송 큐를 사용하여 클러스터의 단일 큐 관리자로 메시지 트래픽을 분리합니다.

187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 솔루션은 추가 클러스터를 사용하여 메시지를 특정 클러스터 큐로 분리합니다.

261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』

클러스터 전송 큐의 선택에 대한 지침입니다. 하나의 공용 기본 큐, 별도의 기본 큐, 또는 수동으로 정의된 큐를 구성할 수 있습니다. 다중 클러스터 전송 큐 구성은 z/OS 이외의 플랫폼에 적용됩니다.

게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 솔루션은 추가 클러스터를 사용하여 메시지를 특정 클러스터 큐로 분리합니다.

시작하기 전에

이 태스크의 단계는 [185 페이지의 그림 33](#)에 설명된 구성을 수정하기 위해 작성됩니다.

1. 게이트웨이 큐 관리자는 Version 7.5 이상 및 z/OS 이외의 플랫폼에 있어야 합니다.
2. 해당 태스크의 단계를 따라 [197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』](#)의 [198 페이지의 그림 37](#)에 표시된 겹쳐진 클러스터를 구성하십시오.
3. [184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』](#)에 있는 [185 페이지의 그림 33](#)의 단계를 수행하여 추가 클러스터 없이 솔루션을 작성하십시오. 이를 이 태스크의 단계를 위한 기반으로 사용하십시오.

이 태스크 정보

[184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』](#)에서 메시지 트래픽을 단일 애플리케이션으로 격리하는 솔루션은 대상 클러스터 큐가 큐 관리자에 있는 유일한 클러스터 큐인 경우에 작동합니다. 그렇지 않은 경우에는 다음 두 가지 선택사항이 있습니다. 큐를 다른 큐 관리자로 이동하거나 큐 관리자의 다른 클러스터 큐에서 큐를 격리하는 클러스터를 작성하십시오.

이 태스크는 대상 큐를 격리하기 위한 클러스터를 추가하는 단계에 대해 설명합니다. 클러스터는 해당 용도로만 추가됩니다. 실제로 클러스터 및 클러스터 이름 지정 체계를 설계하는 과정에 있을 때 특정 애플리케이션을 체계적으로 격리하는 태스크를 시작하십시오. 큐가 격리를 요구할 때마다 클러스터를 추가하면 관리할 클러스터만 많아질 수 있습니다. 이 태스크에서는 QM3의 Q1을 격리하기 위해 클러스터 CL3를 추가하여 184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』에서 구성을 변경합니다. 애플리케이션은 변경 전 과정 동안 계속해서 실행됩니다.

새로운 정의 및 변경된 정의는 188 페이지의 그림 34에서 강조표시됩니다. 변경사항의 요약은 다음과 같습니다. 클러스터를 작성합니다. 새 전체 클러스터 저장소도 작성해야 함을 의미합니다. 예에서 QM3가 CL3의 전체 저장소 중 하나가 됩니다. QM1에 대한 클러스터-송신자 및 클러스터-수신자 채널을 작성하여 게이트웨이 큐 관리자를 새 클러스터에 추가하십시오. Q1의 정의를 변경하여 이 정의를 CL3로 전환하십시오. 게이트웨이 큐 관리자에서 클러스터 이름 목록을 수정하고 새 클러스터 채널을 사용할 클러스터 전송 큐를 추가하십시오. 마지막으로 큐 알리아스 Q1A를 새 클러스터 이름 목록으로 전환하십시오.

IBM WebSphere MQ은 184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』에서 새 전송 큐 XMITQ.CL3.QM3에 자동으로 추가한 전송 큐 XMITQ.CL2.QM3에서 메시지를 전송할 수 없습니다. 두 전송 큐 모두 같은 클러스터 송신자 채널에 의해 지원되는 경우에만 메시지를 자동 전송할 수 있습니다. 대신 이 태스크는 전환을 수동으로 수행하는 한 가지 방식을 설명하며 이 방식이 사용자에게 적합할 수 있습니다. 전송이 완료된 경우 QM3에 있는 다른 CL2 클러스터 큐에 대해 기본 클러스터 전송 큐를 사용하도록 되돌아가는 옵션이 있습니다. 또는 계속해서 XMITQ.CL2.QM3를 사용할 수 있습니다. 기본 클러스터 전송 큐로 되돌아가기로 결정한 경우 게이트웨이 큐 관리자는 전환을 자동으로 관리합니다.

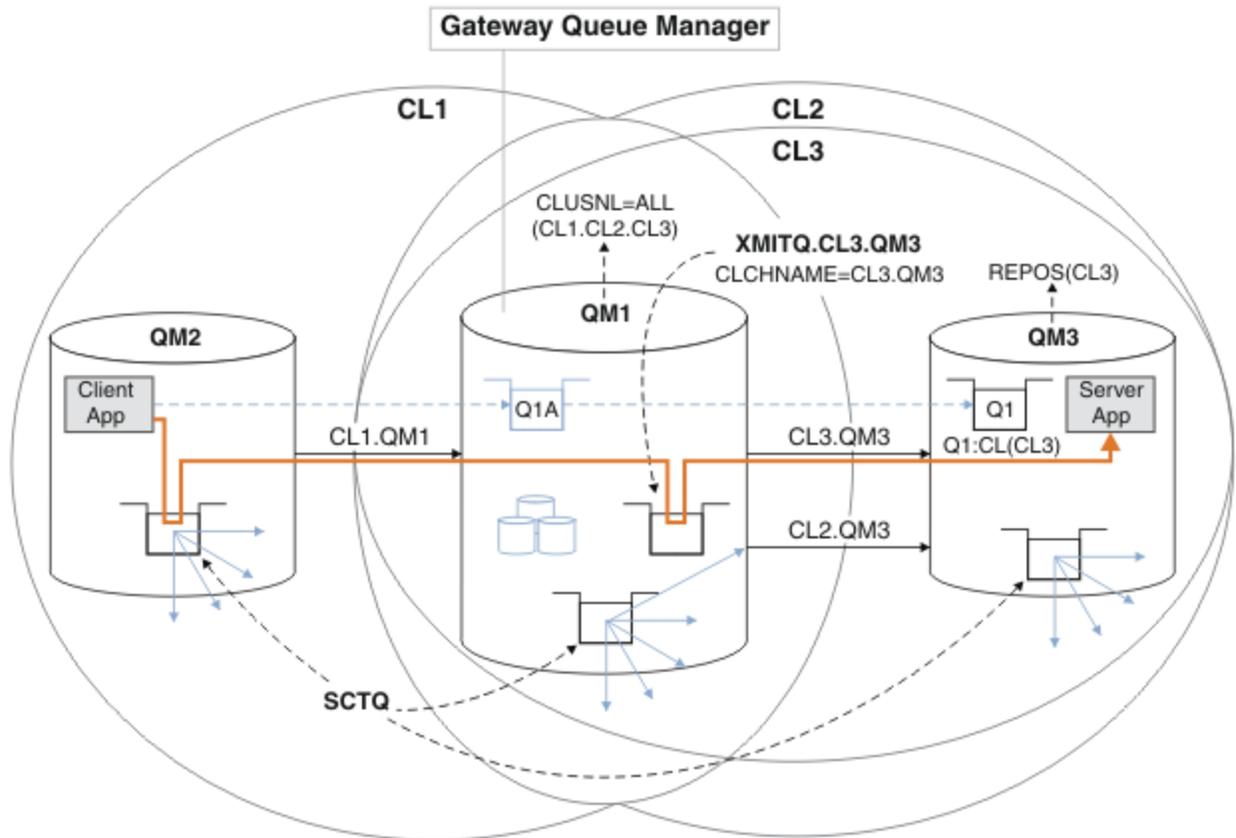


그림 34. 같은 큐 관리자에 있는 다수의 클러스터 중 하나로 이동하는 메시지 트래픽을 게이트웨이 큐 관리자에서 분리하기 위해 추가 클러스터 사용

프로시저

1. 큐 관리자 QM3 및 QM5를 변경하여 CL2 및 CL3 둘 다를 위한 저장소로 만드십시오.

큐 관리자를 다중 클러스터의 멤버가 되게 하려면 큐 관리자가 멤버인 클러스터를 식별하는 클러스터 이름 목록을 사용해야 합니다.

```
*... On QM3 and QM5
DEFINE NAMELIST(CL23) NAMES(CL2, CL3) REPLACE
ALTER QMGR REPOS(' ') REPOSNL(CL23)
```

2. CL3에 대한 큐 관리자 QM3 및 QM5 간 채널을 정의하십시오.

```
*... On QM3
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSRCVR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE

*... On QM5
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
```

3. CL3에 게이트웨이 큐 관리자를 추가하십시오.

QM1을 부분 저장소로 CL3에 추가하여 게이트웨이 큐 관리자를 추가하십시오. 클러스터 송신자 및 클러스터 수신자 채널을 QM1에 추가하여 부분 저장소를 작성하십시오.

또한 CL3를 게이트웨이 큐 관리자에 연결된 모든 클러스터의 이름 목록에 추가하십시오.

```
*... On QM1
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL3) REPLACE
ALTER NAMELIST(ALL) NAMES(CL1, CL2, CL3)
```

4. QM3에서 CL3로 이동하는 메시지에 대해 게이트웨이 큐 관리자 QM1에 클러스터 전송 큐를 추가하십시오.

처음에는 전송 큐를 전환할 준비가 될 때까지 전송 큐에서 메시지를 전송하는 클러스터 송신자 채널을 중지하십시오.

```
*... On QM1
DEFINE QLOCAL(XMITQ.CL3.QM3) USAGE(XMITQ) CLCHNAME(CL3.QM3) GET(DISABLED) REPLACE
```

5. 기존 클러스터 전송 큐 XMITQ.CL2.QM3에서 메시지를 배출하십시오.

이 서브프로시저는 메시지가 게이트웨이 큐 관리자에 도착한 순서와 일치하도록 Q1에서 메시지의 순서를 유지하는 것을 목적으로 합니다. 클러스터에서 메시지 순서 지정은 완전히는 아니더라도 어느 정도는 보장됩니다. 보장된 메시지 순서 지정이 필요한 경우 애플리케이션이 메시지의 순서를 정의해야 합니다. [큐에서 메시지가 검색되는 순서를 참조하십시오.](#)

- a) QM3의 대상 큐 Q1을 CL2에서 CL3로 변경하십시오.

```
*... On QM3
ALTER QLOCAL(Q1) CLUSTER(CL3)
```

- b) 메시지가 전달되기 시작할 때까지 XMITQ.CL3.QM3를 모니터링하십시오.

Q1에서 CL3로의 전환이 게이트웨이 큐 관리자에게 전파되면 메시지가 XMITQ.CL3.QM3에 전달되기 시작합니다.

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL3.QM3) CURDEPTH
```

- c) QM3의 Q1에 전달되기 위해 대기 중인 메시지가 없을 때까지 XMITQ.CL2.QM3를 모니터링하십시오.

참고: XMITQ.CL2.QM3는 CL2의 멤버인 QM3에서 다른 큐에 대한 메시지를 저장할 수 있으며 이 경우에 용량은 0이 되지 않을 수 있습니다.

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL2.QM3) CURDEPTH
```

- d) 새 클러스터 전송 큐 XMITQ.CL3.QM3에서 가져오기를 사용으로 설정하십시오.

```
*... On QM1
ALTER QLOCAL(XMITQ.CL3.QM3) GET(ENABLED)
```

6. 더 이상 필요하지 않은 경우 오래된 클러스터 송신 큐 XMITQ.CL2.QM3를 제거하십시오.

QM3의 CL2에서 클러스터 큐에 대한 메시지는 게이트웨이 큐 관리자 QM1에서 기본 클러스터 전송 큐를 사용하는 것으로 되돌아갑니다. 기본 클러스터 전송 큐는 SYSTEM.CLUSTER.TRANSMIT.QUEUE 또는 SYSTEM.CLUSTER.TRANSMIT.CL2.QM3입니다. 이 중 사용되는 것은 QM1에서 큐 관리자 속성 **DEFCLXQ**의 값이 SCTQ인지 또는 CHANNEL인지에 따라 달라집니다. 큐 관리자는 클러스터 송신자 채널 CL2.QM3가 다음에 시작될 때 XMITQ.CL2.QM3에서 자동으로 메시지를 전송합니다.

a) 전송 큐 XMITQ.CL2.QM3를 클러스터 전송 큐인 상태에서 일반 전송 큐인 상태로 변경하십시오.

이렇게 하면 클러스터 송신자 채널과 전송 큐의 연관이 해제됩니다. 이에 따라 IBM WebSphere MQ는 클러스터 송신자 채널이 다음에 시작될 때 XMITQ.CL2.QM3에서 기본 클러스터 전송 큐로 메시지를 자동 전송합니다. 그 때까지 QM3의 CL2에 대한 메시지는 계속해서 XMITQ.CL2.QM3에 배치됩니다.

```
*... On QM1
ALTER QLOCAL(XMITQ.CL2.QM3) CLCHNAME(' ')
```

b) 클러스터 송신자 채널 CL2.QM3를 중지하십시오.

클러스터 송신자 채널을 중지하고 재시작하면 XMITQ.CL2.QM3에서 기본 클러스터 전송 큐로의 메시지 전송이 시작됩니다. 일반적으로 전송을 시작하려면 수동으로 채널을 중지하고 시작합니다. 연결 끊기 간격이 만료되어 종료된 후에 채널이 재시작되는 경우 전송이 자동으로 시작됩니다.

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

응답은 명령이 승인됨입니다.

AMQ8019: Stop WebSphere MQ channel accepted.

c) 채널 CL2.QM3가 중지되는지 확인하십시오.

채널이 중지되지 않으면 **FORCE** 옵션을 사용하여 **STOP CHANNEL** 명령을 다시 실행할 수 있습니다. **FORCE** 옵션 설정의 예는 채널이 중지되지 않고 채널을 동기화하기 위해 다른 큐 관리자를 재시작할 수 없는 경우입니다.

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

응답은 채널 상태의 요약입니다.

AMQ8417: Display Channel Status details.

CHANNEL(CL2.QM3)	CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))	CURRENT
RQMNAME(QM3)	STATUS(STOPPED)
SUBSTATE(MQGET)	XMITQ(XMITQ.CL2.QM3)

d) 채널 CL2.QM3를 시작하십시오.

```
*... On QM1
START CHANNEL(CL2.QM3)
```

응답은 명령이 승인됨입니다.

AMQ8018: Start WebSphere MQ channel accepted.

e) 채널이 시작되었는지 확인하십시오.

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

응답은 채널 상태의 요약입니다.

```
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM3)          CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1413)) CURRENT
RQMNAME (QM3)             STATUS (RUNNING)
SUBSTATE (MQGET)          XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE/CL2.QM3)
```

f) "AMQ7341 메시지에 대한 게이트웨이 큐 관리자 오류 로그를 모니터링하십시오. 채널 CL2.QM3용 전송 큐는 SYSTEM.CLUSTER.TRANSMIT.QUEUE/CL2.QM3"입니다.

g) 클러스터 전송 큐 XMITQ.CL2.QM3를 삭제하십시오.

```
*... On QM1
DELETE QLOCAL(XMITQ.CL2.QM3)
```

다음에 수행할 작업

큐 알리언스 정의 Q1A를 사용하여 QM2의 메시지를 QM3의 Q1로 전송함으로써 개별적으로 클러스터 큐를 테스트하십시오.

1. 메시지를 넣으려면 QM2에서 샘플 프로그램 **amqsput**를 실행하십시오.

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

2. QM3의 Q1에서 메시지를 가져오려면 샘플 프로그램 **amqsget**를 실행하십시오.

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

관련 개념

[145 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』](#)

애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드는 클러스터 큐에 대해 원격으로 검사, SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사 또는 클러스터 큐나 클러스터 큐 관리자의 로컬 프로파일에 대해 검사입니다.

[258 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』](#)

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

[155 페이지의 『클러스터 전송 큐 및 클러스터-송신자 채널』](#)

클러스터된 큐 관리자 간 메시지는 클러스터 전송 큐에 저장되고 클러스터 송신자 채널에서 전달됩니다.

관련 태스크

[179 페이지의 『클러스터에 큐 관리자 추가: 개별 전송 큐』](#)

다음 지시사항에 따라 사용자가 작성한 클러스터에 큐 관리자를 추가합니다. 클러스터 큐 및 토픽에 대한 메시지는 다중 클러스터 전송 큐를 사용하여 전송됩니다.

[197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』](#)

태스크의 지시사항에 따라 게이트웨이 큐 관리자가 포함된 중첩 클러스터를 구성합니다. 한 애플리케이션에 대한 메시지를 클러스터 내 다른 애플리케이션에 대한 메시지에서 격리하는 다음 예에 대한 시작점으로 이 클러스터를 사용합니다.

[181 페이지의 『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』](#)

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 클러스터된 큐 리모트 정의 및 개별 송신자 채널과 전송 큐를 사용합니다.

202 페이지의 『메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경』

클러스터된 큐 또는 전송 큐의 토픽에 대해 큐 관리자가 메시지를 저장하는 기본 방식을 변경할 수 있습니다. 기본값을 변경하여 게이트웨이 큐 관리자에서 클러스터 메시지를 격리하는 방식을 제공합니다.

184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 추가 클러스터 전송 큐를 사용하여 클러스터의 단일 큐 관리자로 메시지 트래픽을 분리합니다.

187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 솔루션은 추가 클러스터를 사용하여 메시지를 특정 클러스터 큐로 분리합니다.

261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』

클러스터 전송 큐의 선택에 대한 지침입니다. 하나의 공용 기본 큐, 별도의 기본 큐, 또는 수동으로 정의된 큐를 구성할 수 있습니다. 다중 클러스터 전송 큐 구성은 z/OS 이외의 플랫폼에 적용됩니다.

DHCP를 사용하여 클러스터에 큐 관리자 추가

DHCP를 사용하여 클러스터에 큐 관리자를 추가합니다. 이 태스크는 CLUSRCVR 정의에서 CONNAME 값을 생략하는 것을 보여줍니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 태스크를 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

이 태스크는 두 개의 특수 기능을 보여줍니다.

- CLUSRCVR 정의에서 CONNAME 값을 생략하는 기능
- CLUSSDR 정의에서 +QMNAME+을 사용하는 기능

두 기능 모두 z/OS에서 제공되지 않습니다.

시나리오:

- INVENTORY 클러스터는 168 페이지의 『새 클러스터 설정』에 설명된 대로 설정되었습니다. 이 클러스터는 둘 다 전체 저장소를 보유하는 두 개의 큐 관리자 LONDON 및 NEWYORK을 포함합니다.
- 체인 저장소의 새 분기가 Paris에서 설정되고 PARIS라고 하는 큐 관리자를 클러스터에 추가하려고 합니다.
- 큐 관리자 PARIS는 메시지를 INVENTQ 큐에 넣어 New York의 시스템에서 실행되는 애플리케이션에 인벤토리 업데이트를 송신합니다.
- 세 시스템 사이에는 모두 네트워크 연결성이 있습니다.
- 네트워크 프로토콜은 TCP입니다.
- PARIS 큐 관리자 시스템은 DHCP를 사용하여 이는 IP 주소가 시스템 재시작 시 변경될 수 있음을 의미합니다.
- PARIS와 LONDON 시스템 간 채널의 이름은 정의된 이름 지정 규칙에 따라 지정됩니다. 이 규칙에서는 LONDON의 전체 저장소 큐 관리자의 큐 관리자 이름을 사용합니다.
- PARIS 큐 관리자의 관리자에게는 LONDON 저장소에 있는 큐 관리자의 이름에 대한 정보가 없습니다. LONDON 저장소에서 큐 관리자의 이름은 변경될 수 있습니다.

이 태스크 정보

다음 단계에 따라 DHCP를 사용하여 클러스터에 큐 관리자를 추가하십시오.

프로시저

1. PARIS가 처음 참조할 전체 저장소를 결정하십시오.

클러스터 내 모든 큐 관리자는 전체 저장소 중 하나를 참조해야 합니다. 큐 관리자는 전체 저장소에서 클러스터에 대한 정보를 수집하고 자체 부분 저장소를 빌드합니다. 저장소 중 하나를 전체 저장소로 선택하십시오. 새 큐 관리자는 클러스터에 추가되는 즉시 다른 저장소에 대해서도 학습합니다. 큐 관리자 변경에 대한 정보는 두 저장소에 직접 송신됩니다. 이 예에서는 순전히 지리적 이유로 PARIS를 큐 관리자 LONDON에 링크하도록 선택합니다.

참고: 큐 관리자 PARIS가 시작된 후에는 순서에 관계없이 나머지 단계를 수행하십시오.

2. 큐 관리자 PARIS에서 CLUSRCVR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 수신할 수 있는 클러스터 수신자 채널을 정의해야 합니다. PARIS에서 다음을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR)
TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

클러스터 수신자 채널은 클러스터 INVENTORY에 있는 다른 큐 관리자로부터 메시지를 수신할 큐 관리자의 가용성을 알립니다. 클러스터 수신자 채널에서 CONNAME을 지정하지 않아도 됩니다. 쿨명을 생략하거나 CONNAME(' ')를 지정하여 IBM WebSphere MQ 를 요청하여 시스템에서 연결 이름을 찾을 수 있습니다. IBM WebSphere MQ 는 시스템의 현재 IP 주소를 사용하여 CONNAME 값을 생성합니다. CONNAME을 (를) 참조하십시오. 클러스터 수신자 채널 INVENTORY.PARIS에 대한 송신 측의 다른 큐 관리자에서 정의를 작성하지 않아도 됩니다. 기타 정의는 필요한 경우 자동으로 작성됩니다.

3. 큐 관리자 PARIS에서 CLUSSDR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 해당 초기 전체 저장소에 송신할 수 있는 한 개의 클러스터 송신자 채널을 정의해야 합니다. PARIS에서 네트워크 주소가 LONDON.CHSTORE.COM인 큐 관리자에 대한 INVENTORY.+QMNAME+이라고 하는 채널에 대해 다음 정의를 작성하십시오.

```
DEFINE CHANNEL(INVENTORY.+QMNAME+) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. 옵션: 이 큐 관리자가 클러스터에 다시 결합할 경우 일부 추가 단계를 수행하십시오.

a) 이전에 같은 클러스터에서 제거된 큐 관리자를 클러스터에 추가하는 경우, 이제 클러스터 멤버로 표시하는지 확인하십시오. 그렇지 않은 경우 다음 추가 단계를 완료하십시오.

i) 추가할 큐 관리자에서 **REFRESH CLUSTER** 명령을 발행하십시오. 이 단계는 클러스터 채널을 중지하고 로컬 클러스터 캐시에 클러스터의 나머지 부분에서 최신임이 보장되는 새로운 순서 번호 세트를 제공합니다.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

참고: 대형 클러스터의 경우, **REFRESH CLUSTER** 명령을 사용하면 진행 중에 클러스터에 혼란을 줄 수 있으며, 클러스터 오브젝트가 모든 관심 있는 큐 관리자에 자동으로 상태 업데이트를 보낸 이후 27 일 간격으로 다시 수행됩니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오.

ii) CLUSSDR 채널을 다시 시작하십시오(예: **START CHANNEL** 명령 사용).

iii) CLUSRCVR 채널을 재시작하십시오.

b) 클러스터가 발행/구독 클러스터이고 재결합 큐 관리자에게 구독물이 있는 경우 다음 명령을 실행하여 클러스터 전체에서 프록시 구독이 올바르게 동기화되도록 합니다.

```
REFRESH QMGR TYPE(PROXYSUB)
```

결과

이 태스크에서 설정된 클러스터는 177 페이지의 『클러스터에 큐 관리자 추가』의 경우와 동일합니다.

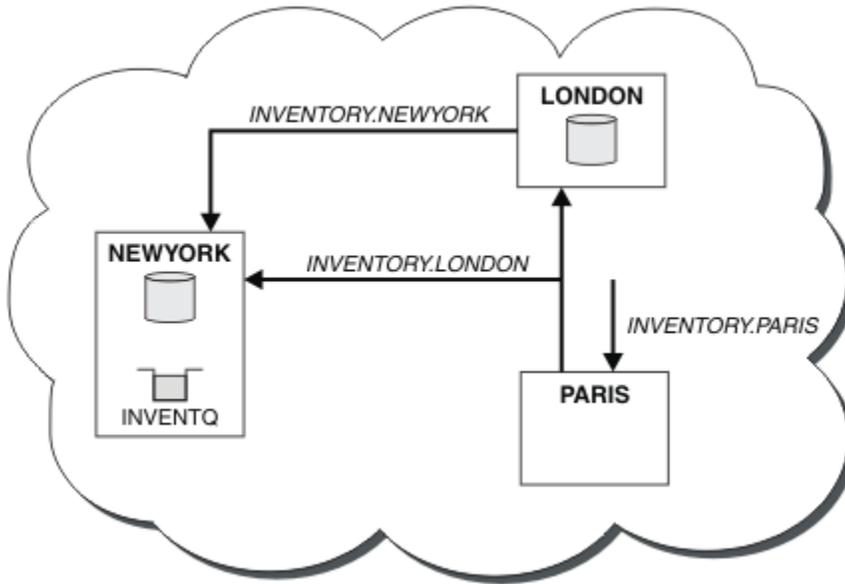


그림 35. 세 개의 큐 관리자가 있는 *INVENTORY* 클러스터

CLUSRCVR 정의 및 CLUSSDR 정의 두 개만 작성하여 큐 관리자 PARIS를 클러스터에 추가했습니다.

PARIS 큐 관리자에서 문자열 +QMNAME+를 포함하는 CLUSSDR이 시작됩니다. LONDON 시스템에서 IBM WebSphere MQ 는 +QMNAME+ 를 큐 관리자 이름 (LONDON) 으로 해석합니다. IBM WebSphere MQ 는 INVENTORY.LONDON 라는 채널에 대한 정의를 해당 CLUSRCVR 정의와 일치시킵니다.

WebSphere MQ는 결정된 채널 이름을 PARIS 큐 관리자로 다시 송신합니다. PARIS에서 INVENTORY.+QMNAME+이라고 하는 채널에 대한 CLUSSDR 채널 정의는 INVENTORY.LONDON에 대해 내부적으로 생성된 CLUSSDR 정의로 바뀝니다. 이 정의는 해석된 채널 이름을 포함하지만 그렇지 않은 경우 사용자가 작성한 +QMNAME+ 정의와 같습니다. 또한 클러스터 저장소는 새로 해석된 채널 이름을 가진 채널 정의로 최신화됩니다.

참고:

1. +QMNAME+ 이름을 사용하여 작성된 채널은 즉시 비활성 상태가 됩니다. 이 채널은 데이터를 전송하는 데에는 사용되지 않습니다.
2. 채널 엑시트는 한 호출과 다음 호출 사이의 채널 이름 변경을 확인할 수 있습니다.

이제 PARIS 큐 관리자는 LONDON의 저장소에서 INVENTQ 큐가 큐 관리자 NEWYORK에 의해 호스트됨을 학습합니다. Paris에서 시스템이 호스트하는 애플리케이션이 INVENTQ, PARIS 에 메시지를 넣으려고 하면 클러스터 수신자 채널 INVENTORY.NEWYORK에 연결할 클러스터 송신자 채널을 자동으로 정의합니다. 이 애플리케이션은 해당 큐 관리자 이름이 대상 큐 관리자로 지정되고 응답 대상 큐가 제공되는 경우에 응답을 수신할 수 있습니다.

관련 참조

[DEFINE CHANNEL](#)

큐를 호스팅하는 큐 관리자 추가

다른 INVENTQ 큐를 호스팅하기 위해 다른 큐 관리자를 클러스터에 추가합니다. 요청은 각 큐 관리자의 큐에 번갈아 송신됩니다. 기존 INVENTQ 호스트에 대한 변경은 필요하지 않습니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 작업을 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- INVENTORY 클러스터가 177 페이지의 『클러스터에 큐 관리자 추가』에서 설명된 대로 설정되었습니다. 이 클러스터는 세 큐 관리자를 포함하며 LONDON 및 NEWYORK은 둘 다 전체 저장소를 보유하고 PARIS는 부분 저장소를 보유하고 있습니다. 인벤토리 애플리케이션은 NEWYORK 큐 관리자에 연결된 New York의 시스템에서 실행됩니다. 애플리케이션은 INVENTQ 큐에 메시지가 도착하면 구동됩니다.
- 새 저장소가 Toronto에서 설정됩니다. 추가 용량을 제공하기 위해 New York뿐만 아니라 Toronto의 시스템에서도 인벤토리 애플리케이션을 실행하려고 합니다.
- 네 시스템 사이에는 모두 네트워크 연결성이 있습니다.
- 네트워크 프로토콜은 TCP입니다.

참고: 큐 관리자 TORONTO는 부분 저장소만 포함합니다. 클러스터에 전체 저장소 큐 관리자를 추가하려는 경우 209 페이지의 『다른 큐 관리자로 전체 저장소 이동』의 내용을 참조하십시오.

이 태스크 정보

다음 단계에 따라 큐를 호스팅하는 큐 관리자를 추가하십시오.

프로시저

1. TORONTO가 처음 참조할 전체 저장소를 결정하십시오.

클러스터 내 모든 큐 관리자는 전체 저장소 중 하나를 참조해야 합니다. 큐 관리자는 전체 저장소에서 클러스터에 대한 정보를 수집하고 자체 부분 저장소를 빌드합니다. 저장소 선택이 특별히 중요한 것은 아닙니다. 이 예에서는 NEWYORK을 선택합니다. 새 큐 관리자는 클러스터에 조인하고 나면 두 저장소와 모두 통신합니다.

2. CLUSRCVR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 수신할 수 있는 클러스터 수신자 채널을 정의해야 합니다. TORONTO에서 CLUSRCVR 채널을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.TORONTO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(TORONTO.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for TORONTO')
```

TORONTO 큐 관리자는 해당 클러스터 수신자 채널을 사용하여 INVENTORY 클러스터에 있는 다른 큐 관리자로부터 메시지를 수신하기 위해 그 가용성을 알립니다.

3. 큐 관리자 TORONTO에서 CLUSSDR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 해당 첫 번째 전체 저장소에 송신할 수 있는 한 개의 클러스터 송신자 채널을 정의해야 합니다. 이 경우에는 NEWYORK을 선택하십시오. TORONTO에는 다음 정의가 필요합니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from TORONTO to repository at NEWYORK')
```

4. 옵션: 이 큐 관리자가 클러스터에 다시 결합할 경우 일부 추가 단계를 수행하십시오.

a) 이전에 같은 클러스터에서 제거된 큐 관리자를 클러스터에 추가하는 경우, 이제 클러스터 멤버로 표시하는지 확인하십시오. 그렇지 않은 경우 다음 추가 단계를 완료하십시오.

i) 추가할 큐 관리자에서 **REFRESH CLUSTER** 명령을 발행하십시오. 이 단계는 클러스터 채널을 중지하고 로컬 클러스터 캐시에 클러스터의 나머지 부분에서 최신임이 보장되는 새로운 순서 번호 세트를 제공합니다.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

참고: 대형 클러스터의 경우, **REFRESH CLUSTER** 명령을 사용하면 진행 중에 클러스터에 혼란을 줄 수 있으며, 클러스터 오브젝트가 모든 관심 있는 큐 관리자에게 자동으로 상태 업데이트를 보낸 이후 27 일 간격으로 다시 수행됩니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오.

ii) CLUSSDR 채널을 다시 시작하십시오(예: **START CHANNEL** 명령 사용).

iii) CLUSRCVR 채널을 재시작하십시오.

b) 클러스터가 발행/구독 클러스터이고 재결합 큐 관리자에게 구독물이 있는 경우 다음 명령을 실행하여 클러스터 전체에서 프록시 구독이 올바르게 동기화되도록 합니다.

```
REFRESH QMGR TYPE(PROXYSUB)
```

5. 메시지 연관관계에 대해 인벤토리 애플리케이션을 검토하십시오.

계속하기 전에 메시지 처리 순서에 따라 인벤토리 애플리케이션에 종속 항목이 없는지 확인하고 이 애플리케이션을 Toronto의 시스템에 설치하십시오.

6. 클러스터 큐 INVENTQ를 정의하십시오.

이미 NEWYORK 큐 관리자가 호스팅하고 있는 INVENTQ 큐도 TORONTO가 호스팅합니다. 다음과 같이 이 큐를 TORONTO 큐 관리자에서 정의하십시오.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

결과

196 페이지의 그림 36에서는 이 태스크에서 설정되는 INVENTORY 클러스터를 보여줍니다.

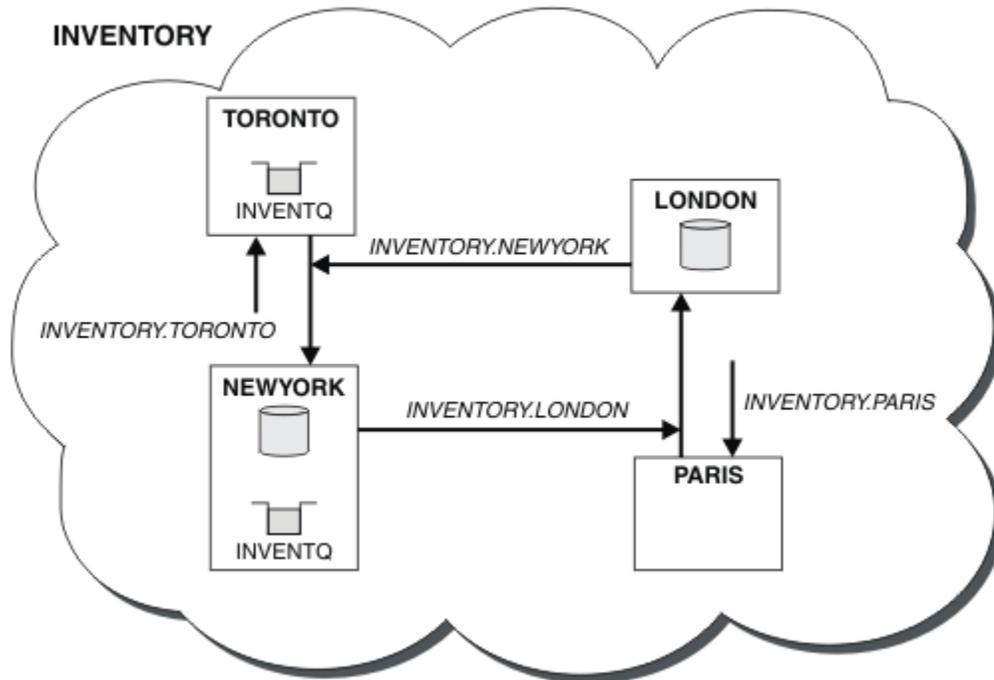


그림 36. 네 개의 큐 관리자가 있는 INVENTORY 클러스터

INVENTQ 큐 및 인벤토리 애플리케이션은 이제 클러스터에 있는 두 개의 큐 관리자에서 호스팅됩니다. 이 경우 두 큐 관리자의 가용성이 증가되고 메시지 처리 속도가 빨라지며 두 큐 관리자 사이에 워크로드를 분산시킬 수 있습니다. TORONTO 또는 NEWYORK에 의해 INVENTQ에 들어진 메시지는 가능할 때마다 로컬 큐 관리자에서 인스턴스에 의해 핸들링됩니다. LONDON 또는 PARIS에 의해 들어진 메시지는 워크로드 밸런싱이 이루어지도록 TORONTO 또는 NEWYORK에 교대로 라우트됩니다.

클러스터에 대한 이러한 수정은 사용자가 큐 관리자 NEWYORK, LONDON 및 PARIS에 대한 정의를 변경하지 않고도 수행할 수 있습니다. 이러한 큐 관리자의 전체 저장소는 메시지를 TORONTO의 INVENTQ에 송신하기 위해 필요한 정보로 자동 업데이트됩니다. 인벤토리 애플리케이션은 NEWYORK 또는 TORONTO 큐 관리자 중 하나가 사용 불가능한 상태가 되더라도 계속해서 기능을 하며 충분한 용량을 가집니다. 인벤토리 애플리케이션은 두 위치 모두에서 호스팅되는 경우에도 올바르게 작동할 수 있어야 합니다.

이 태스크의 결과에서 볼 수 있듯이 둘 이상의 큐 관리자에서 실행되는 동일한 애플리케이션을 가질 수 있습니다. 고른 분산 워크로드를 위해 클러스터링할 수 있습니다.

애플리케이션은 두 위치에서 레코드를 처리하지 못할 수 있습니다. 예를 들어, 고객 계정 조회를 추가하고 LONDON 및 NEWYORK에서 실행 중인 애플리케이션을 업데이트하기로 결정한 것으로 가정하십시오. 계정 레코드는 한 위치에만 보유될 수 있습니다. 데이터 파티션 기술을 사용하여 요청 분배의 제어를 결정할 수 있습니다. 레코드의 분배를 분할할 수 있습니다. 레코드의 절반에 대해 배열할 수 있습니다. 예를 들어, 계정 번호 00000 - 49999는 LONDON에 보유됩니다. 나머지 절반인 50000 - 99999가 NEWYORK에 포함되도록 정렬할 수 있습니다. 그런 다음 모든 메시지의 계정 필드를 조사하고 메시지를 적절한 큐 관리자에게 라우트하는 클러스터 워크로드 엑시트 프로그램을 작성할 수 있습니다.

다음에 수행할 작업

모든 정의를 마쳤으므로 아직 수행하지 않은 경우 IBM WebSphere MQ for z/OS에서 채널 시작기를 시작하십시오. 모든 플랫폼의 큐 관리자 TORONTO에서 리스너 프로그램을 시작하십시오. 리스너 프로그램은 수신되는 네트워크 요청을 대기하고 필요한 경우 클러스터 수신자 채널을 시작합니다.

게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성

태스크의 지시사항에 따라 게이트웨이 큐 관리자가 포함된 중첩 클러스터를 구성합니다. 한 애플리케이션에 대한 메시지를 클러스터 내 다른 애플리케이션에 대한 메시지에서 격리하는 다음 예에 대한 시작점으로 이 클러스터를 사용합니다.

이 태스크 정보

클러스터 메시지 트래픽 격리를 설명하는 데 사용되는 클러스터 구성 예가 198 페이지의 그림 37에 표시되어 있습니다. 예제는 258 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』에 설명되어 있습니다.

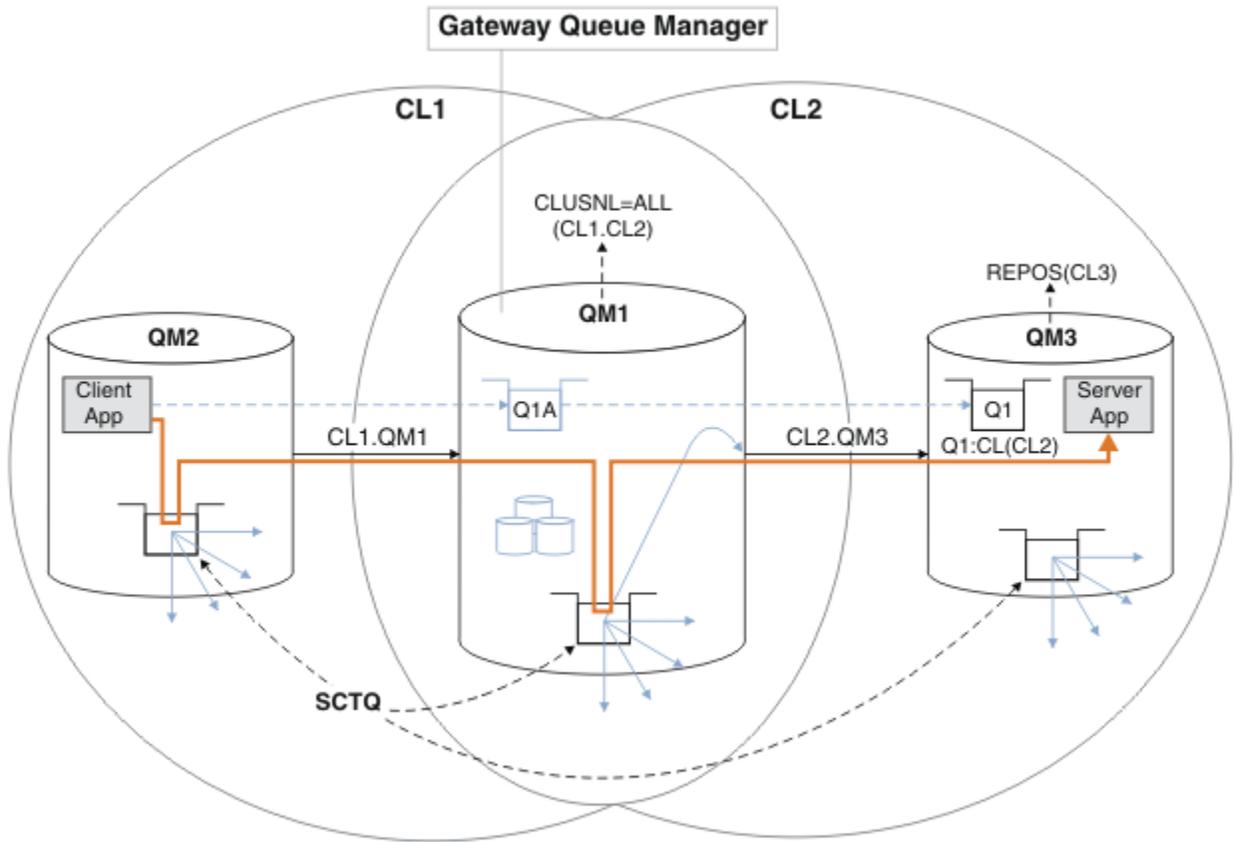


그림 37. IBM WebSphere MQ 클러스터를 사용하여 허브 및 스포크 아키텍처에 배치되는 클라이언트 서버 애플리케이션

예를 구성하기 위한 단계의 수를 최대한 적게 하려면 구성을 실제적이 아닌 단순하게 유지합니다. 이 예는 두 개의 개별 조직에서 작성된 두 개의 클러스터 통합을 나타낼 수 있습니다. 더 현실적인 시나리오는 [261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』](#)의 내용을 참조하십시오.

단계에 따라 클러스터를 구성하십시오. 클러스터는 클라이언트 애플리케이션에서 서버 애플리케이션으로의 메시지 트래픽을 격리하는 다음 예에서 사용됩니다.

지시사항은 각 클러스터가 두 개의 저장소를 가지도록 2개의 추가 큐 관리자를 추가합니다. 게이트웨이 큐 관리자는 성능상의 이유로 저장소로 사용되지 않습니다.

프로시저

1. 큐 관리자 QM1, QM2, QM3, QM4, QM5을(를) 작성하고 시작하십시오.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QMn
strmqm QmgrName
```

참고: QM4 및 QM5은(는) 클러스터에 대한 백업 전체 저장소입니다.

2. 각 큐 관리자에 대해 리스너를 정의하고 시작하십시오.

```
*... On QMn
DEFINE LISTENER(TCP141n) TRPTYPE(TCP) IPADDR(hostname) PORT(141n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141n)
```

3. 모든 클러스터에 대해 클러스터 이름 목록을 작성하십시오.

```
*... On QM1
DEFINE NAMELIST(ALL) NAMES(CL1, CL2) REPLACE
```

4. QM2 및 QM4을(를) CL2용 CL1, QM3 및 QM5 저장소에 대한 전체 저장소로 만드십시오.

a) CL1의 경우:

```
*... On QM2 and QM4
ALTER QMGR REPOS(CL1) DEFCLXQ(SCTQ)
```

b) CL2의 경우:

```
*... On QM3 and QM5
ALTER QMGR REPOS(CL2) DEFCLXQ(SCTQ)
```

5. 각 큐 관리자 및 클러스터의 클러스터 송신자 및 클러스터 수신자 채널을 추가하십시오.

다음 명령을 QM2, QM3, QM4 및 QM5에서 실행하십시오. 여기서, *c*, *n* 및 *m*은 각 큐 관리자의 [199 페이지의 표 26](#)에 표시된 값을 사용합니다.

큐 관리자	클러스터 <i>c</i>	기타 저장소 <i>n</i>	이 저장소 <i>m</i>
QM2	1	4	2
QM4	1	2	4
QM3	2	5	3
QM5	2	3	5

```
*... On Qm
DEFINE CHANNEL(CLc.QMn) CHLTYPE(CLUSSDR) CONNAME('localhost(141n)') CLUSTER(CLc) REPLACE
DEFINE CHANNEL(CLc.Qm) CHLTYPE(CLUSRCVR) CONNAME('localhost(141m)') CLUSTER(CLc) REPLACE
```

6. 게이트웨이 큐 관리자(QM1)를 각 클러스터에 추가하십시오.

```
*... On QM1
DEFINE CHANNEL(CL1.QM2) CHLTYPE(CLUSSDR) CONNAME('localhost(1412)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL1.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL2.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL2) REPLACE
DEFINE CHANNEL(CL2.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL2) REPLACE
```

7. 로컬 큐 Q1을(를) 클러스터의 CL2의 큐 관리자 QM3에서 추가하십시오.

```
*... On QM3
DEFINE QLOCAL(Q1) CLUSTER(CL2) REPLACE
```

8. 클러스터된 큐 관리자 알리어스 Q1A을(를) 게이트웨이 큐 관리자에 추가하십시오.

```
*... On QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(Queue) DEFBIND(NOTFIXED) REPLACE
```

참고: 다른 큐 관리자나 QM1에서 큐 관리자 알리어스를 사용하는 애플리케이션은 알리어스 큐를 열 때 DEFBIND(NOTFIXED)을(를) 지정해야 합니다. **DEFBIND**은(는) 애플리케이션이 큐를 열 때 메시지 헤더의 라우팅 정보가 수정되는지 여부를 지정합니다. 기본값인 **OPEN**으로 설정된 경우 메시지는 Q1@QM1(으)로 라우트됩니다. Q1@QM1이(가) 존재하지 않으므로 다른 큐 관리자의 메시지가 데드-레터 큐에 종료됩니다. 이 큐 속성을 DEFBIND(NOTFIXED)로 설정하면 큐 설정 **DEFBIND**로 기본값 설정되는 **amqsput**과 같은 애플리케이션은 올바른 방식으로 작동합니다.

9. 모든 클러스터된 큐 관리자에 대한 클러스터 큐 관리자 알리어스 정의를 게이트웨이 큐 관리자 QM1에 추가하십시오.

```
*... On QM1
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSNL(ALL) REPLACE
```

팁: 게이트웨이 큐 관리자의 큐 관리자 알리어스 정의는 다른 클러스터에 있는 큐 관리자를 참조하는 메시지를 전송합니다. 클러스터된 큐 관리자 알리어스를 참조하십시오.

다음에 수행할 작업

1. 큐 알리어스 정의 Q1A을(를) 사용하여 QM3 Q1 QM2(으)로 메시지를 전송하여 큐 알리어스 정의를 테스트하십시오.

a. 메시지를 넣으려면 QM2에서 샘플 프로그램 **amqsput**를 실행하십시오.

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

b. QM3의 Q1에서 메시지를 가져오려면 샘플 프로그램 **amqsget**를 실행하십시오.

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. 임시 동적 응답 큐에서 요청 메시지를 송신하고 응답 메시지를 수신하여 큐 관리자 알리어스 정의를 테스트하십시오.

다이어그램은 응답 메시지가 RQ(이)라는 임시 동적 큐로 되돌아가는 경로를 보여줍니다. QM3에 연결된 서버 애플리케이션은 큐 관리자 이름 QM2을(를) 사용하여 응답 큐를 엽니다. 큐 관리자 이름 QM2은(는) QM1에서 클러스터된 큐 관리자 알리어스로 정의됩니다. QM3은(는) 응답 메시지를 QM1(으)로 라우트합니다. QM1은(는) 메시지를 QM2(으)로 라우트합니다.

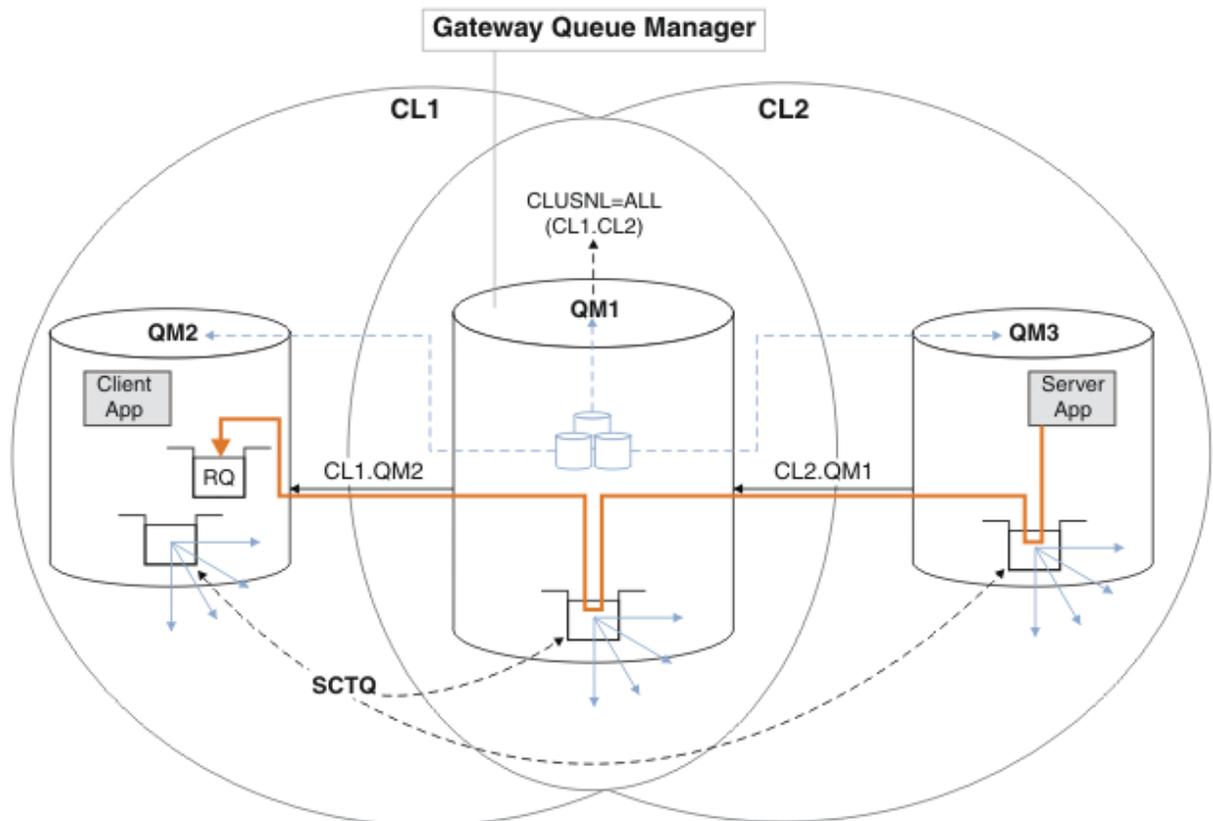


그림 38. 큐 관리자 알리어스를 사용하여 응답 메시지를 다른 클러스터로 리턴

라우팅이 작동하는 방식은 다음과 같습니다. 각 클러스터의 모든 큐 관리자에는 QM1에 대한 큐 관리자 알리어스 정의가 있습니다. 이러한 알리어스는 모든 클러스터에서 클러스터됩니다. 각 알리어스에서 큐 관리자로의

회색 파선 화살표는 각 큐 관리자 알리어스가 최소 하나 이상의 클러스터에 있는 실제 큐 관리자로 해석됨을 표시합니다. 이 경우 QM2 별명은 CL1 및 CL2 클러스터 모두에서 클러스터되며 CL1의 실제 큐 관리자 QM2(으)로 해석됩니다. 서버 애플리케이션은 큐 이름 RQ에 대한 응답을 사용하여 응답 메시지를 작성하고 큐 관리자 이름 QM2에 응답합니다. 큐 관리자 알리어스 정의 QM2이(가) 클러스터 CL2의 QM1에 정의되고 큐 관리자 QM2이(가) 클러스터 CL2에 없기 때문에 메시지가 QM1(으)로 라우트됩니다. 메시지를 대상 큐 관리자로 송신할 수 없으므로 이 메시지는 알리어스 정의가 있는 큐 관리자에게 송신됩니다.

QM1은(는) 트랜스퍼럴에 대한 QM1의 클러스터 트랜스미션 큐에 메시지를 QM2에 둡니다. QM2에 대한 QM1에서 큐 관리자 별명 정의가 QM2을(를) 실제 대상 큐 관리자로 정의하기 때문에 QM1은(는) 메시지를 QM2에 라우트합니다. 정의는 순환되지 않습니다. 알리어스 정의는 실제 정의만 참조할 수 있고 알리어스는 그 자체를 가리킬 수 없기 때문입니다. QM1 및 QM2 모두가 동일한 클러스터 CL1에 있기 때문에 실제 정의는 QM1에 의해 해석됩니다. QM1은(는) CL1의 저장소에서 QM2에 대한 연결 정보를 찾고 메시지를 QM2(으)로 라우트합니다. 메시지가 QM1에 의해 리라우트되도록 하려면, 서버 애플리케이션이 DEFBIND 옵션을 MQBND_BIND_NOT_FIXED(으)로 설정하여 응답 큐를 열어야 합니다. 서버 애플리케이션이 MQBND_BIND_ON_OPEN 옵션을 사용하여 응답 큐를 연 경우, 메시지는 경로 재지정되지 않으며 데드-레터 큐에서 종료됩니다.

- a. QM3에 트리거가 있는 클러스터된 요청 큐를 작성하십시오.

```
*... On QM3
DEFINE QLOCAL(QR) CLUSTER(CL2) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
```

- b. 게이트웨이 큐 관리자 QM1에서 QR의 클러스터된 큐 알리어스 정의를 작성하십시오.

```
*... On QM1
DEFINE QALIAS(QRA) CLUSNL(ALL) TARGET(QR) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

- c. QM3에서 샘플 에코 프로그램 **amqsech**를 시작하기 위한 프로세스 정의를 작성하십시오.

```
*... On QM3
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

- d. 임시 동적 응답 큐를 작성하려면 QM2에서 샘플 프로그램 **amqsreq**에 대한 모델 큐를 작성하십시오.

```
*... On QM2
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

- e. 큐 알리어스 정의 QRA을(를) 사용하여 QM3 QR QM2(으)로 요청을 전송하여 큐 관리자 알리어스 정의를 테스트하십시오.

- i) QM3에서 트리거 모니터 프로그램을 실행하십시오.

```
runmqtrm -m QM3
```

출력은 다음과 같습니다.

```
C:\IBM\MQ>runmqtrm -m QM3
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
01/02/2012 16:17:15: WebSphere MQ trigger monitor started.
```

```
-----
01/02/2012 16:17:15: Waiting for a trigger message
```

- ii) 요청을 넣고 응답을 기다리려면 QM2에서 샘플 프로그램 **amqsreq**를 실행하십시오.

```
C:\IBM\MQ>amqsreq QRA QM2
Sample AMQSREQ0 start
server queue is QRA
replies to 4F2961C802290020
A request message from QM2 to QR on QM3
```

```
response <A request message from QM2 to QR on QM3>
no more replies
Sample AMQSREQ0 end
```

관련 개념

145 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』

애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드는 클러스터 큐에 대해 원격으로 검사, SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사 또는 클러스터 큐나 클러스터 큐 관리자의 로컬 프로파일에 대해 검사입니다.

258 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

관련 태스크

179 페이지의 『클러스터에 큐 관리자 추가: 개별 전송 큐』

다음 지시사항에 따라 사용자가 작성한 클러스터에 큐 관리자를 추가합니다. 클러스터 큐 및 토픽에 대한 메시지는 다중 클러스터 전송 큐를 사용하여 전송됩니다.

197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』

태스크의 지시사항에 따라 게이트웨이 큐 관리자가 포함된 중첩 클러스터를 구성합니다. 한 애플리케이션에 대한 메시지를 클러스터 내 다른 애플리케이션에 대한 메시지로부터 격리하는 다음 예에 대한 시작점으로 이 클러스터를 사용합니다.

181 페이지의 『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 클러스터된 큐 리모트 정의 및 개별 송신자 채널과 전송 큐를 사용합니다.

202 페이지의 『메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경』

클러스터된 큐 또는 전송 큐의 토픽에 대해 큐 관리자가 메시지를 저장하는 기본 방식을 변경할 수 있습니다. 기본값을 변경하여 게이트웨이 큐 관리자에서 클러스터 메시지를 격리하는 방식을 제공합니다.

184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 추가 클러스터 전송 큐를 사용하여 클러스터의 단일 큐 관리자로 메시지 트래픽을 분리합니다.

187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 솔루션은 추가 클러스터를 사용하여 메시지를 특정 클러스터 큐로 분리합니다.

261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』

클러스터 전송 큐의 선택에 대한 지침입니다. 하나의 공용 기본 큐, 별도의 기본 큐, 또는 수동으로 정의된 큐를 구성할 수 있습니다. 다중 클러스터 전송 큐 구성은 z/OS 이외의 플랫폼에 적용됩니다.

메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경

클러스터된 큐 또는 전송 큐의 토픽에 대해 큐 관리자가 메시지를 저장하는 기본 방식을 변경할 수 있습니다. 기본값을 변경하여 게이트웨이 큐 관리자에서 클러스터 메시지를 격리하는 방식을 제공합니다.

시작하기 전에

1. 게이트웨이 큐 관리자는 Version 7.5 이상 및 z/OS 이외의 플랫폼에 있어야 합니다.
2. 해당 태스크의 단계를 따라 197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』 의 198 페이지의 그림 37에 표시된 겹쳐진 클러스터를 구성하십시오.

이 태스크 정보

다중 클러스터 큐가 포함된 아키텍처를 구현하려면 게이트웨이 큐 관리자는 Version 7.5 이상에 있어야 합니다. 다중 클러스터 전송 큐를 사용하기 위해 수행해야 하는 작업은 게이트웨이 큐 관리자에서 기본 클러스터 전송 큐를 변경하는 것입니다. 큐 관리자 속성 **DEFCLXQ**의 값을 QM1에서 SCTQ로부터 CHANNEL로 변경합니다. 203 페이지의 [그림 39](#)의 내용을 참조하십시오. 이 다이어그램은 한 개의 메시지 플로우를 표시합니다. 다른 큐 관리자 또는 다른 클러스터로의 플로우에 대해서 큐 관리자는 추가 영구 동적 클러스터 전송 큐를 작성합니다. 각 클러스터 송신자 채널은 다른 클러스터 전송 큐에서 메시지를 전송합니다.

이러한 변경은 게이트웨이 큐 관리자를 클러스터에 처음으로 연결하지 않는 한 즉시 적용되지 않습니다. 이 태스크는 기존 구성에 대한 일반적인 변경 관리 단계를 포함합니다. 처음 클러스터에 조인할 때 개별 클러스터 전송 큐를 사용하도록 큐 관리자를 설정하려면 179 페이지의 『클러스터에 큐 관리자 추가: 개별 전송 큐』의 내용을 참조하십시오.

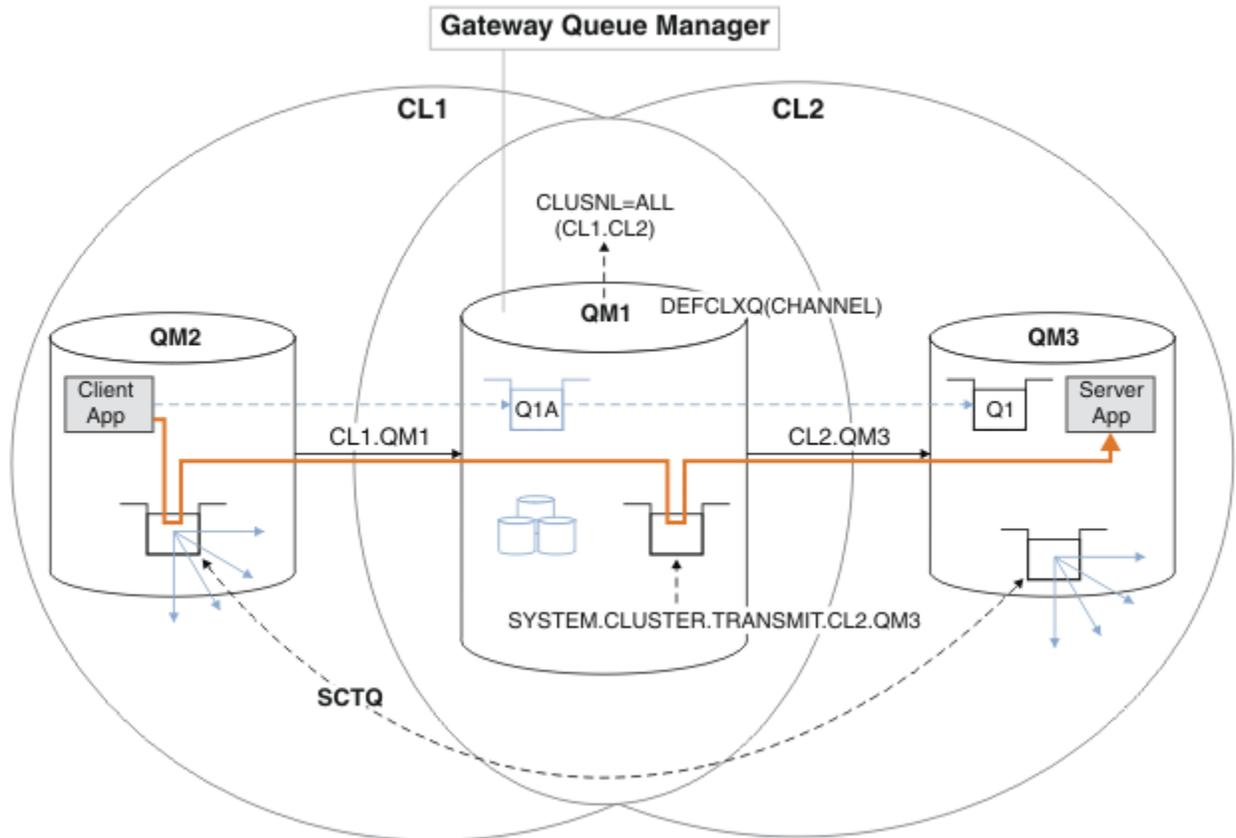


그림 39. 게이트웨이 큐 관리자에서 개별 클러스터 전송 큐를 사용하여 허브 및 스포크 아키텍처에 배치되는 클라이언트 서버 애플리케이션

프로시저

1. 개별 클러스터 전송 큐를 사용하도록 게이트웨이 큐 관리자를 변경하십시오.

```
*... On QM1
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. 개별 클러스터 전송 큐로 전환하십시오.

실행 중이 아닌 클러스터 송신자 채널은 다음에 시작할 때 개별 클러스터 전송 큐를 사용하도록 전환됩니다.

실행 중인 채널을 전환하려면 큐 관리자를 재시작하거나 다음 단계를 따르십시오.

- a) **SYSTEM.CLUSTER.TRANSMIT.QUEUE**로 실행되고 있는 클러스터 송신자 채널을 나열하십시오.

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

응답은 채널 상태 보고서 목록입니다.

```
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM2)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1412)) CURRENT
RQMNAME (QM2)              STATUS (RUNNING)
SUBSTATE (MQGET)           XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM3)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1413)) CURRENT
RQMNAME (QM3)              STATUS (RUNNING)
SUBSTATE (MQGET)           XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM5)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1415)) CURRENT
RQMNAME (QM5)              STATUS (RUNNING)
SUBSTATE (MQGET)           XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM4)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1414)) CURRENT
RQMNAME (QM4)              STATUS (RUNNING)
SUBSTATE (MQGET)           XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

b) 실행 중인 채널을 중지하십시오.

목록에 있는 각 채널에 대해 다음 명령을 실행하십시오.

```
*... On QM1
STOP CHANNEL (ChannelName)
```

여기서, *ChannelName*은 각각 CL1.QM2, CL1.QM4, CL1.QM3, CL1.QM5입니다.

응답은 명령이 승인됨입니다.

```
AMQ8019: Stop WebSphere MQ channel accepted.
```

c) 어떤 채널이 중지되는지 모니터하십시오.

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

응답은 여전히 실행 중인 채널과 중지되는 채널의 목록입니다.

```
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM2)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1412)) CURRENT
RQMNAME (QM2)              STATUS (STOPPED)
SUBSTATE ( )               XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM3)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1413)) CURRENT
RQMNAME (QM3)              STATUS (STOPPED)
SUBSTATE ( )               XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM5)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1415)) CURRENT
RQMNAME (QM5)              STATUS (STOPPED)
SUBSTATE ( )               XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM4)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1414)) CURRENT
```

```
RQMNAME(QM4)          STATUS(STOPPED)
SUBSTATE( )           XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

d) 중지된 각 채널을 시작하십시오.

이 단계를 실행되고 있는 모든 채널에 대해 수행하십시오. 채널이 중지되지 않으면 **STOP CHANNEL** 명령을 **FORCE** 옵션과 함께 다시 실행할 수 있습니다. **FORCE** 옵션 설정의 예는 채널이 중지되지 않고 채널을 동기화하기 위해 다른 큐 관리자를 재시작할 수 없는 경우입니다.

```
*... On QM1
START CHANNEL(CL2.QM5)
```

응답은 명령이 승인됨입니다.

AMQ8018: Start WebSphere MQ channel accepted.

e) 전환되는 전송 큐를 모니터하십시오.

"AMQ7341 메시지에 대한 게이트웨이 큐 관리자 오류 로그를 모니터링하십시오. 채널 CL2.QM3용 전송 큐는 SYSTEM.CLUSTER.TRANSMIT.QUEUE/CL2.QM3"입니다.

f) SYSTEM.CLUSTER.TRANSMIT.QUEUE가 더 이상 사용되지 않는지 확인하십시오.

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
DISPLAY QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE) CURDEPTH
```

응답은 채널 상태 보고서 및 SYSTEM.CLUSTER.TRANSMIT.QUEUE의 용량 목록입니다.

AMQ8420: Channel Status not found.

AMQ8409: Display Queue details.

```
QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)  TYPE(QLOCAL)
CURDEPTH(0)
```

g) 어떤 채널이 시작되는지 모니터하십시오.

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

응답은 이 경우 새 기본 클러스터 전송 큐에서 이미 실행 중인 채널의 목록입니다.

AMQ8417: Display Channel Status details.

```
CHANNEL(CL1.QM2)          CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))  CURRENT
RQMNAME(QM2)             STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM2)
```

AMQ8417: Display Channel Status details.

```
CHANNEL(CL2.QM3)          CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))  CURRENT
RQMNAME(QM3)             STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM3)
```

AMQ8417: Display Channel Status details.

```
CHANNEL(CL2.QM5)          CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1415))  CURRENT
RQMNAME(QM5)             STATUS(RUNNING)
SUBSTATE(MQGET)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)
```

AMQ8417: Display Channel Status details.

```
CHANNEL(CL1.QM4)          CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1414))  CURRENT
RQMNAME(QM4)             STATUS(RUNNING)
```

```
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL1.QM4)
```

다음에 수행할 작업

1. 큐 알리어스 정의 Q1A로 큐 이름을 해석하고 QM2에서 QM3의 Q1으로 메시지를 송신하여 자동 정의된 클러스터 전송 큐를 테스트하십시오.

- a. 메시지를 넣으려면 QM2에서 샘플 프로그램 **amqsput**를 실행하십시오.

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

- b. QM3의 Q1에서 메시지를 가져오려면 샘플 프로그램 **amqsget**를 실행하십시오.

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. 클러스터 큐에 대한 메시지가 시작되는 큐 관리자에서 클러스터 큐에 대한 보안을 구성하여 보안을 재구성할지 여부를 고려해보십시오.

관련 개념

[145 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』](#)

애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드는 클러스터 큐에 대해 원격으로 검사, SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사 또는 클러스터 큐나 클러스터 큐 관리자의 로컬 프로파일에 대해 검사입니다.

[258 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』](#)

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

관련 태스크

[179 페이지의 『클러스터에 큐 관리자 추가: 개별 전송 큐』](#)

다음 지시사항에 따라 사용자가 작성한 클러스터에 큐 관리자를 추가합니다. 클러스터 큐 및 토픽에 대한 메시지는 다중 클러스터 전송 큐를 사용하여 전송됩니다.

[197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』](#)

태스크의 지시사항에 따라 게이트웨이 큐 관리자가 포함된 중첩 클러스터를 구성합니다. 한 애플리케이션에 대한 메시지를 클러스터 내 다른 애플리케이션에 대한 메시지로부터 격리하는 다음 예에 대한 시작점으로 이 클러스터를 사용합니다.

[181 페이지의 『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』](#)

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 클러스터된 큐 리모트 정의 및 개별 송신자 채널과 전송 큐를 사용합니다.

[202 페이지의 『메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경』](#)

클러스터된 큐 또는 전송 큐의 토픽에 대해 큐 관리자가 메시지를 저장하는 기본 방식을 변경할 수 있습니다. 기본값을 변경하여 게이트웨이 큐 관리자에서 클러스터 메시지를 격리하는 방식을 제공합니다.

[184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』](#)

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 추가 클러스터 전송 큐를 사용하여 클러스터의 단일 큐 관리자로 메시지 트래픽을 분리합니다.

187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 솔루션은 추가 클러스터를 사용하여 메시지를 특정 클러스터 큐로 분리합니다.

261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』

클러스터 전송 큐의 선택에 대한 지침입니다. 하나의 공용 기본 큐, 별도의 기본 큐, 또는 수동으로 정의된 큐를 구성할 수 있습니다. 다중 클러스터 전송 큐 구성은 z/OS 이외의 플랫폼에 적용됩니다.

큐 관리자에서 클러스터 큐 제거

Toronto에서 INVENTQ 큐를 사용 안함으로 설정합니다. 모든 인벤토리 메시지를 New York으로 전송하고, 비어 있으면 토론토에서 INVENTQ 큐를 삭제하십시오.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 작업을 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- INVENTORY 클러스터가 194 페이지의 『큐를 호스팅하는 큐 관리자 추가』에서 설명된 대로 설정되었습니다. 네 개의 큐 관리자를 포함합니다. LONDON 및 NEWYORK 둘 다 전체 저장소를 보유하고 있습니다. PARIS 및 TORONTO 는 부분 저장소를 보유하고 있습니다. 인벤토리 애플리케이션은 New York 및 Toronto에 있는 시스템에서 실행되며 INVENTQ 큐에 메시지가 도착하면 구동됩니다.
- 감소된 워크로드로 인해 Toronto에서 인벤토리 애플리케이션을 더 이상 실행하지 않으려고 합니다. 큐 관리자 TORONTO가 호스팅하는 INVENTQ 큐를 사용 안함으로 설정하고 TORONTO가 NEWYORK의 INVENTQ 큐에 메시지를 공급하게 하려고 합니다.
- 네 시스템 사이에는 모두 네트워크 연결성이 있습니다.
- 네트워크 프로토콜은 TCP입니다.

이 태스크 정보

다음 단계에 따라 클러스터 큐를 제거하십시오.

프로시저

1. 큐가 더 이상 사용 가능하지 않음을 표시하십시오.

클러스터에서 큐를 제거하려면 로컬 큐 정의에서 클러스터 이름을 제거하십시오. TORONTO 에서 INVENTQ 을 변경하여 클러스터의 나머지 부분에서 액세스할 수 없도록 하십시오.

```
ALTER QLOCAL(INVENTQ) CLUSTER(' ')
```

2. 큐가 더 이상 사용 가능하지 않은지 확인하십시오.

전체 저장소 큐 관리자 LONDON 또는 NEWYORK에서 다음 명령을 발행하여 큐가 큐 관리자 TORONTO에 의해 더 이상 호스팅되지 않는지 확인하십시오.

```
DIS QCLUSTER (INVENTQ)
```

ALTER 명령이 성공적으로 완료된 경우 TORONTO가 결과에 나열되지 않습니다.

3. 큐를 사용 안함으로 설정하십시오.

추가 메시지가 기록되지 않도록 TORONTO에서 INVENTQ 큐를 사용 안함으로 설정하십시오.

```
ALTER QLOCAL(INVENTQ) PUT(DISABLED)
```

이제 MQ00_BIND_ON_OPEN 를 사용하여 이 큐로 전송 중인 메시지가 데드-레터 큐로 이동합니다. 모든 애플리케이션이 이 큐 관리자의 큐에 명시적으로 메시지를 넣지 못하도록 해야 합니다.

4. 큐가 비워질 때까지 큐를 모니터하십시오.

DISPLAY QUEUE 명령을 사용하고 IPPROCS, OPROCS 및 CURDEPTH 속성을 지정하여 큐를 모니터링하거나 IBM i에서 **WRKMQSTS** 명령을 사용하십시오. 입력 및 출력 프로세스의 수와 큐의 현재 용량이 0인 경우 이 큐는 비어 있는 큐입니다.

5. 채널을 모니터하여 인다우트(in-doubt) 메시지가 없는지 확인하십시오.

채널 INVENTORY.TORONTO에 인다우트 메시지가 없는지 확인하려면 다른 큐 관리자 각각에서 INVENTORY.TORONTO 이라는 클러스터 송신자 채널을 모니터하십시오. 각 큐 관리자에서 INDOUBT 매개 변수를 지정하여 DISPLAY CHSTATUS 명령을 발행하십시오.

```
DISPLAY CHSTATUS(INVENTORY.TORONTO) INDOUBT
```

인다우트(in-doubt) 메시지가 있는 경우 진행하기 전에 이 메시지를 해결해야 합니다. 예를 들어, RESOLVE 채널 명령 발행 또는 채널 중지 및 재시작을 시도할 수 있습니다.

6. 로컬 큐를 삭제하십시오.

TORONTO에서 인벤토리 애플리케이션에 전달할 메시지가 더 이상 없는 것으로 확인되면 큐를 삭제할 수 있습니다.

```
DELETE QLOCAL(INVENTQ)
```

7. 이제 Toronto의 시스템에서 인벤토리 애플리케이션을 제거할 수 있습니다.

이 애플리케이션을 제거하면 중복을 피하고 시스템 공간을 확보할 수 있습니다.

결과

이 태스크에서 설정되는 클러스터는 이전 태스크에서 설정된 클러스터와 유사합니다. 차이점은 큐 관리자 TORONTO에서 INVENTQ 큐가 더 이상 사용 가능하지 않다는 점입니다.

1단계에서 큐를 서비스에서 제거하면, TORONTO 큐 관리자가 두 개의 전체 저장소 큐 관리자에게 메시지를 전송했습니다. 이 메시지는 큐 관리자에게 상태가 변경되었음을 알렸습니다. 전체 저장소 큐 관리자는 INVENTQ에 관한 정보 업데이트를 요청한 클러스터에 있는 다른 큐 관리자에게 이 정보를 전달합니다.

큐 관리자가 메시지를 INVENTQ 큐에 넣으면 업데이트된 부분 저장소가 INVENTQ 큐는 NEWYORK 큐 관리자에게서만 사용 가능함을 표시합니다. 이 메시지는 NEWYORK 큐 관리자에게 송신됩니다.

다음에 수행할 작업

이 태스크에는 제거할 한 개의 큐와 이 큐를 제거할 한 개의 클러스터만 있습니다.

여러 클러스터 이름이 들어 있는 이름 목록을 참조하는 다수의 큐가 있는 것으로 가정하십시오. 예를 들어, TORONTO 큐 관리자는 INVENTQ 뿐만 아니라 PAYROLLQ, SALESQ 및 PURCHASESQ를 호스트할 수 있습니다. TORONTO는 이러한 큐를 적절한 클러스터 INVENTORY, PAYROLL, SALES 및 PURCHASES에서 사용 가능하도록 합니다. TORONTO 큐 관리자에서 클러스터 이름의 이름 목록을 정의하십시오.

```
DEFINE NAMELIST(TOROLIST)
  DESCR('List of clusters TORONTO is in')
  NAMES(INVENTORY, PAYROLL, SALES, PURCHASES)
```

각 큐 정의에 이름 목록을 추가하십시오.

```
DEFINE QLOCAL(INVENTQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PAYROLLQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(SALESQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PURCHASESQ) CLUSNL(TOROLIST)
```

SALES 조작이 PURCHASES 조작에 의해 대체되기 때문에 SALES 클러스터에서 모든 큐를 제거하려고 하는 것으로 가정합니다. 수행해야 하는 유일한 작업은 TOROLIST 이름 목록에서 SALES 클러스터의 이름을 제거하기 위해 이 이름 목록을 대체하는 것입니다.

이름 목록에 있는 클러스터 중 하나에서 단일 큐를 제거하려는 경우 나머지 클러스터 이름 목록을 포함하는 이름 목록을 작성하십시오. 그런 다음 새 이름 목록을 사용하도록 큐 정의를 대체하십시오. INVENTORY 클러스터에서 PAYROLLQ를 제거하려면 다음을 수행하십시오.

1. 다음과 같이 이름 목록을 작성하십시오.

```
DEFINE NAMLIST(TOROSHORTLIST)
DESCR('List of clusters TORONTO is in other than INVENTORY')
NAMES(PAYROLL, SALES, PURCHASES)
```

2. 다음과 같이 PAYROLLQ 큐 정의를 대체하십시오.

```
ALTER QLOCAL(PAYROLLQ) CLUSNL(TOROSHORTLIST)
```

다른 큐 관리자로 전체 저장소 이동

두 번째 저장소에 보유된 정보로부터 새 저장소를 빌드하여 한 큐 관리자에서 다른 큐 관리자로 전체 저장소를 이동합니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 작업을 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- INVENTORY 클러스터가 [177 페이지의 『클러스터에 큐 관리자 추가』](#)에서 설명된 대로 설정되었습니다.
- 비즈니스상의 이유로 큐 관리자 LONDON에서 전체 저장소를 제거하고 이를 큐 관리자 PARIS의 전체 저장소로 대체하려고 합니다. NEWYORK 큐 관리자는 계속해서 전체 저장소를 보유하게 됩니다.

이 태스크 정보

다음 단계에 따라 다른 큐 관리자로 전체 저장소 이동하십시오.

프로시저

1. 전체 저장소 큐 관리자가 되도록 PARIS를 변경하십시오.

PARIS에서 다음 명령을 실행하십시오.

```
ALTER QMGR REPOS(INVENTORY)
```

2. PARIS에서 CLUSSDR 채널을 추가하십시오.

PARIS에는 현재 LONDON을 가리키는 클러스터 송신자 채널이 있습니다. LONDON은 클러스터에 대한 전체 저장소를 더 이상 보유하지 않습니다. PARIS에는 현재 다른 전체 저장소를 보유하고 있는 NEWYORK을 가리키는 새 클러스터 송신자 채널이 있어야 합니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at NEWYORK')
```

3. NEWYORK에서 PARIS를 가리키는 CLUSSDR 채널을 정의하십시오.

현재 NEWYORK에는 LONDON을 가리키는 클러스터 송신자 채널이 있습니다. 다른 전체 저장소가 PARIS로 이동했으므로 PARIS를 가리키는 새 클러스터 송신자 채널을 NEWYORK에서 추가해야 합니다.

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from NEWYORK to repository at PARIS')
```

PARIS에 클러스터 송신자 채널을 추가하면 PARIS는 NEWYORK의 클러스터에 대해 학습합니다. 이는 NEWYORK의 정보를 사용하여 자체 전체 저장소를 빌드합니다.

4. 큐 관리자 PARIS에 이제 전체 저장소가 있는지 확인하십시오.

큐 관리자 PARIS가 큐 관리자 NEWYORK의 전체 저장소로부터 자체 전체 저장소를 빌드했는지 확인하십시오. 다음 명령을 실행하십시오.

```
DIS QCLUSTER(*) CLUSTER (INVENTORY)
DIS CLUSQMGR(*) CLUSTER (INVENTORY)
```

이러한 명령이 이 클러스터에서 NEWYORK에서와 같은 자원의 세부사항을 표시하는지 확인하십시오.

참고: 큐 관리자 NEWYORK이 사용 불가능한 경우 이 정보 빌드를 완료할 수 없습니다. 이 태스크가 완료될 때까지 다음 단계로 이동하지 마십시오.

5. LONDON에서 큐 관리자 정의를 대체하십시오.

마지막으로 클러스터에 대한 전체 저장소를 더 이상 보유하지 않도록 LONDON에서 이 큐 관리자를 대체하십시오. LONDON에서 다음 명령을 실행하십시오.

```
ALTER QMGR REPOS(' ')
```

큐 관리자가 더 이상 클러스터 정보를 수신하지 않습니다. 30일 후에는 해당 전체 저장소에 저장되는 정보가 만료됩니다. 이제 큐 관리자 LONDON이 자체 부분 저장소를 빌드합니다.

6. 미해결 정의를 제거하거나 변경하십시오.

클러스터의 새 배열이 예상대로 작동하는 것으로 확인되면 더 이상 정확하지 않은 수동으로 정의된 CLUSSDR 정의를 제거하거나 변경하십시오.

- PARIS 큐 관리자에서 LONDON에 대한 클러스터 송신자 채널을 중지하고 삭제한 다음 이 클러스터가 자동 채널을 다시 사용할 수 있도록 채널 시작 명령을 발행해야 합니다.

```
STOP CHANNEL (INVENTORY.LONDON)
DELETE CHANNEL (INVENTORY.LONDON)
START CHANNEL (INVENTORY.LONDON)
```

- NEWYORK 큐 관리자에서 LONDON에 대한 클러스터 송신자 채널을 중지하고 삭제한 다음 이 클러스터가 자동 채널을 다시 사용할 수 있도록 채널 시작 명령을 발행해야 합니다.

```
STOP CHANNEL (INVENTORY.LONDON)
DELETE CHANNEL (INVENTORY.LONDON)
START CHANNEL (INVENTORY.LONDON)
```

- LONDON을 가리키는 클러스터의 다른 모든 클러스터 송신자 채널을 NEWYORK 또는 PARIS를 가리키는 채널로 바꾸십시오. 이 작은 예에는 다른 클러스터 송신자 채널이 없습니다. 잊어버린 다른 사용자가 있는지 확인하려면 TYPE (CLUSSDR)를 지정하여 각 큐 관리자에서 DISPLAY CHANNEL 명령을 실행하십시오. 예를 들면, 다음과 같습니다.

```
DISPLAY CHANNEL(*) TYPE (CLUSSDR)
```

전체 저장소를 LONDON에서 PARIS로 이동한 후에 최대한 빨리 이 태스크를 수행하는 것이 중요합니다. 이 태스크를 수행하기 전에 INVENTORY.LONDON이라는 이름의 수동으로 정의된 CLUSSDR 채널이 있는 큐 관리자가 이 채널을 사용하여 정보에 대한 요청을 송신할 수 있습니다.

LONDON이 전체 저장소가 되기 위해 중지된 후에 그러한 요청을 수신하면 해당 큐 관리자 오류 로그에 오류 메시지를 기록합니다. 다음 예는 LONDON에 표시될 수 있는 오류 메시지를 보여줍니다.

- AMQ9428: Unexpected publication of a cluster queue object received
- AMQ9432: Query received by a non-repository queue manager

큐 관리자 LONDON은 더 이상 전체 저장소가 아니므로 정보 요청에 응답하지 않습니다. LONDON에서 정보를 요청하는 큐 관리자는 수동으로 정의된 CLUSSDR 정의가 PARIS를 가리키도록 수정될 때까지 클러스터 정보에 대해 NEWYORK에 의존해야 합니다. 이 상황은 장기적으로는 유효한 구성으로 허용되지 않아야 합니다.

결과

211 페이지의 그림 40에서는 이 태스크에 의해 설정되는 클러스터를 보여줍니다.

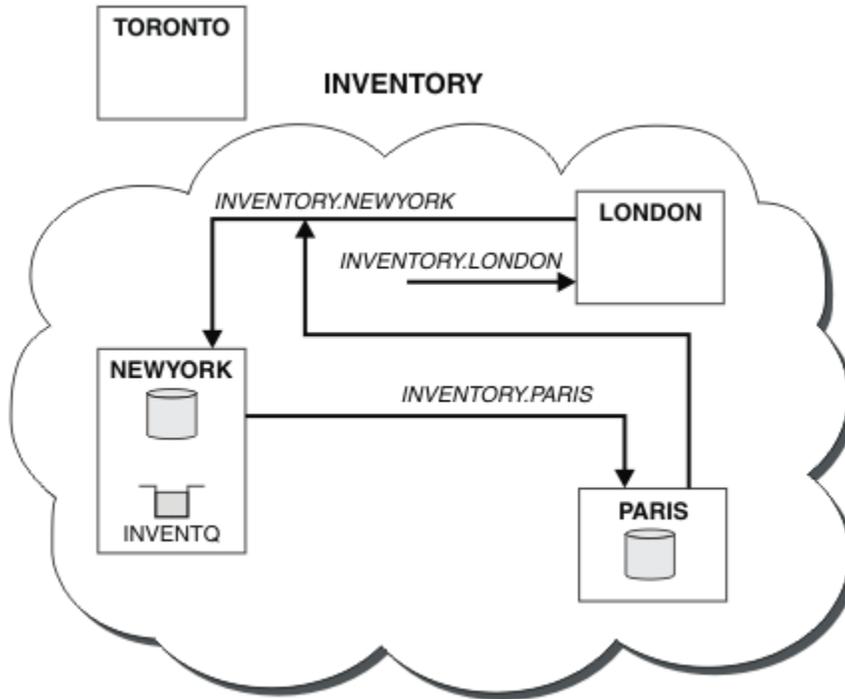


그림 40. 전체 저장소가 PARIS로 이동된 INVENTORY 클러스터

기존 네트워크를 클러스터로 변환

기존 분산 큐잉 네트워크를 클러스터로 변환하고 용량을 늘리기 위해 추가 큐 관리자를 추가합니다.

시작하기 전에

168 페이지의 『새 클러스터 설정』에서 209 페이지의 『다른 큐 관리자로 전체 저장소 이동』까지 새 클러스터를 작성하고 확장했습니다. 다음 두 태스크에서는 큐 관리자의 기존 네트워크를 클러스터로 변환하는 접근법에 대해 알아봅니다.

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 태스크를 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- IBM WebSphere MQ 네트워크가 이미 설정되어 있어 전국 체인점이 연결되어 있습니다. 이 네트워크는 허브 및 스포크 구조를 가지며 모든 큐 관리자는 하나의 중앙 큐 관리자에 연결됩니다. 중앙 큐 관리자는 인벤토리 애플리케이션이 실행되는 시스템에 있습니다. 애플리케이션은 각 큐 관리자가 리모트 큐 정의를 가지는 INVENTQ 큐에 메시지가 도착하면 구동됩니다.

이 네트워크는 212 페이지의 그림 41에서 설명됩니다.

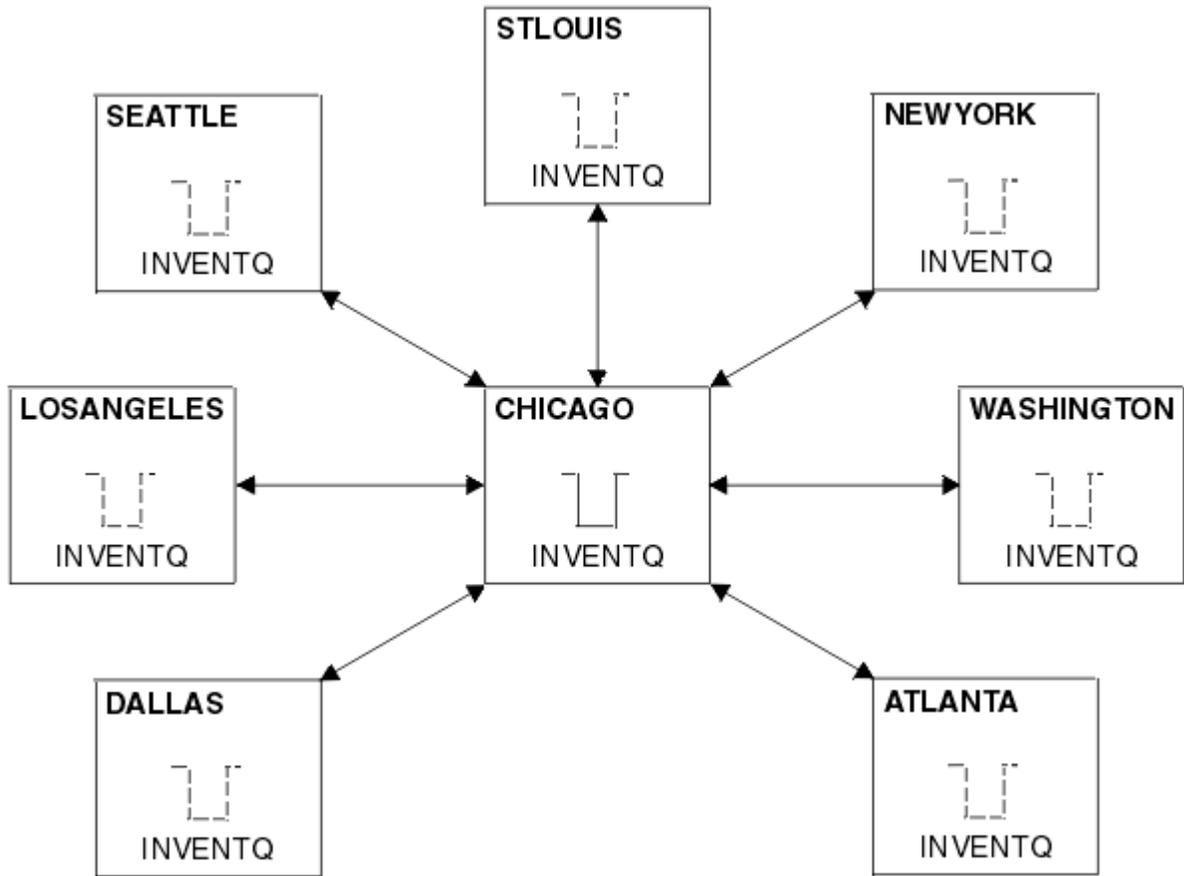


그림 41. 허브 및 스포크 네트워크

- 관리를 쉽게 하기 위해 이 네트워크를 클러스터로 변환하고 중앙 사이트에서 워크로드를 공유할 다른 큐 관리자를 작성합니다.

이 클러스터 이름은 CHNSTORE입니다.

참고: 클러스터 이름 CHNSTORE가 선택되어 *cluster-name.queue-manager* 형식의 이름으로 클러스터 수신자 채널 이름을 작성할 수 있습니다. 이 이름은 최대 20자입니다(예: CHNSTORE.WASHINGTON).

- 중앙 큐 관리자 둘 다 전체 저장소를 호스팅하고 인벤토리 애플리케이션에서 액세스 가능합니다.
- 인벤토리 애플리케이션은 중앙 큐 관리자 중 하나가 호스팅하는 INVENTQ 큐에 메시지가 도착하면 구동됩니다.
- 인벤토리 애플리케이션은 둘 이상의 큐 관리자에 의해 액세스 가능하고 병렬로 실행되는 유일한 애플리케이션이어야 합니다. 다른 모든 애플리케이션은 전과 같이 계속해서 실행됩니다.
- 모든 분기는 두 중앙 큐 관리자에 대한 네트워크 연결성을 가집니다.
- 네트워크 프로토콜은 TCP입니다.

이 태스크 정보

다음 단계에 따라 기존 네트워크를 클러스터로 변환하십시오.

프로시저

1. 메시지 연관관계에 대해 인벤토리 애플리케이션을 검토하십시오.

진행하기 전에 애플리케이션이 메시지 연관관계를 핸들링할 수 있는지 확인하십시오. 메시지 연관관계는 두 애플리케이션 간 교환되는 대화형 메시지 간 관계이며 이 관계에서는 메시지가 특정 큐 관리자 또는 특정 순서로 처리되어야 합니다. 메시지 연관관계에 대한 자세한 정보는 252 페이지의 『메시지 연관관계 핸들링』의 내용을 참조하십시오.

2. 전체 저장소 큐 관리자가 되도록 두 중앙 큐 관리자를 대체하십시오.

이 두 큐 관리자 CHICAGO 및 CHICAGO2는 이 네트워크의 허브에 있습니다. 이러한 체인 저장소 클러스터와 연관된 모든 활동을 이 두 큐 관리자에 집중하기로 결정했습니다. 인벤토리 애플리케이션 및 INVENTQ 큐 외에도 이러한 큐 관리자가 클러스터에 대한 두 개의 전체 저장소를 호스팅하게 하려고 합니다. 두 큐 관리자 각각에서 다음 명령을 실행하십시오.

```
ALTER QMGR REPOS(CHNSTORE)
```

3. 각 큐 관리자에서 CLUSRCVR 채널을 정의하십시오.

클러스터 내 각 큐 관리자에서 클러스터 수신자 채널 및 클러스터 송신자 채널을 정의하십시오. 어떤 채널을 처음 정의할지는 중요하지 않습니다.

CLUSRCVR 정의를 작성하여 각 큐 관리자, 해당 네트워크 주소 및 기타 정보를 클러스터에 알리십시오. 예를 들어, 큐 관리자 ATLANTA의 경우:

```
DEFINE CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(ATLANTA.CHSTORE.COM) CLUSTER(CHNSTORE)
DESCR('Cluster-receiver channel')
```

4. 각 큐 관리자에서 CLUSSDR 채널을 정의하십시오.

각 큐 관리자에서 CLUSSDR 정의를 작성하여 해당 큐 관리자를 전체 저장소 큐 관리자 중 어느 하나에 링크하십시오. 예를 들어, ATLANTA를 CHICAGO2에 링크할 수 있습니다.

```
DEFINE CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO2.CHSTORE.COM) CLUSTER(CHNSTORE)
DESCR('Cluster-sender channel to repository queue manager')
```

5. CHICAGO2에서 인벤토리 애플리케이션을 설치하십시오.

큐 관리자 CHICAGO에 이미 인벤토리 애플리케이션이 있습니다. 이제 큐 관리자 CHICAGO2에서 이 애플리케이션의 사본을 작성해야 합니다.

6. 중앙 큐 관리자에서 INVENTQ 큐를 정의하십시오.

CHICAGO에서 큐 INVENTQ의 로컬 큐 정의를 수정하여 이 큐를 클러스터에서 사용 가능하게 하십시오. 다음 명령을 실행하십시오.

```
ALTER QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

CHICAGO2에서 같은 큐에 대한 정의를 작성하십시오.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

z/OS에서 **CSQUTIL**의 COMMAND 기능의 MAKEDEF 옵션을 사용하여 CHICAGO에 있는 INVENTQ의 CHICAGO2에 정확한 사본을 만들 수 있습니다.

이러한 정의를 작성하면 메시지가 CHICAGO 및 CHICAGO2에 있는 전체 저장소에 송신되어 해당 정보가 업데이트됩니다. INVENTQ에 메시지를 넣을 때 큐 관리자는 전체 저장소에서 메시지의 목적지를 선택할 수 있는지 확인합니다.

7. 클러스터 변경사항이 전파되었는지 확인하십시오.

이전 단계에서 사용자가 작성한 정의가 클러스터 전체에 전파되었는지 확인하십시오. 전체 저장소 큐 관리자에서 다음 명령을 실행하십시오.

```
DIS QCLUSTER(INVENTQ)
```

상호 연결된 새 클러스터 추가

기존 클러스터와 일부 큐 관리자를 공유하는 새 클러스터를 추가합니다.

시작하기 전에

참고:

1. 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 작업을 시작하기 전에 저장소가 사용 가능한지 확인하십시오.
2. 이 작업을 시작하기 전에 큐 이름 충돌이 있는지 확인하고 그 결과를 파악합니다. 진행하기 위해 큐의 이름을 바꾸거나 큐 알리어스를 설정해야 할 수 있습니다.

시나리오:

- [211 페이지의 『기존 네트워크를 클러스터로 변환』](#)에 설명된 대로 WebSphere MQ 클러스터가 설정되어 있습니다.
- MAILORDER라고 하는 새 클러스터가 구현되어야 합니다. 이 클러스터는 CHNSTORE 클러스터에 있는 네 개의 큐 관리자, CHICAGO, CHICAGO2, SEATTLE, ATLANTA 및 두 개의 추가 큐 관리자 (HARTFORD 및 OMAHA)를 포함합니다. MAILORDER 애플리케이션은 큐 관리자 OMAHA에 연결된 Omaha에 있는 시스템에서 실행됩니다. 이는 클러스터의 다른 큐 관리자가 MORDERQ 큐에 메시지를 넣는 방식으로 구동됩니다.
- MAILORDER 클러스터의 전체 저장소는 두 개의 큐 관리자 CHICAGO 및 CHICAGO2에서 유지보수됩니다.
- 네트워크 프로토콜은 TCP입니다.

이 태스크 정보

다음 단계에 따라 상호 연결된 새 클러스터를 추가하십시오.

프로시저

1. 클러스터 이름의 이름 목록을 작성하십시오.

이제 CHICAGO 및 CHICAGO2의 전체 저장소 큐 관리자가 클러스터 CHNSTORE 및 MAILORDER 둘 다에 대한 전체 저장소를 보유하고 있습니다. 먼저 클러스터의 이름을 포함하는 이름 목록을 작성하십시오. CHICAGO 및 CHICAGO2에서 다음과 같이 이름 목록을 정의하십시오.

```
DEFINE NAMLIST(CHAINMAIL)
  DESCR('List of cluster names')
  NAMES(CHNSTORE, MAILORDER)
```

2. 두 큐 관리자 정의를 대체하십시오.

이제 CHICAGO 및 CHICAGO2에서 두 큐 관리자 정의를 대체하십시오. 현재 이러한 정의는 큐 관리자가 클러스터 CHNSTORE의 전체 저장소를 보유함을 표시합니다. 큐 관리자가 CHAINMAIL 이름 목록에 나열된 모든 클러스터에 대해 전체 저장소를 보유함을 표시하도록 정의를 변경하십시오. CHICAGO 및 CHICAGO2 큐 관리자 정의를 변경하십시오.

```
ALTER QMGR REPOS(' ') REPOSNL(CHAINMAIL)
```

3. CHICAGO 및 CHICAGO2에서 CLUSRCVR 채널을 변경하십시오.

CHICAGO 및 CHICAGO2의 CLUSRCVR 채널 정의는 채널이 CHNSTORE 클러스터에서 사용 가능함을 표시합니다. CHAINMAIL 이름 목록에 나열된 모든 클러스터에서 채널을 사용할 수 있음을 표시하도록 클러스터 수신자 정의를 변경해야 합니다. 다음과 같이 CHICAGO에서 클러스터 수신자 정의를 변경하십시오.

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR)
  CLUSTER(' ') CLUSNL(CHAINMAIL)
```

CHICAGO2에서 다음 명령을 입력하십시오.

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSRCVR)
  CLUSTER(' ') CLUSNL(CHAINMAIL)
```

4. CHICAGO 및 CHICAGO2에서 CLUSSDR 채널을 변경하십시오.

이름 목록을 추가하려면 두 CLUSSDR 채널 정의를 변경하십시오. CHICAGO에서 다음 명령을 입력하십시오.

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

CHICAGO2에서 다음 명령을 입력하십시오.

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

5. SEATTLE 및 ATLANTA에 이름 목록을 작성하십시오.

SEATTLE 및 ATLANTA는 둘 이상의 클러스터의 멤버가 되므로 클러스터의 이름이 포함된 이름 목록을 작성해야 합니다. SEATTLE 및 ATLANTA에서 다음과 같이 이름 목록을 정의하십시오.

```
DEFINE NAMELIST(CHAINMAIL)
DESCR('List of cluster names')
NAMES(CHNSTORE, MAILORDER)
```

6. SEATTLE 및 ATLANTA에서 CLUSRCVR 채널을 변경하십시오.

SEATTLE 및 ATLANTA의 CLUSRCVR 채널 정의는 채널이 CHNSTORE 클러스터에서 사용 가능함을 표시합니다. CHAINMAIL 이름 목록에 나열된 모든 클러스터에서 클러스터 수신 채널을 사용할 수 있음을 표시하도록 이 채널의 정의를 변경하십시오. SEATTLE에서 다음 명령을 입력하십시오.

```
ALTER CHANNEL(CHNSTORE.SEATTLE) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

ATLANTA에서 다음 명령을 입력하십시오.

```
ALTER CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

7. SEATTLE 및 ATLANTA에서 CLUSSDR 채널을 변경하십시오.

이름 목록을 추가하려면 두 CLUSSDR 채널 정의를 변경하십시오. SEATTLE에서 다음 명령을 입력하십시오.

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

ATLANTA에서 다음 명령을 입력하십시오.

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

8. HARTFORD와 OMAHA에서 CLUSRCVR 및 CLUSSDR 채널을 정의하십시오.

두 새 큐 관리자 HARTFORD 및 OMAHA에서 클러스터 수신자 및 클러스터 송신자 채널을 정의하십시오. 정의를 작성하는 순서는 문제가 되지 않습니다. HARTFORD에서 다음을 입력하십시오.

```
DEFINE CHANNEL(MAILORDER.HARTFORD) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(HARTFORD.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for HARTFORD')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from HARTFORD to repository at CHICAGO')
```

OMAHA에서 다음을 입력하십시오.

```
DEFINE CHANNEL(MAILORDER.OMAHA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(OMAHA.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for OMAHA')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from OMAHA to repository at CHICAGO')
```

9. OMAHA에서 MORDERQ 큐를 정의하십시오.

이 태스크를 완료하는 마지막 단계는 큐 관리자 OMAHA에서 MORDERQ 큐를 정의하는 것입니다. OMAHA에서 다음을 입력하십시오.

```
DEFINE QLOCAL(MORDERQ) CLUSTER(MAILORDER)
```

10. 클러스터 변경사항이 전파되었는지 확인하십시오.

이전 단계에서 사용자가 작성한 정의가 클러스터 전체에 전파되었는지 확인하십시오. 전체 저장소 큐 관리자에서 다음 명령을 실행하십시오.

```
DIS QCLUSTER (MORDERQ)  
DIS CLUSQMGR
```

11.

결과

이 태스크에서 설정되는 클러스터는 [217 페이지의 그림 42](#)에 표시됩니다.

이제 두 개의 중첩 클러스터가 있습니다. 두 클러스터에 대한 전체 저장소는 CHICAGO 및 CHICAGO2에 있습니다. OMAHA에서 실행되는 메일 오더 애플리케이션은 CHICAGO에서 실행되는 인벤토리 애플리케이션과는 독립적입니다. 그러나 CHNSTORE 클러스터에 있는 일부 큐 관리자는 MAILORDER 클러스터에도 있으므로 메시지를 애플리케이션에 송신할 수 있습니다. 이 태스크를 수행하여 두 클러스터를 중첩하기 전에 큐 이름 충돌 가능성을 확인하십시오.

클러스터 CHNSTORE 의 NEWYORK 및 클러스터 MAILORDER의 OMAHA 에 ACCOUNTQ 라는 큐가 있다고 가정합니다. If you overlap the clusters and then an application on SEATTLE puts a message to the queue ACCOUNTQ , the message can go to either instance of the ACCOUNTQ .

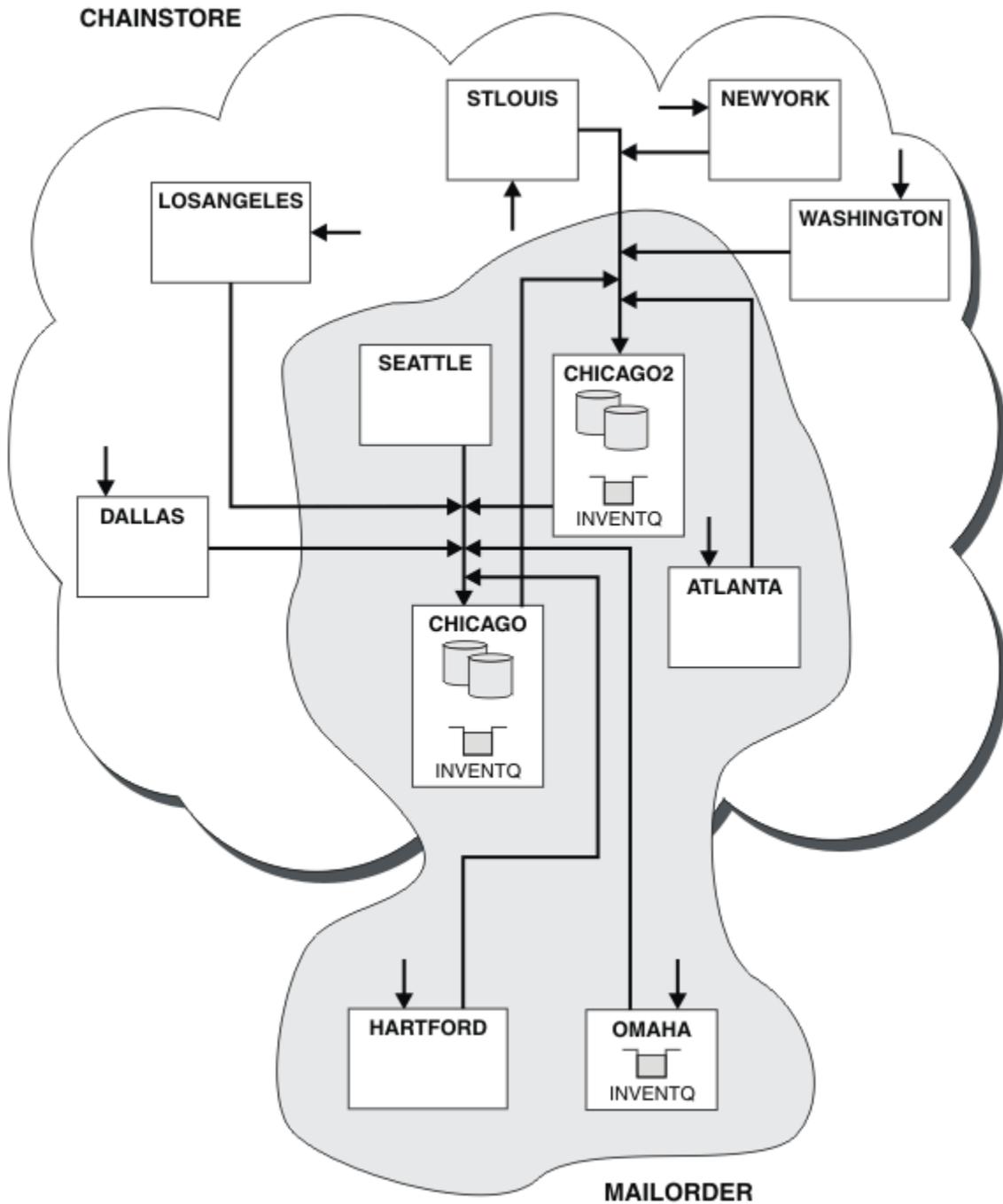


그림 42. 상호 연결된 클러스터

다음에 수행할 작업

CHAINSTORE라고 하는 한 개의 대형 클러스터를 구성하기 위해 MAILORDER 클러스터와 CHAINSTORE 클러스터를 병합하기로 결정했다고 가정하십시오.

MAILORDER 클러스터를 CHAINSTORE 클러스터와 병합하려면 CHICAGO 및 CHICAGO2가 전체 저장소를 보유하도록 하십시오.

- CHICAGO 및 CHICAGO2에 대한 큐 관리자 정의를 변경하고, 이름 목록 (CHAINMAIL) 을 지정하는 REPOSNL 속성을 제거하고 클러스터 이름 (CHNSTORE) 을 지정하는 REPOS 속성으로 바꾸십시오. 예를 들어:

```
ALTER QMGR(CHICAGO) REPOSNL(' ') REPOS(CHNSTORE)
```

- MAILORDER 클러스터의 각 큐 관리자에서 모든 채널 정의 및 큐 정의를 대체하여 CLUSTER 속성의 값을 MAILORDER에서 CHNSTORE로 변경하십시오. 예를 들어, HARTFORD에서 다음을 입력하십시오.

```
ALTER CHANNEL(MAILORDER.HARTFORD) CLUSTER(CHNSTORE)
```

OMAHA에서 다음을 입력하십시오.

```
ALTER QLOCAL(MORDERQ) CLUSTER(CHNSTORE)
```

- Alter all definitions that specify the cluster namelist CHAINMAIL, that is, the CLUSRCVR and 클러스터링 channel definitions at CHICAGO, CHICAGO2, SEATTLE, and ATLANTA, to specify instead the cluster CHNSTORE.

이 예에서 이름 목록 사용의 장점을 확인할 수 있습니다. CHICAGO 및 CHICAGO2 에 대한 큐 관리자 정의를 대체하는 대신 이름 목록 CHAINMAIL의 값을 변경할 수 있습니다. 이와 마찬가지로 CHICAGO, CHICAGO2, SEATTLE 및 ATLANTA에서 CLUSRCVR 및 CLUSSDR 채널 정의를 대체하는 대신 이름 목록을 대체하여 필요한 결과를 달성할 수 있습니다.

클러스터 네트워크 제거

네트워크에서 클러스터를 제거하고 분산 큐잉 구성을 복원합니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 태스크를 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- IBM WebSphere MQ 클러스터가 [211 페이지의 『기존 네트워크를 클러스터로 변환』](#)에서 설명된 대로 설정되었습니다.
- 이 클러스터를 이제 시스템에서 제거하려고 합니다. 큐 관리자의 네트워크는 클러스터가 구현되기 전과 같이 계속해서 작동합니다.

이 태스크 정보

다음 단계에 따라 클러스터 네트워크를 제거하십시오.

프로시저

1. CHNSTORE 클러스터에서 클러스터 큐를 제거하십시오.

CHICAGO 및 CHICAGO2에서 큐 INVENTQ 의 로컬 큐 정의를 수정하여 클러스터에서 큐를 제거하십시오. 다음 명령을 실행하십시오.

```
ALTER QLOCAL(INVENTQ) CLUSTER(' ')
```

큐를 대체하면 전체 저장소의 정보가 업데이트되고 클러스터 전체에 전파됩니다.

MQOO_BIND_NOT_FIXED를 사용하는 활성 애플리케이션 및 큐가 DEFBIND(NOTFIXED)로 정의된 MQOO_BIND_AS_Q_DEF 를 사용하는 애플리케이션은 다음 번 시도된 MQPUT 또는 MQPUT1 호출에서 실패합니다. 이유 코드 MQRC_UNKNOWN_OBJECT_NAME이 리턴됩니다.

단계 1을 처음 수행할 필요는 없지만 대신 단계 4 후에 단계 1을 수행하십시오.

2. 클러스터 큐에 대한 액세스가 있는 모든 애플리케이션을 중지하십시오.

클러스터 큐에 대한 액세스가 있는 모든 애플리케이션을 중지하십시오. 그렇지 않으면 5단계에서 클러스터를 새로 고칠 때 일부 클러스터 정보가 로컬 큐 관리자에 남아 있을 수 있습니다. 모든 애플리케이션이 중지되고 클러스터 채널의 연결이 끊어진 경우 이 정보가 제거됩니다.

3. 전체 저장소 큐 관리자에서 저장소 속성을 제거하십시오.

CHICAGO 및 CHICAGO2에서 큐 관리자 정의를 수정하여 저장소 속성을 제거하십시오. 이를 수행하려면 다음 명령을 실행하십시오.

```
ALTER QMGR REPOS(' ')
```

큐 관리자는 클러스터 내 다른 큐 관리자에게 더 이상 전체 저장소를 보유하지 않음을 알립니다. 다른 큐 관리자가 이 정보를 수신하면 전체 저장소가 종료되었음을 나타내는 메시지가 표시됩니다. 더 이상 CHNSTORE 클러스터에 사용할 수 있는 저장소가 없음을 나타내는 하나 이상의 메시지도 표시됩니다.

4. 클러스터 채널을 제거하십시오.

CHICAGO에서 클러스터 채널을 제거하십시오.

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) CLUSTER(' ')  
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR) CLUSTER(' ')
```

참고: CLUSSDR 명령을 먼저 발행한 다음 CLUSRCVR 명령을 발행하는 것이 중요합니다. CLUSRCVR 명령을 먼저 발행한 다음 CLUSSDR 명령을 발행해서는 안 됩니다. 이렇게 할 경우 상태가 STOPPED인 인다우트(indoubt) 채널이 작성됩니다. 그런 다음 START CHANNEL 명령을 실행하여 중지된 채널을 복구해야 합니다 (예: START CHANNEL(CHNSTORE.CHICAGO)).

클러스터 CHNSTORE에 대한 저장소가 없음을 나타내는 메시지가 표시됩니다.

단계 1에서 설명한 대로 클러스터 큐를 제거하지 않은 경우 지금 제거하십시오.

5. 클러스터 채널을 중지하십시오.

CHICAGO에서 다음 명령을 사용하여 클러스터 채널을 중지하십시오.

```
STOP CHANNEL(CHNSTORE.CHICAGO2)  
STOP CHANNEL(CHNSTORE.CHICAGO)
```

6. 클러스터에 있는 각 큐 관리자에 대해 단계 4와 5를 반복하십시오.
7. 클러스터 채널을 중지한 다음 각 큐 관리자에서 클러스터 채널 및 클러스터 큐에 대한 모든 정의를 제거하십시오.
8. 옵션: 큐 관리자가 보유하는 캐시된 클러스터 정보를 지우십시오.
큐 관리자가 더 이상 클러스터의 멤버가 아니더라도 큐 관리자 각각은 클러스터에 대한 정보의 캐시된 사본을 보유합니다. 이 데이터를 제거하려는 경우 [223 페이지의 『큐 관리자를 해당 클러스터 전 상태로 복원』](#) 태스크를 참조하십시오.
9. INVENTQ에 대한 리모트 큐 정의를 바꾸십시오.

네트워크가 계속해서 작동할 수 있도록 모든 큐 관리자에서 INVENTQ에 대한 리모트 큐 정의를 바꾸십시오.

10. 클러스터를 정리하십시오.

더 이상 필요하지 않은 큐 또는 채널 정의를 삭제하십시오.

클러스터에서 큐 관리자 제거

큐 관리자가 클러스터에서 일반적으로 하나 이상의 전체 저장소와 통신할 수 있는 시나리오에서 클러스터로부터 큐 관리자를 제거합니다.

시작하기 전에

이 메소드는 최소 하나 이상의 전체 저장소가 사용 가능하고 이러한 저장소에 제거할 큐 관리자가 접속할 수 있는 시나리오에 대한 우수 사례입니다. 이 메소드에는 최소한의 수동 개입이 포함되고 큐 관리자가 클러스터에서 제

어떤 방식으로 탈퇴하는 것을 협상할 수 있습니다. 제거할 큐 관리자가 전체 저장소에 접속할 수 없는 경우 [221 페이지의 『클러스터에서 큐 관리자 제거: 대체 메소드』](#)의 내용을 참조하십시오.

클러스터에서 큐 관리자를 제거하기 전에 이 큐 관리자가 클러스터에 필요한 자원을 더 이상 호스팅하지 않음을 확인해야 합니다.

- 큐 관리자가 전체 저장소를 호스팅하는 경우, [209 페이지의 『다른 큐 관리자로 전체 저장소 이동』](#)의 1-4 단계를 완료하십시오.
- 큐 관리자가 클러스터 큐를 호스팅하는 경우 [207 페이지의 『큐 관리자에서 클러스터 큐 제거』](#)의 단계 1 - 7을 완료하십시오.
- 큐 관리자가 클러스터 토픽을 호스팅하는 경우, 토픽을 삭제하거나(예를 들어 `DELETE TOPIC` 명령을 사용하여) 기타 호스트로 이동하십시오.

참고: 클러스터에서 큐 관리자를 제거하고 이 큐 관리자가 여전히 클러스터 토픽을 호스팅하는 경우 이 큐 관리자는 이 토픽이 삭제될 때까지 계속해서 클러스터에 남겨진 발행물을 큐 관리자에게 전달하려고 시도할 수 있습니다.

이 태스크 정보

이 태스크 예에서는 INVENTORY 클러스터에서 큐 관리자 LONDON을 제거합니다. INVENTORY 클러스터는 [177 페이지의 『클러스터에 큐 관리자 추가』](#)에 설명된 대로 설정되고 [207 페이지의 『큐 관리자에서 클러스터 큐 제거』](#)에 설명된 대로 수정됩니다.

클러스터에서 큐 관리자를 제거하기 위한 프로세스는 큐 관리자를 추가하는 프로세스보다 복잡합니다.

큐 관리자가 클러스터에 조인하면 기존 클러스터 멤버는 새 큐 관리자에 대한 지식이 없기 때문에 새 큐 관리자와 상호작용하지 않습니다. 전체 저장소에 연결할 수 있도록 조인한 큐 관리자에 대해 새 송신자 및 수신자 채널이 작성되어야 합니다.

클러스터에서 큐 관리자가 제거되면 이 큐 관리자에 연결된 애플리케이션이 클러스터 외에서 호스팅되는 큐와 같은 오브젝트를 사용할 가능성이 높습니다. 또한 클러스터 내 다른 큐 관리자에 연결되는 애플리케이션은 대상 큐 관리자에서 호스팅되는 오브젝트를 사용할 수 있습니다. 이러한 애플리케이션의 결과로 현재 큐 관리자는 클러스터에 조인할 때 사용했던 전체 저장소 외 클러스터 멤버와의 통신을 설정하기 위해 추가 송신자 채널을 작성할 수 있습니다. 클러스터 내 모든 큐 관리자에는 다른 클러스터 멤버를 설명하는 데이터의 캐시된 사본이 있습니다. 여기에는 제거 중인 것도 포함될 수 있습니다.

프로시저

1. 수동으로 정의된 클러스터 수신자 채널을 대체하여 큐 관리자 LONDON에 있는 클러스터로부터 제거하십시오.

```
ALTER CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) CLUSTER(' ')
```

2. 수동으로 정의된 클러스터 송신자 채널을 대체하여 큐 관리자 LONDON에 있는 클러스터로부터 제거하십시오.

```
ALTER CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) CLUSTER(' ')
```

클러스터 내 다른 큐 관리자는 이 큐 관리자 및 해당 클러스터 자원이 더 이상 클러스터에 속하지 않음을 알게 됩니다.

3. 클러스터의 전체 저장소로 플로우되기 위해 대기 중인 메시지가 없을 때까지 큐 관리자 LONDON에서 클러스터 전송 큐를 모니터링하십시오.

```
DISPLAY CHSTATUS(INVENTORY.LONDON) XQMSGSA
```

메시지가 전송 큐에 남아 있는 경우 계속하기 전에 메시지가 PARIS 및 NEWYORK 전체 저장소로 송신되지 않는 이유를 판별하십시오.

결과

큐 관리자 LONDON은 더 이상 클러스터에 속하지 않습니다. 그러나 이 큐 관리자는 여전히 독립적 큐 관리자로 작동할 수 있습니다.

다음에 수행할 작업

이러한 변경의 결과는 클러스터의 나머지 멤버에서 다음 명령을 발행하여 확인할 수 있습니다.

```
DISPLAY CLUSQMGR(LONDON)
```

큐 관리자에 대해 자동 정의된 클러스터 송신자 채널이 중지될 때까지 큐 관리자는 계속해서 표시됩니다. 이러한 중지가 발생할 때까지 기다리거나 다음 명령을 발행하여 활성 인스턴스에 대해 계속해서 모니터링할 수 있습니다.

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

이 큐 관리자로 메시지가 더 이상 전달되지 않는 것이 확실하면 클러스터의 나머지 멤버에서 다음 명령을 발행하여 LONDON에 대한 클러스터 송신자 채널을 중지할 수 있습니다.

```
STOP CHANNEL(INVENTORY.LONDON) STATUS(INACTIVE)
```

변경이 클러스터 전체에 전파되고 이 큐 관리자에 더 이상 메시지가 전달되지 않으면 LONDON에서 CLUSRCVR 채널을 중지하고 삭제하십시오.

```
STOP CHANNEL(INVENTORY.LONDON)  
DELETE CHANNEL(INVENTORY.LONDON)
```

177 페이지의 『클러스터에 큐 관리자 추가』에 설명된 대로 제거된 큐 관리자는 나중에 해당 클러스터에 다시 추가할 수 있습니다. 제거된 큐 관리자는 최대 90일 동안 클러스터의 나머지 멤버에 대한 지식을 계속해서 캐시합니다. 이 캐시가 만료될 때까지 대기하는 것을 원치 않는 경우 223 페이지의 『큐 관리자를 해당 클러스터 전 상태로 복원』에서 설명된 대로 강제로 제거할 수 있습니다.

클러스터에서 큐 관리자 제거: 대체 메소드

중요한 시스템 또는 구성 문제로 인해 큐 관리자가 클러스터에서 전체 저장소와 통신할 수 없는 시나리오에서 클러스터로부터 큐 관리자를 제거합니다.

시작하기 전에

클러스터에서 큐 관리자를 수동으로 제거하는 이 대체 메소드는 제거된 큐 관리자를 클러스터에 링크하는 모든 클러스터 채널을 중지 및 삭제하고 클러스터에서 이 큐 관리자를 강제로 제거합니다. 이 메소드는 제거할 큐 관리자가 모든 전체 저장소와 통신할 수 없는 시나리오에서 사용됩니다. 이는 예를 들면, 큐 관리자가 작업을 중지했거나 큐 관리자와 클러스터 사이에 지속적 통신 실패가 있기 때문일 수 있습니다. 그렇지 않으면 가장 일반적인 방법: 219 페이지의 『클러스터에서 큐 관리자 제거』를 사용하십시오.

클러스터에서 큐 관리자를 제거하기 전에 이 큐 관리자가 클러스터에 필요한 자원을 더 이상 호스팅하지 않음을 확인해야 합니다.

- 큐 관리자가 전체 저장소를 호스팅하는 경우, 209 페이지의 『다른 큐 관리자로 전체 저장소 이동』의 1-4 단계를 완료하십시오.
- 큐 관리자가 클러스터 큐를 호스팅하는 경우 207 페이지의 『큐 관리자에서 클러스터 큐 제거』의 단계 1 - 7을 완료하십시오.
- 큐 관리자가 클러스터 토픽을 호스팅하는 경우, 토픽을 삭제하거나(예를 들어 `DELETE TOPIC` 명령을 사용하여) 기타 호스트로 이동하십시오.

참고: 클러스터에서 큐 관리자를 제거하고 이 큐 관리자가 여전히 클러스터 토픽을 호스팅하는 경우 이 큐 관리자는 이 토픽이 삭제될 때까지 계속해서 클러스터에 남겨진 발행물을 큐 관리자에게 전달하려고 시도할 수 있습니다.

이 태스크 정보

이 태스크 예에서는 INVENTORY 클러스터에서 큐 관리자 LONDON을 제거합니다. INVENTORY 클러스터는 177 페이지의 『클러스터에 큐 관리자 추가』에 설명된 대로 설정되고 207 페이지의 『큐 관리자에서 클러스터 큐 제거』에 설명된 대로 수정됩니다.

클러스터에서 큐 관리자를 제거하기 위한 프로세스는 큐 관리자를 추가하는 프로세스보다 복잡합니다.

큐 관리자가 클러스터에 조인하면 기존 클러스터 멤버는 새 큐 관리자에 대한 지식이 없기 때문에 새 큐 관리자와 상호작용하지 않습니다. 전체 저장소에 연결할 수 있도록 조인한 큐 관리자에 대해 새 송신자 및 수신자 채널이 작성되어야 합니다.

클러스터에서 큐 관리자가 제거되면 이 큐 관리자에 연결된 애플리케이션이 클러스터 외에서 호스팅되는 큐와 같은 오브젝트를 사용할 가능성이 높습니다. 또한 클러스터 내 다른 큐 관리자에 연결되는 애플리케이션은 대상 큐 관리자에 호스팅되는 오브젝트를 사용할 수 있습니다. 이러한 애플리케이션의 결과로 현재 큐 관리자는 클러스터에 조인할 때 사용했던 전체 저장소 외 클러스터 멤버와의 통신을 설정하기 위해 추가 송신자 채널을 작성할 수 있습니다. 클러스터 내 모든 큐 관리자에게는 다른 클러스터 멤버를 설명하는 데이터의 캐시된 사본이 있습니다. 여기에는 제거 중인 것도 포함될 수 있습니다.

이 프로시저는 큐 관리자가 별다른 문제 없이 클러스터에서 떠날 때까지 대기할 수 없는 비상 시에 적절할 수 있습니다.

프로시저

1. 클러스터 내 다른 큐 관리자와 통신하는 데 사용되는 모든 채널을 중지하십시오. MODE (FORCE)를 사용하여 큐 관리자 LONDON에서 CLUSRCVR 채널을 중지하십시오. 그렇지 않은 경우 송신자 큐 관리자가 채널을 중지할 때까지 대기해야 될 수도 있습니다.

```
STOP CHANNEL (INVENTORY.LONDON) MODE(FORCE)
STOP CHANNEL (INVENTORY.TORONTO)
STOP CHANNEL (INVENTORY.PARIS)
STOP CHANNEL (INVENTORY.NEWYORK)
```

2. 채널이 중지될 때까지 큐 관리자 LONDON에서 채널 상태를 모니터링하십시오.

```
DISPLAY CHSTATUS (INVENTORY.LONDON)
DISPLAY CHSTATUS (INVENTORY.TORONTO)
DISPLAY CHSTATUS (INVENTORY.PARIS)
DISPLAY CHSTATUS (INVENTORY.NEWYORK)
```

채널이 중지된 후에는 클러스터 내 다른 큐 관리자와 애플리케이션 메시지를 주고 받지 않습니다.

3. 큐 관리자 LONDON에서 수동으로 정의된 클러스터 채널을 삭제하십시오.

```
DELETE CHANNEL (INVENTORY.NEWYORK)
DELETE CHANNEL (INVENTORY.TORONTO)
```

4. 클러스터 내 나머지 큐 관리자는 제거된 큐 관리자에 대한 지식을 계속 보유하고 제거된 큐 관리자에게 계속 해서 메시지를 송신할 수도 있습니다. 나머지 큐 관리자에서 이러한 지식을 영구 제거하려면 전체 저장소 중 하나의 클러스터에서 제거된 큐 관리자를 재설정하십시오.

```
RESET CLUSTER (INVENTORY) ACTION (FORCEREMOVE) QMNAME (LONDON) QUEUES (YES)
```

제거된 큐 관리자와 이름이 동일한 클러스터에 다른 큐 관리자가 있는 경우 제거된 큐 관리자의 QMID 를 지정하십시오.

결과

큐 관리자 LONDON은 더 이상 클러스터에 속하지 않습니다. 그러나 이 큐 관리자는 여전히 독립적 큐 관리자로 작동할 수 있습니다.

다음에 수행할 작업

이러한 변경의 결과는 클러스터의 나머지 멤버에서 다음 명령을 발행하여 확인할 수 있습니다.

```
DISPLAY CLUSQMGR(LONDON)
```

큐 관리자에 대해 자동 정의된 클러스터 송신자 채널이 중지될 때까지 큐 관리자는 계속해서 표시됩니다. 이러한 중지가 발생할 때까지 기다리거나 다음 명령을 발행하여 활성 인스턴스에 대해 계속해서 모니터링할 수 있습니다.

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

변경이 클러스터 전체에 전파되고 이 큐 관리자에 더 이상 메시지가 전달되지 않으면 LONDON에서 CLUSRCVR 채널을 삭제하십시오.

```
DELETE CHANNEL(INVENTORY.LONDON)
```

177 페이지의 『클러스터에 큐 관리자 추가』에 설명된 대로 제거된 큐 관리자는 나중에 해당 클러스터에 다시 추가할 수 있습니다. 제거된 큐 관리자는 최대 90일 동안 클러스터의 나머지 멤버에 대한 지식을 계속해서 캐시합니다. 이 캐시가 만료될 때까지 대기하는 것을 원치 않는 경우 223 페이지의 『큐 관리자를 해당 클러스터 전 상태로 복원』에서 설명된 대로 강제로 제거할 수 있습니다.

큐 관리자를 해당 클러스터 전 상태로 복원

큐 관리자가 클러스터에서 제거될 때 큐 관리자는 나머지 클러스터 멤버에 대한 지식을 보유하고 있습니다. 이 지식은 결국 만료되어 자동으로 삭제됩니다. 그러나 이러한 지식을 즉시 삭제하고 싶은 경우에는 이 주제의 단계를 사용할 수 있습니다.

시작하기 전에

큐 관리자가 클러스터에서 제거되었고 클러스터에서 더 이상 작업을 수행하지 않는 것으로 가정합니다. 예를 들어, 해당 큐는 클러스터에서 더 이상 메시지를 수신하지 않고 애플리케이션은 이러한 큐에 메시지가 도착하는 것을 대기하지 않습니다.

중요사항: 클러스터에서 큐 관리자를 제거하고 REPOS(YES)를 사용하여 새로 고치는 경우 CLUSRCVR의 CLUSTER 속성을 대체하여 다시 추가할 수 없습니다. CLUSRCVR의 CLUSTER 속성을 비어 있지 않도록(즉, 클러스터이름) 변경한 후에는 REPOS(NO)로 새로 고치기 클러스터를 발행해야 합니다. 이때 CLUSRCVR에 있는 내부 순서 번호가 최신 상태가 됩니다. 그런 다음 큐 관리자가 전체 저장소 및 나머지 클러스터 멤버에 자신을 다시 소개하는 데 성공합니다. (CLUSRCVR 채널에 올바른 클러스터 이름이 제공된 후에는 명령의 REPOS(NO) 버전을 실행해야 합니다.)

이 제한사항은 IBM WebSphere MQ Version 7.5에만 적용됩니다.

이 태스크 정보

큐 관리자가 클러스터에서 제거될 때 큐 관리자는 최대 90일 동안 나머지 클러스터 멤버에 대한 지식을 보유하고 있습니다. 이는 시스템적 이점을 가질 수 있으며 특히 큐 관리자가 빠르게 클러스터에 다시 조인하는 경우에 유리합니다. 이 지식은 만료되면 자동으로 삭제됩니다. 그러나 이 정보를 수동으로 삭제하는 것을 선호하는 이유가 있습니다. 예를 들면, 다음과 같습니다.

- 이전에 클러스터 자원을 사용했던 모든 애플리케이션이 이 큐 관리자에서 중지되었음을 확인하려고 합니다. 나머지 클러스터 멤버에 대한 지식이 만료될 때까지 그러한 애플리케이션은 계속해서 전송 큐에 기록합니다. 클러스터 지식이 삭제된 후에 그러한 애플리케이션이 클러스터 자원을 사용하려고 시도하면 시스템이 오류 메시지를 생성합니다.
- 큐 관리자에 대한 상태 정보를 표시할 때 나머지 클러스터 멤버에 대한 만기 정보는 표시하지 않으려고 합니다.

이 태스크는 INVENTORY 클러스터를 예로 사용합니다. LONDON 큐 관리자가 219 페이지의 『클러스터에서 큐 관리자 제거』에 설명된 대로 INVENTORY 클러스터에서 제거되었습니다. 클러스터의 나머지 멤버에 대한 지식을 삭제하려면 LONDON 큐 관리자에서 다음 명령을 실행하십시오.

프로시저

1. 이 큐 관리자에서 클러스터 내 다른 큐 관리자의 모든 메모리를 제거하십시오.

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

2. 모든 클러스터 자원이 없어질 때까지 큐 관리자를 모니터하십시오.

```
DISPLAY CLUSQMgr(*) CLUSTER(INVENTORY)
DISPLAY QCLUSTER(*) CLUSTER(INVENTORY)
DISPLAY TOPIC(*) CLUSTER(INVENTORY)
```

관련 개념

클러스터

146 페이지의 『클러스터링과 분산 큐잉의 비교』

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

148 페이지의 『클러스터의 컴포넌트』

클러스터는 큐 관리자, 클러스터 저장소, 클러스터 채널 및 클러스터 큐로 구성됩니다.

167 페이지의 『IBM WebSphere MQ 클러스터 관리』

IBM WebSphere MQ 클러스터를 작성, 확장 및 유지보수할 수 있습니다.

큐 관리자 유지보수

유지보수를 수행하기 위해 클러스터에서 큐 관리자를 일시중단하고 재개합니다.

이 태스크 정보

클러스터의 일부인 큐 관리자에서 유지보수를 수행해야 되는 경우가 있습니다. 예를 들어, 데이터의 백업을 해당 큐에 넣거나 수정사항을 소프트웨어에 적용해야 합니다. 큐 관리자가 큐를 호스팅하는 경우 활동을 일시중단해야 합니다. 유지보수가 완료되면 그 활동이 재개될 수 있습니다.

프로시저

1. SUSPEND QMGR **runmqsc** 명령을 실행하여 큐 관리자를 일시중단하십시오.

```
SUSPEND QMGR CLUSTER(SALES)
```

SUSPEND **runmqsc** 명령은 SALES 클러스터의 큐 관리자에게 이 큐 관리자가 일시중단되었음을 알립니다.

SUSPEND QMGR 명령의 목적은 가능하면 이 큐 관리자에게 메시지를 송신하는 것을 피하도록 다른 큐 관리자에게 권고하는 것입니다. 이는 큐 관리자가 사용 불가능함을 의미하지는 않습니다. 이 큐 관리자가 핸들링해야 하는 일부 메시지가 여전히 이 큐 관리자에 송신됩니다. 예를 들어, 이 큐 관리자가 클러스터된 큐의 유일한 호스트인 경우입니다.

큐 관리자가 일시중단된 경우 워크로드 관리 루틴이 이 큐 관리자에 대한 메시지 송신을 피합니다. 해당 큐 관리자가 핸들링해야 하는 메시지에는 로컬 큐 관리자가 송신하는 메시지가 포함됩니다.

WebSphere MQ에서는 가능할 때마다 로컬 큐 관리자를 선택하는 대신, 워크로드 밸런스 알고리즘을 사용하여 적합한 목적지를 결정합니다.

- a) SUSPEND QMGR 명령에 FORCE 옵션을 사용하여 큐 관리자의 일시중단을 강제 실행하십시오.

```
SUSPEND QMGR CLUSTER(SALES) MODE(FORCE)
```

MODE(FORCE)는 클러스터 내 다른 큐 관리자에서 모든 인바운드 채널을 강제로 중지합니다. MODE(FORCE)를 지정하지 않으면 기본 MODE(QUIESCE)가 적용됩니다.

2. 필요한 유지보수 태스크를 수행하십시오.
3. RESUME QMGR **runmqsc** 명령을 실행하여 큐 관리자를 재개하십시오.

```
RESUME QMGR CLUSTER(SALES)
```

결과

RESUME **runmqsc** 명령은 큐 관리자가 다시 사용 가능함을 전체 저장소에 알립니다. 전체 저장소 큐 관리자는 이 큐 관리자에 관한 정보 업데이트를 요청한 다른 큐 관리자에게 이 정보를 전파합니다.

클러스터 전송 큐 유지보수

클러스터 전송 큐를 사용 가능한 상태로 유지하기 위해 최선을 다해야 합니다. 클러스터 전송 큐는 클러스터의 성능에 필수적입니다.

시작하기 전에

- 클러스터 전송 큐가 가득 차지 않도록 하십시오.
- 클러스터 전송 큐를 가져오기 사용 안함 또는 넣기 사용 안함으로 우발적으로 설정하지 않도록 ALTER **runmqsc** 명령을 발행하지 않게 주의하십시오.
- 클러스터 전송 큐가 저장된 매체가 가득 차지 않았는지 확인하십시오.

클러스터 큐 관리자 새로 고치기

REFRESH CLUSTER 명령을 사용하여 로컬 저장소에서 자동 정의된 채널 또는 자동 정의된 클러스터 오브젝트를 제거할 수 있습니다. 메시지가 손실되지 않습니다.

시작하기 전에

IBM 지원 센터에서 명령을 사용하도록 요청할 수 있습니다. 이 명령은 매우 주의해서 사용해야 합니다. 예를 들어, **REFRESH CLUSTER** 명령을 사용하는 대형 클러스터의 경우에는 진행 중인 클러스터에 지장을 주고 다시 27 일 간격으로 클러스터 오브젝트가 모든 관심있는 큐 관리자에게 자동으로 상태 업데이트를 전송할 수 있습니다. [278 페이지의 『클러스터링: REFRESH CLUSTER 사용 우수 사례』](#)의 내용을 참조하십시오.

이 태스크 정보

큐 관리자는 클러스터에서 새롭게 시작할 수 있습니다. 정상적인 환경에서는 REFRESH CLUSTER 명령을 사용할 필요가 없습니다.

프로시저

큐 관리자에서 REFRESH CLUSTER **MQSC** 명령을 발행하여 로컬 저장소에서 자동 정의된 클러스터 큐 관리자 및 큐 오브젝트를 제거하십시오.

이 명령은 큐 관리자를 참조하는 오브젝트만 제거하고 로컬 큐 관리자와 관련된 오브젝트는 제거하지 않습니다. 이 명령은 자동 정의된 채널도 제거합니다. 이 명령은 클러스터 전송 큐에 메시지가 없고 전제 저장소 큐 관리자에 연결되지 않은 채널을 제거합니다.

결과

REFRESH CLUSTER 명령을 사용하면 큐 관리자가 그 전체 저장소 콘텐츠에 대하여 콜드 스타트될 수 있습니다. IBM WebSphere MQ는 큐에서 데이터가 손실되지 않도록 보장합니다.

관련 개념

[278 페이지의 『클러스터링: REFRESH CLUSTER 사용 우수 사례』](#)

REFRESH CLUSTER 명령을 사용하여 로컬에 보유된 클러스터에 대한 모든 정보를 제거하고 클러스터의 전체 저장소에서 이 정보를 다시 빌드합니다. 예외 상황을 제외하고는 이 명령을 사용해서는 안됩니다. 사용해야 하는 경우 사용 방법에 대한 특수 고려사항이 있습니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

큐 관리자 복구

REFRESH CLUSTER **runmqsc** 명령을 사용하여 큐 관리자에 대한 클러스터 정보를 최신 정보로 고칩니다. 특정 시점 백업에서 큐 관리자를 복구한 후에 이 프로시저를 따릅니다.

시작하기 전에

특정 시점 백업에서 클러스터 큐 관리자를 복원했습니다.

이 태스크 정보

클러스터에서 큐 관리자를 복구하려면 큐 관리자를 복원한 다음 `REFRESH CLUSTER runmqsc` 명령을 사용하여 클러스터 정보를 최신으로 가져오십시오.

참고: 대형 클러스터의 경우, `REFRESH CLUSTER` 명령을 사용하면 진행 중에 클러스터에 혼란을 줄 수 있으며, 클러스터 오브젝트가 모든 관심 있는 큐 관리자에 자동으로 상태 업데이트를 보낸 이후 27일 간격으로 다시 수행됩니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오.

프로시저

큐 관리자가 참여하는 모든 클러스터의 복원된 큐 관리자에서 `REFRESH CLUSTER` 명령을 발행하십시오.

다음에 수행할 작업

다른 모든 큐 관리자에서 `REFRESH CLUSTER` 명령을 발행하지 않아도 됩니다.

관련 개념

278 페이지의 『클러스터링: REFRESH CLUSTER 사용 우수 사례』

`REFRESH CLUSTER` 명령을 사용하여 로컬에 보유된 클러스터에 대한 모든 정보를 제거하고 클러스터의 전체 저장소에서 이 정보를 다시 빌드합니다. 예외 상황을 제외하고는 이 명령을 사용해서는 안 됩니다. 사용해야 하는 경우 사용 방법에 대한 특수 고려사항이 있습니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

가용성을 위한 클러스터 채널 구성

우수 구성 사례에 따라 간헐적 네트워크 중지가 발생할 경우 클러스터 채널이 계속해서 원활하게 실행될 수 있도록 합니다.

시작하기 전에

클러스터는 채널 정의의 필요성을 줄여주지만 여전히 채널을 유지보수해야 합니다. 분산 큐잉에서 사용된 것과 같은 채널 기술이 클러스터 내 큐 관리자 간 통신에 사용됩니다. 클러스터 채널에 대해 이해하려면 다음과 같은 사항을 익혀야 합니다.

- 채널이 작동하는 방식
- 채널의 상태를 찾는 방법
- 채널 엑시트를 사용하는 방법

이 태스크 정보

다음과 같은 점을 특별히 고려하려고 합니다.

프로시저

클러스터 채널을 구성할 때 다음과 같은 점을 고려하십시오.

- 클러스터 송신자 채널 및 클러스터 수신자 채널에서 네트워크에 다량의 하트비트로 부담을 주지 않거나 활성 플로우를 유지하는 `HBINT` 또는 `KAINT`의 값을 선택하십시오. 때때로 네트워크가 느려지고 이 길이의 지연이 생기는 경우 간격이 약 10초 미만이면 `False` 실패가 발생합니다.
- 실패된 채널에서 인다우트(in-doubt)이기 때문에 고립된 메시지를 발생시키는 범위를 줄이려면 `BATCHHB` 값을 설정하십시오. 배치를 채우는 데 더 긴 시간이 제공되는 경우 실패한 채널의 인다우트(in-doubt) 배치가 발생할 가능성이 높아집니다. 채널의 메시지 트래픽이 메시지의 버스트 간 긴 시간 동안 산발적이면 실패한 배치가 발생할 가능성이 높습니다.
- 채널의 클라이언트 송신자 측이 실패한 후 하트비트 또는 활성 유지가 실패를 감지하기 전에 재시작을 시도할 경우 문제점이 발생합니다. 채널의 클러스터 수신자 측이 활성으로 유지되는 경우 채널 송신자 재시작이 거부

됩니다. 실패를 피하려면 클러스터 송신자 채널이 재시작을 시도하면 클러스터 수신자 채널이 종료되고 재시작되도록 설정하십시오.

z/OS 이외의 플랫폼의 경우

qm.ini 파일 또는 Windows NT 레지스트리에서 AdoptNewMCA, AdoptNewMCATimeout 및 AdoptNewMCA검사 속성을 사용하여 활성 상태인 채널의 클러스터 수신자 끝 문제점을 제어하십시오.

메시지를 클러스터로(부터) 라우팅

큐 알리어스, 큐 관리자 알리어스 및 리모트 큐 정의를 사용하여 클러스터를 외부 큐 관리자 및 다른 클러스터에 연결합니다.

메시지를 클러스터로(부터) 라우팅하는 것에 대한 자세한 내용은 다음 하위 주제를 참조하십시오.

관련 개념

[클러스터](#)

[클러스터 작동 방법](#)

[146 페이지의 『클러스터링과 분산 큐잉의 비교』](#)

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

[148 페이지의 『클러스터의 컴포넌트』](#)

클러스터는 큐 관리자, 클러스터 저장소, 클러스터 채널 및 클러스터 큐로 구성됩니다.

[167 페이지의 『IBM WebSphere MQ 클러스터 관리』](#)

IBM WebSphere MQ 클러스터를 작성, 확장 및 유지보수할 수 있습니다.

[237 페이지의 『큐 관리자 알리어스 및 클러스터』](#)

큐 관리자 알리어스를 사용하여 메시지를 클러스터 내외로 송신할 때 큐 관리자의 이름을 숨기고 클러스터에 송신된 메시지를 워크로드 밸런싱합니다.

[239 페이지의 『큐 알리어스 및 클러스터』](#)

큐 알리어스를 사용하여 클러스터 큐의 이름을 숨기거나, 큐를 클러스터링하거나, 다른 속성을 채택하거나 다른 액세스 제어를 채택할 수 있습니다.

[239 페이지의 『응답 대상 큐 알리어스 및 클러스터』](#)

응답 대상 큐 알리어스 정의는 응답 정보에 대한 대체 이름을 지정하는 데 사용됩니다. 응답 대상 큐 알리어스 정의는 분산 큐잉 환경에서와 같이 클러스터에서 사용할 수 있습니다.

관련 태스크

[144 페이지의 『큐 관리자 클러스터 구성』](#)

클러스터가 작업하는 방법, 클러스터 구성을 설계하는 방법을 찾고 단순한 클러스터를 설정하는 방법의 예를 알려면 이 토픽의 링크를 사용하십시오.

[168 페이지의 『새 클러스터 설정』](#)

다음 지시사항에 따라 클러스터 예를 설정합니다. 개별 지시사항에서는 TCP/IP, LU 6.2 및 단일 전송 큐 또는 다중 전송 큐에서의 클러스터 설정을 설명합니다. 한 큐 관리자에서 다른 큐 관리자로 메시지를 송신하여 클러스터가 작동하는지 테스트합니다.

클러스터에 대한 요청/응답 구성

클러스터 외부에 있는 큐 관리자로부터 요청/응답 메시지를 구성합니다. 게이트웨이 큐 관리자를 클러스터에 대한 통신 경로로 사용하여 클러스터의 내부 세부사항을 숨깁니다.

시작하기 전에

228 페이지의 그림 43 은 DEMO라는 클러스터 외부에 있는 QM3 큐 관리자를 표시합니다. QM3는 클러스터를 지원하지 않는 WebSphere MQ 제품의 큐 관리자가 될 수 있습니다. QM3 는 다음과 같이 정의된 Q3라는 큐를 호스트합니다.

```
DEFINE QLOCAL(Q3)
```

클러스터 내부에는 QM1 및 QM2라고 하는 두 큐 관리자가 있습니다. QM2는 Q2라고 하는 클러스터 큐를 호스팅하며 다음과 같이 정의됩니다.

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO)
```

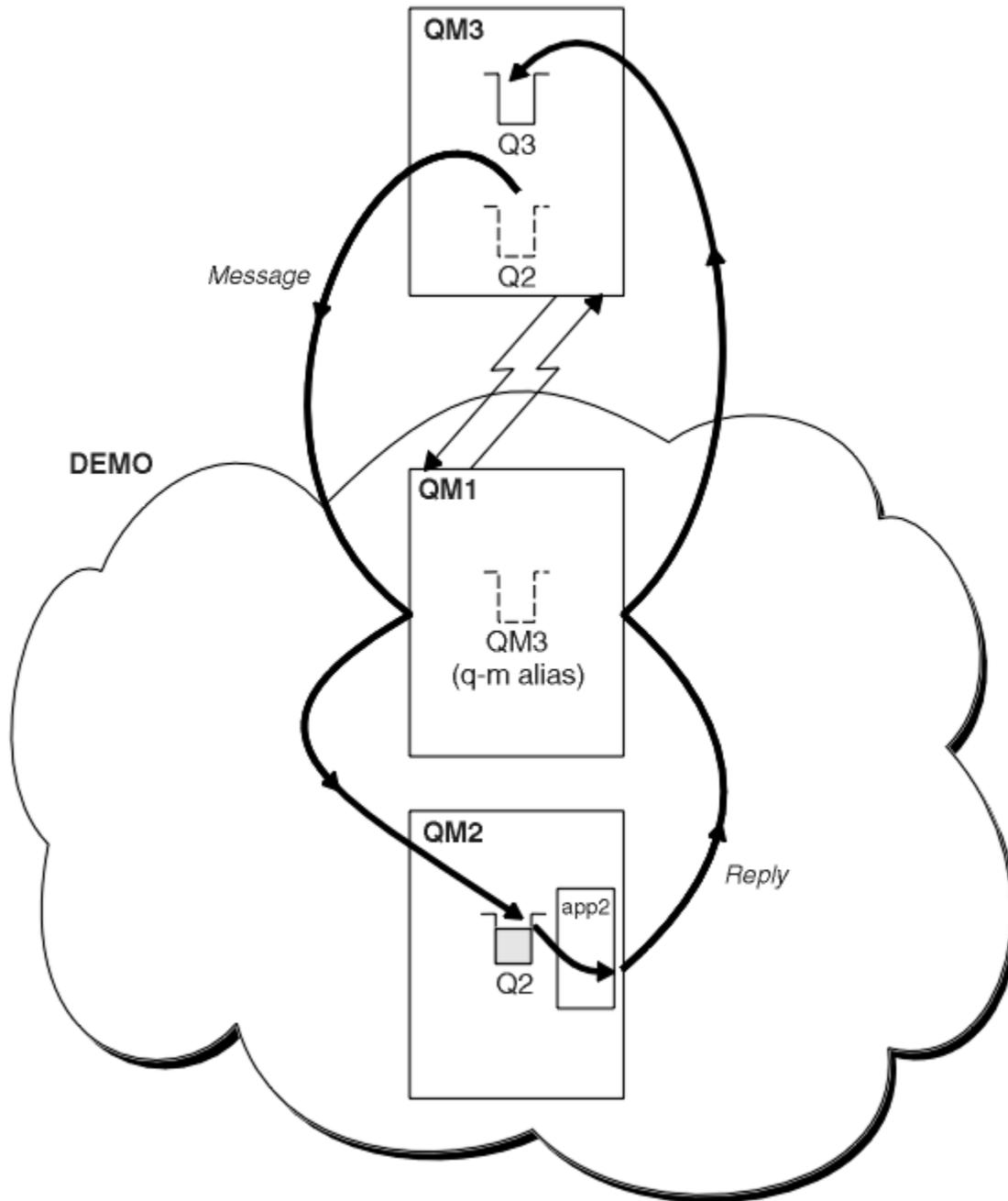


그림 43. 클러스터 외부 큐 관리자에서 넣기

이 태스크 정보

프로시저의 권고사항에 따라 요청 및 응답 메시지에 대한 경로를 설정하십시오.

프로시저

1. 요청 메시지를 클러스터에 송신하십시오.

클러스터 외부에 있는 큐 관리자가 클러스터 내부에 있는 QM2의 큐 Q2에 메시지를 넣는 방법을 고려하십시오. 클러스터 외부에 있는 큐 관리자에는 이 큐 관리자가 메시지를 넣는 클러스터 내 각 큐에 대한 QREMOTE 정의가 있어야 합니다.

a) QM3에서 Q2에 대한 리모트 큐를 정의하십시오.

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(QM2) XMITQ(QM1)
```

QM3는 클러스터의 일부가 아니므로 분산 큐잉 기술을 사용하여 통신해야 합니다. 따라서 송신자 채널 및 QM1에 대한 전송 큐도 있어야 합니다. QM1에는 상응하는 수신자 채널이 필요합니다. 채널 및 전송 큐는 [228 페이지의 그림 43](#)에 명시적으로 표시되지 않습니다.

이 예제에서는 QM3의 애플리케이션이 MQPUT 호출을 실행하여 Q2에 메시지를 넣습니다. QREMOTE 정의는 QM1 전송 큐에서 메시지를 가져오는 송신자 채널을 사용하여 QM2에서 Q2로 메시지를 라우트합니다.

2. 클러스터에서 응답 메시지를 수신하십시오.

큐 관리자 알리아스를 사용하여 클러스터 외부 큐 관리자에 대한 응답의 리턴 경로를 작성하십시오. 게이트웨이 QM1은 클러스터 외부에 있는 큐 관리자의 큐 관리자 알리아스를 알립니다 (QM3). QM3에 대한 큐 관리자 알리아스 정의에 클러스터 속성을 추가하여 클러스터 내부의 큐 관리자에게 QM3를 알립니다. 큐 관리자 알리아스 정의는 리모트 큐 정의와 같지만 공백 RNAME이 있습니다.

a) QM1의 QM3에 대한 큐 관리자 알리아스를 정의하십시오.

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

응답을 QM1에서 QM3로 전달하는 데 사용되는 전송 큐의 이름 선택을 고려해야 합니다. XMITQ 속성의 생략에 의해 QREMOTE 정의에서 암시적이며 전송 큐의 이름은 QM3입니다. 그러나 QM3은 큐 관리자 알리아스를 사용하여 나머지 클러스터에 광고할 것으로 예상되는 동일한 이름입니다. WebSphere MQ에서는 사용자가 전송 큐 및 큐 관리자 알리아스 모두에 동일한 이름을 제공할 수 없습니다. 한 가지 솔루션은 큐 관리자 알리아스에 다른 이름을 사용하여 QM3에 메시지를 전달하는 전송 큐를 작성하는 것입니다.

b) QREMOTE 정의에 전송 큐 이름을 제공하십시오.

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO) XMITQ(QM3.XMIT)
```

새 큐 관리자 알리아스는 QM3.XMIT라는 새 전송 큐를 QM3 큐 관리자 알리아스와 연결합니다. 이는 단순하고 정확한 솔루션이지만 완전히 만족스러운 솔루션은 아닙니다. 이 솔루션은 대상 큐 관리자와 같은 이름이 지정되는 전송 큐에 대한 이름 지정 규칙을 위반했습니다. 전송 큐 이름 지정 규칙을 준수하는 대체 솔루션이 있습니까?

QM3에서 송신되는 요청 메시지에 요청자가 응답 대상 큐 관리자 이름으로 QM3를 전달하도록 기본값 지정되었기 때문에 문제점이 발생합니다. QM2의 서버는 QM3 응답 대상 큐 관리자 이름을 사용하여 응답에서 QM3를 처리합니다. The solution required QM1 to advertise QM3 as the queue manager alias to return reply messages to and prevented QM1 from using QM3 as the name of the transmission queue.

QM3를 응답 대상 큐 관리자 이름으로 제공하는 것으로 기본값 설정하는 대신 QM3의 애플리케이션은 응답 대상 큐 관리자 알리아스를 응답 메시지에 대한 QM1에 전달해야 합니다. 게이트웨이 큐 관리자 QM1은 전송 큐의 이름과 충돌하지 않도록 QM3 자체가 아닌 QM3에 대한 응답에 대해 큐 관리자 알리아스를 알립니다.

c) QM1의 QM3에 대한 큐 관리자 알리아스를 정의하십시오.

```
DEFINE QREMOTE(QM3.ALIAS) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

구성 명령에 대한 두 가지 변경이 필요합니다.

i) 이제 QM1의 QREMOTE은 (는) 큐 관리자 알리아스 QM3.ALIAS를 클러스터의 나머지 부분에 전달하여 실제 큐 관리자 QM3의 이름에 연결합니다. QM3은 다시 QM3로 응답 큐를 전송하는 전송 큐의 이름입니다.

- ii) 클라이언트 애플리케이션은 요청 메시지를 구성할 때 응답 대상 큐 관리자의 이름으로 QM3.ALIAS를 제공해야 합니다. 다음 두 가지 방식 중 하나로 QM3.ALIAS를 클라이언트 애플리케이션에 제공할 수 있습니다.
 - MQMD에서 MQPUT 로 구성된 응답 대상 큐 관리자 이름 필드에 QM3.ALIAS 를 입력하십시오. 응답에 동적 큐를 사용하는 경우 이 방법으로 이를 수행해야 합니다.
 - 응답 대상 큐 이름을 제공할 때 응답 대상 큐가 아닌 응답 대상 큐 알리어스 Q3.ALIAS를 사용하십시오.

```
DEFINE QREMOTE(Q3.ALIAS) RNAME(Q3) RQMNAME(QM3.ALIAS)
```

다음에 수행할 작업

참고: AMQSREQ0와 함께 응답 대상 큐 알리어스를 사용하는 것을 보여줄 수 없습니다. 이는 매개변수 3 또는 기본 SYSTEM.SAMPLE.REPLY 모델 큐에서 제공된 큐 이름을 사용하여 응답 대상 큐를 엽니다. MQPUT에 대한 응답 대상 큐 관리자 알리어스의 이름을 지정하려면 응답 대상 큐 알리어스를 포함하는 다른 매개변수를 제공하여 샘플을 수정해야 합니다.

관련 태스크

230 페이지의 『클러스터 대상 큐 관리자의 이름 숨기기』

큐 관리자의 이름을 지정하지 않고 클러스터 내 임의의 큐 관리자에서 정의되는 클러스터 큐에 메시지를 라우트합니다.

클러스터 대상 큐 관리자의 이름 숨기기

큐 관리자의 이름을 지정하지 않고 클러스터 내 임의의 큐 관리자에서 정의되는 클러스터 큐에 메시지를 라우트합니다.

시작하기 전에

- 클러스터 내부에 있는 큐 관리자의 이름을 클러스터 외부에 있는 큐 관리자에 노출하지 않도록 하십시오.
 - 클러스터 내부에서 큐를 호스팅하는 큐 관리자에 대한 참조를 해석하면 워크로드 밸런싱을 수행하기 위한 유연성이 제거됩니다.
 - 또한 클러스터에서 큐를 호스팅하는 큐 관리자를 변경하는 것을 어렵게 합니다.
 - 대안은 RQMNAME을 클러스터 관리자가 제공한 큐 관리자 알리어스로 바꾸는 것입니다.
 - 230 페이지의 『클러스터 대상 큐 관리자의 이름 숨기기』에서는 클러스터 외부에 있는 큐 관리자를 클러스터 내부에 있는 큐 관리자의 관리로부터 결합 해제하기 위해 큐 관리자 알리어스를 사용하는 것에 대해 설명합니다.
- 그러나 전송 큐의 이름을 지정하기 위해 제안된 방식은 전송 큐에 대상 큐 관리자의 이름을 지정하는 것입니다. 전송 큐의 이름은 클러스터 내 큐 관리자의 이름을 노출시킵니다. 따를 규칙을 선택해야 합니다. 큐 관리자 이름 또는 클러스터 이름을 사용하여 전송 큐의 이름을 지정하도록 선택할 수 있습니다.

게이트웨이 큐 관리자 이름을 사용하여 전송 큐의 이름 지정

게이트웨이 큐 관리자의 이름을 클러스터 외부에 있는 큐 관리자에게 노출하는 것은 클러스터 큐 관리자 이름을 숨기는 규칙에 대한 합리적 예외입니다.

클러스터의 이름을 사용하여 전송 큐의 이름 지정

대상 큐 관리자의 이름으로 전송 큐의 이름을 지정하는 규칙을 준수하지 않는 경우 클러스터 이름을 사용하십시오.

이 태스크 정보

227 페이지의 『클러스터에 대한 요청/응답 구성』 태스크를 수정하여 클러스터 내부에 있는 대상 큐 관리자의 이름을 숨기십시오.

프로시저

이 예제에서 231 페이지의 그림 44 를 참조하고 게이트웨이 큐 관리자 QM1 에 DEMO이라는 큐 관리자 알리언스를 정의하십시오.

```
DEFINE QREMOTE(DEMO) RNAME(' ') RQMNAME(' ')
```

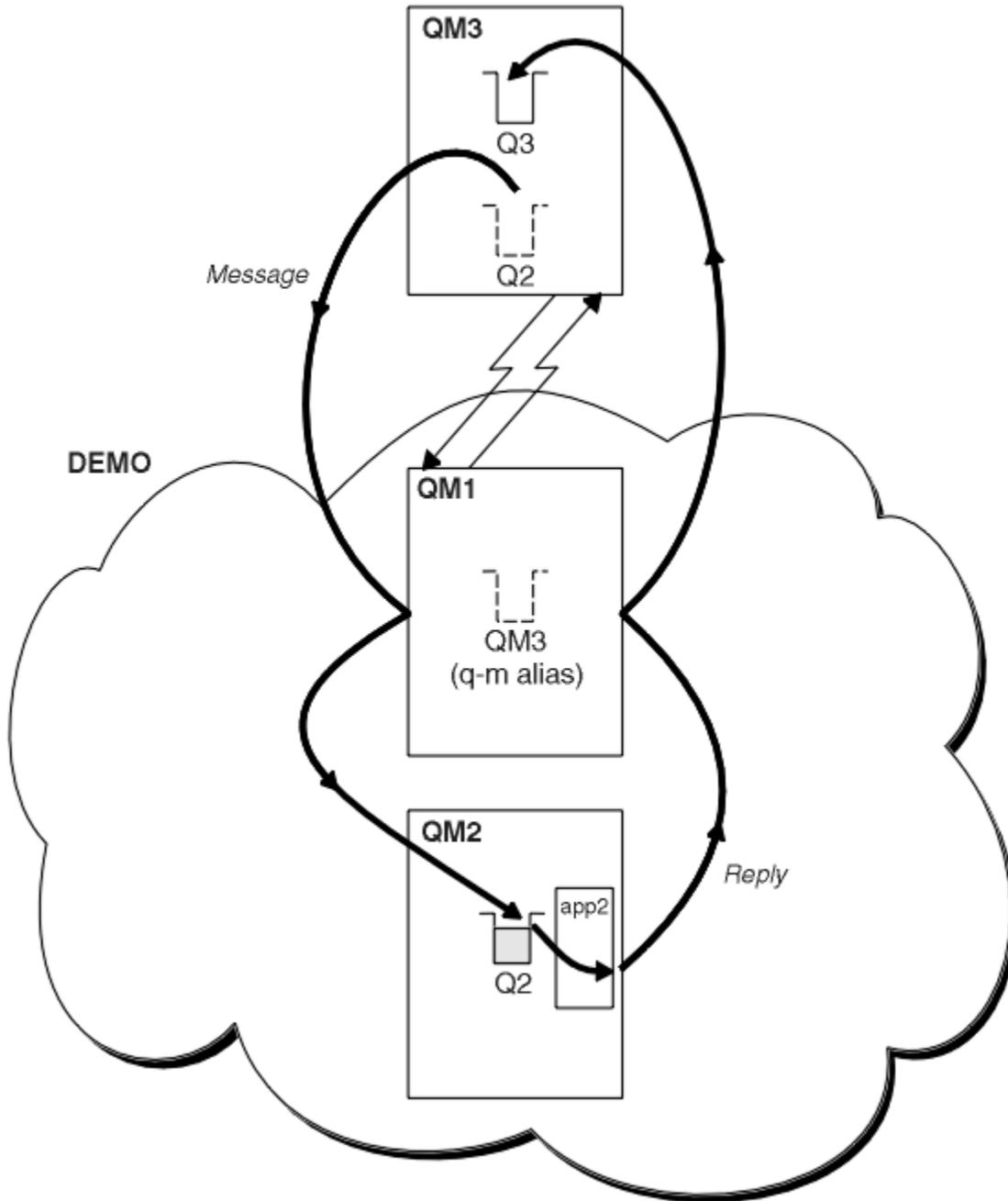


그림 44. 클러스터 외부 큐 관리자에서 넣기

QM1 의 QREMOTE 정의는 게이트웨이 큐 관리자에 알려진 큐 관리자 알리언스 DEMO 를 작성합니다. QM3, 클러스터 외부의 큐 관리자는 실제 큐 관리자 이름을 사용하지 않고 큐 관리자 알리언스 DEMO 를 사용하여 DEMO의 클러스터 큐에 메시지를 송신할 수 있습니다.

클러스터에 연결하는 전송 큐의 이름을 지정하기 위해 클러스터 이름을 사용하는 규칙을 채택하는 경우, Q2 에 대한 리모트 큐 정의는 다음과 같이 됩니다.

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(DEMO)
```

결과

DEMO의 Q2를 목적지로 하는 메시지는 DEMO 전송 큐에 배치됩니다. 전송 큐에서 이러한 메시지는 송신자 채널에 의해 게이트웨이 큐 관리자 QM1으로 전송됩니다. 게이트웨이 큐 관리자는 클러스터 큐 Q2를 호스팅하는 클러스터에 있는 임의의 큐 관리자에게 메시지를 라우트합니다.

클러스터에서 요청/응답 구성

클러스터에서 클러스터 외부에 있는 큐 관리자로 요청/응답 메시지 경로를 구성합니다. 게이트웨이 큐 관리자를 사용하여 클러스터 내부 큐 관리자가 클러스터 외부와 통신하는 방법에 대한 세부사항을 숨깁니다.

시작하기 전에

[233 페이지의 그림 45](#)에서는 클러스터 DEMO 내부에 있는 큐 관리자 QM2를 보여줍니다. 이 큐 관리자는 클러스터 외부에 있는 큐 관리자에서 호스팅되는 큐 Q3에 요청을 송신합니다. 응답이 클러스터 내부에 있는 QM2의 Q2에 리턴됩니다.

클러스터 외부에 있는 큐 관리자와 통신하기 위해 클러스터 내부에서 하나 이상의 큐 관리자가 게이트웨이 역할을 합니다. 게이트웨이 큐 관리자에는 클러스터 외부에 있는 큐 관리자에 대한 통신 경로가 있습니다. 이 예에서 QM1은 게이트웨이입니다.

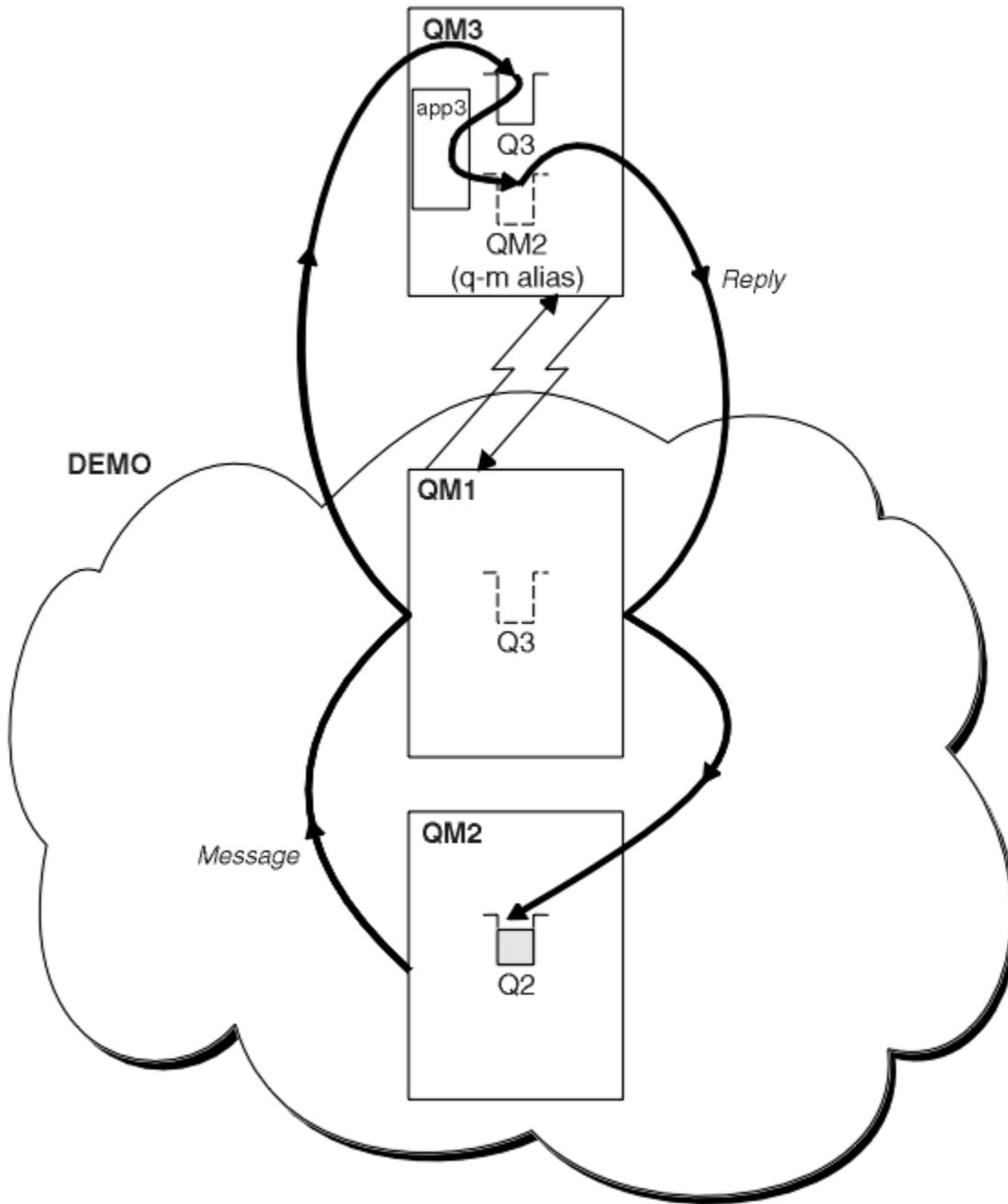


그림 45. 클러스터 외부에 있는 큐 관리자에 넣기

이 태스크 정보

지시사항에 따라 요청 및 응답 메시지에 대한 경로를 설정하십시오.

프로시저

1. 클러스터에서 요청 메시지를 송신하십시오.

클러스터 내부에 있는 큐 관리자 QM2가 클러스터 외부에 있는 QM3의 큐 Q3에 메시지를 넣는 방법을 고려하십시오.

- a) QM1에서 리모트 큐 Q3를 클러스터에 알리는 QREMOTE 정의를 작성하십시오.

```
DEFINE QREMOTE(Q3) RNAME(Q3) RQMNAME(QM3) CLUSTER(DEMO)
```

또한 클러스터 외부에 있는 큐 관리자에 대한 송신자 채널 및 전송 큐가 있습니다. QM3에는 해당 수신자 채널이 있습니다. 채널은 [233 페이지의 그림 45](#)에 표시되지 않습니다.

QM2의 애플리케이션이 대상 큐 및 응답이 송신될 큐를 지정하는 MQPUT 호출을 발행합니다. 대상 큐는 Q3이고 응답 대상 큐는 Q2입니다.

이 메시지는 해당 리모트 큐 정의를 사용하여 큐 이름을 QM3의 Q3로 해석하는 QM1으로 송신됩니다.

2. 클러스터 외부에 있는 큐 관리자에서 응답 메시지를 수신하십시오.

클러스터 외부에 있는 큐 관리자에는 메시지를 송신하는 클러스터에 있는 각 큐 관리자에 대한 큐 관리자 알리어스가 있어야 합니다. 큐 관리자 알리어스는 게이트웨이 큐 관리자에 대한 전송 큐의 이름도 지정해야 합니다. 이 예에서 QM3에는 QM2에 대한 큐 관리자 알리어스 정의가 필요합니다.

- a) QM3에서 큐 관리자 알리어스 QM2를 작성하십시오.

```
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) XMITQ(QM1)
```

QM3에는 QM1에 대한 송신자 채널 및 전송 큐도 필요하고 QM1에는 해당 수신자 채널이 필요합니다.

그러면 QM3의 애플리케이션 **app3**는 MQPUT 호출을 발행하고 큐 이름 Q2 및 큐 관리자 이름 QM2를 지정하여 QM2에 응답을 송신할 수 있습니다.

다음에 수행할 작업

클러스터 외부에 둘 이상의 라우트를 정의할 수 있습니다.

클러스터 외부에서 워크로드 밸런싱 구성

클러스터 외부에 있는 큐 관리자에서 클러스터 큐 사본에 대한 메시지 경로를 구성합니다. 결과는 클러스터 외부 요청을 클러스터 큐의 각 인스턴스로 워크로드 밸런싱하는 것입니다.

시작하기 전에

[227 페이지의 『클러스터에 대한 요청/응답 구성』](#)의 [228 페이지의 그림 43](#)에 표시된 대로 예를 구성하십시오.

이 태스크 정보

이 시나리오에서는 클러스터 외부의 큐 관리자 ([235 페이지의 그림 46](#)의 QM3)가 큐 Q2에 요청을 전송합니다. Q2는 워크로드 밸런싱을 사용하도록 클러스터 DEMO 내의 두 개의 큐 관리자에서 호스트됩니다. 큐 Q2는 큐 관리자 QM2 및 QM4에서 정의되지만 게이트웨이 큐 관리자 QM1에서는 정의되지 않습니다. 클러스터 외부의 큐 관리자 QM3의 요청은 Q2의 인스턴스로 송신됩니다.

QM3는 클러스터의 일부가 아니고 분산 큐잉 기술을 사용하여 통신합니다. 이 큐 관리자에는 QM1에 대한 송신자 채널 및 전송 큐가 있어야 합니다. QM1에는 해당 수신자 채널이 필요합니다. 채널 및 전송 큐는 [235 페이지의 그림 46](#)에 명시적으로 표시되지 않습니다.

프로시저는 [227 페이지의 『클러스터에 대한 요청/응답 구성』](#)의 [228 페이지의 그림 43](#)에 있는 예를 확장합니다.

프로시저

1. QM2 및 QM4 각각에서 Q2라고 하는 로컬 큐를 정의하십시오.

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

2. QM3에서 Q2에 대한 QREMOTE 정의를 작성하십시오.

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(Q3) XMITQ(QM1)
```

QM3가 메시지를 넣는 클러스터의 각 큐에 대해 QREMOTE 정의를 작성하십시오.

3. QM3의 큐 관리자 알리어스 Q3를 작성하십시오.

```
DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ') CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

Q3는 실제 큐 관리자 이름이 아닙니다. 이것은 큐 관리자 알리어스 이름 Q3와 공백 ' '가 같음을 표시하는 클러스터 내 큐 관리자 알리어스 정의의 이름입니다.

4. 게이트웨이 큐 관리자인 QM1에는 특별한 정의가 없습니다.

결과

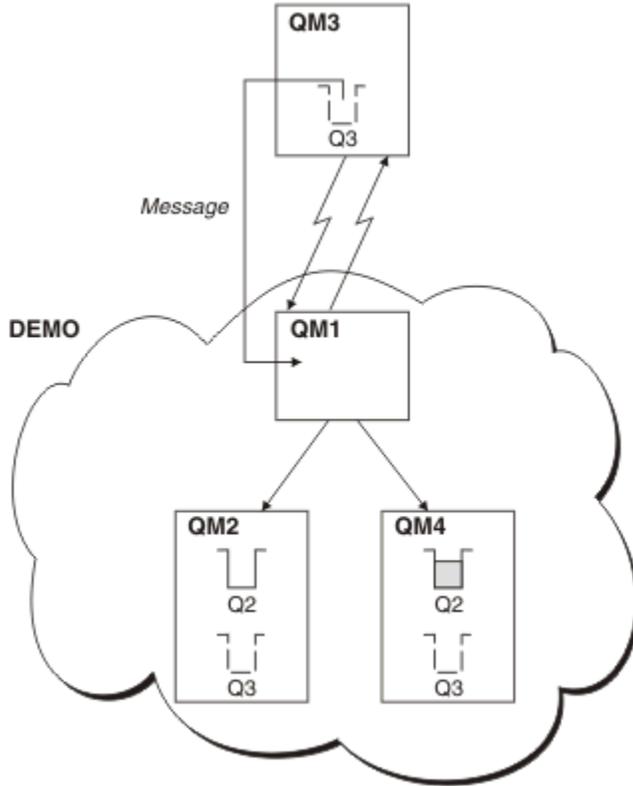


그림 46. 클러스터 외부 큐 관리자에서 넣기

QM3의 애플리케이션이 Q2에 메시지를 넣도록 MQPUT 호출을 발행하는 경우, QREMOTE 정의를 사용하면 게이트웨이 큐 관리자 QM1을 통해 메시지를 라우트할 수 있습니다. QM1은 워크로드 밸런싱을 사용하여 Q3에 대한 클러스터 큐 관리자 알리어스를 가진 두 개의 큐 관리자 QM2와 QM4의 Q2 큐 사이에 Q2로 전달되는 메시지를 분배할 수 있습니다.

클러스터 간 메시지 경로 구성

게이트웨이 큐 관리자를 사용하여 클러스터를 서로 연결합니다. 게이트웨이 큐 관리자에서 클러스터 큐 또는 클러스터 큐 관리자 알리어스를 정의하여 큐 또는 큐 관리자가 모든 클러스터에 표시되도록 합니다.

이 태스크 정보

하나의 대형 클러스터에 모든 큐 관리자를 그룹화하는 대신 여러 작은 클러스터로 구성할 수 있습니다. 각 클러스터에는 브릿지 역할을 함에 있어 하나 이상의 큐 관리자가 있습니다. 이것의 장점은 클러스터 전체에서 큐 및 큐 관리자 이름이 표시되는 것을 제한할 수 있다는 점입니다. 163 페이지의 『중첩 클러스터』의 내용을 참조하십시오. 이름 충돌을 피하거나 로컬 이름 지정 규칙을 준수하려면 알리어스를 사용하여 큐 및 큐의 이름을 변경하십시오.

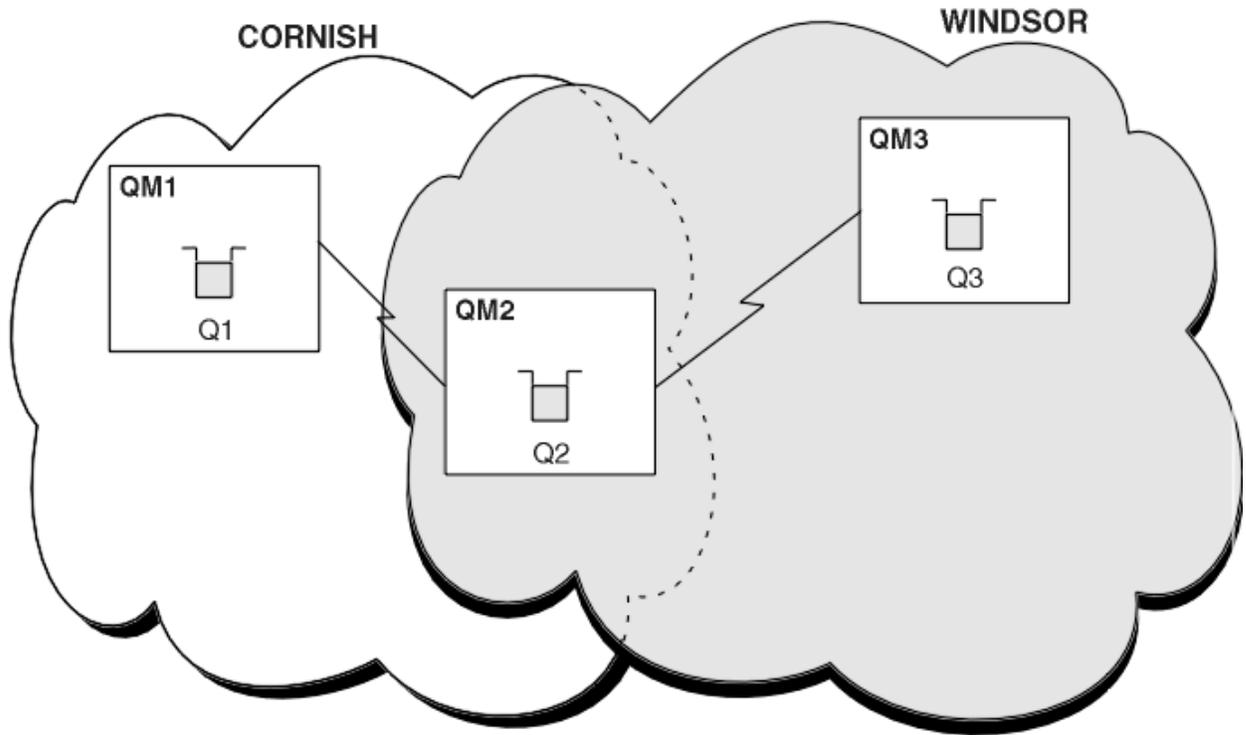


그림 47. 클러스터 전체에서 브릿징

236 페이지의 그림 47에서는 사이에 브릿지가 있는 두 개의 클러스터를 보여줍니다. 둘 이상의 브릿지가 있을 수 있습니다.

다음 프로시저를 사용하여 클러스터를 구성하십시오.

프로시저

1. QM1에서 클러스터 큐 Q1을 정의하십시오.

```
DEFINE QLOCAL(Q1) CLUSTER(CORNISH)
```

2. QM3에서 클러스터 큐 Q3을 정의하십시오.

```
DEFINE QLOCAL(Q3) CLUSTER(WINDSOR)
```

3. QM2에서 두 클러스터의 이름을 모두 포함하는 CORNISHWINDSOR이라고 하는 이름 목록을 작성하십시오.

```
DEFINE NAMELIST(CORNISHWINDSOR) DESCR('CornishWindsor namelist')
  NAMES(CORNISH, WINDSOR)
```

4. QM2에서 클러스터 큐 Q2를 정의하십시오.

```
DEFINE QLOCAL(Q2) CLUSNL(CORNISHWINDSOR)
```

다음에 수행할 작업

QM2는 두 클러스터의 멤버이고 두 클러스터 간 브릿지입니다. 브릿지 전체에 표시되게 하려는 각 큐의 경우 브릿지에 대한 QALIAS 정의가 필요합니다. 예를 들어, 236 페이지의 그림 47에서 QM2에는 다음이 필요합니다.

```
DEFINE QALIAS(MYQ3) TARGET(Q3) CLUSTER(CORNISH) DEFBIND(NOTFIXED)
```

큐 알리아스를 사용하여 CORNISH에서 큐 관리자에 연결된 애플리케이션(예: QM1)은 메시지를 Q3에 넣을 수 있습니다. 이 애플리케이션은 Q3를 MYQ3로 참조합니다. 이 메시지는 QM3의 Q3로 라우트됩니다.

큐를 열 때 DEFBIND를 NOTFIXED 또는 QDEF로 설정해야 합니다. DEFBIND를 기본값 OPEN으로 두면 큐 관리자는 호스팅하는 브릿지 큐 관리자로 알리어스 정의를 해석합니다. 이 브릿지는 메시지를 전달하지 않습니다.

표시되게 하려는 각 큐 관리자에 대해 큐 관리자 알리어스 정의가 필요합니다. 예를 들어, QM2에서 다음이 필요합니다.

```
DEFINE QREMOTE(QM1) RNAME(' ') RQMNAME(QM1) CLUSTER(WINDSOR)
```

WINDSOR에서 큐 관리자(예: QM3)에 연결되는 애플리케이션은 MQOPEN 호출에서 QM1을 명시적으로 이름 지정하여 QM1에 있는 임의의 큐에 메시지를 넣을 수 있습니다.

큐 관리자 알리어스 및 클러스터

큐 관리자 알리어스를 사용하여 메시지를 클러스터 내외로 송신할 때 큐 관리자의 이름을 숨기고 클러스터에 송신된 메시지를 워크로드 밸런싱합니다.

공백 RNAME이 포함된 리모트 큐 정의를 사용하여 작성되는 큐 관리자 알리어스에는 다음과 같은 다섯 가지 용도가 있습니다.

메시지를 송신할 때 큐 관리자 이름 재맵핑

큐 관리자 알리어스는 MQOPEN 호출에서 지정된 큐 관리자 이름을 다른 큐 관리자로 재맵핑하는 데 사용할 수 있습니다. 이는 클러스터 큐 관리자일 수 있습니다. 예를 들어, 큐 관리자에는 큐 관리자 알리어스 정의가 있을 수 있습니다.

```
DEFINE QREMOTE(YORK) RNAME(' ') RQMNAME(CLUSQM)
```

YORK는 CLUSQM이라고 하는 큐 관리자의 알리어스로 사용될 수 있습니다. 이 정의를 작성한 큐 관리자의 애플리케이션이 메시지를 큐 관리자 YORK에 넣으면 로컬 큐 관리자가 이 이름을 CLUSQM로 해석합니다. 로컬 큐 관리자를 CLUSQM이라고 하지 않을 경우 메시지를 CLUSQM으로 이동할 클러스터 전송 큐에 넣습니다. 또한 전송 헤더를 YORK가 아닌 CLUSQM을 표시하도록 변경합니다.

참고: 이 정의는 이 정의를 작성하는 큐 관리자에서만 적용됩니다. 알리어스를 전체 클러스터에 알려려면 CLUSTER 속성을 리모트 큐 정의에 추가해야 합니다. 그 다음 다른 큐 관리자에서 YORK로 보낸 메시지가 CLUSQM으로 송신됩니다.

메시지 송신 시 전송 큐 대체 또는 지정

알리어스는 클러스터를 비클러스터 시스템에 조인하는 데 사용할 수 있습니다. 예를 들어, 클러스터 ITALY의 큐 관리자는 클러스터 외부에 있는 큐 관리자 PALERMO와 통신할 수 있습니다. 통신하려면 클러스터 내 큐 관리자 중 하나가 게이트웨이 역할을 해야 합니다. 게이트웨이 큐 관리자에서 다음 명령을 실행하십시오.

```
DEFINE QREMOTE(ROME) RNAME(' ') RQMNAME(PALERMO) XMITQ(X) CLUSTER(ITALY)
```

이 명령은 큐 관리자 알리어스 정의입니다. It defines and advertises ROME as a queue manager over which messages from any queue manager in the cluster ITALY can multi-hop to reach their destination at PALERMO. ROME으로 설정된 큐 관리자 이름으로 열리는 큐에 넣은 메시지는 큐 관리자 알리어스 정의가 있는 게이트웨이 큐 관리자로 송신됩니다. 그 다음 메시지를 전송 큐 X에 넣고 큐 관리자 PALERMO에 대한 클러스터가 아닌 채널을 통해 이동시킵니다.

이 예에서 이름 ROME을 선택하는 것은 중요하지 않습니다. QREMOTE 및 RQMNAME의 값은 둘 다 동일할 수 있습니다.

메시지를 수신할 때 대상 판별

큐 관리자는 메시지를 수신하면 전송 헤더에서 목적지 큐와 큐 관리자의 이름을 추출합니다. 큐 관리자는 전송 헤더에서 큐 관리자와 같은 이름을 가지는 큐 관리자 알리어스 정의를 찾습니다. 이러한 정의를 찾으면 전송 헤더의 큐 관리자 이름에 대한 큐 관리자 알리어스 정의에서 RQMNAME을 대체합니다.

이 방식으로 큐 관리자 알리어스를 사용하는 두 가지 이유가 있습니다.

- 메시지를 다른 큐 관리자로 경로 지정
- 큐 관리자 이름을 로컬 큐 관리자와 같도록 변경

게이트웨이 큐 관리자에서 큐 관리자 알리언스를 사용하여 서로 다른 클러스터에 있는 큐 관리자 사이에 메시지를 라우트합니다.

애플리케이션은 큐 관리자 알리언스를 사용하여 다른 클러스터에 있는 큐에 메시지를 송신할 수 있습니다. 큐는 꼭 클러스터 큐가 아니어도 됩니다. 큐는 한 개의 클러스터에서 정의됩니다. 애플리케이션이 다른 클러스터에 있는 큐 관리자에 연결됩니다. 게이트웨이 큐 관리자가 두 클러스터를 연결합니다. 큐가 클러스터된 것으로 정의되지 않은 경우 올바른 라우팅이 발생하도록 애플리케이션은 큐 이름 및 클러스터된 큐 관리자 알리언스 이름을 사용하여 큐를 열어야 합니다. 구성 예는 197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』의 내용을 참조하십시오. 여기에서 그림 1에 설명된 응답 메시지 플로우가 선택됩니다.

다이아그램은 응답 메시지가 RQ(이)라는 임시 동적 큐로 되돌아가는 경로를 보여줍니다. QM3에 연결된 서버 애플리케이션은 큐 관리자 이름 QM2을(를) 사용하여 응답 큐를 엽니다. 큐 관리자 이름 QM2은(는) QM1에서 클러스터된 큐 관리자 알리언스로 정의됩니다. QM3은(는) 응답 메시지를 QM1(으)로 라우트합니다. QM1은(는) 메시지를 QM2(으)로 라우트합니다.

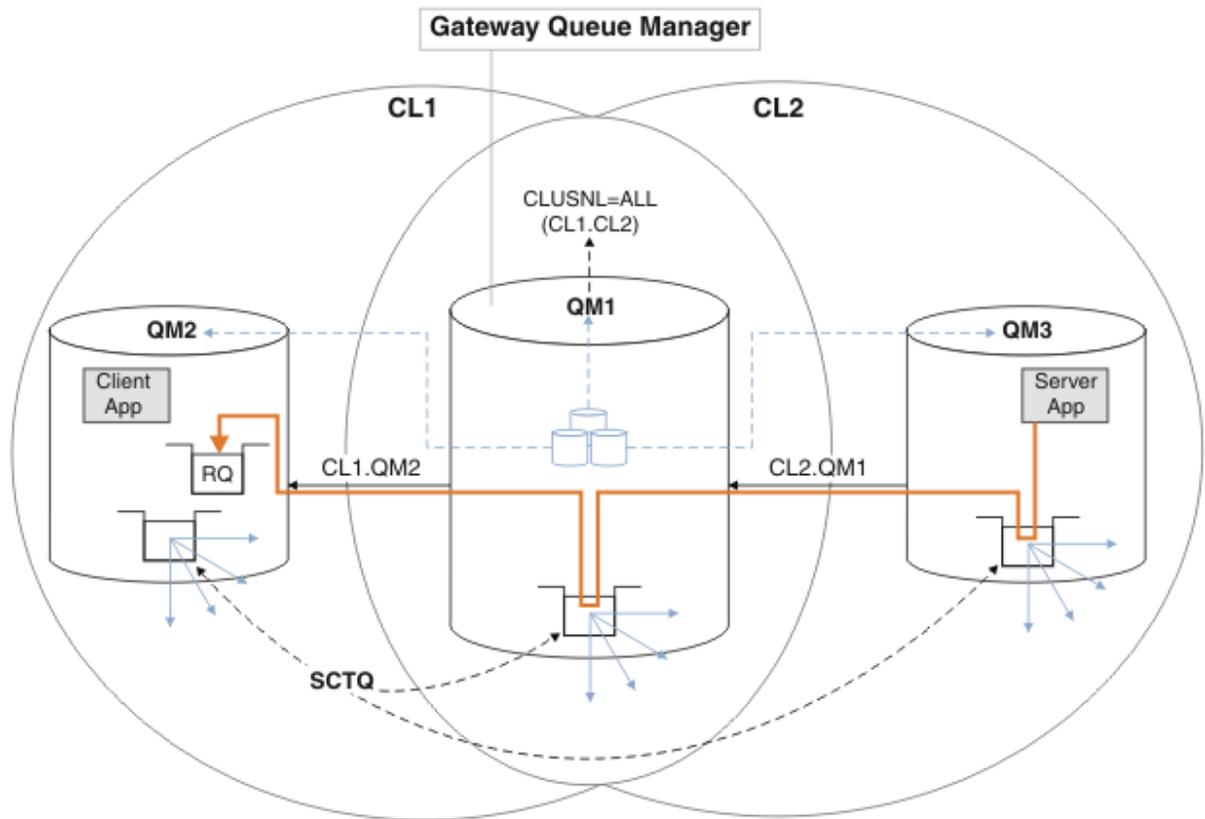


그림 48. 큐 관리자 알리언스를 사용하여 응답 메시지를 다른 클러스터로 리턴

라우팅이 작동하는 방식은 다음과 같습니다. 각 클러스터의 모든 큐 관리자에는 QM1에 대한 큐 관리자 알리언스 정의가 있습니다. 이러한 알리언스는 모든 클러스터에서 클러스터됩니다. 각 알리언스에서 큐 관리자로의 회색 파선 화살표는 각 큐 관리자 알리언스가 최소 하나 이상의 클러스터에 있는 실제 큐 관리자로 해석됨을 표시합니다. 이 경우 QM2 별명은 CL1 및 CL2 클러스터 모두에서 클러스터되며 CL1의 실제 큐 관리자 QM2(으)로 해석됩니다. 서버 애플리케이션은 큐 이름 RQ에 대한 응답을 사용하여 응답 메시지를 작성하고 큐 관리자 이름 QM2에 응답합니다. 큐 관리자 알리언스 정의의 QM2이(가) 클러스터 CL2의 QM1에 정의되고 큐 관리자 QM2이(가) 클러스터 CL2에 없기 때문에 메시지가 QM1(으)로 라우트됩니다. 메시지를 대상 큐 관리자로 송신할 수 없으므로 이 메시지는 알리언스 정의가 있는 큐 관리자에게 송신됩니다.

QM1은(는) 트랜스퍼럴에 대한 QM1의 클러스터 트랜스미션 큐에 메시지를 QM2에 둡니다. QM2에 대한 QM1에서 큐 관리자 별명 정의가 QM2을(를) 실제 대상 큐 관리자로 정의하기 때문에 QM1은(는) 메시지를 QM2에 라우트합니다. 정의는 순환되지 않습니다. 알리언스 정의는 실제 정의만 참조할 수 있고 알리언스는 그 자체를 가리킬 수 없기 때문입니다. QM1 및 QM2 모두가 동일한 클러스터 CL1에 있기 때문에 실제 정의는 QM1에 의해 해석됩니다. QM1은(는) CL1의 저장소에서 QM2에 대한 연결 정보를 찾고 메시지를 QM2(으)로 라우트합니다. 메시지가 QM1에 의해 리라우트되도록 하려면, 서버 애플리케이션이 DEFBIND 옵션을

MQBND_BIND_NOT_FIXED(으)로 설정하여 응답 큐를 열어야 합니다. 서버 애플리케이션이 MQBND_BIND_ON_OPEN 옵션을 사용하여 응답 큐를 연 경우, 메시지는 경로 재지정되지 않으며 데드-레터 큐에서 종료됩니다.

큐 관리자를 클러스터로의 게이트웨이로 사용하여 클러스터 외부에서 수신되는 메시지를 워크로드 밸런싱합니다.

클러스터에 있는 둘 이상의 큐 관리자에서 EDINBURGH라고 하는 큐를 정의합니다. 클러스터링 메커니즘이 클러스터 외부에서 수신되는 메시지에 대해 워크로드 밸런싱을 수행하게 하려고 합니다.

클러스터 외부의 큐 관리자에게는 클러스터 내 한 큐 관리자에 대한 송신자 채널 및 전송 큐가 필요합니다. 이 큐를 게이트웨이 큐 관리자라고 합니다. 기본 워크로드 밸런싱 메커니즘을 이용하려면 다음 규칙 중 하나를 적용해야 합니다.

- 게이트웨이 큐 관리자는 EDINBURGH 큐의 인스턴스를 포함할 수 없습니다.
- 게이트웨이 큐 관리자는 ALTER QMGR에서 CLWLUSEQ(ANY)를 지정합니다.

클러스터 외부에서의 워크로드 밸런싱 예는 234 페이지의 『클러스터 외부에서 워크로드 밸런싱 구성』의 내용을 참조하십시오.

응답 대상 큐 알리어스 및 클러스터

응답 대상 큐 알리어스 정의는 응답 정보에 대한 대체 이름을 지정하는 데 사용됩니다. 응답 대상 큐 알리어스 정의는 분산 큐잉 환경에서와 같이 클러스터에서 사용할 수 있습니다.

예를 들면, 다음과 같습니다.

- 큐 관리자 VENICE의 애플리케이션은 MQPUT 호출을 사용하여 큐 관리자 PISA에 메시지를 송신합니다. 이 애플리케이션은 메시지 디스크립터에서 다음 응답 대상 큐 정보를 제공합니다.

```
ReplyToQ='QUEUE'  
ReplyToQMGr=''
```

- QUEUE에 송신된 응답을 PISA의 OTHERQ에서 수신할 수 있도록 VENICE에서 응답 대상 큐 알리어스로 사용되는 리모트 큐 정의를 작성하십시오. 이 알리어스는 알리어스가 작성된 시스템에서만 유효합니다.

```
DEFINE QREMOTE(QUEUE) RNAME(OTHERQ) RQMNAME(PISA)
```

RQMNAME 및 QREMOTE는 RQMNAME이 그 자체로 클러스터 큐 관리자인 경우라도 같은 이름을 지정할 수 있습니다.

큐 알리어스 및 클러스터

큐 알리어스를 사용하여 클러스터 큐의 이름을 숨기거나, 큐를 클러스터링하거나, 다른 속성을 채택하거나 다른 액세스 제어를 채택할 수 있습니다.

QALIAS 정의는 큐를 나타내는 알리어스를 작성하는 데 사용됩니다. 다음과 같은 여러 이유로 알리어스를 작성할 수 있습니다.

- 다른 큐 사용을 시작하려고 하지만 애플리케이션은 변경하지 않으려고 합니다.
- 애플리케이션이 메시지를 넣는 큐의 실제 이름을 애플리케이션에 알리지 않으려고 합니다.
- 큐가 정의된 것과는 다른 이름 지정 규칙이 있을 수 있습니다.
- 애플리케이션이 실제 이름을 사용할 경우 큐에 대한 액세스 권한이 부여되지 않고 그 알리어스를 사용하는 경우에만 액세스 권한이 부여됩니다.

DEFINE QALIAS 명령을 사용하여 큐 관리자에서 QALIAS 정의를 작성하십시오. 예를 들어, 다음 명령을 실행하십시오.

```
DEFINE QALIAS(PUBLIC) TARGET(LOCAL) CLUSTER(C)
```

명령은 PUBLIC이라는 큐를 클러스터 C의 큐 관리자에 알립니다. PUBLIC은 LOCAL이라는 큐로 해석되는 별명입니다. PUBLIC으로 송신되는 메시지는 LOCAL이라고 하는 큐에 라우트됩니다.

큐 알리어스 정의를 사용하여 큐 이름을 클러스터 큐로 해석할 수도 있습니다. 예를 들어, 다음 명령을 실행하십시오.

```
DEFINE QALIAS(PRIVATE) TARGET(PUBLIC)
```

이 명령을 사용하면 큐 관리자가 이름 PRIVATE을 사용하여 PUBLIC이라는 이름으로 클러스터 다른 위치에 알려진 큐에 액세스할 수 있습니다. 이 정의에는 CLUSTER 속성이 포함되지 않으므로 이를 작성하는 큐 관리자에만 적용됩니다.

워크로드 관리에 클러스터 사용

클러스터에 있는 여러 큐 관리자에서 한 큐의 다중 인스턴스를 정의하여 다중 서버에 큐 서비스 작업을 분산시킬 수 있습니다. 실패가 발생할 경우 메시지가 다른 큐 관리자에 다시 큐잉되지 않게 방지할 수 있는 여러 요소가 있습니다.

시스템 관리를 줄이게 위해 클러스터를 설정할 뿐 아니라 둘 이상의 큐 관리자가 같은 큐의 인스턴스를 호스팅하는 클러스터를 작성할 수 있습니다.

클러스터 안에 있는 큐 관리자가 서로의 복제본이 되도록 클러스터를 구성할 수 있습니다. 각 큐 관리자는 동일한 애플리케이션을 실행하고 동일한 큐의 로컬 정의를 가질 수 있습니다. 애플리케이션의 여러 인스턴스를 사용하여 큐 관리자 사이에 워크로드를 분산시킬 수 있습니다. 애플리케이션의 각 인스턴스는 메시지를 수신하고 서로 독립적으로 실행합니다.

이 방식으로 클러스터를 사용할 때의 장점은 다음과 같습니다.

- 큐 및 애플리케이션의 사용가능성 증가
- 더 빠른 메시지 처리량
- 네트워크에서 워크로드의 보다 균등한 분배

특정 큐의 인스턴스를 호스팅하는 큐 관리자의 어느 하나가 해당 큐를 목적지로 하는 메시지를 처리할 수 있습니다. 애플리케이션은 메시지를 송신할 때 큐 관리자를 이름 지정하지 않습니다. 워크로드 관리 알고리즘이 메시지를 처리할 큐 관리자를 판별합니다.

워크로드 관리를 위한 클러스터 구성의 자세한 정보는 다음 하위 주제를 참조하십시오.

관련 개념

[클러스터](#)

[클러스터 작동 방법](#)

[146 페이지의 『클러스터링과 분산 큐잉의 비교』](#)

분산 큐잉 및 클러스터링을 사용하여 큐 관리자를 연결하기 위해 정의해야 하는 컴포넌트를 비교합니다.

[148 페이지의 『클러스터의 컴포넌트』](#)

클러스터는 큐 관리자, 클러스터 저장소, 클러스터 채널 및 클러스터 큐로 구성됩니다.

[167 페이지의 『IBM WebSphere MQ 클러스터 관리』](#)

IBM WebSphere MQ 클러스터를 작성, 확장 및 유지보수할 수 있습니다.

[227 페이지의 『메시지를 클러스터로\(부터\) 라우팅』](#)

큐 알리어스, 큐 관리자 알리어스 및 리모트 큐 정의를 사용하여 클러스터를 외부 큐 관리자 및 다른 클러스터에 연결합니다.

관련 태스크

[144 페이지의 『큐 관리자 클러스터 구성』](#)

클러스터가 작업하는 방법, 클러스터 구성을 설계하는 방법을 찾고 단순한 클러스터를 설정하는 방법의 예를 알려면 이 토픽의 링크를 사용하십시오.

[168 페이지의 『새 클러스터 설정』](#)

다음 지시사항에 따라 클러스터 예를 설정합니다. 개별 지시사항에서는 TCP/IP, LU 6.2 및 단일 전송 큐 또는 다중 전송 큐에서의 클러스터 설정을 설명합니다. 한 큐 관리자에서 다른 큐 관리자로 메시지를 송신하여 클러스터가 작동하는지 테스트합니다.

[클러스터 워크로드 엑시트 작성 및 컴파일](#)

둘 이상의 큐 인스턴스가 있는 클러스터의 예

둘 이상의 큐 인스턴스가 있는 이 클러스터 예에서 메시지는 큐의 다른 인스턴스로 라우트됩니다. 큐의 특정 인스턴스에 대해 메시지를 강제 실행하고 메시지 시퀀스를 큐 관리자 중 하나에 송신하도록 선택할 수 있습니다.

241 페이지의 그림 49에서는 큐 Q3에 대해 둘 이상의 정의가 있는 클러스터를 보여줍니다. QM1의 애플리케이션이 메시지를 Q3에 넣는 경우 해당 메시지를 처리할 Q3의 인스턴스를 꼭 알아야 하는 것은 아닙니다. Q3의 로컬 인스턴스가 있는 QM2 또는 QM4, 에서 애플리케이션이 실행 중인 경우, Q3의 로컬 인스턴스가 기본적으로 열립니다. CLWLUSEQ 큐 속성을 설정하면 큐의 로컬 인스턴스를 큐의 원격 인스턴스와 동일하게 처리할 수 있습니다.

MQOPEN 옵션 DefBind는 MQOPEN 호출이 발행될 때 또는 메시지가 전송 큐에서 전송될 때 대상 큐 관리자가 선택되는지 여부를 제어합니다.

DefBind를 MQBND_BIND_NOT_FIXED로 설정하면 메시지가 전송될 때 사용 가능한 큐의 인스턴스에 메시지가 송신될 수 있습니다. 이 경우 다음과 같은 문제점을 피할 수 있습니다.

- 메시지가 대상 큐 관리자에 도착할 때 대상 큐가 사용 불가능합니다.
- 큐의 상태가 변경되었습니다.
- 메시지를 클러스터 큐 알리어스를 사용하여 넣었고 클러스터 큐 알리어스의 인스턴스가 정의되는 큐 관리자에 대상 큐의 인스턴스가 존재하지 않습니다.

전송 시 이러한 문제점 중 어느 하나라도 발견되면 대상 큐의 사용 가능한 다른 인스턴스를 찾아 메시지를 다시 라우팅합니다. 사용 가능한 큐의 인스턴스가 없는 경우 메시지가 데드-레터 큐에 배치됩니다.

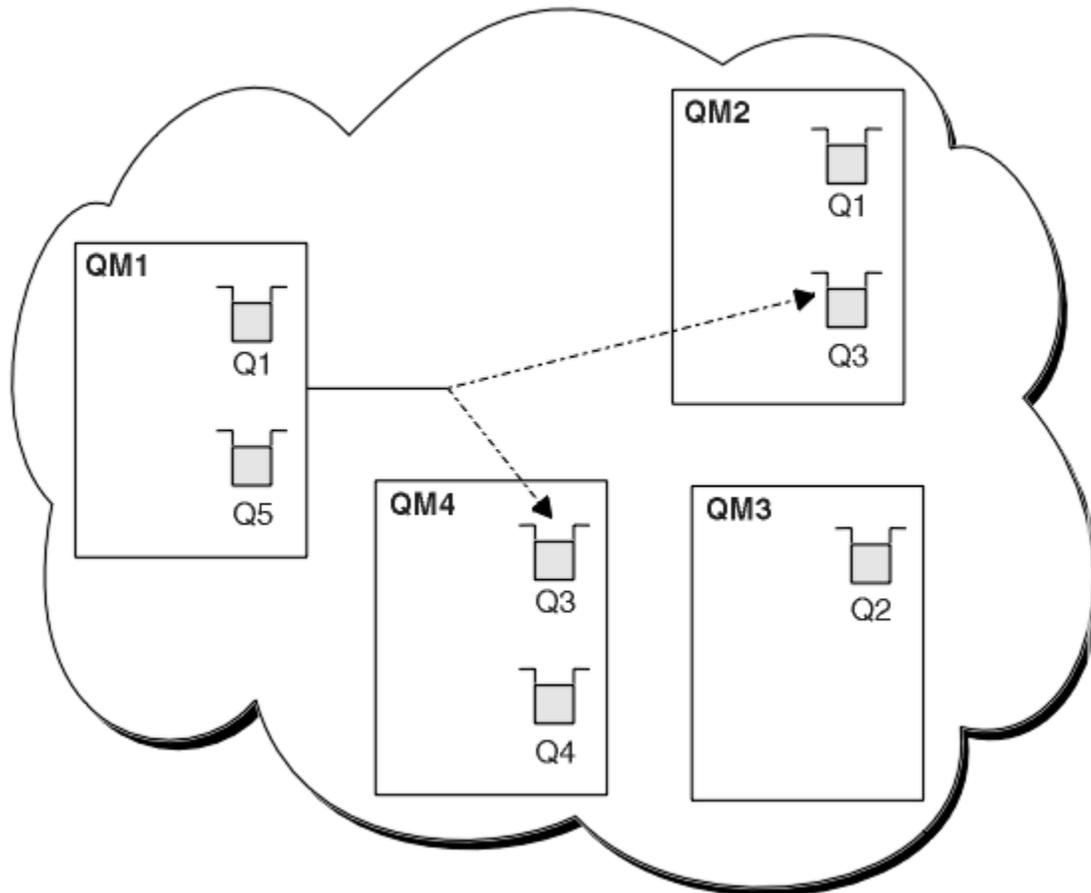


그림 49. 동일한 큐의 다중 인스턴스를 포함하는 클러스터

메시지가 다시 라우팅되는 것을 막을 수 있는 한 가지 요소는 MQBND_BIND_ON_OPEN을 사용하여 메시지를 고정 큐 관리자 또는 채널로 지정하는 경우입니다. MQOPEN에 바인드된 메시지는 다른 채널에 재할당되지 않습니다. 또한 메시지 재할당은 클러스터 채널이 실제로 실패할 때만 발생한다는 점도 유의하십시오. 재할당은 채널이 이미 실패한 경우에는 발생하지 않습니다.

목적지 큐 관리자가 작동하지 않으면 시스템이 메시지를 다시 라우팅하려고 시도합니다. 이러한 시도에서 시스템은 메시지를 손실할 수 있는 위험을 감수하거나 복제본을 작성하여 메시지의 무결성에 영향을 미치지 않습니다. 큐 관리자가 실패하고 인다우트(in-doubt) 상태로 메시지를 둘 경우 해당 메시지는 다시 라우팅되지 않습니다.

큐를 로컬로 호스팅하는 큐 관리자 추가

다음 지시사항에 따라 INVENTQ의 인스턴스를 추가하여 Paris와 New York에서 인벤토리 애플리케이션 시스템을 실행하기 위한 추가 용량을 제공합니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 태스크를 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- INVENTORY 클러스터가 클러스터에 새 큐 관리자 추가에 설명된 대로 설정되었습니다. 이 클러스터는 세 큐 관리자를 포함하며 LONDON 및 NEWYORK은 둘 다 전체 저장소를 보유하고 PARIS는 부분 저장소를 보유합니다. 인벤토리 애플리케이션은 NEWYORK 큐 관리자에 연결된 New York의 시스템에서 실행됩니다. 애플리케이션은 INVENTQ 큐에 메시지가 도착하면 구동됩니다.
- INVENTQ의 인스턴스를 추가하여 Paris와 New York에서 인벤토리 애플리케이션 시스템을 실행하기 위한 추가 용량을 제공하려고 합니다.

이 태스크 정보

다음 단계에 따라 큐를 로컬로 호스팅하는 큐 관리자를 추가하십시오.

프로시저

1. PARIS 큐 관리자를 대체하십시오.

Paris에 있는 애플리케이션이 Paris에 있는 INVENTQ와 New York에 있는 INVENTQ를 사용하게 하려면 큐 관리자에게 알려야 합니다. PARIS에서 다음 명령을 실행하십시오.

```
ALTER QMGR CLWLUSEQ(ANY)
```

2. 메시지 연관관계에 대해 인벤토리 애플리케이션을 검토하십시오.

계속하기 전에 메시지 처리 순서에 따라 인벤토리 애플리케이션에 종속 항목이 없는지 확인하십시오. 추가 정보는 252 페이지의 『[메시지 연관관계 핸들링](#)』의 내용을 참조하십시오.

3. Paris에 있는 시스템에 인벤토리 애플리케이션을 설치하십시오.
4. 클러스터 큐 INVENTQ를 정의하십시오.

이미 NEWYORK 큐 관리자가 호스팅하고 있는 INVENTQ 큐도 PARIS가 호스팅합니다. 다음과 같이 이 큐를 PARIS 큐 관리자에서 정의하십시오.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

이제 모든 정의를 완료했습니다. 아직 완료하지 않은 경우 z/OS용 WebSphere MQ에서 채널 시작기를 시작하십시오. 모든 플랫폼의 큐 관리자 PARIS에서 리스너 프로그램을 시작하십시오. 리스너는 수신되는 네트워크 요청을 청취하고 필요한 경우 클러스터 수신자 채널을 시작합니다.

결과

243 페이지의 그림 50에서는 이 태스크에 의해 설정되는 클러스터를 보여줍니다.

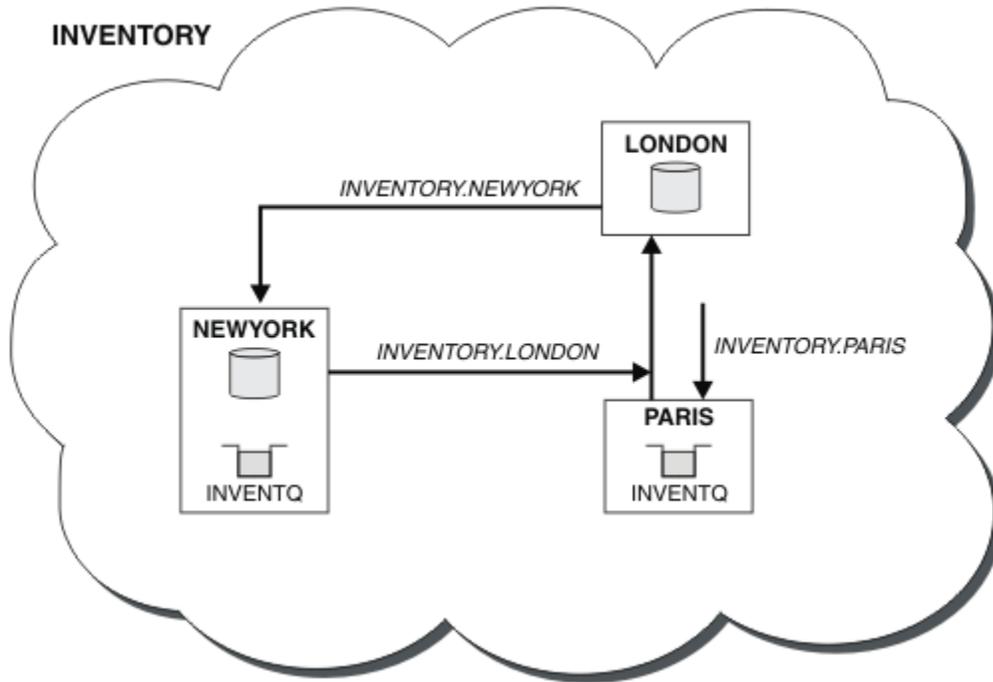


그림 50. 세 개의 큐 관리자가 있는 INVENTORY 클러스터

이 클러스터에 대한 수정은 큐 관리자 NEWYORK 또는 LONDON을 변경하지 않고 수행되었습니다. 이러한 큐 관리자의 전체 저장소는 메시지를 PARIS의 INVENTQ에 송신하기 위해 필요한 정보로 자동 업데이트됩니다.

다음에 수행할 작업

INVENTQ 큐 및 인벤토리 애플리케이션은 이제 클러스터에 있는 두 개의 큐 관리자에서 호스팅됩니다. 이 경우 두 큐 관리자의 가용성이 증가되고 메시지 처리 속도가 빨라지며 두 큐 관리자 사이에 워크로드를 분산시킬 수 있습니다. 큐 관리자 LONDON, NEWYORK 및 PARIS 중 어느 하나가 INVENTQ에 넣은 메시지는 워크로드 밸런싱을 위해 PARIS 또는 NEWYORK에 교대로 라우트됩니다.

클러스터에서 두 개의 네트워크 사용

다음 지시사항에 따라 TOKYO에 두 개의 다른 네트워크가 있는 새 저장소를 추가합니다. 두 네트워크 모두 Tokyo의 큐 관리자와 통신하는 데 사용 가능해야 합니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 작업을 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- INVENTORY 클러스터가 "클러스터에 큐 관리자 추가"에 설명된 대로 설정되었습니다. 이 클러스터는 세 큐 관리자를 포함하며 LONDON 및 NEWYORK은 둘 다 전체 저장소를 보유하고 PARIS는 부분 저장소를 보유하고 있습니다. 인벤토리 애플리케이션은 NEWYORK 큐 관리자에 연결된 New York의 시스템에서 실행됩니다. 애플리케이션은 INVENTQ 큐에 메시지가 도착하면 구동됩니다.
- 새 저장소가 두 개의 다른 네트워크가 있는 TOKYO에 추가됩니다. 두 네트워크 모두 Tokyo의 큐 관리자와 통신하는 데 사용 가능해야 합니다.

이 태스크 정보

다음 단계에 따라 클러스터에서 두 네트워크를 사용하십시오.

프로시저

1. TOKYO가 처음 참조할 전체 저장소를 결정하십시오.

클러스터 내 모든 큐 관리자는 클러스터에 대한 정보를 수집하기 위해 전체 저장소 중 어느 하나를 참조해야 합니다. 그러면 자체 부분 저장소를 빌드합니다. 저장소 선택이 특별히 중요한 것은 아닙니다. 이 예에서는 NEWYORK을 선택합니다. 새 큐 관리자는 클러스터에 조인하고 나면 두 저장소와 모두 통신합니다.

2. CLUSRCVR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 수신할 수 있는 클러스터 수신자를 정의해야 합니다. 이 큐 관리자는 각 네트워크에서 통신할 수 있어야 합니다.

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETB.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network B for TOKYO')
```

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETA.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network A for TOKYO')
```

3. 큐 관리자 TOKYO. 에서 CLUSSDR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 해당 첫 번째 전체 저장소에 송신할 수 있는 한 개의 클러스터 송신자 채널을 정의해야 합니다. 이 경우 NEWYORK을 선택했으므로 TOKYO에는 다음 정의가 필요합니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from TOKYO to repository at NEWYORK')
```

이제 모든 정의를 완료했습니다. 아직 완료하지 않은 경우 z/OS용 WebSphere MQ에서 채널 시작기를 시작하십시오. 모든 플랫폼의 큐 관리자 PARIS에서 리스너 프로그램을 시작하십시오. 리스너 프로그램은 수신되는 네트워크 요청을 청취하고 필요한 경우 클러스터 수신자 채널을 시작합니다.

결과

244 페이지의 그림 51에서는 이 태스크에 의해 설정되는 클러스터를 보여줍니다.

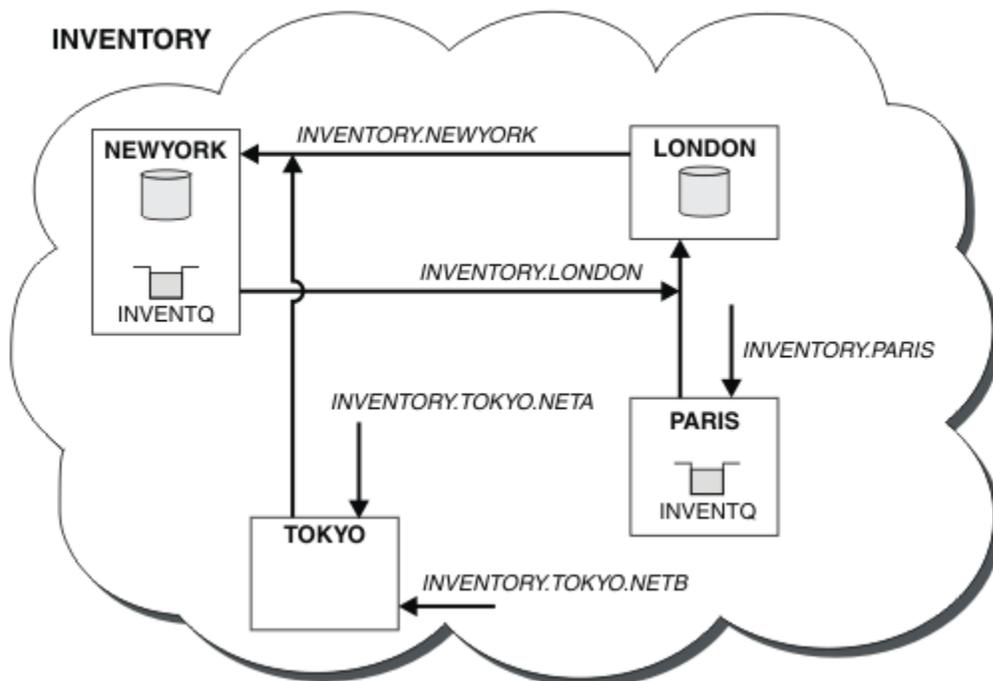


그림 51. 네 개의 큐 관리자가 있는 INVENTORY 클러스터

세 개의 정의만 작성하여 큐 관리자 TOKYO를 사용 가능한 두 개의 다른 네트워크 루트가 있는 클러스터에 추가했습니다.

관련 태스크

177 페이지의 『클러스터에 큐 관리자 추가』

다음 지시사항에 따라 사용자가 작성한 클러스터에 큐 관리자를 추가합니다. 클러스터 큐 및 토픽에 대한 메시지는 단일 클러스터 전송 큐 SYSTEM.CLUSTER.TRANSMIT.QUEUE를 사용하여 전송됩니다.

클러스터에서 1차 및 보조 네트워크 사용

다음 지시사항에 따라 한 네트워크는 1차 네트워크로 하고 다른 네트워크는 백업 네트워크로 구성합니다. 1차 네트워크에 문제점이 있을 경우 백업 네트워크를 사용합니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 태스크를 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- INVENTORY 클러스터가 243 페이지의 『클러스터에서 두 개의 네트워크 사용』에서 설명된 대로 설정되었습니다. 이 클러스터는 네 개의 큐 관리자를 포함하며 LONDON 및 NEWYORK은 둘 다 전체 저장소를 보유하고 PARIS 및 TOKYO는 부분 저장소를 보유하고 있습니다. 인벤토리 애플리케이션은 큐 관리자 NEWYORK에 연결된 New York의 시스템에서 실행됩니다. TOKYO 큐 관리자에는 통신에 사용할 수 있는 두 개의 다른 네트워크가 있습니다.
- 이러한 네트워크 중 하나는 1차 네트워크로 하고 다른 네트워크는 백업 네트워크로 구성하려고 합니다. 1차 네트워크에 문제점이 있을 경우 백업 네트워크를 사용할 계획입니다.

이 태스크 정보

NETPRTY 속성을 사용하여 클러스터에서 1차 및 보조 네트워크를 구성하십시오.

프로시저

TOKYO에서 기존 CLUSRCVR 채널을 대체하십시오.

네트워크 A 채널이 1차 채널이고 네트워크 B가 보조 채널임을 표시하려면 다음 명령을 사용하십시오.

- a) ALTER CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) NETPRTY(2) DESCR('Main cluster-receiver channel for TOKYO')
- b) ALTER CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) NETPRTY(1) DESCR('Backup cluster-receiver channel for TOKYO')

다음에 수행할 작업

서로 다른 네트워크 우선순위로 채널을 구성함으로써 클러스터에 대해 1차 네트워크 및 보조 네트워크가 있음을 구성했습니다. 이러한 채널을 사용하는 클러스터의 큐 관리자는 사용 가능한 경우 자동으로 1차 네트워크를 사용합니다. 1차 네트워크가 사용 불가능한 경우 큐 관리자는 보조 네트워크를 사용하기 위해 장애를 복구합니다.

백업 역할을 하는 큐 추가

다음 지시사항에 따라 New York에서 실행되는 인벤토리 시스템에 대한 백업을 Chicago에 제공합니다. 이 Chicago 시스템은 New York 시스템에 문제점이 있는 경우에만 사용됩니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 태스크를 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- INVENTORY 클러스터가 177 페이지의 『클러스터에 큐 관리자 추가』에서 설명된 대로 설정되었습니다. 이 클러스터는 세 큐 관리자를 포함하며 LONDON 및 NEWYORK은 둘 다 전체 저장소를 보유하고 PARIS는 부분 저장소를 보유하고 있습니다. 인벤토리 애플리케이션은 NEWYORK 큐 관리자에 연결된 New York의 시스템에서 실행됩니다. 애플리케이션은 INVENTQ 큐에 메시지가 도착하면 구동됩니다.
- New York에서 실행되는 인벤토리 시스템에 대한 백업을 제공하기 위해 Chicago에서 새 저장소가 설정됩니다. 이 Chicago 시스템은 New York 시스템에 문제점이 있는 경우에만 사용됩니다.

이 태스크 정보

다음 단계에 따라 백업 역할을 할 큐를 추가하십시오.

프로시저

1. CHICAGO가 처음 참조할 전체 저장소를 결정하십시오.

클러스터 내 모든 큐 관리자는 클러스터에 대한 정보를 수집하기 위해 전체 저장소 중 어느 하나를 참조해야 합니다. 그러면 자체 부분 저장소를 빌드합니다. 특정 큐 관리자에 대해 선택하는 저장소가 특별히 중요하지는 않습니다. 이 예에서는 NEWYORK을 선택합니다. 새 큐 관리자는 클러스터에 조인하고 나면 두 저장소와 모두 통신합니다.

2. CLUSRCVR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 수신할 수 있는 클러스터 수신자를 정의해야 합니다. CHICAGO에서 다음을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.CHICAGO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(CHICAGO.CMSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel for CHICAGO')
```

3. 큐 관리자 CHICAGO에서 CLUSSDR 채널을 정의하십시오.

클러스터 내 모든 큐 관리자는 메시지를 해당 첫 번째 전체 저장소에 송신할 수 있는 한 개의 클러스터 송신자 채널을 정의해야 합니다. 이 경우 NEWYORK을 선택했으므로 CHICAGO에는 다음 정의가 필요합니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from CHICAGO to repository at NEWYORK')
```

4. 기존 클러스터 큐 INVENTQ를 대체하십시오.

이미 NEWYORK 큐 관리자가 호스팅하고 있는 INVENTQ가 큐의 기본 인스턴스입니다.

```
ALTER QLOCAL(INVENTQ) CLWLPRTY(2)
```

5. 메시지 연관관계에 대해 인벤토리 애플리케이션을 검토하십시오.

계속하기 전에 메시지 처리 순서에 따라 인벤토리 애플리케이션에 종속 항목이 없는지 확인하십시오.

6. CHICAGO에 있는 시스템에 인벤토리 애플리케이션을 설치하십시오.

7. 백업 클러스터 큐 INVENTQ를 정의하십시오.

이미 NEWYORK 큐 관리자가 호스팅하고 있는 INVENTQ 큐도 CHICAGO에 의해 백업으로 호스팅됩니다. 다음과 같이 이 큐를 CHICAGO 큐 관리자에서 정의하십시오.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) CLWLPRTY(1)
```

이제 모든 정의를 완료했습니다. 아직 완료하지 않은 경우 z/OS용 WebSphere MQ에서 채널 시작기를 시작하십시오. 모든 플랫폼의 큐 관리자 CHICAGO에서 리스너 프로그램을 시작하십시오. 리스너 프로그램은 수신되는 네트워크 요청을 청취하고 필요한 경우 클러스터 수신자 채널을 시작합니다.

결과

247 페이지의 그림 52에서는 이 태스크에 의해 설정되는 클러스터를 보여줍니다.

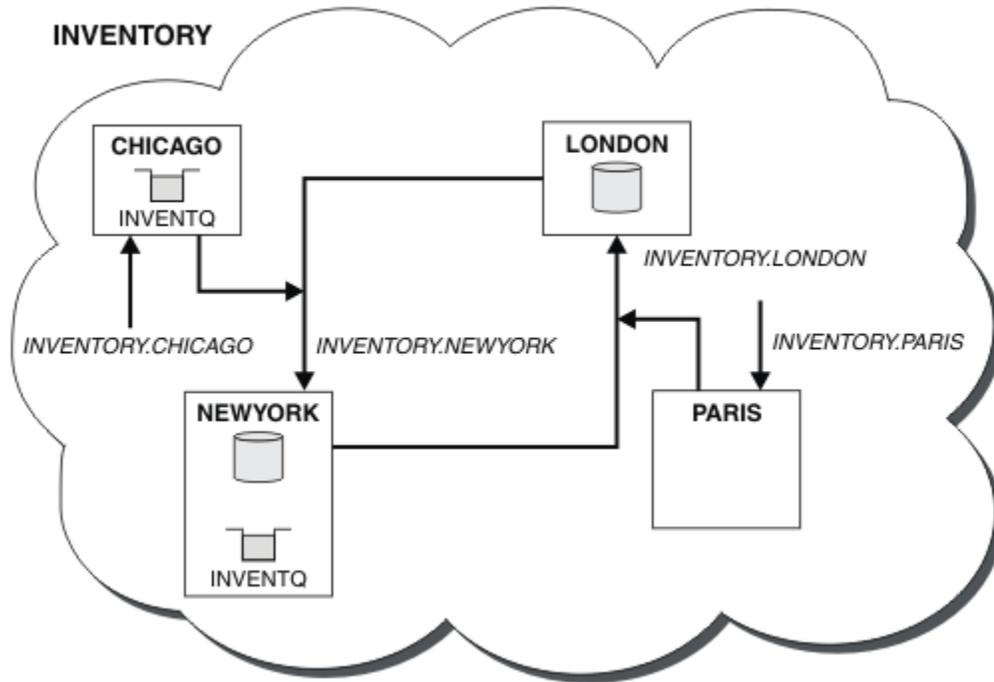


그림 52. 네 개의 큐 관리자가 있는 *INVENTORY* 클러스터

INVENTQ 큐 및 인벤토리 애플리케이션은 이제 클러스터에 있는 두 개의 큐 관리자에서 호스팅됩니다. *CHICAGO* 큐 관리자는 백업입니다. *INVENTQ*에 넣은 메시지는 *CHICAGO*로 대신 송신될 때 사용 불가능하지 않으면 *NEWYORK*에 라우트됩니다.

참고:

리모트 큐 관리자의 가용성은 해당 큐 관리자에 대한 채널의 상태를 기준으로 합니다. 채널이 시작될 때 해당 상태는 여러 번 변경되며, 일부 상태는 클러스터 워크로드 관리 알고리즘에 대한 우선권이 낮습니다. 실제로 이는 상위 우선순위(1차) 목적지에 대한 채널이 시작되는 동안 하위 우선순위(백업) 목적지를 선택할 수 있음을 의미합니다.

백업 목적지로 메시지가 이동하지 않도록 해야 하는 경우 *CLWLRPTY*를 사용하지 마십시오. 1차에서 백업으로의 수동 전환에 개별 큐 또는 *CLWLRANK* 사용을 고려해 보십시오.

사용되는 채널 수 제한

다음 지시사항에 따라 다양한 큐 관리자에 가격 검사 애플리케이션이 설치될 때 각 서버가 실행하는 활성 채널의 수를 제한합니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 작업을 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- 가격 검사 애플리케이션은 다양한 큐 관리자에 설치됩니다. 사용되는 채널의 수를 적은 수로 유지하기 위해 각 서버가 실행하는 활성 채널의 수가 제한됩니다. 애플리케이션은 *PRICEQ* 큐에 메시지가 도착하면 구동됩니다.
- 네 개의 서버 큐 관리자가 가격 검사 애플리케이션을 호스팅합니다. 두 조회 큐 관리자가 가격을 조회하기 위한 메시지를 *PRICEQ*에 송신합니다. 두 개의 추가 큐 관리자가 전체 저장소로 구성됩니다.

이 태스크 정보

다음 단계에 따라 사용되는 채널의 수를 제한하십시오.

프로시저

1. 두 개의 전체 저장소를 선택하십시오.

가격 검사 클러스터의 전체 저장소가 될 두 개의 큐 관리자를 선택하십시오. 이를 REPOS1 및 REPOS2라고 합니다.

다음 명령을 실행하십시오.

```
ALTER QMGR REPOS(PRICECHECK)
```

2. 각 큐 관리자에서 CLUSRCVR 채널을 정의하십시오.

클러스터 내 각 큐 관리자에서 클러스터 수신자 채널 및 클러스터 송신자 채널을 정의하십시오. 어떤 채널이 처음 정의되는지는 중요하지 않습니다.

```
DEFINE CHANNEL(PRICECHECK.SERVE1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNNAME(SERVER1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-receiver channel')
```

3. 각 큐 관리자에서 CLUSSDR 채널을 정의하십시오.

각 큐 관리자에서 CLUSSDR 정의를 작성하여 해당 큐 관리자를 전체 저장소 큐 관리자 중 어느 하나에 링크하십시오.

```
DEFINE CHANNEL(PRICECHECK.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNNAME(REPOS1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-sender channel to  
repository queue manager')
```

4. 가격 검사 애플리케이션을 설치하십시오.
5. 모든 서버 큐 관리자에서 PRICEQ 큐를 정의하십시오.

각각에서 다음 명령을 실행하십시오.

```
DEFINE QLOCAL(PRICEQ) CLUSTER(PRICECHECK)
```

6. 조회에 사용되는 채널의 수를 제한하십시오.

조회 큐 관리자에서 각각에 다음 명령을 발행하여 사용되는 활성 채널의 수를 제한합니다.

```
ALTER QMGR CLWLMRUC(2)
```

7. 아직 완료하지 않은 경우 z/OS용 WebSphere MQ에서 채널 시작기를 시작하십시오. 모든 플랫폼에서 리스너 프로그램을 시작하십시오.

리스너 프로그램은 수신되는 네트워크 요청을 청취하고 필요한 경우 클러스터 수신자 채널을 시작합니다.

결과

[249 페이지의 그림 53](#)에서는 이 태스크에 의해 설정되는 클러스터를 보여줍니다.

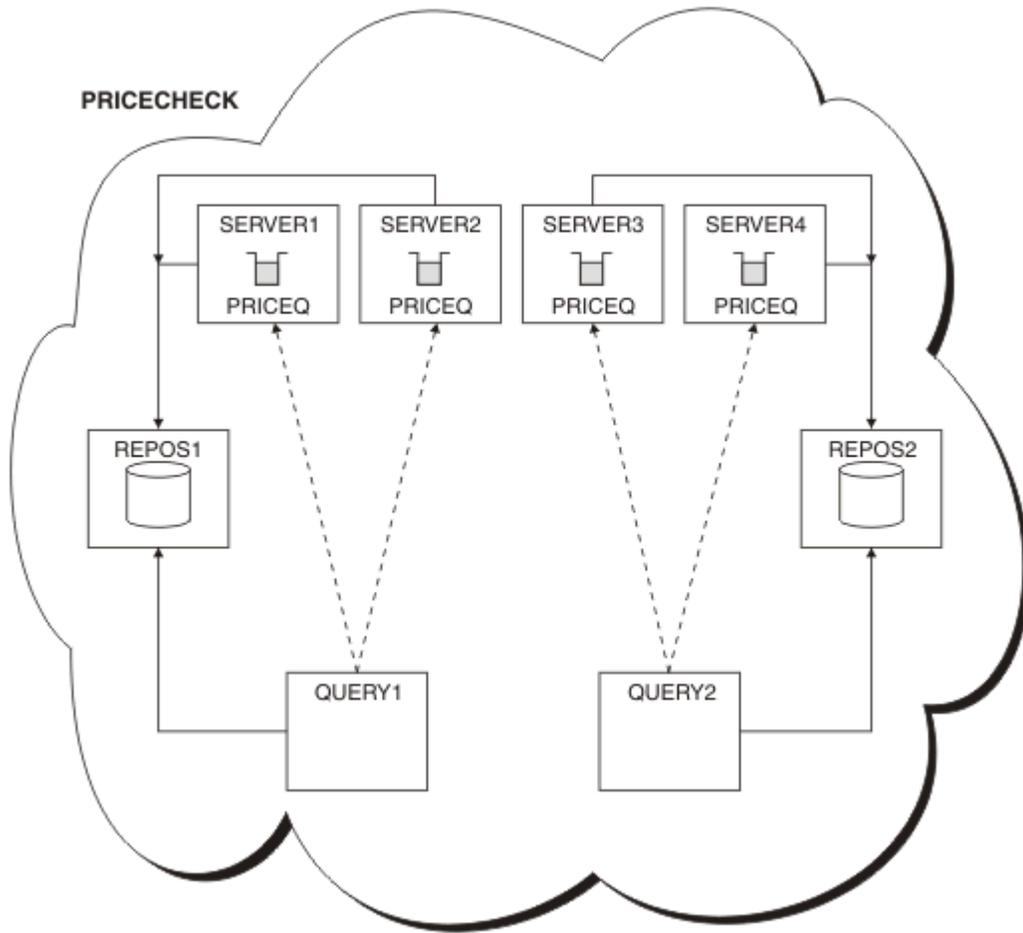


그림 53. 네 개의 서버 큐 관리자, 두 개의 저장소 및 두 개의 조회 큐 관리자가 포함된 *PRICHECK* 클러스터

PRICHECK 클러스터에 사용 가능한 네 개의 *PRICEQ* 큐 인스턴스가 있지만 각 조회 큐 관리자는 그 중 두 개의 인스턴스만 사용합니다. 예를 들어, *QUERY1* 큐 관리자에만 *SERVER1* 및 *SERVER2* 큐 관리자에 대한 활성 채널이 있습니다. *SERVER1*이 사용 불가능한 상태가 되면 *QUERY1* 큐 관리자는 다른 큐 관리자(예: *SERVER3*)를 사용하기 시작합니다.

다음에 수행할 작업

PRICHECK 클러스터에 사용 가능한 네 개의 *PRICEQ* 큐 인스턴스가 있지만 각 조회 큐 관리자는 그 중 두 개의 인스턴스만 사용합니다. 예를 들어, *QUERY1* 큐 관리자에만 *SERVER1* 및 *SERVER2* 큐 관리자에 대한 활성 채널이 있습니다. *SERVER1*이 사용 불가능한 상태가 되면 *QUERY1* 큐 관리자는 다른 큐 관리자(예: *SERVER3*)를 사용하기 시작합니다.

큐를 호스팅하는 더 강력한 큐 관리자 추가

다음 지시사항에 따라 인벤토리 시스템을 New York뿐만 아니라 Los Angeles에서도 실행하여 추가 용량을 제공합니다. 여기서 Los Angeles는 New York에 비해 메시지를 두 배로 처리할 수 있습니다.

시작하기 전에

참고: 클러스터에 대한 변경이 클러스터 전체에 전파되도록 하려면 최소한 하나 이상의 전체 저장소가 항상 사용 가능해야 합니다. 이 작업을 시작하기 전에 저장소가 사용 가능한지 확인하십시오.

시나리오:

- 177 페이지의 『클러스터에 큐 관리자 추가』에 설명된 대로 INVENTORY 클러스터가 설정되어 있습니다. 이 클러스터는 세 큐 관리자를 포함하며 LONDON 및 NEWYORK은 둘 다 전체 저장소를 보유하고 PARIS는 부분 저장소를 보유하고 INVENTQ에서 메시지를 넣습니다. 인벤토리 애플리케이션은 NEWYORK 큐 관리자에 연결된 New York의 시스템에서 실행됩니다. 애플리케이션은 INVENTQ 큐에 메시지가 도착하면 구동됩니다.
- 새 저장소가 Los Angeles에서 설정됩니다. 추가 용량을 제공하기 위해 New York뿐만 아니라 Los Angeles에서도 인벤토리 시스템을 실행하려고 합니다. 새 큐 관리자는 New York에 비해 두 배 많은 메시지를 처리할 수 있습니다.

이 태스크 정보

다음 단계에 따라 큐를 호스팅하는 보다 강력한 큐 관리자를 추가하십시오.

프로시저

1. LOSANGELES가 처음 참조할 전체 저장소를 결정하십시오.
2. 클러스터 내 모든 큐 관리자는 클러스터에 대한 정보를 수집하기 위해 전체 저장소 중 어느 하나를 참조해야 합니다. 그러면 자체 부분 저장소를 빌드합니다. 저장소 선택이 특별히 중요한 것은 아닙니다. 이 예에서는 NEWYORK을 선택합니다. 새 큐 관리자는 클러스터에 조인하고 나면 두 저장소와 모두 통신합니다.

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from LOSANGELES to repository at NEWYORK')
```

3. 큐 관리자 LOSANGELES에서 CLUSRCVR 채널을 정의하십시오.
클러스터 내 모든 큐 관리자는 메시지를 수신할 수 있는 클러스터 수신자 채널을 정의해야 합니다. LOSANGELES에서 다음을 정의하십시오.

```
DEFINE CHANNEL(INVENTORY.LOSANGELES) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LOSANGELES.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager LOSANGELES')
CLWLWGT(2)
```

클러스터 수신자 채널은 클러스터 INVENTORY에 있는 다른 큐 관리자로부터 메시지를 수신할 큐 관리자의 가용성을 알립니다. CLWLWGT를 2로 설정하면 Los Angeles 큐 관리자가 New York에 비해 두 배 많은 인벤토리 메시지를 가져오도록 보장합니다(NEWYORK에 대한 채널이 1로 설정된 경우).

4. 큐 관리자 NEWYORK에서 CLUSRCVR 채널을 대체하십시오.
Los Angeles 큐 관리자가 New York에 비해 두 배 많은 인벤토리 메시지를 가져오도록 하십시오. 클러스터 수신자 채널의 정의를 대체하십시오.

```
ALTER CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) CLWLWGT(1)
```

5. 메시지 연관관계에 대해 인벤토리 애플리케이션을 검토하십시오.
계속하기 전에 메시지 처리 순서에 따라 인벤토리 애플리케이션에 종속 항목이 없는지 확인하십시오.
6. Los Angeles에 있는 시스템에 인벤토리 애플리케이션을 설치하십시오.
7. 클러스터 큐 INVENTQ를 정의하십시오.

이미 NEWYORK 큐 관리자가 호스팅하고 있는 INVENTQ 큐도 LOSANGELES가 호스팅합니다. 다음과 같이 이 큐를 LOSANGELES 큐 관리자에서 정의하십시오.

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

Now that you have completed all the definitions, if you have not already done so start the channel initiator on WebSphere MQ for z/OS. 모든 플랫폼의 큐 관리자 LOSANGELES에서 리스너 프로그램을 시작 하십시오. 리스너 프로그램은 수신되는 네트워크 요청을 청취하고 필요한 경우 클러스터 수신자 채널을 시작 합니다.

결과

249 페이지의 『큐를 호스팅하는 더 강력한 큐 관리자 추가』에서는 이 태스크에 의해 설정되는 클러스터를 보여줍니다.

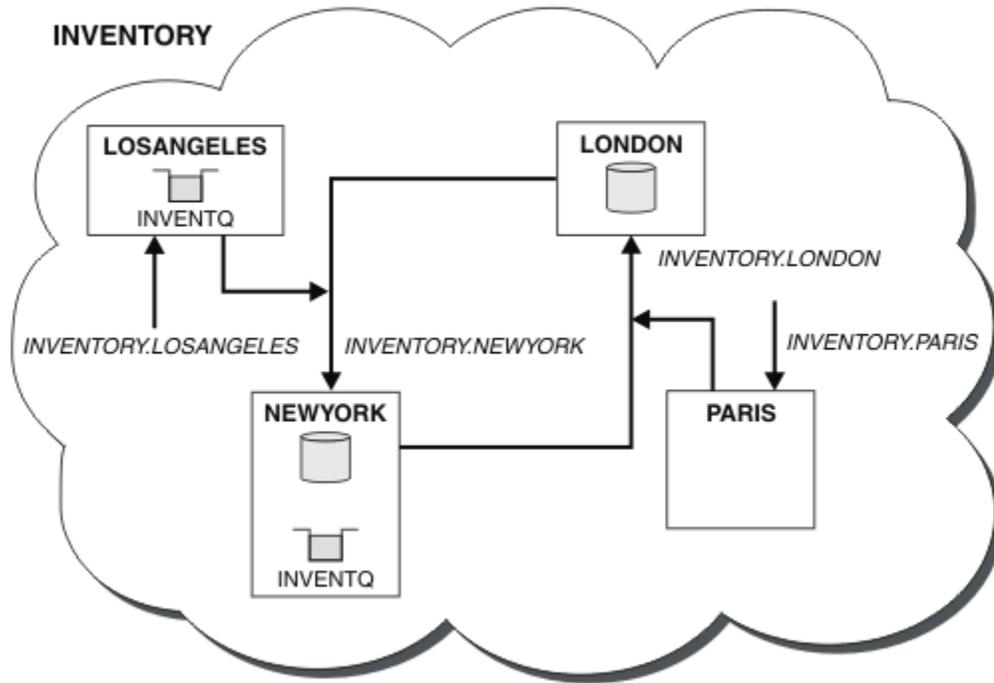


그림 54. 네 개의 큐 관리자가 있는 INVENTORY 클러스터

클러스터에 대한 이러한 수정은 큐 관리자 LONDON 및 PARIS를 변경하지 않고도 수행할 수 있습니다. 이러한 큐 관리자의 저장소는 메시지를 LOSANGELES의 INVENTQ에 송신하기 위해 필요한 정보로 자동 업데이트됩니다.

다음에 수행할 작업

INVENTQ 큐 및 인벤토리 애플리케이션은 클러스터에 있는 두 개의 큐 관리자에서 호스팅됩니다. 이 구성은 두 큐 관리자의 가용성을 증가시키고 메시지 처리 속도를 높이고 두 큐 관리자 사이에 워크로드를 분산시킬 수 있습니다. LOSANGELES 또는 NEWYORK에 의해 INVENTQ에 놓여진 메시지는 가능할 때마다 로컬 큐 관리자에서 인스턴스에 의해 처리됩니다. LONDON 또는 PARIS에 의해 놓여진 메시지가 LOSANGELES 또는 NEWYORK로 라우트되며 두 배 많은 메시지가 LOSANGELES로 송신됩니다.

애플리케이션 프로그래밍 및 클러스터

동일한 큐의 다중 인스턴스를 이용하기 위해 프로그래밍 변경을 작성하지 않아도 됩니다. 그러나 일부 프로그램은 메시지의 시퀀스가 큐의 동일 인스턴스로 송신되지 않으면 올바르게 작동하지 않습니다.

애플리케이션은 MQOPEN 호출을 사용하여 큐를 열 수 있습니다. 애플리케이션은 MQPUT 호출을 사용하여 메시지를 열린 큐에 넣습니다. 애플리케이션은 MQPUT1 호출을 사용하여 아직 열리지 않은 큐에 단일 메시지를 넣을 수 있습니다.

동일 큐의 다중 인스턴스가 있는 클러스터를 설정하는 경우 특정 애플리케이션 프로그래밍 고려사항이 없습니다. 그러나 클러스터링의 워크로드 관리 측면에서 이득을 얻기 위해 애플리케이션을 수정해야 할 수도 있습니다. 동일한 큐의 다중 정의가 있는 네트워크를 설정할 경우 메시지 연관관계에 대해 애플리케이션을 검토하십시오.

예를 들어, 질문과 응답 양식으로 그 사이에서 플로우하는 일련의 메시지에 의존하는 두 애플리케이션이 있다고 가정하십시오. 질문을 송신한 큐 관리자에게 응답이 돌아가게 하려고 할 수 있습니다. 워크로드 관리 루틴이 응답 큐의 사본을 호스팅하는 어떤 큐 관리자에게도 메시지를 송신하지 않는 것이 중요합니다.

메시지가 순서대로 처리하도록 요구하는 애플리케이션(예: 순서대로 검색되어야 하는 메시지의 배치를 송신하는 데이터베이스 복제 애플리케이션)이 있을 수 있습니다. 세그먼트화된 메시지의 사용은 연관관계 문제점의 원인이 될 수도 있습니다.

로컬 또는 리모트 버전의 대상 큐 열기

큐 관리자가 대상 큐의 로컬 버전을 사용할지 또는 리모트 버전을 사용할지 선택하는 방식을 알아야 합니다.

1. 큐 관리자는 로컬 버전 대상 큐를 열어 메시지를 읽거나 큐의 속성을 설정합니다.
2. 큐 관리자는 다음 조건 중 최소한 하나 이상이 true이면 대상 큐의 인스턴스를 열어 메시지를 기록합니다.
 - 로컬 버전 대상 큐는 존재하지 않습니다.
 - 큐 관리자는 ALTER QMGR에서 CLWLUSEQ(ANY) 를 지정합니다.
 - 큐 관리자의 큐가 CLWLUSEQ(ANY)를 지정합니다.

메시지 연관관계 핸들링

메시지 연관관계는 우수한 프로그래밍 설계에는 거의 포함되지 않습니다. 클러스터링을 완전하게 사용하려면 메시지 연관관계를 제거해야 합니다. 메시지 연관관계를 제거할 수 없는 경우 같은 채널을 사용하여 관련 메시지가 같은 큐 관리자에 전달되도록 강제 실행할 수 있습니다.

메시지 연관관계가 포함된 애플리케이션이 있는 경우 클러스터 사용을 시작하기 전에 이 연관관계를 제거하십시오.

메시지 연관관계를 제거하면 애플리케이션의 가용성을 향상시킵니다. 애플리케이션이 메시지 연관관계가 있는 메시지의 배치를 큐 관리자에게 송신합니다. 큐 관리자가 배치의 일부만 수신한 경우에는 실패합니다. 송신 큐 관리자가 추가 메시지를 송신하려면 먼저 불완전한 메시지 배치가 복구되어 처리될 때까지 대기해야 합니다.

메시지 연관관계를 제거하면 애플리케이션의 확장성도 향상시킵니다. 연관관계가 있는 메시지 배치는 후속 메시지를 대기하는 동안 목적지 큐 관리자에서 자원을 잠글 수 있습니다. 이러한 자원은 다른 애플리케이션이 해당 작업을 수행하지 못하도록 장기간 동안 잠금 상태로 남아 있을 수 있습니다.

또한 메시지 연관관계는 클러스터 워크로드 관리 루틴이 최적의 큐 관리자를 선택하지 못하도록 합니다.

연관관계를 제거하려면 다음과 같은 가능성을 고려하십시오.

- 메시지로 상태 정보 운반
- 모든 큐 관리자가 액세스할 수 있는 비휘발성 스토리지(예: Db2 데이터베이스)에서 상태 정보 유지보수
- 둘 이상의 큐 관리자가 액세스할 수 있도록 읽기 전용 데이터 복제

메시지 연관관계를 제거하기 위해 애플리케이션을 수정하는 것이 적절하지 않은 경우 이 문제점에 대해 가능한 솔루션이 다수 있습니다.

MQOPEN 호출에서 특정 목적지의 이름 지정

MQOPEN 호출 시마다 리모트 큐 이름 및 큐 관리자 이름을 지정하면 해당 오브젝트 핸들을 사용하여 큐에 넣어진 모든 메시지가 로컬 큐 관리자일 수도 있는 같은 큐 관리자로 이동합니다.

MQOPEN 호출 시마다 리모트 큐 이름 및 큐 관리자 이름을 지정하는 것에는 다음과 같은 단점이 있습니다.

- 워크로드 밸런싱이 수행되지 않습니다. 클러스터 워크로드 밸런싱의 이점을 이용하지 않습니다.
- 대상 큐 관리자가 리모트이고 이 큐 관리자에 대해 둘 이상의 채널이 있을 경우 메시지는 다른 라우트를 선택할 수 있고 메시지 시퀀스는 보존되지 않습니다.
- 큐 관리자에 목적지 큐 관리자와 같은 이름을 가지는 전송 큐에 대한 정의가 있는 경우 메시지는 클러스터 전송 큐가 아닌 해당 전송 큐로 이동합니다.

응답 대상 큐 관리자 필드에 큐 관리자 이름 리턴

배치의 첫 번째 메시지를 수신하는 큐 관리자가 해당 이름을 해당 응답으로 리턴할 수 있게 합니다. 이는 메시지 디스크립터의 ReplyToQMGr 필드를 사용하여 수행됩니다. 그러면 송신 측 큐 관리자가 응답 대상 큐 관리자 이름을 추출하고 이를 모든 후속 메시지에 지정할 수 있습니다.

응답의 ReplyToQMGr 정보를 사용하는 것에는 다음과 같은 단점이 있습니다.

- 요청 큐 관리자는 첫 번째 메시지에 대한 응답을 대기해야 합니다.
- 후속 메시지를 송신하기 전에 ReplyToQMGr 정보를 찾고 사용하기 위한 추가 코드를 작성해야 합니다.
- 큐 관리자에 대한 라우트가 둘 이상 있는 경우 메시지 시퀀스가 보존되지 않을 수 있습니다.

MQOPEN 호출에서 MQOO_BIND_ON_OPEN 옵션을 설정하십시오.

MQOPEN 호출에서 MQOO_BIND_ON_OPEN 옵션을 사용하여 모든 메시지가 같은 목적지에 들어가도록 강제 실행하십시오. 클러스터에서 메시지 그룹을 사용하여 그룹의 모든 메시지가 동일한 대상에서 처리되도록 하려면 MQOO_BIND_ON_OPEN 또는 MQOO_BIND_ON_GROUP 를 지정해야 합니다.

큐를 열고 MQOO_BIND_ON_OPEN을 지정하여 이 큐로 송신되는 모든 메시지가 동일한 큐 인스턴스로 송신되도록 강제 실행합니다. MQOO_BIND_ON_OPEN은 모든 메시지를 같은 큐 관리자와 같은 라우트에 바인드합니다. 예를 들어, 동일한 목적지에 대한 IP 라우트 및 NetBIOS 라우트가 있는 경우 큐가 열릴 때 이들 중 하나가 선택되며 이러한 선택은 확보된 오브젝트 핸들을 사용하여 같은 큐에 들어가는 모든 메시지에 대해 준수됩니다.

MQOO_BIND_ON_OPEN을 지정하여 모든 메시지가 같은 목적지로 라우트되도록 강제 실행합니다. 따라서 메시지 연관관계가 있는 애플리케이션은 중단되지 않습니다. 목적지가 사용 불가능한 경우에는 사용 가능해질 때까지 메시지가 전송 큐에 남게 됩니다.

MQOO_BIND_ON_OPEN은 큐를 열 때 큐 관리자 이름이 오브젝트 디스크립터에서 지정되는 경우에도 적용됩니다. 이름 지정된 큐 관리자에 대한 라우트가 둘 이상 있을 수 있습니다. 예를 들어, 다중 네트워크 경로가 있거나 다른 큐 관리자가 알리어스를 정의했을 수 있습니다. MQOO_BIND_ON_OPEN을 지정하는 경우 큐가 열리면 라우트가 선택됩니다.

참고: 이 방법이 권장되는 기술입니다. 그러나 이 방법은 큐 관리자가 클러스터 큐에 대한 알리어스를 알리는 멀티홉 구성에서는 작동하지 않습니다. 또한 애플리케이션이 여러 메시지 그룹에 대해 같은 큐 관리자에서 다른 큐를 사용하는 상황에서도 도움이 되지 않습니다.

MQOPEN 호출에서 MQOO_BIND_ON_OPEN을 지정하는 것에 대한 대안은 큐 정의를 수정하는 것입니다. 큐 정의에서 DEFBIND(OPEN)을 지정하고 MQOPEN 호출의 DefBind 옵션이 MQOO_BIND_AS_Q_DEF로 기본값 설정되도록 하십시오.

MQOPEN 호출에서 MQOO_BIND_ON_GROUP 옵션을 설정하십시오.

Force all your messages in a group to be put to the same destination using the MQOO_BIND_ON_GROUP option on the MQOPEN call. 클러스터에서 메시지 그룹을 사용하여 그룹의 모든 메시지가 동일한 대상에서 처리되도록 하려면 MQOO_BIND_ON_OPEN 또는 MQOO_BIND_ON_GROUP 를 지정해야 합니다.

큐를 열고 MQOO_BIND_ON_GROUP을 지정하여 이 큐로 송신되는 모든 메시지가 동일한 큐 인스턴스로 송신되도록 강제 실행합니다. MQOO_BIND_ON_GROUP은 그룹 내 모든 메시지를 같은 큐 관리자와 같은 라우트에 바인드합니다. 예를 들어, 동일한 목적지에 대한 IP 라우트 및 NetBIOS 라우트가 있는 경우 큐가 열릴 때 이들 중 하나가 선택되며 이러한 선택은 확보된 오브젝트 핸들을 사용하여 같은 큐에 들어가는 그룹 내 모든 메시지에 대해 준수됩니다.

MQOO_BIND_ON_GROUP을 지정하여 그룹 내 모든 메시지가 같은 목적지로 라우트되도록 강제 실행합니다. 따라서 메시지 연관관계가 있는 애플리케이션은 중단되지 않습니다. 목적지가 사용 불가능한 경우에는 사용 가능해질 때까지 메시지가 전송 큐에 남게 됩니다.

MQOO_BIND_ON_GROUP은 큐를 열 때 큐 관리자 이름이 오브젝트 디스크립터에서 지정되는 경우에도 적용됩니다. 이름 지정된 큐 관리자에 대한 라우트가 둘 이상 있을 수 있습니다. 예를 들어, 다중 네트워크 경로가 있거나 다른 큐 관리자가 알리어스를 정의했을 수 있습니다. MQOO_BIND_ON_GROUP을 지정하는 경우 큐가 열리면 라우트가 선택됩니다.

MQOO_BIND_ON_GROUP이 유효하려면 MQPUT에서 MQPMO_LOGICAL_ORDER Put 옵션을 포함해야 합니다. 메시지의 MQMD에서 **GroupId**를 MQGI_NONE으로 설정할 수 있으며, 메시지의 MQMD **MsgFlags** 필드 내에 다음을 메시지 플래그를 포함해야 합니다.

- 그룹의 마지막 메시지: MQMF_LAST_MSG_IN_GROUP
- 그룹의 기타 모든 메시지: MQMF_MSG_IN_GROUP

MQOO_BIND_ON_GROUP 가 지정되었지만 메시지가 그룹화되지 않은 경우 동작은 MQOO_BIND_NOT_FIXED 와 동등합니다.

참고: 이는 그룹 내 메시지가 같은 목적지에 송신되도록 보장하기 위해 권장되는 기술입니다. 그러나 큐 관리자가 클러스터 큐에 대한 별명을 알리는 다중 홉 구성에서는 작동하지 않습니다.

MQOPEN 호출에서 MQOO_BIND_ON_GROUP을 지정하는 것에 대한 대안은 큐 정의를 수정하는 것입니다. 큐 정의에서 DEFBIND(GROUP)을 지정하고 MQOPEN 호출의 DefBind 옵션이 MQOO_BIND_AS_Q_DEF로 기본값 설정되도록 하십시오.

사용자 정의된 클러스터 워크로드 엑시트 프로그램 작성

애플리케이션을 수정하는 대신 클러스터 워크로드 엑시트 프로그램을 작성하여 메시지 연관관계 문제점을 피할 수 있습니다. 클러스터 워크로드 엑시트 프로그램을 작성하는 것은 쉽지 않고 권장되는 솔루션이 아닙니다. 프로그램은 메시지의 콘텐츠를 검사하여 연관관계를 인식하도록 설계되어야 합니다. 연관관계를 인식하면 프로그램이 모든 관련 메시지를 동일한 큐 관리자에 라우트하기 위한 워크로드 관리 유틸리티를 강제 실행합니다.

클러스터링: 우수 사례

클러스터는 큐 관리자를 상호 연결하기 위한 메커니즘을 제공합니다. 이 절에 설명된 우수 사례는 테스트 및 고객의 피드백에 기반을 둔 것입니다.

성공적인 클러스터 설정은 양질의 애플리케이션 관리 및 네트워크 설계와 같은 IBM WebSphere MQ 기본사항에 대한 완벽한 이해와 우수한 계획에 의존합니다. 계속하기 전에 나열된 관련 항목에 있는 정보를 숙지해야 합니다.

관련 개념

[클러스터링](#)

[상호통신의 개념](#)

[클러스터 작동 방법](#)

클러스터링: 중첩 클러스터에 대한 특수 고려사항

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

클러스터 소유권

다음 정보를 읽기 전에 중첩 클러스터를 숙지하십시오. 필요한 정보는 [163 페이지의 『중첩 클러스터』](#) 및 [235 페이지의 『클러스터 간 메시지 경로 구성』](#)의 내용을 참조하십시오.

중첩 클러스터로 이루어진 시스템을 구성하고 관리할 때에는 다음 사항을 준수하는 것이 가장 좋습니다.

- IBM WebSphere MQ 클러스터가 이전에 설명된 대로 '느슨하게 결합'되어 있어도 클러스터를 단일 관리 단위로 간주하는 것이 유용합니다. 개별 큐 관리자의 정의 간 상호작용이 클러스터의 원활한 기능에 중요하기 때문에 이 개념이 사용됩니다. 예를 들어, 워크로드 밸런스가 조절된 클러스터 큐를 사용할 때에는 단일 관리자나 팀이 클러스터 전반에 퍼진 정의에 따라 달라지는 메시지의 가능한 전체 목적지 세트를 이해하는 것이 중요합니다. 보다 구체적으로는 클러스터 송신자/수신자 채널 쌍이 도처에서 호환 가능해야 합니다.
- 여러 클러스터(개별 팀/개인이 관리할)가 모이는 이전 개념을 고려할 경우 게이트웨이 큐 관리자의 관리를 제어할 명확한 정책을 적절하게 보유하는 것이 중요합니다.
- 중첩 클러스터를 단일 네임스페이스로 처리하면 유용합니다. 즉, 채널 이름과 큐 관리자 이름이 단일 클러스터 전반에서 고유해야 합니다. 전체 토폴로지에서 이름이 고유하면 관리가 훨씬 용이합니다. 최상의 방법은 적합한 이름 지정 규칙을 따르는 것입니다. [162 페이지의 『클러스터 이름 지정 규칙』](#)에 가능한 규칙이 설명되어 있습니다.
- 때로 관리 및 시스템 관리 협업은 필수적/불가피합니다. 예를 들어, 중첩시켜야 하는 여러 다른 클러스터를 소유한 조직 간의 협업이 있습니다. 누가 무엇을 소유하는지 이해하고 시행 가능한 규칙이 있으면 중첩 클러스터 시 클러스터링이 원활하게 실행됩니다.

중첩 클러스터: 게이트웨이

일반적으로 단일 클러스터는 다중 클러스터보다 관리가 더 용이합니다. 따라서 많은 수의 작은 클러스터를 작성(예를 들어, 모든 애플리케이션에 하나씩)하는 방식은 대개 피하는 방법입니다.

하지만 서비스 클래스를 제공하려면 중첩 클러스터를 구현할 수 있습니다. 예를 들면, 다음과 같습니다.

- 집중형 클러스터가 있는 경우, 더 작은 클러스터가 발행/구독을 위한 것입니다. 자세한 정보는 [시스템 크기 조절 방법을 참조하십시오](#).
- 일부 큐 관리자가 다른 팀에 의해 관리되는 경우. 자세한 정보는 이전 [섹션 254 페이지의 『클러스터 소유권』](#)을 참조하십시오.
- 조직 또는 지리적 관점에서 타당할 경우.
- 동일한 클러스터가 이름 분석에 대해 작업하는 경우(예: 기존 클러스터에서 SSL 또는 TLS를 구현할 때).

중첩 클러스터에 따른 보안 혜택은 없습니다. 다른 두 팀에서 관리하는 클러스터를 중첩시키면 토폴로지 및 팀이 효과적으로 조인됩니다. 모든:

- 이러한 클러스터에 광고된 이름은 다른 클러스터에 액세스 가능합니다.
- 한 클러스터에 광고된 이름을 다른 클러스터에 광고하여 적합한 메시지를 빼낼 수 있습니다.
- 게이트웨이에 인접한 큐 관리자의 광고되지 않은 오브젝트를 게이트웨이가 멤버인 클러스터로부터 분석할 수 있습니다.

네임스페이스는 두 클러스터 모두의 유니온이며 단일 네임스페이스로 처리해야 합니다. 따라서 중첩 클러스터의 소유권은 두 클러스터의 모든 관리자가 공유합니다.

시스템에 여러 클러스터가 있으면 한 클러스터의 큐 관리자에서 다른 클러스터의 큐 관리자로 메시지를 라우팅하기 위한 요구사항이 있을 수도 있습니다. 이 경우 다수의 클러스터를 일정한 방식으로 서로 연결시켜야 합니다. 우수한 패턴은 클러스터 간 게이트웨이 큐 관리자를 사용하는 것입니다. 이렇게 배열하면 관리하기 어려운 포인트-투-포인트 채널의 메시지가 빌드되지 않고 보안 정책으로서 이러한 문제를 관리할 좋은 위치가 제공됩니다. 이 배열을 완성하는 두 가지 특징적 방법이 있습니다.

1. 두 번째 클러스터 수신자 정의를 사용하여 하나(이상)의 큐 관리자를 두 클러스터 모두에 두십시오. 이 배열은 관리 정의 수가 더 적지만 이전에 언급한 것처럼, 중첩 클러스터의 소유권을 두 클러스터의 모든 관리자가 공유함을 의미합니다.
2. 일반적인 포인트-투-포인트 채널을 사용하여 클러스터 1의 큐 관리자와 클러스터 2의 큐 관리자가 쌓이 되게 하십시오.

둘 중 어느 경우든, 다양한 도구를 사용하여 트래픽을 적절하게 라우팅할 수 있습니다. 특히, 큐 또는 큐 관리자 알리어스를 사용하여 다른 클러스터로 라우팅할 수 있으며, 비어 있는 **RQMNAME** 특성이 있는 큐 관리자 알리어스는 원하는 경우 워크로드 밸런싱을 다시 구동합니다.

관련 개념

[162 페이지의 『클러스터 이름 지정 규칙』](#)

큐 관리자가 속한 클러스터를 식별하는 이름 지정 규칙을 사용하여 동일한 클러스터의 큐 관리자 이름을 지정할 것을 고려하십시오. 채널 이름에 유사한 이름 지정 규칙을 사용하고 채널 특성을 설명하도록 확장하십시오.

클러스터링: 토폴로지 설계 고려사항

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

사용자 애플리케이션과 내부 관리 프로세스가 위치할 곳을 미리 생각해 두면 많은 문제점을 피하고 나중에 최소화할 수 있습니다. 이 주제는 성능을 개선하고 유지보수 태스크를 클러스터 배율로 단순화할 수 있는 설계 의사결정에 대한 정보를 포함합니다.

- [256 페이지의 『클러스터링 인프라의 성능』](#)
- [256 페이지의 『전체 저장소』](#)
- [257 페이지의 『애플리케이션에 전체 저장소의 큐를 사용해야 합니까?』](#)
- [257 페이지의 『채널 정의 관리』](#)
- [258 페이지의 『다중 채널에 대한 워크로드 밸런싱』](#)

클러스터링 인프라의 성능

애플리케이션이 클러스터의 큐 관리자에서 큐를 열려고 시도하면 클러스터에서 큐가 존재하는 위치를 알 수 있도록 큐 관리자는 해당 큐의 전체 저장소에 관심을 등록합니다. 큐 위치 또는 구성에 대한 업데이트는 전체 저장소를 통해 관심 있는 큐 관리자에 자동으로 송신됩니다. 이러한 관심 등록을 내부적으로 구독이라 합니다(이 구독은 IBM WebSphere MQ의 발행/구독 메시징에 사용되는 IBM WebSphere MQ 구독과 동일하지 않음).

클러스터에 대한 모든 정보는 모든 전체 저장소를 통과합니다. 따라서 전체 저장소는 관리 메시지 트래픽을 위해 항상 클러스터에 사용됩니다. 이 구독을 관리할 때의 많은 시스템 자원 사용과 구독 및 결과적인 구성 메시지의 전송은 클러스터링 인프라에 상당한 로드를 야기시킬 수 있습니다. 이 로드가 이해되고 가능한 위치에서 최소화되는지 확인할 때 고려할 많은 사항이 있습니다.

- 클러스터 큐를 사용하는 개별 큐 관리자가 더 많을수록 시스템에 더 많은 구독이 있으므로 특히 전체 저장소 큐 관리자에서 변경이 발생하고 관심 있는 구독을 알려야 할 때에는 관리 오버헤드가 더 커집니다. 불필요한 트래픽과 전체 저장소 로드를 최소화하기 위한 한 가지 방법은 유사한 애플리케이션(즉, 동일한 큐에 대해 작업하는 애플리케이션)을 더 작은 수의 큐 관리자에 연결시키는 것입니다.
- 성능에 영향을 미치는 시스템의 구독 수 외에도, 클러스터된 큐 구성의 빈번한 변경과 같은 클러스터된 오브젝트의 구성 변화율은 성능에 영향을 줄 수 있습니다.
- 큐 관리자가 다중 클러스터의 멤버(즉, 중첩 클러스터 시스템의 일부)일 때에는 동일한 큐 관리자가 둘 이상의 클러스터에 대한 전체 저장소일지라도 큐에서 등록한 관심으로 인해 멤버로 속해 있는 각 클러스터에 대한 구독이 발생합니다. 이 배열은 시스템의 로드를 증가시키며 단일 클러스터보다 여러 중첩 클러스터가 필요할지 여부를 고려하는 한 가지 이유입니다.
- 애플리케이션 메시지 트래픽(즉, IBM WebSphere MQ 애플리케이션이 클러스터 큐에 보내는 메시지)은 목적지 큐 관리자에 도달하기 위해 전체 저장소를 거치지 않습니다. 이 메시지 트래픽은 클러스터에 메시지를 넣는 큐 관리자와 클러스터 큐가 존재하는 큐 관리자 간에 직접 송신됩니다. 따라서 전체 저장소 큐 관리자가 우연히 언급한 두 개의 큐 관리자 중 하나인 경우가 아니면 전체 저장소 큐 관리자에 관하여 애플리케이션 메시지 트래픽을 높은 비율로 유지할 필요가 없습니다. 이러한 이유로, 클러스터링 인프라 로드가 중요한 클러스터의 애플리케이션 메시지 트래픽에는 전체 저장소 큐 관리자를 사용하지 않는 것이 좋습니다.

전체 저장소

저장소는 클러스터의 멤버인 큐 관리자에 대한 정보의 컬렉션입니다. 클러스터에서 모든 큐 관리자에 대한 전체 정보 세트를 호스팅하는 큐 관리자에 전체 저장소가 있습니다. 전체 저장소와 부분 저장소에 대한 자세한 정보는 149 페이지의 『클러스터 저장소』의 내용을 참조하십시오.

신뢰할 수 있고 가능한 한 가용성이 높으며 단일 장애점을 피해야 하는 서버에 전체 저장소를 보유해야 합니다. 클러스터는 두 개의 전체 저장소를 갖도록 설계되어야 합니다. 한 전체 저장소에 장애가 있어도 클러스터가 여전히 작동할 수 있습니다.

클러스터의 큐 관리자가 수행한 클러스터 자원 업데이트의 세부사항(예를 들어, 클러스터된 큐)은 해당 큐 관리자에서 이 클러스터의 많아야 두 개인 전체 저장소로(또는 클러스터에 전체 저장소 큐 관리자가 하나만 있으면 전체 저장소로) 송신됩니다. 이 전체 저장소는 정보를 보유하며 관심을 표시하는(즉, 구독하는) 클러스터의 큐 관리자에 이를 전파합니다. 클러스터의 각 멤버에 클러스터 자원의 최신 보기가 있는지 확인하려면 각 큐 관리자가 언제든지 적어도 하나의 전체 저장소 큐 관리자와 통신할 수 있어야 합니다.

큐 관리자가 어떠한 이유로 전체 저장소와 통신할 수 없는 경우 이미 캐싱된 정보 레벨에 따라 일정 기간 클러스터에서 계속하여 작동할 수 있지만 이전에 사용하지 않은 클러스터 자원에 대한 액세스 또는 새 업데이트는 사용할 수 없습니다.

이러한 이유로 항상 두 개의 전체 저장소를 사용 가능하게 유지해야 합니다. 그렇다고 해서 이 배열이 전체 저장소가 없는 잠시 동안 클러스터가 적절하게 작동하기 때문에 극단적인 조치를 취해야 함을 의미하는 것은 아닙니다.

클러스터 정보의 가용성 외에, 두 개의 전체 저장소 큐 관리자를 보유해야 하는 또 다른 이유가 있습니다. 이 이유는 전체 저장소 캐시에 보유된 클러스터 정보가 복구 용도로 두 위치에 존재하게 하는 것입니다. 전체 저장소가 하나만 있고 클러스터에 대한 정보를 잃으면 클러스터를 작동시키기 위해 클러스터 내 모든 큐 관리자에 대한 수동 개입이 필요합니다. 하지만 두 개의 전체 저장소가 있으면 두 개의 전체 저장소에서 항상 정보가 발행되고 구독되므로 최소의 노력으로 실패한 전체 저장소를 복구할 수 있습니다.

- 두 개의 전체 저장소 클러스터 설계에서는 클러스터의 사용자에게 영향을 주지 않고 전체 저장소 큐 관리자에게서 유지보수를 수행할 수 있습니다. 클러스터는 한 저장소만 있어도 계속해서 기능하므로 저장소가 작동 중지될 경우 유지보수가 적용되고 한 번에 다시 백업됩니다. 두 번째 전체 저장소가 가동 중단되어도 실행 애플리케이션에 최소 3일 동안은 영향이 미치지 않습니다.
- 지리적 이유로 로컬 전체 저장소를 사용하는 것처럼, 세 번째 저장소를 사용해야 하는 타당한 이유가 없으면 두 개의 저장소 설계를 사용하십시오. 세 개의 전체 저장소를 사용한다는 것은 현재 사용 중인 두 개를 결코 알지 못하며 여러 워크로드 관리 매개변수 간의 상호작용으로 인한 관리 문제가 있을 수 있음을 의미합니다. 전체 저장소를 셋 이상은 보유하지 않는 것이 좋습니다.
- 여전히 가용성을 개선할 필요가 있으면 전체 저장소 큐 관리자를 다중 인스턴스 큐 관리자로서 호스트하거나 플랫폼별 고가용성 지원을 사용하여 가용성을 개선할 것을 고려하십시오.
- 수동으로 정의된 클러스터 송신자 채널과 모든 전체 저장소 큐 관리자가 완전히 상호 연결되어 있어야 합니다. 일부 정당한 사유로 클러스터에 셋 이상의 전체 저장소를 사용할 때에는 특히 주의해야 합니다. 이 경우 하나 이상의 채널이 빈번히 누락될 수 있으며 이는 바로 드러나지 않습니다. 완전히 상호 연결되지 않은 경우에는 자주 발생하는 문제점을 진단하기 어렵습니다. 일부 전체 저장소는 모든 저장소 데이터를 보유하지 않아서 결과적으로 클러스터에 있는 큐 관리자가 연결하는 전체 저장소에 따라 클러스터의 보기가 상이하기 때문에 진단이 어렵습니다.

애플리케이션에 전체 저장소의 큐를 사용해야 합니까?

전체 저장소는 대부분 다른 큐 관리자와 유사하므로 전체 저장소의 애플리케이션 큐를 호스트하고 이 큐 관리자에 직접 애플리케이션을 연결시킬 수 있습니다. 애플리케이션에 전체 저장소의 큐를 사용해야 합니까?

일반적으로 허용된 대답은 "아니오"입니다. 이 구성이 가능하더라도 많은 고객은 이 큐 관리자를 전체 저장소 클러스터 캐시의 유지보수 전용으로 사용하는 것을 선호합니다. 어느 옵션을 사용할지 결정할 때 고려할 사항은 여기에 설명되어 있지만 궁극적으로는 클러스터 아키텍처가 환경의 특정 요구에 적합해야 합니다.

- 업그레이드: 일반적으로 IBM WebSphere MQ의 새 릴리스에서 새 클러스터 기능을 사용하려면 이 클러스터의 전체 저장소 큐 관리자를 먼저 업그레이드해야 합니다. 클러스터의 애플리케이션에 새 기능을 사용하려는 경우 같은 위치에 있는 많은 애플리케이션을 테스트하지 않고 전체 저장소(및 부분 저장소의 일부 서브세트)를 업데이트하는 것이 유용할 수 있습니다.
- 유지보수: 이와 유사한 방식으로 전체 저장소에 긴급 유지보수를 적용해야 하는 경우 애플리케이션 없이 **REFRESH** 명령으로 전체 저장소를 재시작하거나 새로 고칠 수 있습니다.
- 성능: 클러스터가 커지고 전체 저장소 클러스터 캐시 유지보수에 대한 요구가 늘어남에 따라 애플리케이션을 별도로 유지하면 시스템 자원에 대한 경합을 통해 애플리케이션 성능에 영향을 미칠 위험성이 감소합니다.
- 하드웨어 요구사항: 일반적으로 전체 저장소가 강력할 필요는 없습니다. 예를 들어, 가용성 기대치가 양호한 단순 UNIX 서버면 충분합니다. 또는 매우 크거나 끊임없이 변경되는 클러스터의 경우 전체 저장소 컴퓨터의 성능을 고려해야 합니다.
- 소프트웨어 요구사항: 요구사항은 대개 전체 저장소에서 애플리케이션 큐를 호스트하도록 선택하는 주된 이유입니다. 작은 클러스터에서 공동 위치는 전체적으로 더 작은 수의 큐 관리자/서버에 대한 요구사항을 의미하는 것일 수 있습니다.

채널 정의 관리

단일 클러스터 내에도 두 개의 큐 관리자 간에 다중 라우트를 제공하는 여러 채널 정의가 존재할 수 있습니다.

단일 클러스터 내에 병렬 채널을 사용하면 때로 이점이 있지만 이 설계 의사결정을 철저히 고려해야 합니다. 이 설계는 복잡도를 더하는 것 외에 채널 이용을 감소시켜서 성능을 저하시킬 수 있습니다. 테스트 시에 일반적으로 많은 메시지를 일정한 비율로 보내서 병렬 채널이 완전히 사용되기 때문에 이러한 상황이 발생합니다. 하지만 메시지 스트림이 일정하지 않은 실세계 조건에서는 워크로드 밸런싱 알고리즘으로 인해 메시지 플로우가 한 채널에서 다른 채널로 전환될 때 성능이 떨어집니다.

큐 관리자가 다중 클러스터의 멤버인 경우 각 클러스터마다 별도의 CLUSRCVR 채널을 정의하지 않고 클러스터 이름 목록과 함께 단일 채널 정의를 사용하는 옵션이 있습니다. 하지만 이 설정은 나중에 관리의 어려움을 유발할 수 있습니다. 예를 들어, SSL을 한 클러스터에 적용하고 두 번째 클러스터에는 적용하지 않을 경우를 고려해 보십시오. 별도의 정의를 작성하는 것이 바람직하며 [162 페이지의 『클러스터 이름 지정 규칙』](#)에 제안된 이름 지정 규칙이 이를 지원합니다.

다중 채널에 대한 워크로드 밸런싱

이 정보는 주제에 대한 고급 이해를 돕기 위한 것입니다. 이 정보를 사용하기 전에 알아야 하는 이 주제에 대한 기본 설명은 240 페이지의 『워크로드 관리에 클러스터 사용』, 워크로드 밸런싱 및 클러스터 워크로드 관리 알고리즘을 참조하십시오.

클러스터 워크로드 관리 알고리즘은 큰 도구 세트를 제공하지만 도구의 작동 및 상호작용 방식을 완전히 이해하지 않은 채로 도구를 함께 사용해서는 안 됩니다. 워크로드 밸런싱 프로세스에 채널이 얼마나 중요한지는 바로 드러나지 않을 수도 있습니다. 워크로드 관리 라운드로빈 알고리즘은 클러스터된 큐를 소유하는 큐 관리자에 대한 다중 클러스터 채널이 해당 큐의 다중 인스턴스로 처리되는 것처럼 작동합니다. 이 프로세스는 다음 예에 보다 자세히 설명되어 있습니다.

1. 클러스터에서 큐를 호스팅하는 두 개의 큐 관리자가 있습니다(QM1 및 QM2).
2. QM1에는 다섯 개의 클러스터 수신자 채널이 있습니다.
3. QM2에는 클러스터 수신자 채널이 하나만 있습니다.
4. When **MQPUT** or **MQOPEN** on QM3 chooses an instance, the algorithm is five times more likely to send the message to QM1 than to QM2.
5. 4 단계의 상황은 알고리즘이 QM1에 대한 모든 채널 및 QM2에 대한 단일 채널 전체의 라운드 로빈과 (5+1) 중에서 선택할 수 있는 6가지 옵션을 볼 수 있기 때문에 발생합니다.

감지하기 힘든 또 다른 작동은 로컬 큐 관리자에 우연히 한 인스턴스가 구성되어 있는 클러스터된 큐에 메시지를 넣을 때조차도 IBM WebSphere MQ가 로컬 클러스터 수신자 채널의 상태를 사용하여 큐의 로컬 인스턴스나 큐의 원격 인스턴스에 메시지를 넣을지 결정하는 경우입니다. 이 시나리오에서,

1. 메시지를 넣을 때 워크로드 관리 알고리즘은 개별 클러스터 큐를 보지 않고 이 목적지에 도달할 수 있는 클러스터 채널을 봅니다.
2. 로컬 목적지에 도달하기 위해 로컬 수신자 채널이 이 목록에 포함되어 있습니다(메시지를 보내는 데 사용되지 않지만).
3. 로컬 수신자 채널이 중지될 때 CLUSRCVR이 중지되지 않으면 워크로드 관리 알고리즘은 기본적으로 대체 인스턴스를 선호합니다. 목적지에 대한 여러 로컬 CLUSRCVR 인스턴스가 있고 최소 하나가 중지되지 않을 경우 로컬 인스턴스를 적합하다고 간주합니다.

클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

애플리케이션을 배치할 때에는 다른 애플리케이션과 공유하는 IBM WebSphere MQ 자원 및 공유하지 않는 자원을 선택합니다. 공유할 수 있는 여러 유형의 자원이 있으며 주된 자원은 서버 자체, 큐 관리자, 채널, 큐입니다. 별도의 큐, 채널, 큐 관리자 또는 심지어 서버를 개별 애플리케이션에 할당하여 공유 자원 수가 작은 애플리케이션을 구성하도록 선택할 수 있습니다. 이러한 경우에는 전체 시스템 구성이 더 커지고 복잡해집니다. IBM WebSphere MQ 클러스터를 사용하면 더 많은 서버, 큐 관리자, 큐 및 채널을 관리하는 복잡도가 감소되지만 다른 공유 자원인 클러스터 전송 큐(SYSTEM.CLUSTER.TRANSMIT.QUEUE)도 도입됩니다.

259 페이지의 그림 55은 SYSTEM.CLUSTER.TRANSMIT.QUEUE 공유의 중요성을 설명하는 대규모 IBM WebSphere MQ 배치 중 하나입니다. 다이어그램에서 Client App 애플리케이션은 CL1 클러스터의 QM2 큐 관리자에 연결됩니다. Client App의 메시지는 Server App 애플리케이션에서 처리합니다. 메시지는 CLUSTER2의 QM3 큐 관리자에 있는 Q1 클러스터 큐에서 Server App을(를) 통해 검색합니다. 클라이언트 및 서버 애플리케이션이 동일한 클러스터에 있지 않기 때문에 게이트웨이 큐 관리자 QM1이(가) 메시지를 전송합니다.

클러스터 게이트웨이를 구성하는 일반적인 방법은 게이트웨이 큐 관리자를 모든 클러스터의 멤버로 만드는 것입니다. 게이트웨이 큐 관리자에는 모든 클러스터의 클러스터 큐에 대한 클러스터된 알리어스 큐가 정의되어 있습니다. 클러스터된 큐 알리어스는 모든 클러스터에서 사용 가능합니다. 클러스터 큐 알리어스에 넣은 메시지는 게이트웨이 큐 관리자를 통해 올바른 목적지로 라우팅됩니다. 게이트웨이 큐 관리자는 클러스터링된 알리어스 큐에 송신된 메시지를 QM1의 공통 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 넣습니다.

허브 및 스포크 아키텍처에서는 클러스터 간의 모든 메시지가 게이트웨이 큐 관리자를 통과해야 합니다. 결과적으로 모든 메시지는 QM1, SYSTEM.CLUSTER.TRANSMIT.QUEUE의 단일 클러스터 전송 큐를 통해 플로워됩니다.

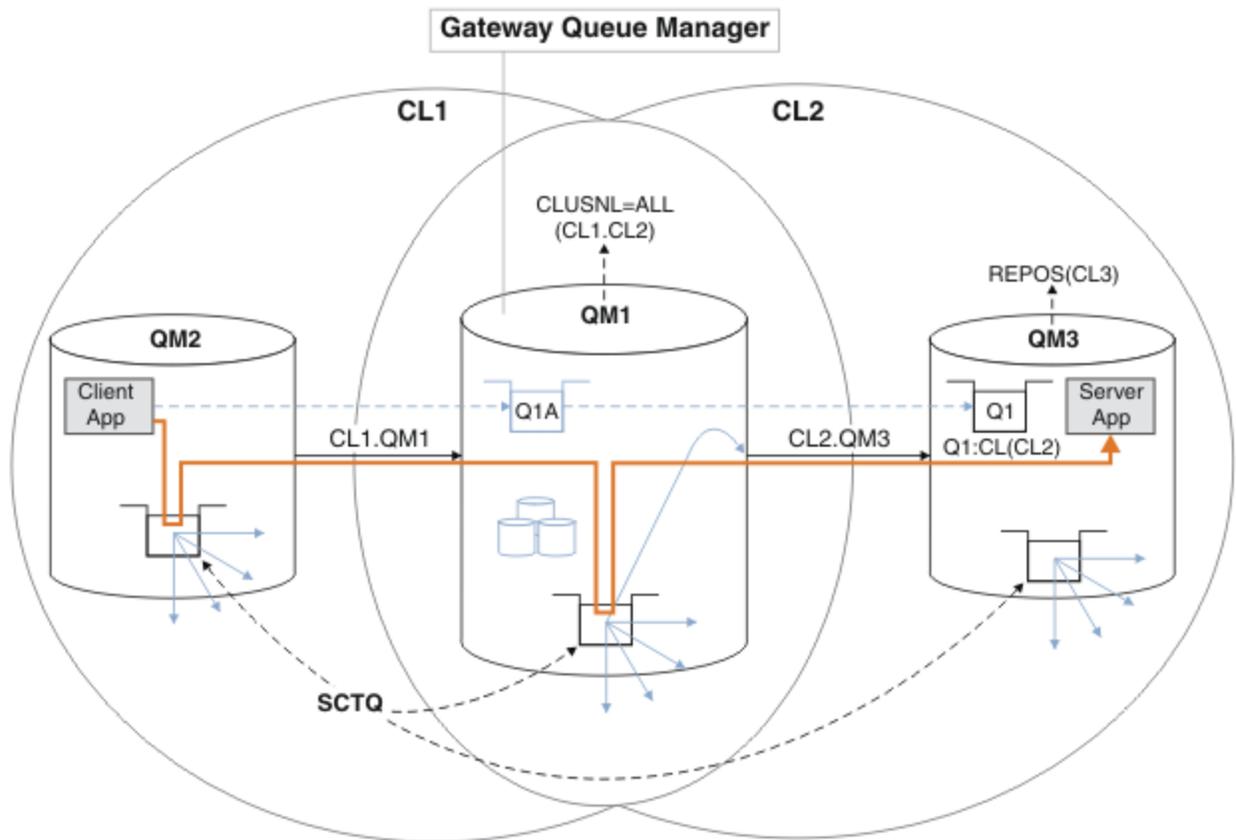
성능면에서 보면 단일 큐는 문제가 되지 않습니다. 공용 전송 큐는 일반적으로 성능 병목을 나타내지 않습니다. 게이트웨이의 메시지 처리량이 대부분 연결된 채널의 성능에 따라 판별됩니다. 처리량은 대개 큐의 수나 채널을 사용하는 큐의 메시지 수에 영향을 받지 않습니다.

다른 관점에서는, 다중 애플리케이션에 단일 전송 큐를 사용할 경우 문제가 있습니다.

- 한 목적지로 향하는 메시지 플로우를 다른 목적지로 향하는 메시지 플로우와 분리시킬 수 없습니다. 목적지가 다른 큐 관리자의 다른 클러스터에 있어도 전달 전에 메시지의 스토리지를 구분할 수 없습니다.

한 클러스터 목적지가 사용 불가능하게 되면 이 목적지에 대한 메시지가 단일 전송 큐에 빌드되어 결과적으로 메시지가 이 큐를 가득 채우게 됩니다. 전송 큐는 일단 가득 차면 어느 클러스터 목적지에 대한 전송 큐에도 메시지를 받지 않습니다.

- 다른 클러스터 목적지로의 메시지 전송을 모니터링하기 쉽지 않습니다. 모든 메시지가 단일 전송 큐에 있습니다. 전송 큐 용량을 표시해도 메시지가 모든 목적지로 전송되고 있는지 거의 표시되지 않습니다.



참고: 259 페이지의 그림 55의 화살표와 다음 그림은 여러 가지 유형입니다. 단색 화살표는 메시지 플로우를 표시합니다. 단색 화살표의 레이블은 메시지 채널 이름입니다. 회색 실선 화살표는 SYSTEM.CLUSTER.TRANSMIT.QUEUE에서 클러스터 송신자 채널로의 잠재적 메시지 플로우입니다. 검은색 파선은 레이블을 대상에 연결합니다. 회색 파선은 참조입니다(예를 들어, Client App의 MQOPEN 호출에서 클러스터 알리어스 큐 정의의 Q1A까지의 파선).

그림 55. IBM WebSphere MQ 클러스터를 사용하여 허브 및 스포크 아키텍처에 배치되는 클라이언트 서버 애플리케이션

259 페이지의 그림 55에서 Server App의 클라이언트가 Q1A 큐를 엽니다. 메시지는 QM2의 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 배치되고 QM1의 SYSTEM.CLUSTER.TRANSMIT.QUEUE(으)로 전송된 다음 QM3의 Q1(으)로 전송되어 Server App 애플리케이션이 수신합니다.

Client App의 메시지는 QM2 및 QM1의 시스템 클러스터 전송 큐를 통과합니다. 259 페이지의 [그림 55](#)의 목적은 메시지가 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 저장되지 않도록 클라이언트 애플리케이션에서 게이트웨이 큐 관리자의 메시지 플로우를 분리하는 것입니다. 클러스터된 다른 모든 큐 관리자에서 플로우를 분리시킬 수 있습니다. 또한 클라이언트로 돌아오는 다른 방향의 플로우도 분리시킬 수 있습니다. 솔루션 설명을 간략하게 보여 주기 위해 설명은 클라이언트 애플리케이션으로부터의 단일 플로우만 고려합니다.

클러스터 게이트웨이 큐 관리자의 클러스터 메시지 트래픽 분리를 위한 솔루션

문제 해결을 위한 한 가지 방법은 클러스터 간 브릿지로 큐 관리자 알리어스 또는 리모트 큐 정의를 사용하는 것입니다. 클러스터된 리모트 큐 정의, 전송 큐 및 채널을 작성하여 게이트웨이 큐 관리자에서 각 메시지 플로우를 분리하십시오. 181 페이지의 [『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』](#)의 내용을 참조하십시오.

Version 7.5부터는, 클러스터 큐 관리자가 단일 클러스터 전송 큐로 제한되지 않습니다. 다음 두 가지 선택사항이 있습니다.

1. 추가 클러스터 전송 큐를 수동으로 정의하고 각 전송 큐에서 메시지를 전송하는 클러스터-송신자 채널을 정의하십시오. 184 페이지의 [『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』](#)의 내용을 참조하십시오.
2. 큐 관리자가 추가 클러스터 전송 큐를 자동으로 작성하고 관리하게 하십시오. 각 클러스터-송신자 채널에 대해 다른 클러스터 전송 큐를 정의합니다. 202 페이지의 [『메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경』](#)의 내용을 참조하십시오.

일부 클러스터 송신자 채널의 정의된 클러스터 전송 큐를 나머지를 관리하는 큐 관리자와 수동으로 결합시켜야 합니다. 전송 큐 결합은 184 페이지의 [『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』](#)에서 사용된 방법입니다. 이 솔루션에서 클러스터 간 대부분의 메시지는 공통 SYSTEM.CLUSTER.TRANSMIT.QUEUE(를) 사용합니다. 한 중요 애플리케이션은 수동으로 정의된 클러스터 전송 큐를 사용하여 모든 메시지 플로우를 다른 플로우와 분리시킵니다.

184 페이지의 [『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』](#)의 구성이 제한됩니다. 이 구성은 다른 클러스터 큐와 동일한 클러스터에 있는 동일한 큐 관리자의 클러스터 큐로 향하는 메시지 트래픽을 분리시키지 않습니다. 분산 큐잉의 일부인 리모트 큐 정의를 사용하여 메시지 트래픽을 개별 큐로 분리할 수 있습니다. 클러스터의 경우에는 다수의 클러스터 전송 큐를 사용하여 다른 클러스터 송신자 채널로 향하는 메시지 트래픽을 분리시킬 수 있습니다. 동일한 클러스터의 동일한 큐 관리자에 있는 여러 클러스터 큐는 클러스터 송신자 채널을 공유합니다. 이러한 큐에 대한 메시지는 게이트웨이 큐 관리자로부터 전달되기 전에 동일한 전송 큐에 저장됩니다. 187 페이지의 [『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』](#)의 구성에서, 다른 클러스터를 추가하고 새 클러스터 멤버로 큐 관리자와 클러스터 큐를 작성하여 제한을 피해줍니다. 새 큐 관리자가 클러스터의 유일한 큐 관리자일 수 있습니다. 더 많은 큐 관리자를 클러스터에 추가하고 동일한 클러스터를 사용하여 큐 관리자의 클러스터 큐를 분리시킬 수도 있습니다.

관련 개념

[145 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』](#)

애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드는 클러스터 큐에 대해 원격으로 검사, SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사 또는 클러스터 큐나 클러스터 큐 관리자의 로컬 프로파일에 대해 검사입니다.

[254 페이지의 『클러스터링: 중첩 클러스터에 대한 특수 고려사항』](#)

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

[155 페이지의 『클러스터 전송 큐 및 클러스터-송신자 채널』](#)

클러스터된 큐 관리자 간 메시지는 클러스터 전송 큐에 저장되고 클러스터 송신자 채널에서 전달됩니다.

[163 페이지의 『중첩 클러스터』](#)

중첩 클러스터는 추가 관리 기능을 제공합니다. 이름 목록을 사용하여 중첩 클러스터를 관리하는 데 필요한 명령의 수를 줄이십시오.

관련 태스크

[리모트 클러스터 큐에 메시지를 넣는 권한 부여](#)

[181 페이지의 『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』](#)

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 클러스터된 큐 리모트 정의 및 개별 송신자 채널과 전송 큐를 사용합니다.

184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 추가 클러스터 전송 큐를 사용하여 클러스터의 단일 큐 관리자로 메시지 트래픽을 분리합니다.

187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 솔루션은 추가 클러스터를 사용하여 메시지를 특정 클러스터 큐로 분리합니다.

202 페이지의 『메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경』

클러스터된 큐 또는 전송 큐의 토픽에 대해 큐 관리자가 메시지를 저장하는 기본 방식을 변경할 수 있습니다. 기본값을 변경하여 게이트웨이 큐 관리자에서 클러스터 메시지를 격리하는 방식을 제공합니다.

197 페이지의 『게이트웨이 큐 관리자로 두 개의 중첩 클러스터 작성』

태스크의 지시사항에 따라 게이트웨이 큐 관리자가 포함된 중첩 클러스터를 구성합니다. 한 애플리케이션에 대한 메시지를 클러스터 내 다른 애플리케이션에 대한 메시지로부터 격리하는 다음 예에 대한 시작점으로 이 클러스터를 사용합니다.

235 페이지의 『클러스터 간 메시지 경로 구성』

게이트웨이 큐 관리자를 사용하여 클러스터를 서로 연결합니다. 게이트웨이 큐 관리자에서 클러스터 큐 또는 클러스터 큐 관리자 알리어스를 정의하여 큐 또는 큐 관리자가 모든 클러스터에 표시되도록 합니다.

261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』

클러스터 전송 큐의 선택에 대한 지침입니다. 하나의 공용 기본 큐, 별도의 기본 큐, 또는 수동으로 정의된 큐를 구성할 수 있습니다. 다중 클러스터 전송 큐 구성은 z/OS 이외의 플랫폼에 적용됩니다.

관련 참조

392 페이지의 『보안』

qm.ini 파일의 Security 스탠자를 사용하여 오브젝트 권한 관리자(OAM)에 대한 옵션을 지정합니다.

setmqaut

클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획

클러스터 전송 큐의 선택에 대한 지침입니다. 하나의 공용 기본 큐, 별도의 기본 큐, 또는 수동으로 정의된 큐를 구성할 수 있습니다. 다중 클러스터 전송 큐 구성은 z/OS 이외의 플랫폼에 적용됩니다.

시작하기 전에

264 페이지의 『사용할 클러스터 전송 큐의 유형 선택 방법』의 내용을 검토하십시오.

이 태스크 정보

다음은 클러스터 전송 큐를 선택하도록 큐 관리자를 구성하는 방법을 계획할 때 사용할 몇 가지 선택사항입니다.

- 클러스터 메시지 전송을 위한 기본 클러스터 전송 큐는 무엇입니까?
 - 공용 클러스터 전송 큐, SYSTEM.CLUSTER.TRANSMIT.QUEUE.
 - 별도의 클러스터 전송 큐. 큐 관리자가 별도의 클러스터 전송 큐를 관리합니다. 모델 큐 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE로부터 영구 동적 큐로 작성합니다. 사용하는 각 클러스터 송신자 채널마다 클러스터 전송 큐를 하나씩 작성합니다.
- 수동으로 작성하도록 결정한 클러스터 전송 큐의 경우 추가로 다음 두 선택사항이 있습니다.
 - 수동으로 구성하도록 결정한 각 클러스터 송신자 채널마다 별도의 전송 큐를 정의합니다. 이 경우 전송 큐의 **CLCHNAME** 큐 속성을 클러스터 송신자 채널의 이름으로 설정하십시오. 이 전송 큐에서 메시지를 전송할 클러스터 송신자 채널을 선택하십시오.

- b. 클러스터 송신자 채널의 그룹에 대한 메시지 트래픽을 동일한 클러스터 전송 큐로 결합하십시오(262 페이지의 그림 56 참조). 이 경우 각 공용 전송 큐의 **CLCHNAME** 큐 속성을 일반 클러스터 송신자 채널 이름으로 설정하십시오. 일반 클러스터 송신자 채널 이름은 클러스터 송신자 채널 이름을 그룹화하는 필터입니다. 예를 들어, SALES.*는 이름이 SALES.로 시작하는 모든 클러스터 송신자 채널을 그룹화합니다. 필터 문자열의 어디에나 여러 와일드카드 문자를 넣을 수 있습니다. 와일드카드 문자는 별표, "*"입니다. 0(영)부터 임의의 수의 문자를 나타냅니다.

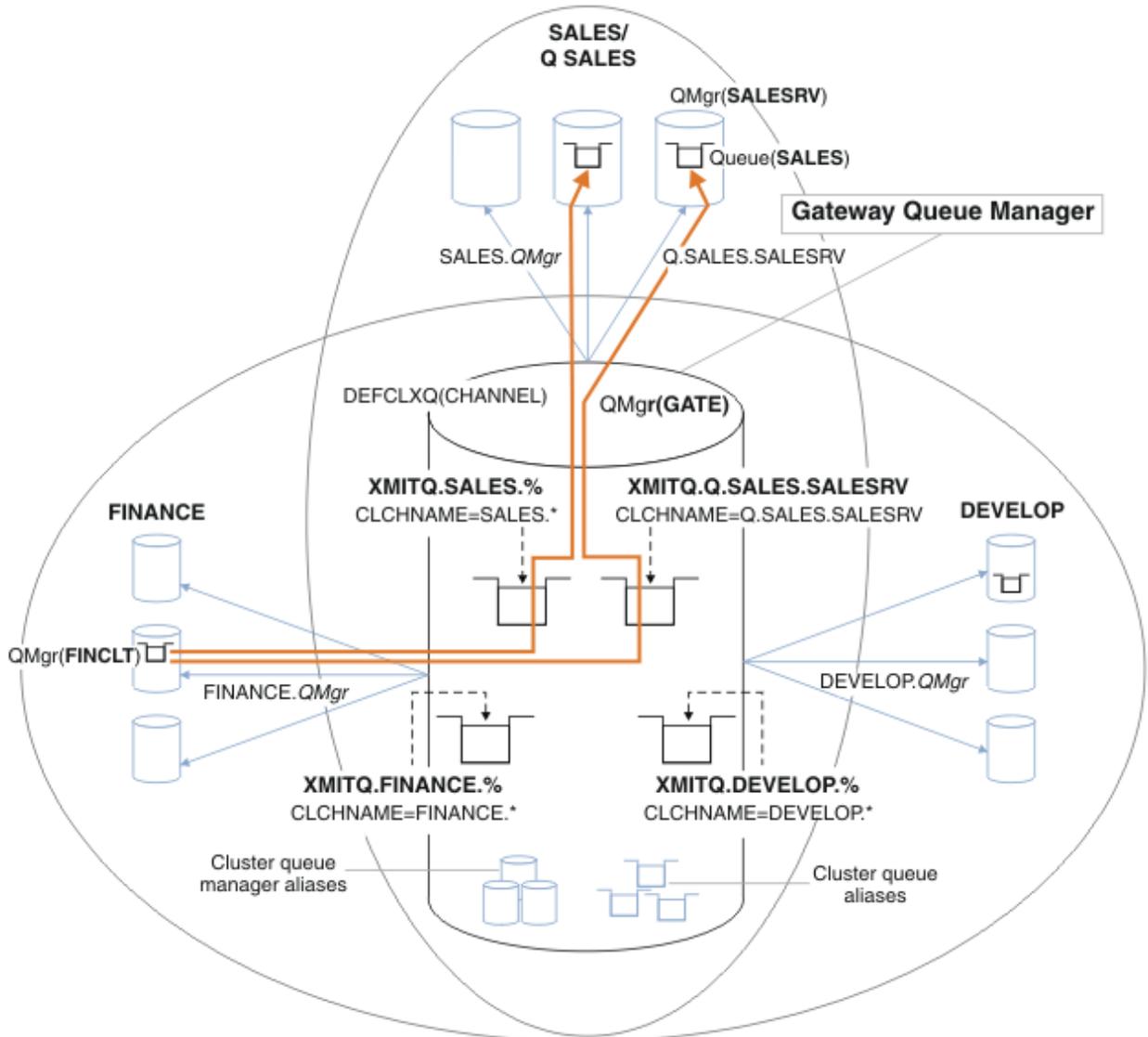


그림 56. 다른 부서 IBM WebSphere MQ 클러스터의 특정 전송 큐 예

프로시저

1. 사용할 기본 클러스터 전송 큐 유형을 선택하십시오.
 - 각 클러스터 연결에 대해 단일 클러스터 전송 큐 또는 별도의 큐를 선택하십시오.
 기본 설정을 그대로 두거나 **MQSC** 명령을 실행하십시오.

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. 다른 플로우와 클러스터 전송 큐를 공유해서는 안되는 메시지 플로우를 분리시키십시오.
 - 266 페이지의 『클러스터링: 다중 클러스터 전송 큐 구성 예』의 내용을 참조하십시오. 예에서, 분리시켜야 하는 SALES 큐는 SALESRV의 SALES 클러스터 멤버입니다. SALES 큐를 분리시키려면 새 클러스터

Q.SALES를 작성하고 SALESRV 큐 관리자를 멤버로 지정한 후 Q.SALES에 속하도록 SALES 큐를 수정하십시오.

- SALES에 메시지를 보내는 큐 관리자도 새 클러스터의 멤버여야 합니다. 클러스터된 큐 알리어스 및 게이트웨이 큐 관리자를 사용할 때에는 예에서처럼, 대부분의 경우 게이트웨이 큐 관리자가 새 클러스터 멤버에 수행하는 변경을 제한할 수 있습니다.
- 하지만 게이트웨이에서 목적지로 향하는 플로우를 분리시키면 소스 큐 관리자에서 게이트웨이로의 플로우가 분리되지 않습니다. 때로는 게이트웨이로 향하는 플로우가 아닌 게이트웨이에서 나오는 플로우만 분리시켜도 충분한 경우가 있습니다. 충분하지 않은 경우에는 소스 큐 관리자를 새 클러스터에 추가하십시오. 메시지가 게이트웨이를 통과하게 하려면 클러스터 알리어스를 새 클러스터로 이동시킨 후 대상 큐 관리자가 아닌 게이트웨이의 클러스터 알리어스로 계속해서 직접 메시지를 보내십시오.

다음 단계에 따라 메시지 플로우를 분리시키십시오.

- a) 각 대상 큐가 큐 관리자의 특정 클러스터에서 유일한 큐가 되도록 플로우 목적지를 구성하십시오.
 - b) 체계적인 이름 지정 규칙을 따라 작성한 새 클러스터에 대해 클러스터 송신자 및 클러스터 수신자 채널을 작성하십시오.
 - 254 페이지의 『클러스터링: 중첩 클러스터에 대한 특수 고려사항』의 내용을 참조하십시오.
 - c) 대상 큐로 메시지를 보내는 모든 큐 관리자의 분리된 각 목적지마다 클러스터 전송 큐를 정의하십시오.
 - 클러스터 전송 큐의 이름 지정 규칙은 XMITQ. 접두부가 있는 클러스터 채널 이름 속성 CLCHNAME의 값을 사용하는 것입니다.
3. 제어 또는 모니터링 요구사항이 충족된 클러스터 전송 큐를 작성하십시오.
- 일반적인 제어 및 모니터링 요구사항은 클러스터당 하나의 전송 큐 또는 큐 관리자당 하나의 전송 큐입니다. 클러스터 채널, *ClusterName.QueueManagerName*에 대한 이름 지정 규칙을 따르는 경우, 큐 관리자의 클러스터를 선택하거나 큐 관리자의 모든 클러스터가 멤버인 일반적인 채널 이름을 작성하는 것이 용이합니다. 266 페이지의 『클러스터링: 다중 클러스터 전송 큐 구성 예』의 내용을 참조하십시오.
 - 퍼센트 부호로 별표 기호를 바꿔서 일반 채널 이름을 제공하도록 클러스터 전송 큐의 이름 지정 규칙을 확장하십시오. 예를 들면 다음과 같습니다.

```
DEFINE QLOCAL(XMITQ.SALES.%) USAGE(XMITQ) CLCHNAME(SALES.*)
```

관련 개념

[155 페이지의 『클러스터 전송 큐 및 클러스터-송신자 채널』](#)

클러스터된 큐 관리자 간 메시지는 클러스터 전송 큐에 저장되고 클러스터 송신자 채널에서 전달됩니다.

[258 페이지의 『클러스터링: 다중 클러스터 전송 큐를 사용하여 애플리케이션 격리』](#)

클러스터의 큐 관리자 간에 메시지 플로우를 분리시킬 수 있습니다. 다른 클러스터 송신자 채널에서 전송되는 메시지를 상이한 클러스터 전송 큐에 둘 수 있습니다. 단일 클러스터 또는 중첩 클러스터에 이 접근법을 사용할 수 있습니다. 이 주제는 사용할 접근법을 선택하는 데 유용한 예와 몇 가지 우수 사례를 제공합니다.

[145 페이지의 『액세스 제어 및 다중 클러스터 전송 큐』](#)

애플리케이션이 리모트 클러스터 큐에 메시지를 넣을 때 검사하는 세 가지 모드 중에서 선택하십시오. 모드는 클러스터 큐에 대해 원격으로 검사, SYSTEM.CLUSTER.TRANSMIT.QUEUE에 대해 로컬로 검사 또는 클러스터 큐나 클러스터 큐 관리자의 로컬 프로파일에 대해 검사입니다.

[254 페이지의 『클러스터링: 중첩 클러스터에 대한 특수 고려사항』](#)

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

[163 페이지의 『중첩 클러스터』](#)

중첩 클러스터는 추가 관리 기능을 제공합니다. 이름 목록을 사용하여 중첩 클러스터를 관리하는 데 필요한 명령의 수를 줄이십시오.

관련 태스크

[181 페이지의 『게이트웨이 큐 관리자에서 송신된 메시지를 격리하기 위한 리모트 큐 정의 추가』](#)

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 클러스터된 큐 리모트 정의 및 개별 송신자 채널과 전송 큐를 사용합니다.

184 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 이 솔루션은 추가 클러스터 전송 큐를 사용하여 클러스터의 단일 큐 관리자로 메시지 트래픽을 분리합니다.

187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』

게이트웨이 큐 관리자를 사용하는 겹치는 클러스터의 구성을 수정하십시오. 게이트웨이 큐 관리자를 사용하는 중첩 클러스터의 구성을 수정합니다. 솔루션은 추가 클러스터를 사용하여 메시지를 특정 클러스터 큐로 분리합니다.

202 페이지의 『메시지 트래픽을 격리하기 위해 클러스터 전송 큐를 분리하도록 기본값 변경』

클러스터된 큐 또는 전송 큐의 토픽에 대해 큐 관리자가 메시지를 저장하는 기본 방식을 변경할 수 있습니다. 기본값을 변경하여 게이트웨이 큐 관리자에서 클러스터 메시지를 격리하는 방식을 제공합니다.

197 페이지의 『게이트웨이 큐 관리자 두 개의 중첩 클러스터 작성』

태스크의 지시사항에 따라 게이트웨이 큐 관리자가 포함된 중첩 클러스터를 구성합니다. 한 애플리케이션에 대한 메시지를 클러스터 내 다른 애플리케이션에 대한 메시지로부터 격리하는 다음 예에 대한 시작점으로 이 클러스터를 사용합니다.

235 페이지의 『클러스터 간 메시지 경로 구성』

게이트웨이 큐 관리자를 사용하여 클러스터를 서로 연결합니다. 게이트웨이 큐 관리자에서 클러스터 큐 또는 클러스터 큐 관리자 알리아스를 정의하여 큐 또는 큐 관리자가 모든 클러스터에 표시되도록 합니다.

사용할 클러스터 전송 큐의 유형 선택 방법

다른 클러스터 전송 큐 구성 옵션 간에 선택 방법.

Version 7.5부터는, 클러스터 송신자 채널과 연관되는 클러스터 전송 큐를 선택할 수 있습니다.

1. 단일 기본 클러스터 전송 큐 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`과(와) 연관된 모든 클러스터 송신자 채널이 있을 수 있습니다. 이 옵션은 기본값이며 Version 7.1 이하 버전을 실행하는 큐 관리자의 유일한 선택 사항입니다.
2. 모든 클러스터 송신자 채널을 개별 클러스터 전송 큐와 자동으로 연관시키도록 설정할 수 있습니다. 큐는 모델 큐 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`의 큐 관리자가 작성하고 이름은 `SYSTEM.CLUSTER.TRANSMIT.ChannelName`입니다. 큐 관리자 속성 **DEFCLXQ**(가) CHANNEL로 설정된 경우 채널은 고유하게 이름 지정된 클러스터 전송 큐를 사용합니다.



주의: 제품의 이전 버전에서 업그레이드된 큐 관리자와 함께 전용 `SYSTEM.CLUSTER.TRANSMIT.QUEUES`을 사용하는 경우, `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`에 **공유/공유하기** 옵션이 **공유**로 설정되어 있는지 확인하십시오.

3. 단일 클러스터 전송 큐에서 제공할 특정 클러스터 송신자 채널을 설정할 수 있습니다. 전송 큐를 작성하고 **CLCHNAME** 속성을 클러스터 송신자 채널의 이름으로 설정하여 이 옵션을 선택하십시오.
4. 단일 클러스터 전송 큐에서 제공할 클러스터 송신자 채널 그룹을 선택할 수 있습니다. 전송 큐를 작성하고 **CLCHNAME** 속성을 일반 채널 이름(예: `ClusterName.*`)으로 설정하여 이 옵션을 선택하십시오. 254 페이지의 『클러스터링: 중첩 클러스터에 대한 특수 고려사항』의 이름 지정 규칙을 따라 클러스터 채널의 이름을 지정하는 경우, 이 이름을 통해 클러스터 `ClusterName`의 큐 관리자에 연결된 모든 클러스터 채널을 선택합니다.

일부 클러스터 송신자 채널에 대한 기본 클러스터 전송 큐 옵션 중 하나와 임의의 수의 특정 및 일반 클러스터 전송 큐 구성을 결합할 수 있습니다.

우수 사례

대부분의 경우 기존 IBM WebSphere MQ 설치에는 기본 구성이 최상의 선택입니다. 클러스터 큐 관리자는 클러스터 메시지를 단일 클러스터 전송 큐 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`에 저장합니다. 상이한 큐 관리자와 상이한 클러스터에 대한 메시지를 별도의 전송 큐에 저장하거나 자신의 전송 큐를 정의하도록 기본값을 변경하는 선택사항이 있습니다.

대부분의 경우에는 새 IBM WebSphere MQ 설치에도 기본 구성이 최상의 선택입니다. 기본 구성에서 각 클러스터 송신자 채널마다 전송 큐가 하나씩 있는 대체 구성으로 전환하는 프로세스는 자동입니다. 재전환도 자동입니다. 어느 것을 선택하든 중요하지 않으며 되돌릴 수 있습니다.

다른 구성을 선택하는 이유는 기능성이나 성능보다는 제어 및 관리와 더 관련이 있습니다. 몇 가지 예외를 제외하면, 다수의 전송 큐를 구성해도 큐 관리자 작동에 이점이 없습니다. 큐가 더 많아져서 단일 전송 큐를 참조하는, 이미 설정한 모니터링 및 관리 프로시저를 수정해야 합니다. 이는 다른 선택사항에 대한 강력한 제어 또는 관리 사유가 없으면 모든 것을 감안할 때 기본 구성을 그대로 사용하는 것이 최상의 선택이기 때문입니다.

두 예외 모두 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 저장된 메시지 수가 증가하는 경우 발생하는 상황과 관련이 있습니다. 한 목적지의 메시지를 다른 목적지의 메시지와 분리시키기 위한 모든 단계를 수행한 경우에는 한 목적지에 대한 채널 및 전달 문제가 다른 목적지로의 전달에 영향을 미치면 안됩니다. 그러나 한 대상에 충분히 빠르게 메시지를 전달하지 않아 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 저장된 메시지 수가 증가할 수 있습니다. 한 대상에 대한 SYSTEM.CLUSTER.TRANSMIT.QUEUE의 메시지 수는 다른 대상으로의 메시지 전달에 영향을 줄 수 있습니다.

단일 전송 큐가 가득 차서 발생하는 문제를 방지하려면 구성에 충분한 용량을 빌드하십시오. 그렇게 하면 한 목적지가 실패하고 메시지 백로그가 빌드되기 시작할 경우 문제점을 수정할 시간이 있습니다.

메시지가 허브 큐 관리자(예: 클러스터 게이트웨이)를 통해 라우팅되는 경우, 공통 전송 큐, SYSTEM.CLUSTER.TRANSMIT.QUEUE을(를) 공유합니다. 게이트웨이 큐 관리자의 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 저장된 메시지 수가 최대 용량에 도달하면 깊이가 줄어들 때까지 큐 관리자가 전송 큐에 대한 새 메시지를 거부하기 시작합니다. 게이트웨이를 통해 라우팅된 모든 목적지에 대한 메시지에 정체(정체)의 영향이 미칩니다. 메시지는 게이트웨이에 메시지를 보내는 다른 큐 관리자의 전송 큐를 백업합니다. 큐 관리자 오류 로그에 기록된 메시지, 메시지 처리량 감소, 메시지를 송신하고 대상에 메시지가 도착하는 시간 사이의 연장된 경과 시간 등으로 문제점이 드러납니다.

단일 전송 큐의 정체 효과는 큐가 가득 차기 전에도 명백해질 수 있습니다. 일부는 큰 비지속 메시지가 있고 일부는 작은 메시지가 있는 혼합 메시지 트래픽의 경우 전송 큐가 가득 차면 작은 메시지를 전달하는 시간이 늘어납니다. 이 지연은 일반적으로 디스크에 쓰지 않는 큰 비지속 메시지를 디스크에 쓰기 때문입니다. 시간에 민감한 메시지 플로우가 있는 경우, 클러스터 전송 큐를 혼합된 다른 메시지 플로우와 공유하면 다른 메시지 플로우에서 격리시 키도록 특별한 메시지 경로를 구성할 가치가 있습니다. [187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』](#)의 내용을 참조하십시오.

별도의 클러스터 전송 큐를 구성하는 다른 이유는 제어 요구사항을 충족시키거나 상이한 클러스터 목적지로 송신되는 메시지 모니터링을 단순화하기 위해서입니다. 예를 들어, 한 목적지에 대한 메시지가 다른 목적지에 대한 메시지와 전송 큐를 결코 공유하지 않음을 증명해야 할 수 있습니다.

기본 클러스터 전송 큐를 제어하는 큐 관리자 속성 **DEFCLXQ**을(를) 변경하여 모든 클러스터 송신자 채널에 사용할 다른 클러스터 전송 큐를 작성하십시오. 여러 목적지가 클러스터 송신자 채널을 공유할 수 있으므로 클러스터가 이 목표를 완전히 충족시키도록 계획해야 합니다. [메소드 187 페이지의 『게이트웨이 큐 관리자에서 송신된 클러스터 메시지 트래픽을 격리하기 위한 클러스터 및 클러스터 전송 큐 추가』](#)를 모든 클러스터 큐에 체계적으로 적용하십시오. 목표로 한 결과는 클러스터 목적지가 클러스터 송신자 채널을 다른 클러스터 목적지와 공유하지 않는 것입니다. 결과적으로 클러스터 목적지에 대한 메시지가 다른 목적지에 대한 메시지와 클러스터 전송 큐를 공유하지 않습니다.

일부 특정 메시지 플로우에 대한 별도의 클러스터 전송 큐를 작성하면 목적지로 향하는 메시지 플로우를 쉽게 모니터링할 수 있습니다. 새 클러스터 전송 큐를 사용하려면 큐를 정의하고 클러스터 송신자 채널과 연관시키고 나서 채널을 중지한 후 시작하십시오. 변경사항이 영구적일 필요는 없습니다. 잠시 동안 메시지 플로우를 분리시켜서 전송 큐를 모니터링한 후 기본 전송 큐를 다시 사용하도록 되돌릴 수 있습니다.

관련 태스크

[클러스터링: 다중 클러스터 전송 큐 구성 예](#)

이 태스크에서는 세 개의 중첩 클러스터에 대한 다중 클러스터 전송 큐를 계획하는 단계를 적용합니다. 요구사항은 한 클러스터 큐로 가는 메시지 플로우를 다른 모든 메시지 플로우와 분리시키고 여러 다른 클러스터에 대한 메시지를 상이한 클러스터 전송 큐에 저장해야 하는 점입니다.

[클러스터링: 클러스터 전송 큐 전환](#)

기존 프로덕션 큐 관리자의 클러스터 전송 큐 변경사항을 적용할 방법을 계획합니다.

클러스터링: 다중 클러스터 전송 큐 구성 예

이 태스크에서는 세 개의 중첩 클러스터에 대한 다중 클러스터 전송 큐를 계획하는 단계를 적용합니다. 요구사항은 한 클러스터 큐로 가는 메시지 플로우를 다른 모든 메시지 플로우와 분리시키고 여러 다른 클러스터에 대한 메시지를 상이한 클러스터 전송 큐에 저장해야 하는 점입니다.

이 태스크 정보

이 태스크의 단계는 261 페이지의 『클러스터링: 클러스터 전송 큐를 구성하는 방법을 계획』의 프로시저를 적용할 방법 및 266 페이지의 그림 57에 표시된 구성에 이르는 방법을 표시합니다. 이는 별도의 클러스터 전송 큐가 구성된 게이트웨이 큐 관리자가 있는 세 개의 중첩 클러스터 예입니다. 클러스터를 정의할 MQSC 명령은 268 페이지의 『예 클러스터 작성』에 설명되어 있습니다.

예를 들어, 두 개의 요구사항이 있습니다. 하나는 게이트웨이 큐 관리자에서 판매를 기록하는 판매 애플리케이션으로 향하는 메시지 플로우를 분리시키는 것입니다. 두 번째는 임의의 시점에 다른 부서 영역으로의 송신을 대기 중인 메시지 수를 조회하는 것입니다. SALES, FINANCE 및 DEVELOP 클러스터가 이미 정의되어 있습니다. 클러스터 메시지는 현재 SYSTEM.CLUSTER.TRANSMIT.QUEUE로부터 전달됩니다.

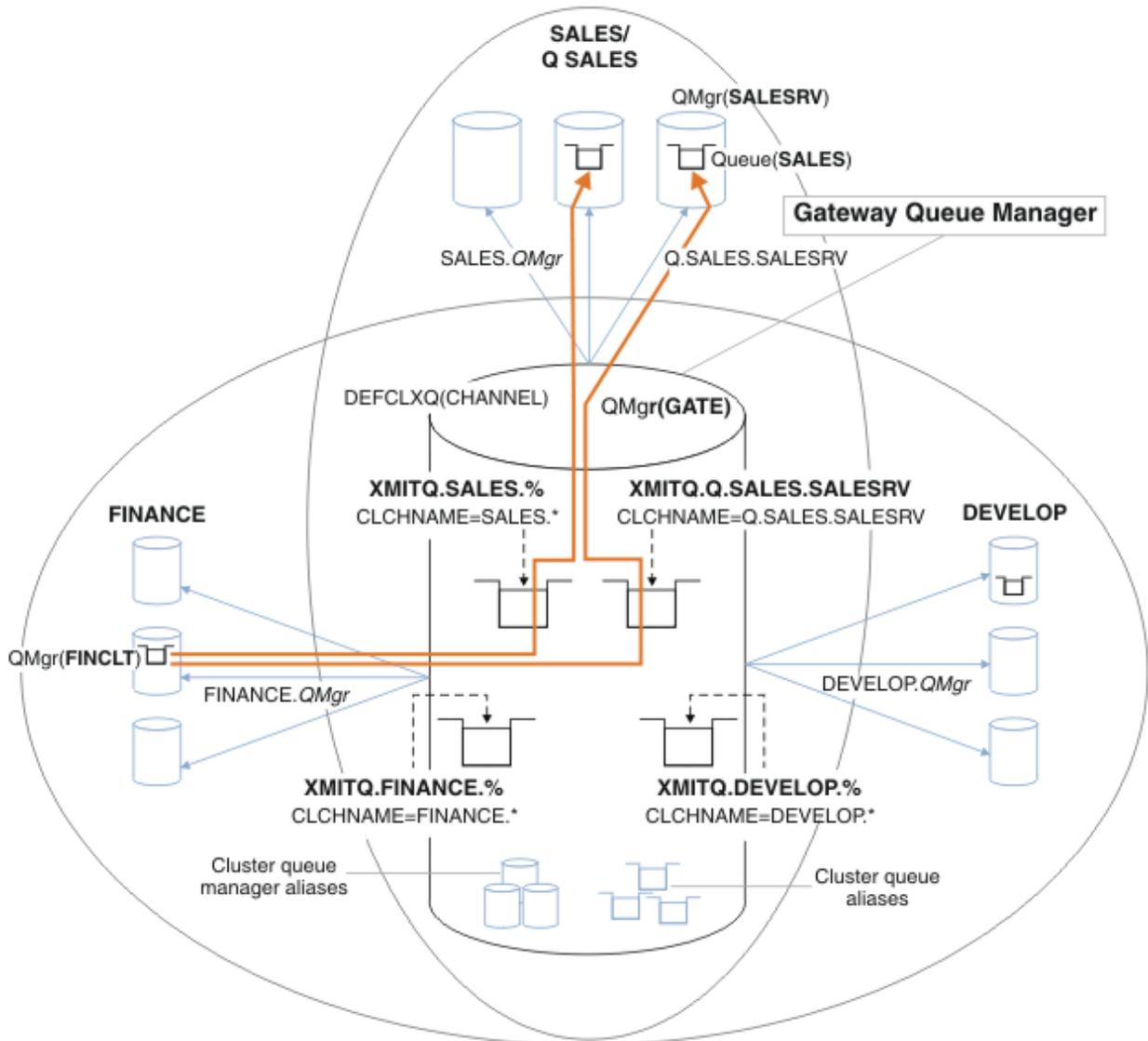


그림 57. 다른 부서 IBM WebSphere MQ 클러스터의 특정 전송 큐 예

클러스터를 수정하는 단계는 다음과 같습니다. 변경사항을 참조하여 새 클러스터의 판매 큐를 분리하고 정의에 대한 게이트웨이 클러스터 전송 큐를 분리 를 참조하십시오.

프로시저

1. 첫 번째 구성 단계는 "사용할 기본 클러스터 전송 큐 유형을 선택하십시오"입니다.

결정 사항은 GATE 큐 관리자에서 다음 **MQSC** 명령을 실행하여 별도의 기본 클러스터 전송 큐를 작성하는 것입니다.

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

클러스터 전송 큐를 수동으로 정의하려고 하기 때문에 이 기본값을 선택할 이유가 많지 않습니다. 선택사항은 약한 진단 값이 있습니다. 수동 정의를 잘못 수행하고 메시지가 아래의 기본 클러스터 전송 큐로 플로우되는 경우 이는 영구 동적 클러스터 전송 큐의 작성으로 나타납니다.

2. 두 번째 구성 단계는 "다른 플로우와 클러스터 전송 큐를 공유해서는 안되는 메시지 플로우를 분리시키십시오"입니다.

이 경우 SALES의 SALESRV 큐에서 메시지를 수신하는 판매 애플리케이션을 격리시킬 필요가 있습니다. 게이트웨이 큐 관리자에서 나오는 메시지만 격리시키면 됩니다. 세 가지 하위 단계는 다음과 같습니다.

- a) "각 대상 큐가 큐 관리자의 특정 클러스터에서 유일한 큐가 되도록 플로우 목적지를 구성하십시오".

예에서는 큐 관리자 SALESRV를 영업부서 내의 새 클러스터에 추가해야 합니다. 격리해야 하는 큐가 거의 없으면 SALES 큐에 대한 특정 클러스터를 작성하도록 결정할 수 있습니다. 클러스터 이름의 가능한 이름 지정 규칙은 이러한 클러스터의 이름을 지정하는 것입니다 Q. *QueueName: Q.SALES*). 많은 수의 큐를 격리시켜야 하는 경우에 보다 실제적일 수 있는 대체 방법은 필요한 시기 및 위치에서 격리된 큐의 클러스터를 작성하는 것입니다. 클러스터 이름은 QUEUES.n일 수 있습니다.

예에서 새 클러스터를 Q.SALES라 합니다. 새 클러스터를 추가하려면 변경사항의 정의를 참조하여 새 클러스터에서 판매 큐를 분리하고 게이트웨이 클러스터 전송 큐를 분리하십시오. 정의 변경 요약은 다음과 같습니다.

- i) Q.SALES를 저장소 큐 관리자의 클러스터 이름 목록에 추가하십시오. 이름 목록은 큐 관리자 **REPOSNL** 매개변수에 참조됩니다.
- ii) Q.SALES를 게이트웨이 큐 관리자의 클러스터 이름 목록에 추가하십시오. 이름 목록은 게이트웨이 큐 관리자의 모든 클러스터 큐 알리어스 및 클러스터 큐 관리자 알리어스 정의에 참조됩니다.
- iii) 멤버인 클러스터 모두에 대해 큐 관리자 SALESRV에 이름 목록을 작성하고, SALES 큐의 클러스터 멤버십을 변경하십시오.

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES) REPLACE  
ALTER QLOCAL(SALES) CLUSTER(' ') CLUSNL(SALESRV.CLUSTERS)
```

SALES 큐는 전이만을 위한 두 클러스터의 멤버입니다. 새 구성이 일단 실행되면 SALES 큐를 SALES 클러스터에서 제거합니다(271 페이지의 그림 58 참조).

- b) "체계적인 이름 지정 규칙을 따라 작성한 새 클러스터에 대해 클러스터 송신자 및 클러스터 수신자 채널을 작성하십시오".

- i) 클러스터-송신자 채널 Q.SALES.*RepositoryQMGr* 를 각 저장소 큐 관리자에 추가합니다.
- ii) 클러스터-송신자 채널 Q.SALES.*OtherRepositoryQMGr* 를 각 저장소 큐 관리자에 추가하여 다른 저장소 관리자에 연결하십시오. 이 채널을 시작하십시오.
- iii) 클러스터 수신자 채널 Q.SALES.SALESRV 및 Q.SALES.GATE를 실행 중인 저장소 큐 관리자 중 하나에 추가하십시오.
- iv) 클러스터-송신자 채널 Q.SALES.SALESRV 및 Q.SALES.GATE를 SALESRV 및 GATE 큐 관리자에 추가하십시오. 클러스터 송신자 채널을 클러스터 수신자 채널이 작성되는 저장소 큐 관리자에 연결하십시오.

- c) "대상 큐로 메시지를 보내는 모든 큐 관리자의 분리된 각 목적지마다 클러스터 전송 큐를 정의하십시오".

게이트웨이 큐 관리자에서 Q.SALES.SALESRV 클러스터 송신자 채널에 대한 클러스터 전송 큐 XMITQ.Q.SALES.SALESRV을(를) 정의하십시오.

```
DEFINE QLOCAL(XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
```

3. 세 번째 구성 단계는 "제어 또는 모니터링 요구사항이 충족된 클러스터 전송 큐를 작성하십시오"입니다.

게이트웨이 큐 관리자에서 클러스터 전송 큐를 정의하십시오.

```
DEFINE QLOCAL(XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
DEFINE QLOCAL(XMITQ.DEVELOP) USAGE(XMITQ) CLCHNAME(DEVELOP.*) REPLACE
DEFINE QLOCAL(XMITQ.FINANCE) USAGE(XMITQ) FINANCE(SALES.*) REPLACE
```

다음에 수행할 작업

게이트웨이 큐 관리자의 새 구성으로 전환하십시오.

새 채널을 시작하고 현재 상이한 전송 큐와 연관된 채널을 다시 시작하면 전환이 트리거됩니다. 또는 게이트웨이 큐 관리자를 중지한 후 시작할 수 있습니다.

1. 게이트웨이 큐 관리자에서 다음 채널을 중지하십시오.

```
SALES.Qmgr
DEVELOP.Qmgr
FINANCE.Qmgr
```

2. 게이트웨이 큐 관리자에서 다음 채널을 시작하십시오.

```
SALES.Qmgr
DEVELOP.Qmgr
FINANCE.Qmgr
Q.SALES.SAVESRV
```

스위치가 완료되면 SALES 클러스터에서 SALES 큐를 제거하십시오. [271 페이지의 그림 58](#)을 참조하십시오.

관련 개념

[사용할 클러스터 전송 큐의 유형 선택 방법](#)
다른 클러스터 전송 큐 구성 옵션 간에 선택 방법.

관련 태스크

[클러스터링: 클러스터 전송 큐 전환](#)
기존 프로덕션 큐 관리자의 클러스터 전송 큐 변경사항을 적용할 방법을 계획합니다.

예 클러스터 작성

예 클러스터를 작성하고 게이트웨이 큐 관리자의 개별 메시지 및 SALES 큐를 분리시키도록 수정하는 지시사항 및 정의를 설명합니다.

이 태스크 정보

FINANCE, SALES 및 Q.SALES 클러스터를 작성하기 위한 전체 **MQSC** 명령은 [기본 클러스터의 정의](#), 새 클러스터에서 판매 큐를 분리하고 게이트웨이 클러스터 전송 큐를 분리하도록 변경 및 판매 클러스터에서 큐 관리자 SALESRV의 판매 큐 제거에 제공됩니다. DEVELOP 클러스터는 정의를 더 짧게 유지하기 위해 정의에서 생략됩니다.

프로시저

1. SALES 및 FINANCE 클러스터와 게이트웨이 큐 관리자를 작성하십시오.

a) 큐 관리자를 작성하십시오.

[269 페이지의 표 27](#)의 각 큐 관리자 이름에 대해 `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QmgrName` 명령을 실행하십시오.

표 27. 큐 관리자 이름 및 포트 번호		
설명	큐 관리자 이름	포트 번호
재무 저장소	FINR1	1414
재무 저장소	FINR2	1415
재무 클라이언트	FINCLT	1418
판매 저장소	SALER1	1416
판매 저장소	SALER2	1417
판매 서버	SALESRV	1419
게이트웨이	GATE	1420

b) 모든 큐 관리자를 시작하십시오.

269 페이지의 표 27의 각 큐 관리자 이름으로 `strmqm QmgrName` 명령을 실행하십시오.

c) 각 큐 관리자에 대한 정의를 작성하십시오.

Run the command: `runmqsc QmgrName <filename` where the files are listed in [기본 클러스터에 대한 정의](#), and the file name matches the queue manager name.

기본 클러스터에 대한 정의 **finr1.txt**

```
DEFINE LISTENER(1414) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1414) REPLACE
START LISTENER(1414)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
```

finr2.txt

```
DEFINE LISTENER(1415) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1415) REPLACE
START LISTENER(1415)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
```

finclt.txt

```
DEFINE LISTENER(1418) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1418) REPLACE
START LISTENER(1418)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)')
CLUSTER(FINANCE) REPLACE
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

saler1.txt

```
DEFINE LISTENER(1416) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1416) REPLACE
START LISTENER(1416)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
```

saler2.txt

```
DEFINE LISTENER(1417) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1417) REPLACE
START LISTENER(1417)
```

```
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
```

salesrv.txt

```
DEFINE LISTENER(1419) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1419) REPLACE
START LISTENER(1419)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(SALES) REPLACE
DEFINE QLOCAL(SALES) CLUSTER(SALES) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

gate.txt

```
DEFINE LISTENER(1420) TRPTYPE(TCP) IPADDR(LOCALHOST) CONTROL(QMGR) PORT(1420) REPLACE
START LISTENER(1420)
DEFINE NAMELIST(ALL) NAMES(SALES, FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(SALES) REPLACE
DEFINE QALIAS(A.SALES) CLUSNL(ALL) TARGET(SALES) TARGTYPE(Queue) DEFBIND(NOTFIXED)
REPLACE
DEFINE QREMOTE(FINCLT) RNAME(' ') RQMNAME(FINCLT) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(SALESRV) RNAME(' ') RQMNAME(SALESRV) CLUSNL(ALL) REPLACE
```

2. 샘플 요청 프로그램을 실행하여 구성을 테스트하십시오.

a) SALESRV 큐 관리자에서 트리거 모니터 프로그램을 시작하십시오.

Windows에서 명령 창을 열고 명령 `runmqtrm -m SALESRV`를 실행하십시오.

b) 샘플 요청 프로그램을 실행하고 요청을 송신하십시오.

Windows에서 명령 창을 열고 `amqsreq A.SALES FINCLT` 명령을 실행하십시오.

요청 메시지가 다시 에코되고 15초 후에 샘플 프로그램이 완료됩니다.

3. 정의를 작성하여 Q.SALES 클러스터의 SALES 큐를 격리시키고 게이트웨이 큐 관리자에서 SALES 및 FINANCE 클러스터에 대한 개별 클러스터 메시지를 분리하십시오.

Run the command: `runmqsc QmgrName < filename` where the files are listed in the following list, and the file name almost matches the queue manager name.

새 클러스터의 판매 큐를 격리시키고 게이트웨이 클러스터 전송 큐를 분리하기 위한 변경사항
chgsaler1.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
```

chgsaler2.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
```

chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(Q.SALES) REPLACE
ALTER QLOCAL (SALES) CLUSTER(' ') CLUSNL(CLUSTERS)
```

chgate.txt

```
ALTER NAMELIST(ALL) NAMES(SALES, FINANCE, Q.SALES)
ALTER QMGR DEFCLXQ(CHANNEL)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('localhost(1420)')
CLUSTER(Q.SALES) REPLACE
DEFINE QLOCAL (XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
DEFINE QLOCAL (XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
DEFINE QLOCAL (XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(FINANCE.*) REPLACE
```

4. SALES 클러스터에서 SALES 큐를 제거하십시오.

271 페이지의 [그림 58](#)에서 **MQSC** 명령을 실행하십시오.

```
ALTER QLOCAL(SALES) CLUSTER('Q.SALES') CLUSNL(' ')
```

그림 58. 판매 클러스터에서 큐 관리자 SALESRV의 판매 큐 제거

5. 새 전송 큐로 채널을 전환하십시오.

요구사항은 GATE 큐 관리자가 사용 중인 모든 채널을 중지한 후 시작하는 것입니다. 최소의 명령 수로 이를 수행하려면 큐 관리자를 중지한 후 시작하십시오.

```
endmqm -i GATE
strmqm GATE
```

다음에 수행할 작업

1. 샘플 요청 프로그램을 재실행하여 새 구성 작업을 확인하십시오(270 페이지의 『2』 단계 참조).
2. GATE 큐 관리자의 모든 클러스터 전송 큐를 통해 플로우되는 메시지를 모니터하십시오.

- a. 각 클러스터 전송 큐 정의를 변경하여 큐 모니터링을 켜십시오.

```
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.
name) STATQ(ON)
```

- b. 큐 관리자 통계 모니터링이 OFF인지 확인하여, 출력을 최소화하고 모니터링 간격을 낮은 값으로 설정하여 편리하게 다중 테스트를 수행하십시오.

```
ALTER QMGR STATINT(60) STATCHL(OFF) STATQ(OFF) STATMQI(OFF) STATACLS(OFF)
```

- c. GATE 큐 관리자를 재시작하십시오.

- d. 샘플 요청 프로그램을 몇 번 실행하여 동일한 수의 메시지가 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV 및 SYSTEM.CLUSTER.TRANSMIT.QUEUE를 통해 플로우되는지 확인하십시오. 요청은 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV를 통해 플로우되고 응답은 SYSTEM.CLUSTER.TRANSMIT.QUEUE를 통해 플로우됩니다.

```
amqsmn -m GATE -t statistics
```

- e. 두 간격의 결과는 다음과 같습니다.

```
C:\Documents and Settings\Admin>amqsmn -m GATE -t statistics
MonitoringType: QueueStatistics
```

```

QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '14.59.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.00.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
  CreateDate: '2012-02-24'
  CreateTime: '15.58.15'
  ...
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [435, 0]
  GetCount: [1, 0]
  GetBytes: [435, 0]
  ...
QueueStatistics: 1
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
  CreateDate: '2012-02-24'
  CreateTime: '16.37.43'
  ...
  PutCount: [1, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [435, 0]
  GetCount: [1, 0]
  GetBytes: [435, 0]
  ...
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '15.00.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.01.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
  CreateDate: '2012-02-24'
  CreateTime: '15.58.15'
  ...
  PutCount: [2, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [863, 0]
  GetCount: [2, 0]
  GetBytes: [863, 0]
  ...
QueueStatistics: 1
  QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
  CreateDate: '2012-02-24'
  CreateTime: '16.37.43'
  ...
  PutCount: [2, 0]

```

```
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [863, 0]
GetCount: [2, 0]
GetBytes: [863, 0]
```

...
2 Records Processed.

하나의 요청 및 응답 메시지가 첫 번째 간격에 송신되고 두 개의 요청 및 응답 메시지가 두 번째 간격에 송신되었습니다. SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV에 배치된 요청 메시지 및 SYSTEM.CLUSTER.TRANSMIT.QUEUE에 있는 응답 메시지를 추론할 수 있습니다.

클러스터링: 클러스터 전송 큐 전환

기존 프로덕션 큐 관리자의 클러스터 전송 큐 변경사항을 적용할 방법을 계획합니다.

시작하기 전에

전환 프로세스가 새 전송 큐에 전송해야 하는 메시지 수를 줄이면 전환이 보다 빨리 완료됩니다. 계속해서 진행하기 전에 전송 큐를 비우려는 이유를 보려면 157 페이지의 『클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식』을 읽어보십시오.

이 태스크 정보

클러스터 전송 큐의 변경사항을 적용할 두 가지 방법이 있습니다.

1. 큐 관리자가 자동으로 변경을 수행할 수 있게 하십시오. 기본값입니다. 클러스터 송신자 채널이 다음 번에 시작될 때 큐 관리자가 보류 중인 전송 큐 변경사항이 있는 클러스터 송신자 채널을 전환합니다.
2. 수동으로 변경을 수행하십시오. 클러스터 송신자 채널이 중지되었을 때 이 채널을 변경할 수 있습니다. 클러스터 송신자 채널이 시작되기 전에 한 클러스터 전송 큐에서 다른 클러스터 전송 큐로 이를 전환할 수 있습니다.

둘 중 어느 옵션을 선택할지 결정할 때 어떤 요인을 고려하며 전환을 어떻게 관리합니까?

프로시저

- 옵션 1: 큐 관리자가 자동으로 변경을 수행할 수 있게 하십시오(274 페이지의 『활성 클러스터 송신자 채널을 다른 클러스터 전송 큐 세트에 전환』 참조).

큐 관리자가 전환하게 하려는 경우 이 옵션을 선택하십시오.

이 옵션을 설명하는 대체 방법은 큐 관리자에게 채널 중지를 강제 실행하지 않은 채로 클러스터 송신자 채널을 전환하라고 알리는 것입니다. 채널이 중지되도록 강제 실행한 후 채널을 시작해서 즉시 전환이 발생하게 하는 옵션이 있습니다. 옵션 2와 다르게, 전환은 채널이 시작될 때 시작되고 채널이 실행 중인 동안에 실행됩니다. 옵션 2에서는 채널이 중지될 때 전환이 발생합니다.

전환이 자동으로 발생하도록 이 옵션을 선택하는 경우 클러스터 송신자 채널이 시작될 때 전환 프로세스가 시작합니다. 채널이 중지되지 않은 경우 처리할 메시지가 있으면 채널이 비활성화된 후에 프로세스가 시작합니다. 채널이 중지된 경우 START CHANNEL 명령을 사용하여 시작하십시오.

채널이 제공하는 전송 큐의 클러스터 송신자 채널에 메시지가 남아 있지 않으면 그 즉시 전환 프로세스가 완료됩니다. 그러한 경우 클러스터 송신자 채널의 새로 도착한 메시지가 새 전송 큐에 직접 저장됩니다. 그 때까지는, 메시지가 이전 전송 큐에 저장되고 전환 프로세스는 이전 전송 큐에서 새 전송 큐로 메시지를 전송합니다. 클러스터 송신자 채널은 전체 전환 프로세스 중 새 클러스터 전송 큐에서 메시지를 전달합니다. 전환 프로세스가 완료되는 시기는 시스템의 상태에 따라 다릅니다. 유지보수 창에서 변경 중인 경우 전환 프로세스가 제 시간에 완료될지 미리 가늠해보십시오. 제 시간에 완료될지 여부는 이전 전송 큐에서 전송을 대기 중인 메시지 수가 0이 되는지 여부에 따라 다릅니다.

첫 번째 메소드의 장점은 자동이라는 것입니다. 단점은 구성 변경을 수행하는 시간이 유지보수 창으로 제한된 경우 유지보수 창에서 전환 프로세스를 완료하도록 시스템을 제어할 수 있는 확신이 있어야 한다는 점입니다. 확실하지 않다면 옵션 2가 더 나은 선택일 수 있습니다.

- 옵션 2: 수동으로 변경을 수행하십시오(275 페이지의 『중지된 클러스터 송신자 채널을 다른 클러스터 전송 큐로 전환』 참조).

전체 전환 프로세스를 수동으로 제어하려는 경우나 비활성 또는 중지된 채널을 전환하려는 경우 이 옵션을 선택하십시오. 몇 개의 클러스터 송신자 채널을 전환 중이며 유지보수 창에서 전환을 수행하려는 경우에 좋은 방법입니다.

이 옵션의 대체 설명은 클러스터 송신자 채널이 중지된 동안에 클러스터 송신자 채널을 전환하는 것입니다.

이 옵션을 선택하면 전환의 발생 시기를 완전히 제어합니다.

유지보수 창에서 정해진 시간 안에 전환 프로세스를 완료한다고 확신할 수 있습니다. 전환 발생 시기는 한 전송 큐에서 다른 전송 큐로 전송해야 하는 메시지 수에 따라 다릅니다. 메시지가 계속 도착하는 경우에는 프로세스가 모든 메시지를 전송하기까지 시간이 걸릴 수 있습니다.

이전 전송 큐에서 메시지를 전송하지 않고 채널을 전환하는 옵션이 있습니다. 이는 "즉석" 전환입니다.

클러스터 송신자 채널을 다시 시작하면 최근에 지정한 전송 큐의 메시지를 채널이 처리하기 시작합니다.

두 번째 방법의 장점은 전환 프로세스를 제어한다는 점입니다. 단점은 전환할 클러스터 송신자 채널을 식별하고, 필요한 명령을 실행하며, 클러스터 송신자 채널이 중지되지 못하게 할 수 있는 인다우트(in-doubt) 채널을 해결해야 하는 점입니다.

관련 개념

사용할 클러스터 전송 큐의 유형 선택 방법
다른 클러스터 전송 큐 구성 옵션 간에 선택 방법.

관련 태스크

클러스터링: 다중 클러스터 전송 큐 구성 예

이 태스크에서는 세 개의 중첩 클러스터에 대한 다중 클러스터 전송 큐를 계획하는 단계를 적용합니다. 요구사항은 한 클러스터 큐로 가는 메시지 플로우를 다른 모든 메시지 플로우와 분리시키고 여러 다른 클러스터에 대한 메시지를 상이한 클러스터 전송 큐에 저장해야 하는 점입니다.

274 페이지의 『활성 클러스터 송신자 채널을 다른 클러스터 전송 큐 세트로 전환』

이 태스크는 활성 클러스터 송신자 채널을 전환하기 위한 세 가지 옵션을 제공합니다. 한 옵션은 큐 관리자가 자동으로 전환하게 하는 방법으로, 실행 중인 애플리케이션에 영향을 주지 않습니다. 다른 옵션은 수동으로 채널을 중지한 후 시작하거나 큐 관리자를 다시 시작하는 방법입니다.

275 페이지의 『중지된 클러스터 송신자 채널을 다른 클러스터 전송 큐로 전환』

관련 정보

157 페이지의 『클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식』

활성 클러스터 송신자 채널을 다른 클러스터 전송 큐 세트로 전환

이 태스크는 활성 클러스터 송신자 채널을 전환하기 위한 세 가지 옵션을 제공합니다. 한 옵션은 큐 관리자가 자동으로 전환하게 하는 방법으로, 실행 중인 애플리케이션에 영향을 주지 않습니다. 다른 옵션은 수동으로 채널을 중지한 후 시작하거나 큐 관리자를 다시 시작하는 방법입니다.

시작하기 전에

클러스터 전송 큐 구성을 변경하십시오. **DEFCLXQ** 큐 관리자 속성을 변경하거나 전송 큐의 **CLCHNAME** 속성을 추가 또는 수정할 수 있습니다.

전환 프로세스가 새 전송 큐에 전송해야 하는 메시지 수를 줄이면 전환이 보다 빨리 완료됩니다. 계속해서 진행하기 전에 전송 큐를 비우려는 이유를 보려면 157 페이지의 『클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식』을 읽어보십시오.

이 태스크 정보

이 태스크의 단계를 클러스터 전송 큐 구성 변경을 수행하기 위한 계획을 세우는 기반으로 삼으십시오.

프로시저

1. 옵션: 현재 채널 상태를 기록하십시오.

클러스터 전송 큐를 서비스 중인 현재 및 저장된 채널의 상태를 기록하십시오. 다음 명령은 시스템 클러스터 전송 큐와 연관된 상태를 표시합니다. 정의한 클러스터 전송 큐와 연관된 상태를 표시하려면 명령을 직접 추가하십시오. `XMITQ.ChannelName`과 같은 규칙을 사용하여 전송 큐의 채널 상태를 쉽게 표시할 수 있도록 정의한 클러스터 전송 큐의 이름을 지정하십시오.

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

2. 전송 큐를 전환하십시오.

- 아무 것도 하지 마십시오. 큐 관리자가 중지 또는 비활성화된 후 다시 시작할 때 클러스터 송신자 채널을 전환합니다.
큐 관리자 구성 변경에 대한 관심 또는 규칙이 없으면 이 옵션을 선택하십시오. 실행 중인 애플리케이션은 변경의 영향을 받지 않습니다.
- 큐 관리자를 재시작하십시오. 모든 클러스터 송신자 채널은 중지되고 요구가 있으면 자동으로 다시 시작됩니다.
모든 변경사항을 즉시 시작하려면 이 옵션을 선택하십시오. 큐 관리자가 시스템 종료된 후 재시작할 때 실행 애플리케이션을 인터럽트합니다.
- 개별 클러스터-송신자 채널을 중지한 후 재시작하십시오.
몇 개의 채널을 즉시 변경하려면 이 옵션을 선택하십시오. 메시지 채널을 중지한 후 다시 시작하는 사이에 실행 중인 애플리케이션은 메시지 전송 중 잠시 지연됩니다. 클러스터 송신자 채널은 중지된 시간을 제외하고, 실행 상태를 유지합니다. 전환 프로세스 중 메시지는 이전 전송 큐로 전달되고, 전환 프로세스를 통해 새 전송 큐에 전송되며, 클러스터 송신자 채널을 통해 새 전송 큐에서 전달됩니다.

3. 옵션: 전환하는 채널을 모니터하십시오.

전환 중 채널 상태 및 전송 큐 용량을 표시하십시오. 다음 예는 시스템 클러스터 전송 큐의 상태를 표시합니다.

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

- 옵션: 큐 관리자 오류 로그에 기록되는 "`AMQ7341 ChannelName` 채널의 전송 큐가 `QueueName` 큐에서 `QueueName` 로 전환되었습니다." 메시지를 모니터하십시오.

중지된 클러스터 송신자 채널을 다른 클러스터 전송 큐로 전환

시작하기 전에

몇 가지 구성 변경을 수행했으며 이제 영향이 미치는 클러스터 송신자 채널을 시작하지 않은 채로 변경사항을 적용시키려 합니다. 또는 필요한 구성 변경을 태스크의 단계 중 하나로 수행합니다.

전환 프로세스가 새 전송 큐에 전송해야 하는 메시지 수를 줄이면 전환이 보다 빨리 완료됩니다. 계속해서 진행하기 전에 전송 큐를 비우려는 이유를 보려면 [157 페이지의 『클러스터 송신자 채널을 다른 전송 큐로 전환하는 프로세스의 작동 방식』](#)을 읽어보십시오.

이 태스크 정보

이 태스크는 비활성 또는 중지된 클러스터 송신자 채널에 제공된 전송 큐를 전환합니다. 클러스터 송신자 채널이 중지되어서 전송 큐를 복시 전환하려 하기 때문에 이 태스크를 수행할 수 있습니다. 예를 들어, 어떠한 이유로 클러스터 송신자 채널이 시작되지 않거나 일부 다른 구성 문제가 있는 경우입니다. 문제를 해결하려면 클러스터 송신자 채널을 작성하고 이전 클러스터 송신자 채널의 전송 큐를 새 클러스터 송신자 채널의 전송 큐와 연관시키도록 결정합니다.

더 가능성 있는 시나리오는 클러스터 전송 큐의 재구성이 수행되는 시기를 제어하려는 경우입니다. 재구성을 완전히 제어하려면 채널을 중지하고 구성을 변경한 후 전송 큐를 전환시킵니다.

프로시저

- 전환하려는 채널을 중지하십시오.

- a) 전환하려는 비활성 또는 실행 중인 채널을 중지하십시오. 비활성 클러스터 송신자 채널을 중지하면 구성 변경을 수행하는 동안에 이 채널이 시작되지 않습니다.

```
STOP CHANNEL(ChannelName) MODE(QUIESCSE) STATUS(STOPPED)
```

2. 옵션: 구성 변경을 수행하십시오.

예를 들어, 266 페이지의 『클러스터링: 다중 클러스터 전송 큐 구성 예』의 내용을 참조하십시오.

3. 클러스터 송신자 채널을 새 클러스터 전송 큐로 전환하십시오.

```
runswchl -m QmgrName -c ChannelName
```

runswchl 명령은 이전 전송 큐의 메시지를 새 전송 큐로 전송합니다. 이 채널에 대한 이전 전송 큐의 메시지 수가 0이 되면 전환이 완료됩니다. 명령은 동기식입니다. 명령은 전환 프로세스 중 장애 진행 메시지를 씁니다.

전송 단계 중 클러스터 송신자 채널로 예정된 기존 및 새 메시지는 새 전송 큐에 순서대로 전송됩니다.

클러스터 송신자 채널이 중지되었기 때문에 새 전송 큐에 메시지가 빌드됩니다. 중지된 클러스터 송신자 채널을 274 페이지의 『활성 클러스터 송신자 채널을 다른 클러스터 전송 큐 세트로 전환』의 275 페이지의 『2』 단계와 대조시키십시오. 이 단계에서는 클러스터 송신자 채널이 실행 중이므로 새 전송 큐에 메시지가 반드시 빌드되지는 않습니다.

4. 옵션: 전환하는 채널을 모니터하십시오.

다른 명령 창에서 전환 중의 전송 큐 용량을 표시하십시오. 다음 예는 시스템 클러스터 전송 큐의 상태를 표시합니다.

```
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

5. 옵션: 큐 관리자 오류 로그에 기록되는 "AMQ7341 ChannelName 채널의 전송 큐가 QueueName 큐에서 QueueName 로 전환되었습니다." 메시지를 모니터하십시오.

6. 중지한 클러스터 송신자 채널을 재시작하십시오.

채널이 중지되어 STOPPED 상태에 있으므로 자동으로 시작되지 않습니다.

```
START CHANNEL(ChannelName)
```

관련 참조

[runswchl](#)

[RESOLVE CHANNEL](#)

[STOP CHANNEL](#)

클러스터링: 마이그레이션 및 수정 우수 사례

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

- 276 페이지의 『클러스터의 오브젝트 이동』 (IBM WebSphere MQ의 새 버전이나 수정팩을 설치하지 않은 채, 클러스터 내에서 오브젝트를 이동시키는 우수 사례).
- 277 페이지의 『업그레이드 및 유지보수 설치』 (유지보수 또는 업그레이드를 적용하고 새 아키텍처를 테스트하는 동안 작업 중인 클러스터 아키텍처를 계속 가동하고 실행하는 우수 사례).

클러스터의 오브젝트 이동

애플리케이션 및 큐

하나의 큐 관리자에서 호스트되는 큐 인스턴스를 다른 큐 관리자로 이동해야 할 때에는 워크로드 밸런싱 매개변수에 대해 작업하여 원활하게 전이할 수 있습니다.

새로 호스트할 큐의 인스턴스를 작성하되, 클러스터 워크로드 밸런싱 설정을 사용하여 애플리케이션이 전환할 준비가 될 때까지 원래 인스턴스로 계속 메시지를 보내십시오. 이를 위해 다음 단계를 따르십시오.

1. 기존 큐의 **CLWLRANK** 특성을 5와 같은 높은 값으로 설정하십시오.
2. 큐의 새 인스턴스를 작성하고 **CLWLRANK** 특성을 0으로 설정하십시오.
3. 큐의 새 인스턴스에 대해 이용 애플리케이션을 배치하고 시작하는 것과 같은 새 시스템의 추가 구성을 완료하십시오.
4. 새 큐 인스턴스의 **CLWLRANK** 특성을 9와 같이 원래 인스턴스보다 높게 설정하십시오.
5. 원래 큐 인스턴스가 시스템의 큐에 대기된 메시지를 처리하도록 한 후 큐를 삭제하십시오.

전체 큐 관리자 이동

큐 관리자가 동일한 호스트에 있지만 IP 주소가 변경되는 경우 프로세스는 다음과 같습니다.

- 제대로 사용할 경우 DNS가 프로세스를 단순화하는 데 유용할 수 있습니다. 연결 이름 (CONNAME) 채널 속성을 설정하여 DNS 사용에 대한 정보는 ALTER CHANNEL 을 참조하십시오.
- 전체 저장소를 이동하는 경우 변경을 수행하기 전에 최소 하나의 다른 전체 저장소가 순조롭게(예를 들어, 채널 상태에 대한 문제 없이) 실행 중인지 확인하십시오.
- 트래픽이 빌드되지 않도록 SUSPEND QMGR 명령을 사용하여 큐 관리자를 일시중단하십시오.
- 컴퓨터의 IP 주소를 수정하십시오. CLUSRCVR 채널 정의에 CONNAME 필드의 IP 주소가 사용되는 경우에는 이 IP 주소 항목을 수정하십시오. 어디에서나 이 업데이트를 사용할 수 있도록 DNS 캐시를 비워야 할 수 있습니다.
- 큐 관리자가 전체 저장소에 다시 연결할 때 채널 자동 정의가 자동으로 자체 분석됩니다.
- 큐 관리자가 전체 저장소를 호스트했으며 IP 주소가 변경되면 수동으로 정의된 CLUSSDR 채널을 새 위치로 가리키기 위해 일부분이 가능한 빨리 전환되는지 확인하는 것이 중요합니다. 이 전환이 이행될 때까지는 이 큐 관리자가 나머지(변경되지 않은) 전체 저장소에만 접속할 수 있으며 올바르게 작동하는 채널 정의에 관한 경고 메시지가 표시될 수 있습니다.
- RESUME QMGR 명령을 사용하여 큐 관리자를 재개하십시오.

큐 관리자를 새 호스트로 이동시켜야 하는 경우 큐 관리자 데이터를 복사하고 백업에서 복원할 수 있습니다. 하지만 다른 옵션이 있을 경우 이 프로세스는 권장되지 않습니다. 이전 절에 설명된 대로 새 시스템에 큐 관리자를 작성하고 큐와 애플리케이션을 복제하는 것이 더 나을 수도 있습니다. 이렇게 하면 롤오버/롤백 메커니즘이 순조롭게 작동합니다.

백업을 사용하여 전체 큐 관리자를 이동시키기로 판별된 경우 다음 우수 사례를 따르십시오.

- 운영 체제 환경에 적합한 시스템 복구에 일반적으로 사용하는 프로세스를 적용하여, 백업에서 큐 관리자를 복원하는 것으로 전체 프로세스를 처리하십시오.
- 마이그레이션 후에 REFRESH CLUSTER 명령을 사용하여 로컬로 보유된 모든 클러스터 정보 (인다우트 상태인 자동 정의 채널 포함) 를 버리고 다시 빌드하도록 강제 실행하십시오.

참고: 대형 클러스터의 경우, REFRESH CLUSTER 명령을 사용하면 진행 중에 클러스터에 혼란을 줄 수 있으며, 클러스터 오브젝트가 모든 관심 있는 큐 관리자에 자동으로 상태 업데이트를 보낸 이후 27일 간격으로 다시 수행됩니다. 대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음을 참조하십시오.

큐 관리자를 작성하고 클러스터의 기존 큐 관리자에서 설정을 복제할 때에는(이 주제에서 이전에 설명한 대로) 두 개의 상이한 큐 관리자를 실제로 동일한 큐 관리자인 것처럼 처리하지 마십시오. 특히, 새 큐 관리자에 동일한 큐 관리자 이름 및 IP 주소를 제공하지 마십시오. 대체 큐 관리자를 '드롭 인'하려는 시도가 IBM WebSphere MQ 클러스터 문제의 원인인 경우가 빈번합니다. 캐시는 QMID 속성을 포함하여 업데이트를 수신할 것으로 예상하며 상태가 손상될 수 있습니다.

두 개의 큐 관리자가 우연히 동일한 이름으로 작성된 경우에는 RESET CLUSTER QMID 명령을 사용하여 올바르게 작동하는 항목을 클러스터에서 빼내는 것이 좋습니다.

업그레이드 및 유지보수 설치

"빅뱅 시나리오"(예를 들어, 모든 클러스터와 큐 관리자 활동을 중지하고 전체 큐 관리자에 모든 업그레이드 및 유지보수를 적용한 후 모든 것을 동시에 시작함)를 피하십시오. 클러스터는 큐 관리자의 여러 버전이 공존하는 상황에서 여전히 작동하도록 설계되어 있으므로 계획이 잘 짜인 단계별 유지보수 접근법을 권장합니다.

백업을 계획하십시오.

- z/OS에서 역방향 마이그레이션 PTF를 적용했습니까?
- 백업을 했습니까?
- 새 클러스터 기능을 바로 사용하지 마십시오. 모든 큐 관리자가 새 레벨로 업그레이드되었음이 확실하고 어느 큐 관리자도 다시 롤백하지 않을 것이라 확신이 있을 때까지 대기하십시오. 일부 큐 관리자가 여전히 이전 레벨인 클러스터에서 새 클러스터 기능을 사용하면 정의되지 않은 작동이 발생할 수 있습니다. 예를 들어, IBM WebSphere MQ Version 6.0에서 IBM WebSphere MQ Version 7.1로 이동하면 큐 관리자가 클러스터 토픽을 정의하면 IBM WebSphere MQ Version 6.0 큐 관리자가 정의를 이해하지 못하거나 이 토픽을 발행할 수 있습니다.

전체 저장소를 먼저 마이그레이션하십시오. 전체 저장소는 이해하지 않는 정보를 전달할 수 있지만 지속할 수 없으므로 절대적으로 필요한 경우가 아니면 이는 권장하는 접근법이 아닙니다. 자세한 정보는 [큐 관리자 클러스터 마이그레이션을 참조하십시오](#).

관련 개념

278 페이지의 『클러스터링: REFRESH CLUSTER 사용 우수 사례』

REFRESH CLUSTER 명령을 사용하여 로컬에 보유된 클러스터에 대한 모든 정보를 제거하고 클러스터의 전체 저장소에서 이 정보를 다시 빌드합니다. 예외 상황을 제외하고는 이 명령을 사용해서는 안 됩니다. 사용해야 하는 경우 사용 방법에 대한 특수 고려사항이 있습니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

클러스터링: REFRESH CLUSTER 사용 우수 사례

REFRESH CLUSTER 명령을 사용하여 로컬에 보유된 클러스터에 대한 모든 정보를 제거하고 클러스터의 전체 저장소에서 이 정보를 다시 빌드합니다. 예외 상황을 제외하고는 이 명령을 사용해서는 안 됩니다. 사용해야 하는 경우 사용 방법에 대한 특수 고려사항이 있습니다. 이 정보는 고객의 피드백과 테스트에 기반을 둔 지침입니다.

필요한 경우 REFRESH CLUSTER만 실행

IBM WebSphere MQ 클러스터 기술은 클러스터된 큐의 변경과 같은 클러스터 구성 변경을 해당 정보를 알아야 하는 클러스터의 멤버에게 자동으로 알릴 수 있습니다. 이 정보를 전파하기 위해 추가로 관리 단계를 수행할 필요가 없습니다.

애플리케이션이 처음으로 열기를 시도할 때 클러스터의 다른 큐 관리자가 클러스터된 큐를 알지 못하는 경우처럼, 이러한 정보가 필요로 하는 클러스터의 큐 관리자에 도달하지 않는 경우 이는 클러스터 인프라에 문제가 있음을 암시합니다. 예를 들어, 큐 관리자와 전체 저장소 큐 관리자 간에 채널을 시작하지 못할 수 있습니다. 따라서 불일치가 관찰되는 상황을 조사해야 합니다. 가능하면 **REFRESH CLUSTER** 명령을 사용하지 않고 상황을 해결하십시오.

이 제품 문서의 다른 위치에 문서화되어 있거나 IBM 요청한 경우, **REFRESH CLUSTER** 명령을 사용하여 클러스터에 대한 모든 로컬 보유 정보를 버리고 클러스터의 전체 저장소에서 해당 정보를 다시 빌드할 수 있습니다.

대형 클러스터를 새로 고치면 클러스터의 성능 및 가용성에 영향을 줄 수 있음

REFRESH CLUSTER 명령을 사용하면 큐 관리자 클러스터 자원의 재전파를 처리할 때 전체 저장소에 대한 작업의 갑작스런 증가를 작성하여 진행 중인 동안 클러스터에 지장을 줄 수 있습니다. 대형 클러스터(즉, 수백 개의 큐 관리자가 있는)를 새로 고치는 경우 가능하면 일상적인 작업에 명령을 사용해서는 안 되며 대체 메소드를 사용하여 특정 불일치를 정정하십시오. 예를 들어, 클러스터 큐가 클러스터에서 제대로 전파되지 않으면 클러스터된 큐 정의 업데이트(정의의 설명 변경과 같은)의 초기 조사 기술을 통해 클러스터에 큐 구성을 재전파합니다. 이 프로세스는 문제점을 식별하고 잠재적으로 일시적 불일치를 해결하는 데 도움이 될 수 있습니다.

대체 메소드를 사용할 수 없고 큰 클러스터에서 **REFRESH CLUSTER**를 실행해야 하는 경우 사용자 워크로드에 대한 영향을 피하기 위해 피크가 아닌 시간 또는 유지보수 창 중에 이를 수행해야 합니다. 단일 배치에서도 대형 클러스터의 새로 고치기를 피해야 하며 대신에 278 페이지의 『클러스터 오브젝트가 자동 업데이트를 보낼 때 성능 및 가용성 문제 피하기』에 설명된 대로 활동의 시차를 두십시오.

클러스터 오브젝트가 자동 업데이트를 보낼 때 성능 및 가용성 문제 피하기

큐 관리자에 새 클러스터 오브젝트가 정의되고 나면 이 오브젝트에 대한 업데이트가 정의한 순간부터 27일마다 생성되어 클러스터의 모든 전체 저장소 및 관심이 있는 큐 관리자에게 송신됩니다. **REFRESH CLUSTER** 명령을

큐 관리자에게 발행하면 지정된 클러스터에 로컬로 정의된 모든 오브젝트에서 이 자동 업데이트를 위한 시계가 재설정됩니다.

단일 배치에서 또는 구성 백업으로부터 시스템을 재작성하는 것과 같은 상황에서 대형 클러스터(즉, 수백 개의 큐 관리자)를 새로 고치면 27일 후에 모든 큐 관리자가 동시에 전체 오브젝트 정의를 전체 저장소에 다시 광고합니다. 이로 인해 시스템이 다시 상당히 느려지거나 모든 업데이트가 완료될 때까지 사용 불가능하게 될 수도 있습니다. 따라서 대형 클러스터에서 다수의 큐 관리자를 새로 고치거나 재작성해야 할 때에는 후속 자동 업데이트가 정기적으로 시스템 성능에 영향을 주지 않도록 몇 시간 또는 며칠 간의 활동의 시차를 두어야 합니다.

시스템 클러스터 실행 기록 큐

REFRESH CLUSTER가 수행되면 큐 관리자는 새로 고치기 이전의 클러스터 상태 스냅샷을 작성하고 큐 관리자에 정의된 경우 **SYSTEM.CLUSTER.HISTORY.QUEUE(SCHQ)**에 스냅샷을 저장합니다. 이 스냅샷은 추후에 시스템에 관한 문제가 발생할 경우 IBM 서비스 용도로만 사용됩니다. SCHQ는 기본적으로 시동 시 분산 큐 관리자에 정의됩니다. z/OS 마이그레이션의 경우, SCHQ를 수동으로 정의해야 합니다. SCHQ에 대한 메시지는 3개월 후에 만기됩니다.

관련 개념

[REFRESH CLUSTER를 실행할 때 표시되는 애플리케이션 문제](#)

[발행/구독 클러스터에 대한 REFRESH CLUSTER 고려사항](#)

관련 참조

MQSC 명령 참조: [REFRESH CLUSTER](#)

클러스터링: 가용성, 다중 인스턴스, 재해 복구

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

IBM WebSphere MQ 클러스터링 자체는 고가용성 솔루션이 아니지만 일부 상황에서 IBM WebSphere MQ를 사용하여(예를 들어, 큐의 여러 인스턴스를 각기 다른 큐 관리자에 배치하여) 서비스 가용성을 개선할 수 있습니다. 이 절은 IBM WebSphere MQ 인프라를 이러한 아키텍처에 사용할 수 있도록 가능한 고가용 상태가 되게 하기 위한 지침을 제공합니다.

클러스터 자원의 가용성

두 개의 전체 저장소를 유지보수하도록 일반적으로 권장하는 이유는 한 전체 저장소가 유실되어도 클러스터의 원활한 실행에 문제가 되지 않기 때문입니다. 둘 다 사용할 수 없게 되는 경우에도 부분 저장소에 기존 정보를 보유할 60일의 유예 기간이 있지만 이 경우에는 새 자원이나 이전에 액세스하지 않은 자원(예: 큐)을 사용할 수 없습니다.

클러스터를 사용한 애플리케이션 가용성 개선

클러스터는 큐와 애플리케이션의 여러 인스턴스를 사용하여 고가용성 애플리케이션(예를 들어, 요청/응답 유형 서버 애플리케이션)을 설계하는 데 도움을 줄 수 있습니다. 큐 관리자나 채널이 사용 불가능한 경우가 아니면 필요에 따라 우선순위 속성이 '라이브' 애플리케이션에 우선권을 부여할 수 있습니다. 이는 문제가 발생할 때 새 메시지 처리를 계속하도록 빠르게 전환시키는 데 있어서 강력한 요인이 됩니다.

하지만 클러스터의 특정 큐 관리자에 전달된 메시지는 해당 큐 인스턴스에만 보유하고 이 큐 관리자가 복구될 때까지는 처리할 수 없습니다. 이러한 이유로 데이터 고가용성을 위해서는 다중 인스턴스 큐 관리자와 같은 다른 기술을 고려해 볼 수 있습니다.

다중 인스턴스 큐 관리자

소프트웨어 고가용성(다중 인스턴스)은 기존 메시지를 사용 가능하게 유지하기 위한 최상의 기본 제공 오픈링입니다. 자세한 정보는 287 페이지의 [『고가용성 구성으로 WebSphere MQ 사용』](#), 314 페이지의 [『다중 인스턴스 큐 관리자 작성』](#) 및 다음 절을 참조하십시오. 클러스터의 모든 큐 관리자가 최소 IBM WebSphere MQ Version 7.0.1에서 실행 중이면 이 기술을 사용하여 클러스터의 큐 관리자를 고가용성 상태가 되게 할 수 있습니다. 이전 레벨인 클러스터의 큐 관리자가 있는 경우 보조 IP로 장애 복구하면 다중 인스턴스 큐 관리자와의 연결이 유실될 수 있습니다.

이 주제에서 이전에 논의한 바와 같이, 두 개의 전체 저장소가 구성되어 있으면 그 자체로 거의 고가용성 상태가 됩니다. 필요한 경우 IBM WebSphere MQ 소프트웨어 고가용성/다중 인스턴스 큐 관리자를 전체 저장소에 사용할 수 있습니다. 이 방법을 사용해야 하는 강력한 이유는 없으며 사실, 일시적 가동 중단은 이 방법은 장애 복구 중에 추가 성능 비용을 유발할 수 있습니다. 예를 들어, 단일 채널 가동 중단이 발생할 경우 반

드시 장애 복구할 필요는 없지만 부분 저장소가 클러스터 자원을 조회할 수 없는 상태가 되므로, 두 개의 부분 저장소를 실행하는 대신 소프트웨어 HA를 사용하는 것은 권장되지 않습니다.

재해 복구

큐 관리자의 데이터를 저장하는 디스크가 손상되었을 때의 복구처럼 재해 복구는 잘 수행하기 어려운 작업입니다. IBM WebSphere MQ가 이를 도울 수 있지만 자동으로 수행할 수는 없습니다. IBM WebSphere MQ의 유일하게 '진정한' 재해 복구 옵션은(운영 체제 또는 기타 기본 복제 기술을 제외하고) 백업에서 복원하는 것입니다. 다음 상황에서 고려할 클러스터별 몇 가지 사항이 있습니다.

- 재해 복구 시나리오를 테스트할 때 주의하십시오. 예를 들어, 백업 큐 관리자의 조작을 테스트하는 경우 동일한 네트워크에서 온라인으로 불러올 때 주의하십시오. 우발적으로 라이브 클러스터를 조인하여 라이브 클러스터 큐 관리자와 동일하게 이름 지정된 큐를 호스트해서 메시지 '도용'을 시작할 수 있기 때문입니다.
- 재해 복구 테스트 시 실행 중인 라이브 클러스터를 간섭하면 안됩니다. 간섭을 피하기 위한 기법은 다음과 같습니다.
 - 완전한 네트워크 분리 또는 방화벽 레벨의 분리.
 - 실제 재해 복구 시나리오가 발생할 때까지 또는 발생하지 않으면 재해 복구 시스템에 라이브 SSL 인증서를 발행하지 않음.
- 클러스터의 큐 관리자 백업을 복원하는 경우 백업이 나머지 클러스터와 비동기화될 수 있습니다. **REFRESH CLUSTER** 명령은 업데이트를 해결하고 클러스터와 동기화하지만 **REFRESH CLUSTER** 명령을 최후의 수단으로 사용해서는 안됩니다. 278 페이지의 『클러스터링: REFRESH CLUSTER 사용 우수 사례』의 내용을 참조하십시오. 명령을 사용하기 전에 간단한 단계가 누락된 것이 아닌지 확인하려면 내부 프로세스 문서 및 IBM WebSphere MQ 문서를 검토하십시오.
- 복구를 위해서는 애플리케이션이 재실행과 데이터 손실을 처리해야 합니다. 큐를 알려진 상태로 지울지 또는 재생을 관리할 충분한 정보가 어딘가에 있는지 결정해야 합니다.

클러스터링: 모니터링

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

클러스터의 애플리케이션 메시지 모니터링

일반적으로, 큐 관리자를 떠나는 모든 클러스터 메시지는 해당 메시지를 전송하는 데 사용되고 있는 클러스터 송신자 채널과 상관없이 SYSTEM.CLUSTER.TRANSMIT.QUEUE를 통해 전달됩니다. 각 채널은 해당 채널을 대상으로 하는 메시지를 다른 모든 클러스터 송신자 채널과 병렬로 내보내고 있습니다. 이 큐에서 점점 쌓여가는 메시지는 하나 이상의 채널의 문제점을 표시할 수 있으며 조사해야 합니다.

- 큐의 용량이 클러스터 설계에 대해 적절하게 모니터링되어야 합니다.
- 다음 명령은 전송 큐에서 대기 중인 둘 이상의 메시지를 갖는 모든 채널을 리턴합니다.

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

모든 클러스터 메시지가 단일 큐에 있을 때, 항상 채널이 채우기 시작할 때 문제가 있는 채널을 확인하는 것이 쉽지는 않습니다. 이 명령의 사용은 책임있는 채널을 확인하는 쉬운 방법입니다.

복수 전송 큐를 갖도록 클러스터 큐 관리자를 구성할 수 있습니다. 큐 관리자 속성 DEFCLXQ를 CHANNEL로 변경하는 경우, 모든 클러스터 송신자 채널이 서로 다른 클러스터 전송 큐와 연관됩니다. 그렇지 않으면 별도의 전송 큐를 수동으로 구성할 수 있습니다. 클러스터 송신자 채널과 연관된 모든 클러스터 전송 큐를 표시하려면 다음 명령을 실행하십시오.

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

왼쪽에 큐 이름의 고정된 어간을 갖는 패턴을 따르도록 클러스터 전송 큐를 정의하십시오. 그런 다음 일반 큐 이름을 사용하여 **DISPLAY CLUSMGR** 명령으로 리턴된 모든 클러스터 전송 큐의 깊이를 조회할 수 있습니다.

```
DISPLAY QUEUE (qname*) CURDEPTH
```

클러스터의 제어 메시지 모니터링

SYSTEM.CLUSTER.COMMAND.QUEUE 큐는 로컬 큐 관리자에 의해 생성되거나 클러스터의 다른 큐 관리자로부터 이 큐 관리자로 전송되는, 큐 관리자에 대한 모든 클러스터 제어 메시지를 처리하는 데 사용됩니다. 큐 관리자가 그의 클러스터 상태를 올바르게 유지보수 중일 때, 이 큐는 0으로 가는 경향이 있습니다. 그러나 이 큐의 메시지 용량이 일시적으로 커질 수 있는 상황이 있습니다.

- 큐에 많은 메시지를 갖는 것은 클러스터 상태의 뒤섞임을 표시합니다.
- 상당한 변경을 수행할 때, 큐가 해당 변경 사이에서 안정할 수 있게 하십시오. 예를 들어, 저장소를 이동할 때 두 번째 저장소를 이동하기 전에 큐가 0에 도달할 수 있도록 허용하십시오.

메시지의 백로그가 이 큐에 존재할 때, 클러스터 상태 업데이트나 클러스터 관련 명령이 처리되지 않습니다. 이 큐에서 메시지가 오랫동안 제거되지 않고 있는 경우 추가적인 조사가 필요한데, 처음에는 이런 상황의 원인이 되는 프로세스의 단서를 제공할 수도 있는 큐 관리자 오류 로그를 검사하십시오.

SYSTEM.CLUSTER.REPOSITORY.QUEUE는 클러스터 저장소 캐시 정보를 메시지 수로 보유하고 있습니다. 메시지가 항상 이 큐에 존재하고 더 큰 클러스터의 경우 더 많이 존재하는 것이 유용합니다. 그러므로 이 큐의 메시지 용량은 관심이 가는 문제가 아닙니다.

모니터링 로그

문제점이 원래 정보의 캐싱 및 클러스터링의 분산 본질로 인해 발생한 후 여러 날(심지어 여러 달) 동안 클러스터에서 발생하는 문제점이 애플리케이션에 외부 증상을 보이지 않을 수 있습니다. 하지만, 원래 문제는 IBM WebSphere MQ 오류 로그에 종종 보고됩니다. 이 때문에 클러스터링과 관련하여 작성되는 모든 메시지에 대해 이들 오류를 능동적으로 모니터링하는 것이 중요합니다. 이들 메시지를 읽고 이해하고, 필요한 경우 조치를 취해야 합니다.

예를 들어, 클러스터에 있는 큐 관리자와의 통신 중단은 클러스터가 정보를 재발행하여 클러스터 자원을 정기적으로 다시 유효성 검증하는 방식으로 인해 삭제되고 있는 특정 클러스터 자원을 알 수 있게 합니다. 이러한 이벤트에 대한 경고가 [AMQ9465](#) 메시지에 의해 보고됩니다. 이 메시지는 문제점을 조사해야 함을 표시합니다.

로드 밸런싱을 위한 특별 고려사항

클러스터가 한 큐의 둘 이상의 인스턴스 사이에서 로드 밸런싱할 때, 이용하는 애플리케이션이 각 인스턴스의 메시지를 처리 중이어야 합니다. 이용 애플리케이션의 하나 이상이 종료하거나 메시지 처리를 중지하는 경우, 클러스터링이 큐의 해당 인스턴스로 메시지를 계속 송신할 수 있습니다. 이 상황에서는 애플리케이션이 다시 올바르게 기능 중일 때까지 해당 메시지가 처리되지 않습니다. 이 때문에 애플리케이션의 모니터링은 솔루션의 중요한 부분이며 해당 상황에서 메시지를 경로 재지정하는 조치를 취해야 합니다. 이러한 모니터링을 자동화하는 메커니즘의 예는 다음 샘플에서 찾을 수 있습니다. [클러스터 큐 모니터링 샘플 프로그램 \(AMQSCLM\)](#).

가용성, 복구 및 재시작

큐 관리자가 실패한 경우 큐 가용성을 유지보수하여 애플리케이션의 가용성을 높여 서버 또는 스토리지 실패 후 메시지가 복구되도록 합니다.

큐 관리자가 실패하면 클라이언트를 자동으로 큐 관리자 그룹 사이에서 또는 다중 인스턴스 큐 관리자의 새 활성 인스턴스로 전환시키기 위해 클라이언트 애플리케이션 가용성을 향상시키십시오. 자동 클라이언트 재연결이 Java용 WebSphere MQ 클래스에서 지원되지 않습니다.

Windows, UNIX, Linux 및 IBM i 플랫폼에서 서버 애플리케이션을 다중 서버에서 단일 큐 관리자로 실행되도록 구성된 멀티 인스턴스 큐 관리자로 배치하십시오. 활성 인스턴스를 실행하는 서버가 실패한 경우 다른 서버에 있는 같은 큐 관리자의 대기 인스턴스로 실행이 자동 전환됩니다. 서버 애플리케이션이 큐 관리자 서비스로 실행되도록 구성된 경우 대기 인스턴스가 활발히 실행하는 큐 관리자 인스턴스가 될 때 이 서버 애플리케이션은 재시작됩니다.

Microsoft Cluster Server 또는 AIX용 PowerHA(공식적으로는 AIX의 HACMP) 및 기타 UNIX and Linux 클러스터링 솔루션과 같은 플랫폼별 클러스터링 솔루션의 일부로 WebSphere MQ를 구성할 수 있습니다.

서버 애플리케이션 가용성을 높이는 다른 방법은 서버 애플리케이션을 큐 관리자 클러스터에 있는 여러 컴퓨터에 배치하는 것입니다.

메시징 시스템은 시스템에 입력된 메시지가 목적지에 전달되도록 합니다. WebSphere MQ는 **dspmqrte** 명령을 사용하여 메시지가 하나의 큐 관리자에서 다른 큐 관리자로 이동할 때 메시지의 라우트를 추적할 수 있습니다. 시스템이 실패하면 메시지는 실패 유형 및 시스템이 구성되는 방식에 따라 다양한 방식으로 복구될 수 있습니다.

WebSphere MQ는 메시지 수신, 전송 및 전달을 핸들링하는 큐 관리자의 활동에 대한 복구 로그를 유지보수하여 메시지가 손실되지 않았는지 확인합니다. 이 로그 유형은 세 유형의 복구에 이러한 로그를 사용합니다.

1. 복구 재시작(WebSphere MQ를 예정된 방식으로 중지한 경우)
2. 실패 복구(실패가 WebSphere MQ를 중지시킨 경우)
3. 매체 복원, 손상된 오브젝트를 복원하려는 경우

인플라이트 트랜잭션이 롤백되는 경우를 제외한 모든 경우에 복구는 큐 관리자가 중지될 때 인플라이트였던 모든 업데이트를 큐에서 제거하여 큐 관리자가 중지되었을 때의 상태로 큐 관리자를 복원합니다. 복구는 모든 지속 메시지를 복원하고 이 프로세스 중에 비지속 메시지는 손실될 수 있습니다.

자동 클라이언트 다시 연결

여러 개의 컴포넌트를 구성하여 추가 코드를 작성하지 않고 클라이언트 애플리케이션이 자동으로 다시 연결되도록 할 수 있습니다.

자동 클라이언트 다시 연결은 인라인입니다. 클라이언트 애플리케이션 프로그램의 어느 지점에서든지 연결이 자동으로 복원되며, 오브젝트를 열기 위한 핸들이 모두 복원됩니다.

반대로, 수동 재연결에서는 MQCONN 또는 MQCONNX를 사용하여 연결을 다시 작성하고 오브젝트를 다시 열기 위해 클라이언트 응용프로그램이 필요합니다. 자동 클라이언트 다시 연결은 대부분의 클라이언트 애플리케이션에 적합하지만, 그렇다고 해서 모든 클라이언트 애플리케이션에 적합한 것은 아닙니다.

283 페이지의 표 28에서는 클라이언트 워크스테이션에 설치되어야 하는 IBM WebSphere MQ 클라이언트 지원의 가장 초기 릴리스를 나열합니다. 애플리케이션이 자동 클라이언트 다시 연결을 사용하게 하려면 클라이언트 워크스테이션을 다음 레벨 중 하나로 업그레이드해야 합니다. 283 페이지의 표 29에서는 자동 클라이언트 다시 연결을 사용 가능하게 하기 위한 기타 요구사항을 나열합니다.

다시 연결 옵션에 대한 프로그램 액세스가 있는 클라이언트 애플리케이션은 다시 연결 옵션을 설정할 수 있습니다. JMS 및 XMS 클라이언트를 제외하고는 클라이언트 애플리케이션이 재연결에 대한 액세스 옵션을 가지고 있는 경우 재연결 이벤트를 핸들링하기 위한 이벤트 핸들러도 작성할 수 있습니다.

기존 클라이언트 애플리케이션은 재컴파일 및 링크 없이도 다시 연결 지원의 이점을 얻을 수 있습니다.

- 비JMS 클라이언트의 경우 재연결 옵션을 설정하려면 `mqclient.ini` 환경 변수 `DefRecon`을 설정하십시오. CCDT를 사용하여 큐 관리자에 연결하십시오. 클라이언트가 다중 인스턴스 큐 관리자에 연결해야 하는 경우 CCDT에서 활성 및 대기 큐 관리자 인스턴스의 네트워크 주소를 제공하십시오.
- JMS 클라이언트의 경우 연결 팩토리 구성에 있는 재연결 옵션을 설정하십시오. Java EE 환경에 통합된 JMS 클라이언트 또는 WebSphere MQ 자원 어댑터를 사용하는 경우 자동 클라이언트 재연결이 사용 불가능할 수 있습니다. 일부 관리 환경에는 제한사항이 있습니다. 자세한 정보는 [Java SE 및 Java EE 환경에서 자동 클라이언트 재연결 사용](#)을 참조하십시오.

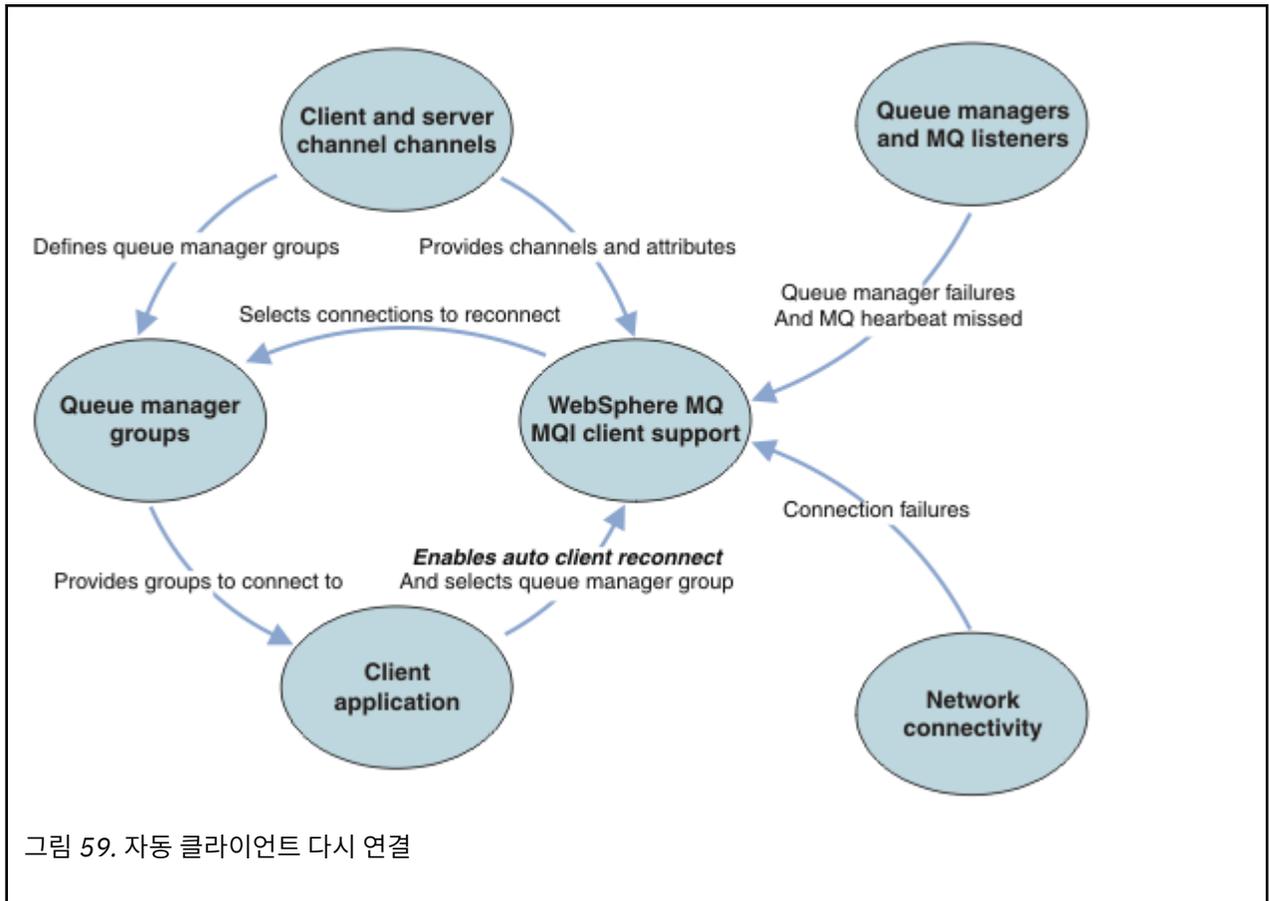
참고: 자동 클라이언트 재연결은 Java의 WebSphere MQ 클래스에서 지원되지 않습니다.

클라이언트 인터페이스	클라이언트	다시 연결 옵션에 대한 프로그램 액세스	다시 연결 지원
메시징 API	C, C++, COBOL, 관리되지 않는 Visual Basic, XMS(Windows의 관리되지 않는 XMS)	7.0.1	7.0.1
	JMS (JSE 및 Java EE 클라이언트 컨테이너 및 관리 컨테이너)	7.0.1.3	7.0.1.3
	Java의 WebSphere MQ 클래스	지원되지 않음	지원되지 않음
	관리되는 XMS 및 관리되는 .NET 클라이언트: C#, Visual Basic	7.1	7.1
기타 API	Windows Communication Foundation(관리되지 않는 ¹)	지원되지 않음	7.0.1
	Windows Communication Foundation(관리되는 ¹)	지원되지 않음	지원되지 않음
	Axis 1	지원되지 않음	지원되지 않음
	Axis 2	지원되지 않음	7.0.1.3
	HTTP(웹 2.0)	지원되지 않음	7.0.1.3

1. WCF 바인딩 구성에서 관리 또는 비관리 모드를 설정하십시오.
 자동으로 다시 연결에는 다음 구성 요구사항이 있습니다.

컴포넌트	요구사항	요구사항을 충족시키지 못하는 경우의 결과
WebSphere MQ MQI 클라이언트 설치	283 페이지의 표 28를 참조하십시오.	MQRC_OPTIONS_ERROR
WebSphere MQ 서버 설치	레벨 7.0.1	MQRC_OPTIONS_ERROR
채널	SHARECNV > 0	MQRC_ENVIRONMENT_ERROR
애플리케이션 환경	스레드되어야 함	MQRC_ENVIRONMENT_ERROR
MQI	다음 중 하나입니다. <ul style="list-style-type: none"> MQCONN 는 MQCNO 옵션 을 MQCNO_RECONNECT 또는 MQCNO_RECONNECT_Q_MGR로 설정합니다. mqclient.ini의 Defrecon=YES QMGR JMS에서 연결 팩토리의 CLIENTRECONNECTOPTIONS 특성 설정. 	연결이 끊어지거나 큐 관리자가 종료되거나 실패할 때 MQCC_FAILED.

284 페이지의 그림 59에서는 클라이언트 다시 연결에 포함된 컴포넌트 간 기본 상호작용을 보여줍니다.



클라이언트 애플리케이션

클라이언트 애플리케이션은 IBM WebSphere MQ MQI 클라이언트입니다.

- 기본적으로 클라이언트는 자동으로 다시 연결되지 않습니다. MQCONNX MQCNO Option MQCNO_RECONNECT 또는 MQCNO_RECONNECT_Q_MGR을 설정하여 자동 클라이언트 다시 연결을 사용 가능하게 하십시오.
- 다수의 애플리케이션은 추가 코드화 없이 자동 연결을 이용할 수 있는 방식으로 작성됩니다. mqclient.ini 구성 파일의 채널 스탠자에서 DefRecon 속성을 설정하여 코딩을 변경하지 않고 기존 프로그램에 대해 자동 재연결을 사용 가능하게 하십시오.
- 다음 세 가지 옵션 중 하나를 사용하십시오.

1. 논리가 다시 연결의 영향을 받지 않도록 프로그램을 수정하십시오. 예를 들어, 동기점 내에서 MQI 호출을 발행하고 백아웃된 트랜잭션을 다시 제출해야 할 수도 있습니다.
2. 다시 연결을 감지하기 위한 이벤트 핸들러를 추가하고 연결이 재설정될 때 클라이언트 애플리케이션의 상태를 복원하십시오.
3. 자동으로 다시 연결을 사용 설정하지 마십시오. 대신 클라이언트의 연결을 끊고 새 MQCONN 또는 MQCONNX MQI 호출을 발행하여 같은 큐 관리자 그룹에서 실행되고 있는 다른 큐 관리자 인스턴스를 찾으십시오.

이러한 세 가지 옵션에 대한 추가 세부사항은 357 페이지의 『애플리케이션 복구』의 내용을 참조하십시오.

- 같은 이름의 큐 관리자에 다시 연결한다고 해도 이 큐 관리자의 같은 인스턴스에 다시 연결되었음을 보장하지는 않습니다.

MQCNO 옵션 MQCNO_RECONNECT_Q_MGR를 사용하여 동일한 큐 관리자의 인스턴스로 재연결하십시오.

- 클라이언트는 다시 연결의 상태에 대한 알림을 받을 수 있도록 이벤트 핸들러에 등록할 수 있습니다. 이벤트 핸들러에 전달된 MQHCONN(를) 사용할 수 없습니다. 다음 이유 코드가 제공됩니다.

MQRC_RECONNECTING

연결에 실패했으며 시스템이 다시 연결을 시도합니다. 다중 재연결 시도가 수행되는 경우 여러 개의 MQRC_RECONNECTING 이벤트가 수신됩니다.

MQRC_RECONNECTED

다시 연결 모드 및 모든 핸들이 성공적으로 재설정되었습니다.

MQRC_RECONNECT_FAILED

다시 연결에 실패했습니다.

MQRC_RECONNECT_QMID_MISMATCH

재연결 가능한 연결 MQCNO_RECONNECT_Q_MGR이(가) 지정되었으며 연결이 다른 큐 관리자에 다시 연결하려고 시도했습니다.

MQRC_RECONNECT_Q_MGR_REQD

MQGET 호출에서 이러한 MQMO_MATCH_MSG_TOKEN 옵션이 동일한 큐 관리자에 다시 연결해야 하는 클라이언트 프로그램에 지정되었습니다.

- 다시 연결 가능 클라이언트는 연결 후에만 자동으로 다시 연결할 수 있습니다. 즉, MQCONNX 호출 자체는 실패한 경우 다시 시도되지 않습니다. 예를 들어, MQCONNX에서 리턴 코드 2543 - MQRC_STANDBY_Q_MGR을 수신하는 경우 짧은 지연 후에 호출을 재발행하십시오.

MQRC_RECONNECT_INCOMPATIBLE

이 이유 코드는 애플리케이션이 재연결 옵션이 설정된 경우 MQPMO_LOGICAL_ORDER (MQPUT 및 MQPUT1포함) 또는 MQGMO_LOGICAL_ORDER (MQGET포함) 을 사용하려고 할 때 리턴됩니다. 이 이유 코드를 리턴하는 이유는 애플리케이션이 그러한 경우에 다시 연결을 사용하지 않도록 하기 위해서입니다.

MQRC_CALL_INTERRUPTED

이 이유 코드는 커밋 호출을 실행하는 중에 연결이 끊기고 클라이언트가 다시 연결되면 리턴됩니다. 동기점 외부 지속 메시지의 MQPUT에 인해서도 같은 이유 코드가 애플리케이션에 리턴됩니다.

다중 인스턴스 큐 관리자

자동 클라이언트 재연결을 사용하여 멀티 인스턴스 큐 관리자가 대기 인스턴스를 활성화한 후 WebSphere MQ MQI 클라이언트 애플리케이션의 재시작을 단순화합니다.

다중 인스턴스 큐 관리자의 대기 인스턴스는 일반적으로 활성 인스턴스에 대한 다른 네트워크 주소에 있습니다. 두 인스턴스 모두의 네트워크 주소를 클라이언트 연결 정의 테이블(CCDT)에 포함시키십시오. **CONNAME** 매개변수에 대한 네트워크 주소 목록을 제공하거나 CCDT에서 큐 관리자에 대한 다중 행을 정의하십시오.

흔히 WebSphere MQ MQI 클라이언트는 큐 관리자 그룹에 있는 큐 관리자에 다시 연결합니다. WebSphere MQ MQI 클라이언트가 동일한 큐 관리자에만 다시 연결하게 하려고 할 수 있습니다. 이는 큐 관리자에 대한 연관관계가 있을 수도 있습니다. 클라이언트가 다른 큐 관리자에 다시 연결하는 것을 막을 수 있습니다. MQCNO 옵션 (MQCNO_RECONNECT_Q_MGR) 을 설정하십시오. WebSphere MQ MQI 클라이언트는 다른 큐 관리자로 재연결할 경우 실패합니다. MQCNO 옵션인 MQCNO_RECONNECT_Q_MGR을 설정하는 경우 같은 큐 관리자 그룹에 다른 큐 관리자를 포함시키지 마십시오. 다시 연결하는 큐 관리자가 원래 연결했던 큐 관리자와 다른 경우 클라이언트는 오류를 리턴합니다.

큐 관리자 그룹

같은 이름의 큐 관리자, 동일한 큐 관리자 또는 클라이언트 연결 테이블에서 같은 QMNAME 값으로 정의되는 일련의 큐 관리자 중 클라이언트 애플리케이션이 항상 연결 및 다시 연결할 대상을 선택할 수 있습니다.

- 클라이언트 채널 정의에 있는 큐 관리자 이름 속성 QMNAME은 큐 관리자 그룹의 이름입니다.
- 클라이언트 응용프로그램에서 MQCONN 또는 MQCONNX QmgrName 매개변수의 값을 큐 관리자 이름으로 설정하면 클라이언트는 해당 이름의 큐 관리자에만 연결됩니다. 큐 관리자 이름에 별표(*)로 접두부를 지정하는 경우 클라이언트는 큐 관리자 그룹에 있는 큐 관리자 중 같은 QMNAME 값을 가지는 큐 관리자에 연결됩니다. 자세한 설명은 [CCDT의 큐 관리자 그룹](#)을 참조하십시오.

큐 공유 그룹

z/OS 큐 공유 그룹에 대한 자동 클라이언트 재연결은 다른 환경과 동일한 재연결 메커니즘을 사용합니다. 클라이언트는 원래 연결에 대해 구성된 것과 같이 선택된 큐 관리자에 다시 연결됩니다. 예를 들어, 클라이언트 채널 정의 테이블을 사용할 때 관리자는 테이블에 있는 모든 입력 항목이 동일한 z/OS 큐 공유 그룹으로 해석되는지 확인해야 합니다.

클라이언트 및 서버 채널 정의

클라이언트 및 서버 채널 정의는 클라이언트 애플리케이션이 다시 연결할 수 있는 큐 관리자 그룹을 정의합니다. 이러한 정의는 다시 연결 선택 및 시간 선택, 그리고 보안과 같은 기타 요소도 결정합니다. 관련 주제를 참조하십시오. 다시 연결에 고려해야 할 가장 관련된 채널 속성이 다음 두 그룹에 나열되어 있습니다.

클라이언트 연결 속성

연결 연관관계(AFFINITY)AFFINITY

연결 연관관계입니다.

클라이언트 채널 가중치(CLNTWGHT)CLNTWGHT

클라이언트 채널 위젯입니다.

연결 이름(CONNAME)CONNAME

연결 정보입니다.

하트비트 간격(HBINT)HBINT

하트비트 간격 서버 연결 채널에서 하트비트 간격을 설정하십시오.

활성 유지(keepalive) 간격(KAINT)KAINT

활성 유지(keepalive) 간격입니다. 서버 연결 채널에서 활성유지(keepalive) 간격을 설정하십시오.

KAINT는 z/OS에만 적용되는 점을 참고하십시오.

큐 관리자 이름(QMNAME)QMNAME

큐 관리자 이름.

서버 연결 속성

하트비트 간격(HBINT)HBINT

하트비트 간격 클라이언트 연결 채널에서 하트비트 간격을 설정하십시오.

활성 유지(keepalive) 간격(KAINT)KAINT

활성 유지(keepalive) 간격입니다. 클라이언트 연결 채널에서 활성유지(keepalive) 간격을 설정하십시오.

KAINT는 z/OS에만 적용되는 점을 참고하십시오.

KAINT는 네트워크 계층 하트비트이고 HBINT는 클라이언트와 큐 관리자 간의 WebSphere MQ 하트비트입니다. 이러한 하트비트를 짧은 시간으로 설정하는 것에는 다음과 같은 두 가지 목적이 있습니다.

1. 연결에서 활동을 시뮬레이션하면 비활성 연결을 닫는 것을 담당하는 네트워크 계층 소프트웨어가 연결을 닫을 가능성이 낮아집니다.
2. 연결이 종료되면 연결이 끊긴 연결이 감지되기 전까지의 지연이 단축됩니다.

기본 TCP/IP 활성 유지(keepalive) 간격은 2시간입니다. KAINT 및 HBINT 속성을 짧게 설정하는 것을 고려해 보십시오. 네트워크의 정상적 작동이 자동으로 다시 연결의 요구에 적합한 것으로 가정하지 마십시오. 예를 들어, 일부 방화벽은 10분 정도 후에 활성 TCP/IP 연결을 종료할 수 있습니다.

네트워크 연결성

네트워크에 의해 WebSphere MQ MQI 클라이언트로 전달된 네트워크 실패만 클라이언트의 자동 재연결 기능으로 핸들링됩니다.

- 전송에 의해 자동으로 수행되는 다시 연결은 IBM WebSphere MQ에 표시되지 않습니다.
- HBINT를 설정하면 WebSphere MQ에 표시되지 않은 네트워크 실패를 처리하는 데 도움이 됩니다.

큐 관리자 및 WebSphere MQ 리스너

클라이언트 다시 연결은 서버 실패, 큐 관리자 실패, 네트워크 연결 실패 및 다른 큐 관리자 인스턴스로 전환하는 관리자에 의해 트리거됩니다.

- 다중 인스턴스 큐 관리자를 사용하는 경우 활성 큐 관리자 인스턴스를 대기 인스턴스로 제어를 전환할 때 클라이언트 다시 연결의 추가 원인이 발생합니다.
- 기본 `endmqm` 명령을 사용하여 큐 관리자를 종료하면 자동 클라이언트 다시 연결을 트리거하지 않습니다. `endmqm` 명령에 `-r` 옵션을 추가하여 자동 클라이언트 재연결을 요청하거나 `-s` 옵션을 추가하여 시스템 종료 후 대기 큐 관리자 인스턴스로 전송하십시오.

WebSphere MQ MQI 클라이언트 자동 재연결 지원

WebSphere MQ MQI 클라이언트에서 자동 클라이언트 재연결 지원을 사용하는 경우, 큐 관리자에 다시 연결하기 위해 MQCONN 또는 MQCONNX MQI 호출을 발행하지 않아도 클라이언트 애플리케이션이 자동으로 다시 연결하여 처리를 계속 수행합니다.

- 자동 클라이언트 다시 연결은 다음의 경우 중 하나에 의해 처리됩니다.
 - 큐 관리자 실패
 - 큐 관리자가 종료되고 **endmqm** 명령에 다시 연결 옵션인 **-r** 지정
- MQCONNX MQCNO 옵션은 자동 클라이언트 다시 연결을 사용 설정했는지 여부를 제어합니다. 이 옵션은 [재연결 옵션](#)에 설명되어 있습니다.
- 자동 클라이언트 다시 연결은 프로그램이 끊어진 연결로 인해 발생한 MQI 오류를 처리한 후에 정상적인 처리를 계속할 수 있도록 애플리케이션을 대신해서 MQI 호출을 발행하여 연결 핸들 및 다른 열린 오브젝트에 대한 핸들을 복원합니다. [359 페이지의 『자동으로 다시 연결된 클라이언트의 복구』](#)를 참조하십시오.
- 연결에 대한 채널 엑시트 프로그램을 작성한 경우 엑시트는 이러한 추가 MQI 호출을 수신합니다.
- 다시 연결이 시작되고 끝날 때 트리거되는 다시 연결 이벤트 핸들러를 등록할 수 있습니다.

재연결에 걸리는 시간이 1분을 초과하지 않지만 큐 관리자가 관리할 자원이 많을 수 있으므로 재연결 시간이 더 오래 걸릴 수 있습니다. 이 시간 동안 클라이언트 애플리케이션은 WebSphere MQ 자원에 속하지 않은 자원에 잠금을 보유하고 있을 수 있습니다. 클라이언트가 다시 연결을 대기하는 시간을 제한하기 위해 구성할 수 있는 제한 시간 값이 있습니다. 값(초)은 `mqclient.ini` 파일에서 설정됩니다.

```
Channels:  
MQReconnectTimeout = 1800
```

제한시간이 만료된 후에는 다시 연결 시도가 이루어지지 않습니다. 시스템이 제한시간이 만료되었음을 감지하면 MQRC_RECONNECT_FAILED 오류를 리턴합니다.

콘솔 메시지 모니터링

큐 관리자 또는 채널 시작기가 발행하는 특별히 중요하게 고려해야 하는 다양한 정보 메시지가 있습니다. 이러한 메시지는 본래 문제점을 표시하지는 않지만 잠재적으로 처리가 필요한 잠재적인 문제점을 나타내기 때문에 추적에 유용할 수 있습니다.

또한 이 메시지가 존재하는 것은 사용자 애플리케이션이 다수의 메시지를 페이지 세트에 넣는 중임을 표시할 수 있으며 이는 더 큰 문제점의 증상일 수 있습니다.

- 메시지를 넣는 사용자 애플리케이션(예: 제어되지 않은 루프)에 대한 문제점
- 큐에서 메시지를 가져오는 사용자 애플리케이션은 더 이상 작동하지 않습니다.

고가용성 구성으로 WebSphere MQ 사용

고가용성 (HA) 구성에서 WebSphere MQ 큐 관리자를 조작하려는 경우, 큐 관리자가 고가용성 관리자 (예: PowerHA AIX (이전에는 HACMP) 또는 마이크로소프트 클러스터 서비스 (MSCS)) 또는 WebSphere MQ 다중 인스턴스 큐 관리자를 사용하여 작업할 수 있도록 설정할 수 있습니다.

다음 구성 정의에 대해 알고 있어야 합니다.

큐 관리자 클러스터

하나 이상의 컴퓨터에 있는 둘 이상의 큐 관리자 그룹으로, 자동 상호연결을 제공하고 로드 밸런싱 및 중복성을 위해 큐가 서로 공유될 수 있도록 합니다.

HA 클러스터

HA 클러스터는 두 개 이상의 컴퓨터 및 자원(예: 디스크 및 네트워크)의 그룹이며 함께 연결되어 있고 하나가 실패하면 고가용성 관리자(예: HACMP(UNIX) 또는 MSCS(Windows))가 장애 복구를 수행하는 방법으로 구성되어 있습니다. 장애 복구는 애플리케이션의 상태 데이터를 실패한 컴퓨터에서 클러스터 내 다른 컴퓨터로 전송하고 그 위치에서 해당 조작을 다시 시작합니다. 이는 HA 클러스터 내에서 실행되는 서비스의 고가용성을 제공합니다. IBM WebSphere MQ 클러스터와 HA 클러스터 간 관계는 [288 페이지의 『큐 관리자 클러스터에 대한 HA 클러스터의 관계』](#)에 설명되어 있습니다.

다중 인스턴스 큐 관리자

둘 이상의 컴퓨터에서 구성된 동일한 큐 관리자의 인스턴스입니다. 다중 인스턴스를 시작하면 한 인스턴스가 활성 인스턴스가 되고 나머지 다른 인스턴스는 대기 인스턴스가 됩니다. 활성 인스턴스가 실패할 경우 다른 컴퓨터에서 실행 중인 대기 인스턴스가 자동으로 인계받습니다. 멀티 인스턴스 큐 관리자를 사용하여 HACMP 또는 MSCS와 같은 클러스터 기술이 필요 없이 WebSphere MQ에 기반하여 고유의 고가용성 메시징 시스템을 구성할 수 있습니다. HA 클러스터 및 다중 인스턴스 큐 관리자는 큐 관리자의 가용성을 높이기 위한 대체 방법입니다. HA 클러스터에 다중 인스턴스 큐 관리자를 넣어 결합하지 마십시오.

다중 인스턴스 큐 관리자와 HA 클러스터 간 차이점

다중 인스턴스 큐 관리자와 HA 클러스터는 큐 관리자에 대한 높은 가용성을 얻기 위한 대체 방법입니다. 다음은 두 가지 접근법 간 차이점을 강조하는 몇 가지 포인트입니다.

다중 인스턴스 큐 관리자에는 다음과 같은 기능이 있습니다.

- WebSphere MQ에 통합된 기본 장애 복구 지원
- HA 클러스터보다 빠른 장애 복구
- 단순한 구성 및 조작
- WebSphere MQ Explorer와 통합

다중 인스턴스 큐 관리자의 제한사항은 다음과 같습니다.

- 가용성이 높고 고성능 네트워크 스토리지가 필요함
- 큐 관리자가 장애 복구 시 IP 주소를 변경함으로써 인한 복잡한 네트워크 구성

HA 클러스터에는 다음과 같은 기능이 있습니다.

- 애플리케이션 서버 또는 데이터베이스와 같은 여러 자원을 조정하는 기능
- 셋 이상의 노드로 구성된 클러스터 등 보다 유연한 구성 옵션
- 운영자의 개입 없이도 여러 번 장애 복구 가능
- 장애 복구의 일부로 큐 관리자의 IP 주소 인계

HA 클러스터의 제한사항은 다음과 같습니다.

- 추가 제품 구매 및 기술이 필요함
- 클러스터의 노드 사이에서 전환할 수 있는 디스크가 필요함
- HA 클러스터의 구성이 상대적으로 복잡함
- 과거에는 장애 복구가 비교적 느렸지만 최근 HA 클러스터 제품에서는 개선됨
- 큐 관리자와 같은 자원을 모니터링하는 데 사용되는 스크립트에 결함이 있는 경우 불필요한 장애 복구가 발생할 수 있음

큐 관리자 클러스터에 대한 HA 클러스터의 관계

큐 관리자 클러스터는 관리 작업을 줄이고 큐 관리자 클러스터 큐의 인스턴스 전체에서 메시지의 로드 밸런싱을 제공합니다. 이는 또한 단일 큐 관리자보다 높은 가용성을 제공합니다. 큐 관리자의 실패 후에도 메시징 애플리케이션이 큐 관리자 클러스터 큐의 남아 있는 인스턴스에 대해 계속해서 액세스할 수 있기 때문입니다. 그러나 큐 관리자 클러스터만으로는 큐 관리자 실패의 자동 감지 및 큐 관리자 재시작 또는 장애 복구의 자동 트리거를 제공하지 않습니다. HA 클러스터는 이러한 기능을 제공합니다. 좋은 효과를 얻기 위해 두 유형의 클러스터를 함께 사용할 수 있습니다.

UNIX and Linux의 고가용성 클러스터에서 WebSphere MQ 사용

UNIX and Linux 플랫폼에서 고가용성 (HA) 클러스터와 함께 WebSphere MQ 를 사용할 수 있습니다. 예를 들어, AIX (이전에는 HACMP), Veritas Cluster Server, HP Serviceguard 또는 Red Hat Cluster Suite와 함께 Red Hat Enterprise Linux 클러스터를 사용할 PowerHA 있습니다.

WebSphere MQ 버전 7.0.1 이전에는 HA 클러스터 구성을 지원하기 위해 SupportPac MC91를 제공했습니다. WebSphere MQ 버전 7.0.1은 큐 관리자가 데이터를 저장하는 이전 버전보다 큰 등급의 제어를 제공했습니다.

이는 HA 클러스터에서 큐 관리자를 구성하는 것을 더욱 용이하게 합니다. SupportPac MC91에서 제공되는 스크립트의 대부분은 더 이상 필요하지 않으며 SupportPac의 제공은 중단됩니다.

이 절에서는 289 페이지의 『HA 클러스터 구성』, 큐 관리자 클러스터에 대한 HA 클러스터의 관계, 289 페이지의 『WebSphere MQ 클라이언트』 및 289 페이지의 『HA 클러스터에서 WebSphere MQ 조작』에 대해 소개하며 단계를 통해 안내하고 HA 클러스터와 큐 관리자 구성에 맞게 조정할 수 있는 스크립트 예를 제공합니다.

이 절에서 설명하는 구성 단계에 대한 지원은 사용자 환경에 맞는 HA 클러스터 문서를 참조하십시오.

HA 클러스터 구성

이 절에서 용어 노드는 운영 체제 및 HA 소프트웨어를 실행하는 엔티티를 참조하는 데 사용되며 "컴퓨터", "시스템" 또는 "머신" 또는 "파티션" 또는 "블레이드"는 이 사용법에서 동의어로 간주될 수 있습니다. 모든 클러스터 노드가 WebSphere MQ 워크로드를 실행하고 있는 상호 인수를 포함하여 대기 또는 인수 구성을 설정하는 데 도움이 되도록 WebSphere MQ를 사용할 수 있습니다.

대기 구성은 한 노드가 작업을 수행하는 동안 다른 노드는 대기 역할만 하는 가장 기본적인 HA 클러스터입니다. 대기 노드는 작업을 수행하지 않으며 유휴로 참조됩니다. 이 구성은 경우에 따라 콜드 대기라고 합니다. 이러한 구성에는 높은 수준의 하드웨어 중복성이 필요합니다. 하드웨어를 절약하기 위해 단일 대기 노드와 다중 작업자 노드를 가지도록 이 구성을 확장할 수 있습니다. 요점은 대기 노드가 다른 작업자 노드의 작업을 인수할 수 있다는 것입니다. 이 구성은 계속해서 대기 구성으로 참조되며 "N+1" 구성이라고 하는 경우도 있습니다.

인수 구성은 모든 노드가 일부 작업을 수행하고 노드 실패가 발생할 경우 중요한 작업은 인계될 수 있는 고급 구성입니다.

한 측 인수 구성은 대기 노드가 일부 추가 작업, 비중요 작업 및 이동 불가능한 작업을 수행하는 구성입니다. 이 구성은 대기 구성과 비슷하지만 (비중요) 작업이 대기 노드에 의해 수행됩니다.

상호 인수 구성은 모든 노드가 높은 가용성의 (이동 가능) 작업을 수행하는 구성입니다. 이 유형의 HA 클러스터 구성은 가끔 "활성/활성"으로도 참조되어 모든 노드가 중요한 워크로드를 활발하게 처리함을 표시합니다.

확장된 대기 구성 또는 인수 구성 중 하나에서 다른 노드의 작업을 인수할 수 있는 노드에 배치될 수 있는 피크 로드를 고려하는 것은 중요합니다. 그러한 노드는 허용 가능한 수준의 성능을 유지하기 위해 충분한 용량을 소유해야 합니다.

큐 관리자 클러스터에 대한 HA 클러스터의 관계

큐 관리자 클러스터는 관리 작업을 줄이고 큐 관리자 클러스터 큐의 인스턴스 전체에서 메시지의 로드 밸런싱을 제공합니다. 이는 또한 단일 큐 관리자보다 높은 가용성을 제공합니다. 큐 관리자의 실패 후에도 메시징 애플리케이션이 큐 관리자 클러스터 큐의 남아 있는 인스턴스에 대해 계속해서 액세스할 수 있기 때문입니다. 그러나 큐 관리자 클러스터만으로는 큐 관리자 실패의 자동 감지 및 큐 관리자 재시작 또는 장애 복구의 자동 트리거를 제공하지 않습니다. HA 클러스터는 이러한 기능을 제공합니다. 좋은 효과를 얻기 위해 두 유형의 클러스터를 함께 사용할 수 있습니다.

WebSphere MQ 클라이언트

재시작 또는 인수에 사용될 수 있는 큐 관리자와 통신 중인 WebSphere MQ 클라이언트는 끊어진 연결을 허용하도록 작성되어야 하며 반복적으로 재연결을 시도해야 합니다. WebSphere MQ 버전 7은 연결 사용 가능성 및 워크로드 밸런싱을 지원하는 클라이언트 채널 정의 테이블(CCDT) 처리에서의 기능을 소개했습니다. 그러나 이러한 기능은 장애 복구 시스템에서 작업할 때 직접적으로 관련되지 않습니다.

WebSphere MQ MQI 클라이언트가 2단계 트랜잭션에 참여할 수 있도록 하는 ETC(확장 트랜잭션 클라이언트)는 항상 동일한 큐 관리자에 연결해야 합니다. ETC는 큐 관리자 목록에서 선택하기 위해 IP 로드 밸런서와 같은 기술을 사용할 수 없습니다. HA 제품을 사용할 때 큐 관리자가 어떤 노드에서 실행 중이든 해당 ID(이름 및 주소)를 유지보수하므로 ETC는 HA 제어 아래에 있는 큐 관리자에서 사용될 수 있습니다.

HA 클러스터에서 WebSphere MQ 조작

모든 HA 클러스터에는 장애 복구 단위의 개념이 있습니다. 이는 고가용성 서비스를 구성하는 모든 자원을 포함하는 일련의 정의입니다. 장애 복구 단위는 서비스 자체 및 서비스가 의존하는 다른 모든 자원을 포함합니다.

HA 솔루션은 장애 복구 단위에 다른 용어를 사용합니다.

- AIX용 PowerHA에서 장애 복구 단위는 자원 그룹이라고 합니다.
- Veritas Cluster Server에서는 이를 서비스 그룹이라고 합니다.
- Serviceguard에서는 패키지라고 합니다.

이 주제에서는 장애 복구 단위를 의미하기 위해 자원 그룹이라는 용어를 사용합니다.

WebSphere MQ의 최소 장애 복구 단위는 큐 관리자입니다. 일반적으로 큐 관리자를 포함하는 자원 그룹은 자원 그룹에서 사용하도록 독점 예약된 볼륨 그룹 또는 디스크 그룹의 공유 디스크 및 큐 관리자에 연결하는 데 사용되는 IP 주소도 포함합니다. 또한 기타 WebSphere MQ 자원(예를 들어, 동일한 자원 그룹의 리스너 또는 트리거 모니터)을 분리된 자원으로 또는 큐 관리자 자체의 제어 아래에 포함할 수 있습니다.

HA 클러스터에서 사용되는 큐 관리자는 클러스터의 노드 사이에서 공유되는 디스크에 그 데이터 및 로그가 있어야 합니다. HA 클러스터는 한 번에 클러스터의 한 노드만 디스크에 쓸 수 있도록 합니다. HA 클러스터는 모니터 스크립트를 사용하여 큐 관리자의 상태를 모니터링할 수 있습니다.

큐 관리자와 관련된 데이터 및 로그 둘 다에 대해 단일 공유 디스크를 사용할 수 있습니다. 그러나 개별적으로 크기를 지정하고 조정할 수 있도록 개별 공유 파일 시스템을 사용하는 것이 일반적인 사례입니다.

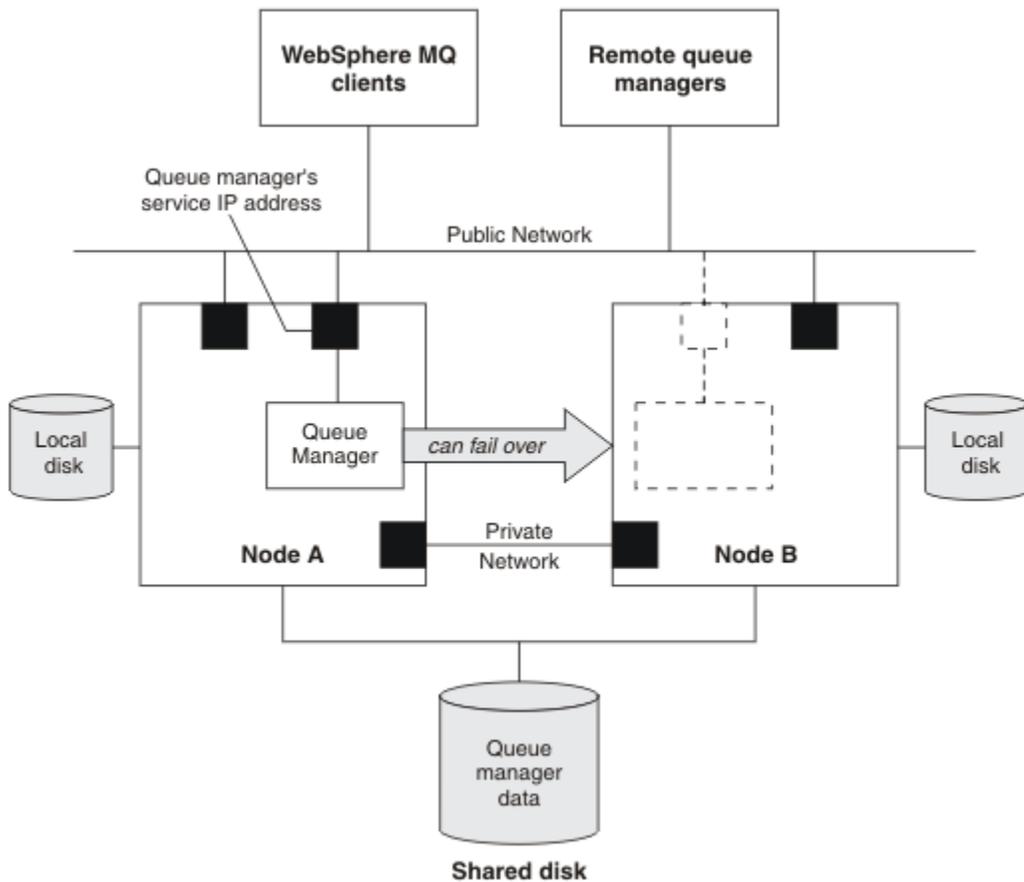


그림 60. HA 클러스터(HA cluster)

그림 1은 두 개의 노드가 있는 HA 클러스터를 보여줍니다. HA 클러스터는 자원 그룹에서 정의된 큐 관리자의 가용성을 관리합니다. 이는 활성/수동 또는 콜드 대기 구성입니다. 한 개의 노드인 노드 A가 현재 큐 관리자를 실행하고 있기 때문입니다. 이 큐 관리자는 공유 디스크에 있는 데이터와 로그 파일로 작성되었습니다. 큐 관리자에는 HA 클러스터가 관리하는 서비스 IP 주소가 있습니다. 큐 관리자는 공유 디스크 및 그 서비스 IP 주소에 따라 달라집니다. HA 클러스터가 노드 A에서 노드 B까지 큐 관리자의 장애를 복구하는 경우 먼저 큐 관리자의 종속 자원을 노드 B로 이동한 다음 큐 관리자를 시작합니다.

HA 클러스터에 둘 이상의 큐 관리자가 있을 경우 사용자의 HA 클러스터 구성으로 인해 장애 복구 후 동일 노드에서 둘 이상의 큐 관리자가 실행될 수 있습니다. HA 클러스터의 각 큐 관리자에는 큐 관리자가 특정 시간에 활성이 될 수도 있는 클러스터 노드에서 사용하는 자체 포트 번호가 지정되어야 합니다.

일반적으로 HA 클러스터는 루트 사용자로 실행됩니다. WebSphere MQ는 mqm 사용자로 실행합니다. WebSphere MQ의 관리는 mqm 그룹의 멤버에 부여됩니다. mqm 사용자 및 그룹 둘 다 모든 HA 클러스터 노드에 존재해야 합니다. 사용자 ID 및 그룹 ID는 클러스터 전체에서 일관되어야 합니다. 루트 사용자에 의한 WebSphere MQ 관리는 허용되지 않습니다. 스크립트를 시작, 중지 또는 모니터하는 스크립트는 mqm 사용자로 전환해야 합니다.

참고: WebSphere MQ는 모든 노드에 올바르게 설치되어야 하며, 사용자는 제품 실행 파일을 공유할 수 없습니다.

공유 디스크 구성

HA 클러스터에 있는 WebSphere MQ 큐 관리자는 데이터 파일 및 로그 파일이 공유 디스크에서 공용으로 이름 지정된 원격 파일 시스템에 있어야 합니다.

공유 디스크를 구성하려면 다음 단계를 완료하십시오.

1. 큐 관리자의 파일 시스템의 마운트 지점 이름을 결정하십시오. 예를 들어, 큐 관리자 데이터 파일의 경우 /MQHA/qmgrname/data이고 해당 로그 파일의 경우 /MQHA/qmgrname/log입니다.
2. 큐 관리자의 데이터 및 로그 파일이 포함될 볼륨 그룹(또는 디스크 그룹)을 작성하십시오. 이 볼륨 그룹은 동일한 자원 그룹의 고가용성(HA) 클러스터에 의해 큐 관리자로 관리됩니다.
3. 볼륨 그룹의 로그 파일과 큐 관리자의 데이터를 위한 파일 시스템을 작성하십시오.
4. 각 노드에 대해 차례대로 파일 시스템의 마운트 지점을 작성하고 파일 시스템을 마운트할 수 있는지 확인하십시오. mqm 사용자는 마운트 지점을 소유해야 합니다.

그림 1은 HA 클러스터에서 큐 관리자에 대해 가능한 레이아웃을 표시합니다. 큐 관리자의 데이터 및 로그 디렉토리는 둘 다 /MQHA/QM1에 마운트되는 공유 디스크에 있습니다. 이 디스크는 장애 복구가 발생하면 큐 관리자가 재시작되는 위치에 관계없이 데이터가 사용 가능하도록 HA 클러스터의 노드 사이에서 전환됩니다. mqs.ini 파일에는 QM1 큐 관리자에 대한 스탠자가 있습니다. qm.ini 파일의 Log 스탠자에는 LogPath의 값이 있습니다.

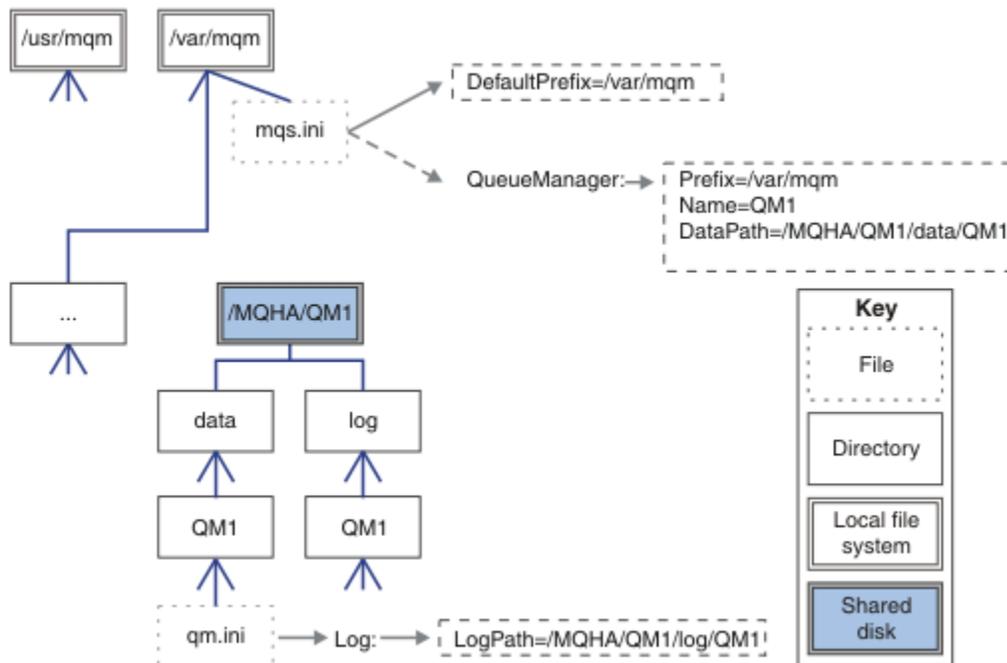


그림 61. 공유 이름 지정된 data 및 log 디렉토리

고가용성(HA) 클러스터에서 사용할 큐 관리자 작성

고가용성 클러스터에서 큐 관리자를 사용하기 위한 첫 번째 단계는 노드 중 하나에서 큐 관리자를 작성하는 것입니다.

HA 클러스터에서 사용할 큐 관리자를 작성하려면 클러스터에 있는 노드 중 큐 관리자를 작성할 한 노드를 선택하십시오. 이 노드에서 다음 단계를 완료하십시오.

1. 큐 관리자의 파일 시스템을 노드에 마운트하십시오.

2. **crtmqm** 명령을 사용하여 큐 관리자를 작성하십시오. 예를 들면, 다음과 같습니다.

```
crtmqm -md /MQHA/qmgrname/data -ld /MQHA/qmgrname/log qmgrname
```

3. **strmqm** 명령을 사용하여 큐 관리자를 수동으로 시작하십시오.

4. 큐 및 채널 작성, 큐 관리자를 시작할 때 리스너가 자동으로 시작되도록 큐 관리자 설정과 같은 큐 관리자의 초기 구성을 완료하십시오.

5. **endmqm** 명령을 사용하여 큐 관리자를 중지하십시오.

6. **dspmqinf** 명령을 사용하여 나중 태스크에서 사용할 수 있는 **addmqinf** 명령을 표시하십시오. 이 명령은 292 페이지의 『큐 관리자 구성 정보를 고가용성(HA) 클러스터에 있는 다른 노드에 추가』에 문서화되어 있습니다.

```
dspmqinf -o command qmgrname
```

여기서, **qmgrname**은 큐 관리자의 이름입니다.

7. 표시되는 **addmqinf** 명령은 다음 예와 비슷합니다.

```
addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \  
-vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
```

표시되는 명령을 주의하여 기록해 두십시오.

8. 큐 관리자의 파일 시스템을 마운트 취소하십시오.

이제 292 페이지의 『큐 관리자 구성 정보를 고가용성(HA) 클러스터에 있는 다른 노드에 추가』에 설명된 단계를 완료할 준비가 되었습니다.

큐 관리자 구성 정보를 고가용성(HA) 클러스터에 있는 다른 노드에 추가

큐 관리자 구성을 HA 클러스터에 있는 다른 노드에 추가해야 합니다.

이 태스크를 완료하기 전에 291 페이지의 『고가용성(HA) 클러스터에서 사용할 큐 관리자 작성』의 단계를 이미 완료한 상태여야 합니다.

큐 관리자의 구성 정보를 HA 클러스터에 있는 다른 노드 각각에 추가하려면 각 노드에 대해 다음 단계를 완료하십시오.

1. 큐 관리자 파일 시스템을 마운트하십시오.

2. `/var/mqm/mqs.ini`를 직접 편집하거나 291 페이지의 『고가용성(HA) 클러스터에서 사용할 큐 관리자 작성』의 단계 6과 7에서 **dspmqinf** 명령을 통해 표시된 **addmqinf** 명령을 발행하여 큐 관리자 구성 정보를 노드에 추가하십시오.

3. 큐 관리자를 시작 및 중지하여 구성을 확인하십시오.

큐 관리자를 시작 및 중지하는 데 사용되는 명령은 **addmqinf** 명령과 동일한 IBM WebSphere MQ 설치에서 실행해야 합니다. 다른 설치에서 큐 관리자를 시작하고 중지하려면 먼저 **setmqm** 명령을 사용하여 큐 관리자와 연관된 설치를 설정해야 합니다. 자세한 정보는 `setmqm`을 참조하십시오.

4. 큐 관리자 파일 시스템을 마운트 취소하십시오.

고가용성(HA) 클러스터의 제어 하에 큐 관리자 시작

큐 관리자는 HA 클러스터에서 자원으로 표시됩니다. HA 클러스터는 큐 관리자를 시작하고 중지할 수 있어야 합니다. 대부분의 경우 셸 스크립트를 사용하여 큐 관리자를 시작할 수 있습니다. 네트워크 파일 시스템을 사용하거나 각 로컬 디스크에 복사하여 클러스터에 있는 모든 노드의 같은 위치에서 이러한 스크립트가 사용 가능하게 해야 합니다.

참고: 실패한 큐 관리자를 재시작하기 전에 큐 관리자의 해당 인스턴스로부터 애플리케이션의 연결을 끊어야 합니다. 이를 수행하지 않으면 큐 관리자가 올바르게 재시작하지 않을 수 있습니다.

적당한 셸 스크립트의 예가 여기에 제공됩니다. 이러한 셸 스크립트를 필요에 맞게 조정하여 HA 클러스터의 제어 하에 큐 관리자를 시작하는 데 사용할 수 있습니다.

다음 셸 스크립트는 큐 관리자가 성공적으로 시작할 수 있도록 HA 클러스터 사용자에서 mqm 사용자로 전환하는 방법의 예입니다.

```
#!/bin/ksh
#
# A simple wrapper script to switch to the mqm user.
su mqm -c name_of_your_script $*
```

다음 셸 스크립트는 큐 관리자의 현재 상태에 대한 어떠한 가정을 하지 않고도 큐 관리자를 시작하는 방법의 예입니다. 이 셸 스크립트는 큐 관리자에 속한 프로세스를 종료하는 데 매우 돌발적인 메소드를 사용함을 참고하십시오.

```
#!/bin/ksh
#
# This script robustly starts the queue manager.
#
# The script must be run by the mqm user.

# The only argument is the queue manager name. Save it as QM variable
QM=$1

if [ -z "$QM" ]
then
    echo "ERROR! No queue manager name supplied"
    exit 1
fi

# End any queue manager processes which might be running.

srchstr="( |-m)$QM *.*$"
for process in amqzmuc0 amqzma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrsta amqirmfa amqrmppa \
               amqzfuma amqzmaa amqmuf0 amqzmur0 amqzmgr0
do
    ps -ef | tr "\t" " " | grep $process | grep -v grep | \
    egrep "$srchstr" | awk '{print $2}' | \
    xargs kill -9 > /dev/null 2>&1
done

# It is now safe to start the queue manager.
# The stmqm command does not use the -x flag.
stmqm ${QM}
```

다른 관련된 프로그램을 시작하도록 스크립트를 수정할 수 있습니다.

고가용성(HA) 클러스터의 제어 하에 큐 관리자 중지

대부분의 경우 셸 스크립트를 사용하여 큐 관리자를 중지할 수 있습니다. 적당한 셸 스크립트의 예가 여기에 제공 됩니다. 이러한 셸 스크립트를 필요에 맞게 조정하여 HA 클러스터의 제어 하에 큐 관리자를 중지하는 데 사용할 수 있습니다.

다음 스크립트는 큐 관리자의 현재 상태에 대한 어떠한 가정을 하지 않고도 즉시 중지하는 방법의 예입니다. 이 스크립트는 mqm 사용자에게 의해 실행되어야 합니다. 따라서 사용자를 HA 클러스터 사용자에서 mqm으로 전환 하려면 이 스크립트를 셸 스크립트에 넣을 필요가 있습니다(셸 스크립트 예는 [292 페이지의 『고가용성\(HA\) 클러스터의 제어 하에 큐 관리자 시작』](#)에 제공되었습니다).

```
#!/bin/ksh
#
# The script ends the QM by using two phases, initially trying an immediate
# end with a time-out and escalating to a forced stop of remaining
# processes.
#
# The script must be run by the mqm user.
#
# There are two arguments: the queue manager name and a timeout value.
QM=$1
TIMEOUT=$2

if [ -z "$QM" ]
then
    echo "ERROR! No queue manager name supplied"
    exit 1
fi
```

```

if [ -z "$TIMEOUT" ]
then
  echo "ERROR! No timeout specified"
  exit 1
fi

for severity in immediate brutal
do
  # End the queue manager in the background to avoid
  # it blocking indefinitely. Run the TIMEOUT timer
  # at the same time to interrupt the attempt, and try a
  # more forceful version. If the brutal version fails,
  # nothing more can be done here.

  echo "Attempting ${severity} end of queue manager '${QM}'"
  case $severity in

immediate)
  # Minimum severity of endmqm is immediate which severs connections.
  # HA cluster should not be delayed by clients
  endmqm -i ${QM} &
  ;;

brutal)
  # This is a forced means of stopping queue manager processes.

  srchstr="(|-m)$QM *.*$"
  for process in amqzmuc0 amqzxma0 amqfcxba amqfqpub amqpcsea amqzlaa0 \
    amqzlsa0 runmqchi runmqlsr amqcrsta amqirmfa amqimppa \
    amqzfuma amqzmaa amqzmuf0 amqzmur0 amqzmgr0
  do
    ps -ef | tr "\t" " " | grep $process | grep -v grep | \
      egrep "$srchstr" | awk '{print $2}' | \
        xargs kill -9 > /dev/null 2>&1
  done

  esac

  TIMED_OUT=yes
  SECONDS=0
  while (( $SECONDS < ${TIMEOUT} ))
  do
    TIMED_OUT=yes
    i=0
    while [ $i -lt 5 ]
    do
      # Check for execution controller termination
      srchstr="(|-m)$QM *.*$"
      cnt=`ps -ef | tr "\t" " " | grep amqzxma0 | grep -v grep | \
        egrep "$srchstr" | awk '{print $2}' | wc -l`
      i=`expr $i + 1`
      sleep 1
      if [ $cnt -eq 0 ]
      then
        TIMED_OUT=no
        break
      fi
    done

    if [ ${TIMED_OUT} = "no" ]
    then
      break
    fi

    echo "Waiting for ${severity} end of queue manager '${QM}'"
    sleep 1
  done # timeout loop

  if [ ${TIMED_OUT} = "yes" ]
  then
    continue # to next level of urgency
  else
    break # queue manager is ended, job is done
  fi

done # next phase

```

큐 관리자 모니터링

일반적으로 고가용성(HA) 클러스터가 큐 관리자의 상태를 주기적으로 모니터링하는 방법이 제공됩니다. 대부분의 경우 이에 셸 스크립트를 사용할 수 있습니다. 적당한 셸 스크립트의 예가 여기에 제공됩니다. 이러한 스크립트를 필요에 맞게 조정하여 사용자 환경에 맞는 추가 모니터링 검사를 수행하는 데 사용할 수 있습니다.

WebSphere MQ 버전 7.1에는 시스템에 WebSphere MQ 설치가 여러 개 공존할 수 있습니다. 다중 설치에 대한 자세한 정보는 [다중 설치](#)를 참조하십시오. 버전 7.1 이상에서의 설치를 포함한 다중 설치에서 모니터링 스크립트를 사용하려면 몇 가지 추가적인 단계를 수행해야 할 수 있습니다. 기본 설치를 수행했거나 7.1 이전 버전에서 스크립트를 사용하고 있는 경우에는 스크립트를 사용하도록 `MQ_INSTALLATION_PATH`를 지정하지 않아도 됩니다. 그렇지 않으면 다음 단계에서 `MQ_INSTALLATION_PATH`가 올바르게 식별되는지 확인하십시오.

1. 버전 7.1 설치에서 `crtmqenv` 명령을 사용하여 큐 관리자에 대한 올바른 `MQ_INSTALLATION_PATH`를 식별하십시오.

```
crtmqenv -m qmname
```

이 명령은 `qmname`에 지정된 큐 관리자에 대해 올바른 `MQ_INSTALLATION_PATH` 값을 리턴합니다.

2. 적절한 `qmname` 및 `MQ_INSTALLATION_PATH` 매개변수를 사용하여 모니터링 스크립트를 실행하십시오.

참고: AIX용 PowerHA는 큐 관리자를 위한 모니터링 프로그램에 매개변수를 제공하는 방법을 제공하지 않습니다. 각 큐 관리자에 대해 큐 관리자 이름을 캡슐화하는 개별 모니터링 프로그램을 작성해야 합니다. 다음은 큐 관리자 이름을 요약하는 AIX에서 사용되는 스크립트의 예입니다.

```
#!/bin/ksh
su mqm -c name_of_monitoring_script qmname MQ_INSTALLATION_PATH
```

여기서 `MQ_INSTALLATION_PATH`는 `qmname` 큐 관리자와 연관된 IBM WebSphere MQ의 설치 경로를 지정하는 선택적 매개변수입니다.

다음 스크립트는 `runmqsc`가 정지될 가능성에 대해 확신하지 않습니다. 일반적으로 HA 클러스터는 정지 모니터링 스크립트를 실패로 처리하고 스스로 이 가능성에 대해 확신합니다.

그러나 이 스크립트는 큐 관리자가 시작 상태인 것은 허용합니다. 이는 HA 클러스터가 큐 관리자가 시작되자마자 큐 관리자를 모니터링하기 시작하는 것이 일반적이기 때문입니다. 일부 HA 클러스터는 시작 단계와 자원에 대한 실행 단계를 구별하지만 시작 단계의 지속 기간을 구성해야 합니다. 큐 관리자를 시작하는 데 걸리는 시간은 수행해야 하는 작업의 양에 따라 다르기 때문에 큐 관리자를 시작하는 데 걸리는 최대 시간을 선택하기는 어렵습니다. 너무 낮은 값을 선택하면 HA 클러스터는 큐 관리자가 시작을 완료하지 않은 경우에 실패한 것으로 잘못 가정합니다. 이로 인해 장애 복구가 끝없이 연속되는 결과가 나타날 수 있습니다.

이 스크립트는 `mqm` 사용자에게 의해 실행되어야 하므로 사용자를 HA 클러스터에서 `mqm`으로 전환하기 위해 셸 스크립트에 이 스크립트를 래핑해야 할 수도 있습니다(셸 스크립트에는 [292 페이지의 『고가용성\(HA\) 클러스터의 제어 하에 큐 관리자 시작』](#)에 제공되어 있음).

```
#!/bin/ksh
#
# This script tests the operation of the queue manager.
#
# An exit code is generated by the runmqsc command:
# 0 => Either the queue manager is starting or the queue manager is running and responds.
#     Either is OK.
# >0 => The queue manager is not responding and not starting.
#
# This script must be run by the mqm user.
QM=$1
MQ_INSTALLATION_PATH=$2

if [ -z "$QM" ]
then
echo "ERROR! No queue manager name supplied"
exit 1
fi

if [ -z "$MQ_INSTALLATION_PATH" ]
then
# No path specified, assume system primary install or MQ level < 7.1.0.0
echo "INFO: Using shell default value for MQ_INSTALLATION_PATH"
else
echo "INFO: Prefixing shell PATH variable with $MQ_INSTALLATION_PATH/bin"
```

```

PATH=$MQ_INSTALLATION_PATH/bin:$PATH
fi

# Test the operation of the queue manager. Result is 0 on success, non-zero on error.
echo "ping qmgr" | runmqsc ${QM} > /dev/null 2>&1
pingresult=$?

if [ $pingresult -eq 0 ]
then # ping succeeded

    echo "Queue manager '${QM}' is responsive"
    result=0

else # ping failed

    # Don't condemn the queue manager immediately, it might be starting.
    srchstr="(|-m)$QM *.*$"
    cnt=`ps -ef | tr "\t" " " | grep stirmqm | grep "$srchstr" | grep -v grep \
        | awk '{print $2}' | wc -l`
    if [ $cnt -gt 0 ]
    then
        # It appears that the queue manager is still starting up, tolerate
        echo "Queue manager '${QM}' is starting"
        result=0
    else
        # There is no sign of the queue manager starting
        echo "Queue manager '${QM}' is not responsive"
        result=$pingresult
    fi
fi

fi

exit $result

```

고가용성(HA) 클러스터의 제어 아래 큐 관리자 넣기

큐 관리자 IP 주소 및 공유 디스크를 사용하여 HA 클러스터의 제어 아래 큐 관리자를 구성해야 합니다.

큐 관리자 및 연관된 모든 자원을 포함하도록 자원 그룹을 정의하려면 다음 단계를 완료하십시오.

1. 큐 관리자, 큐 관리자의 볼륨 또는 디스크 그룹 및 큐 관리자의 IP 주소를 포함하는 자원 그룹을 작성하십시오. IP 주소는 컴퓨터의 IP 주소가 아닌 가상 IP 주소입니다.
2. HA 클러스터가 자원을 클러스터 노드 사이에서 올바르게 전환하고 큐 관리자를 제어할 준비가 되었는지 확인하십시오.

고가용성(HA) 클러스터 노드에서 큐 관리자 삭제

더 이상 큐 관리자를 실행하지 않아도 되는 노드에서 큐 관리자를 제거하려고 합니다.

HA 클러스터 내 노드에서 큐 관리자를 제거하려면 다음 단계를 완료하십시오.

1. HA 클러스터가 노드에서 큐 관리자를 활성화하려고 시도하지 않도록 HA 클러스터에서 이 노드를 제거하십시오.
2. 다음 **rmvmqinf** 명령을 사용하여 큐 관리자의 구성 정보를 제거하십시오.

```
rmvmqinf qmgrname
```

큐 관리자를 완전히 삭제하려면 **dltmqm** 명령을 사용하십시오. 그러나 이 명령은 큐 관리자의 데이터 및 로그 파일까지 완전히 삭제함을 알아야 합니다. 큐 관리자 삭제를 마쳤으면 **rmvmqinf** 명령을 사용하여 다른 노드에서 남아 있는 구성 정보를 제거할 수 있습니다.

MSCS(Microsoft Cluster Service) 지원

가상 서버의 장애 복구를 지원하는 MSCS를 소개하고 설정합니다.

This information applies to WebSphere MQ for 윈도우 only.

MSCS(Microsoft Cluster Service)를 사용하면 서버를 클러스터에 연결하여 보다 높은 사용가능성의 데이터 및 애플리케이션을 제공하고 시스템 관리를 용이하게 할 수 있습니다. MSCS는 서버 또는 애플리케이션 실패를 자동으로 감지하고 복구할 수 있습니다.

MSCS는 애플리케이션, 웹 사이트, 인쇄 큐 또는 파일 공유에 해당하는 가상 서버의 장애 복구를 지원합니다(예: 해당 디스크 스핀들, 파일 및 IP 주소 등).

장애 복구는 MSCS가 클러스터의 한 컴퓨터에서 애플리케이션의 실패를 감지하고 순서대로 문제가 발생한 애플리케이션을 종료하며 해당 상태 데이터를 다른 컴퓨터에 전송하고 그 위치에서 애플리케이션을 다시 시작하는데 사용하는 프로세스입니다.

이 절에서는 MSCS 클러스터를 소개하고 다음 절에서는 MSCS 지원 설정에 대해 설명합니다.

- [297 페이지의 『MSCS 클러스터 소개』](#)
- [298 페이지의 『MSCS 클러스터링을 위한 IBM WebSphere MQ 설정』](#)

그런 다음 MSCS 클러스터링에 대한 WebSphere MQ 구성 방법을 다음 절에서 설명합니다.

- [300 페이지의 『MSCS에서 사용할 큐 관리자 작성』](#)
- [300 페이지의 『MSCS 스토리지로 큐 관리자 이동』](#)
- [302 페이지의 『MSCS 제어 아래 큐 관리자 넣기』](#)
- [307 페이지의 『MSCS 제어에서 큐 관리자 제거』](#)

그리고 WebSphere MQ와 함께 MSCS를 사용하는 것에 대한 여러 유용한 힌트 및 WebSphere MQ MSCS 지원 유틸리티 프로그램에 대한 자세한 내용을 다음 절에서 제공합니다.

- [309 페이지의 『MSCS 사용의 힌트 및 팁』](#)
- [311 페이지의 『IBM WebSphere MQ MSCS 지원 유틸리티 프로그램』](#)

MSCS 클러스터 소개

MSCS 클러스터는 둘 이상의 컴퓨터로 구성된 그룹이며 이 두 컴퓨터는 서로 연결되어 하나가 실패하면 MSCS가 애플리케이션의 상태 데이터를 실패한 컴퓨터에서 클러스터 내 다른 컴퓨터로 전송하고 해당 조작을 그 위치에서 다시 시작하여 장애 복구를 수행하는 방식으로 구성됩니다.

[287 페이지의 『고가용성 구성으로 WebSphere MQ 사용』](#)에는 MSCS 클러스터, 멀티 인스턴스 큐 관리자 및 WebSphere MQ 클러스터 간의 비교가 포함됩니다.

이 절 및 하위 주제에서 용어 클러스터는 단독으로 사용될 때 **항상** MSCS 클러스터를 의미합니다. 이는 이 안내서의 다른 부분에서 설명된 WebSphere MQ 클러스터와 구별됩니다.

두 시스템 클러스터는 가상 IP 주소를 사용한 클라이언트 액세스를 위해 네트워크에 함께 연결된 두 대의 컴퓨터(예: A 및 B)로 구성됩니다. 이 두 컴퓨터는 하나 이상의 사실 네트워크를 통해 서로 연결될 수도 있습니다. A와 B는 각 서버 애플리케이션이 사용할 디스크를 하나 이상 공유합니다. MSCS의 독점 사용을 위해 RAID(Redundant Array of Independent Disks) 레벨 1인 다른 공유 디스크도 있습니다. 이 디스크는 퀴럼 디스크로 알려집니다. MSCS는 두 컴퓨터를 모두 모니터링하여 하드웨어 및 소프트웨어가 올바르게 실행되고 있는지 검사합니다.

이와 같은 단순 설정에서 두 컴퓨터 모두에는 애플리케이션이 설치되어 있지만 컴퓨터 A만 라이브 애플리케이션과 함께 실행되고 컴퓨터 B는 실행 및 대기만 합니다. 컴퓨터 A에서 다양한 문제점 중 하나가 발견되면 MSCS가 문제점이 발생한 애플리케이션을 순서대로 종료하고 그 상태 데이터를 다른 컴퓨터에 전송하며 그 위치에서 애플리케이션을 다시 시작합니다. 이를 장애 복구라고 합니다. 애플리케이션이 MSCS와 완전히 상호작용하고 원활하게 장애 복구할 수 있도록 애플리케이션을 클러스터 인지로 설정할 수 있습니다.

두 컴퓨터 클러스터에 대한 일반적인 설정은 [298 페이지의 그림 62](#)에 표시된 대로입니다.

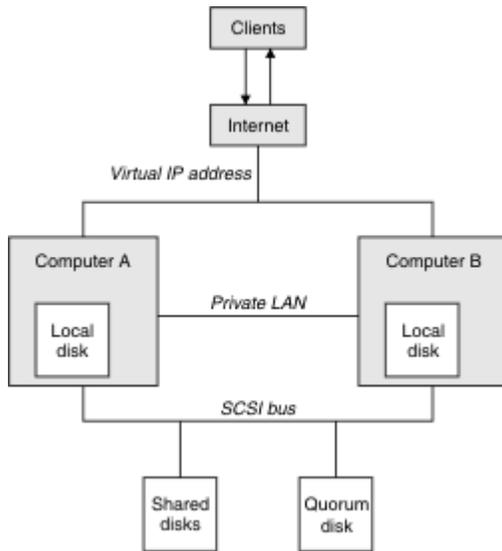


그림 62. 두 컴퓨터 MSCS 클러스터

각 컴퓨터는 공유 디스크에 액세스할 수 있지만 MSCS의 제어 하에 한 번에 하나씩만 가능합니다. 장애 복구 시 MSCS는 액세스를 다른 컴퓨터로 전환합니다. 공유 디스크 자체는 일반적으로 RAID지만 반드시 RAID일 필요는 없습니다.

각 컴퓨터는 클라이언트 액세스를 위해 외부 네트워크에 연결되고 각각 IP 주소를 가지고 있습니다. 그러나 이 클러스터와 통신하는 외부 클라이언트는 한 개의 가상 IP 주소만 인식하고 MSCS는 클러스터 내 IP 트래픽을 적절하게 라우트합니다.

MSCS는 예를 들어, 하트비트를 사용하여 상태를 모니터링하고 해당 데이터베이스를 동기화하기 위해 하나 이상의 개인용 연결 또는 공용 네트워크를 통해 두 컴퓨터 간 자체 통신도 수행합니다.

MSCS 클러스터링을 위한 IBM WebSphere MQ 설정

큐 관리자를 MSCS에 대한 장애 복구 단위로 설정하여 클러스터링을 위한 IBM WebSphere MQ를 구성합니다. 큐 관리자를 MSCS에 대한 자원으로 정의하면 큐 관리자를 모니터링하고 문제점이 있을 경우 클러스터 내 다른 컴퓨터로 전송할 수 있습니다.

이를 위해 시스템을 설정하려면 클러스터 내 각 컴퓨터에서 IBM WebSphere MQ를 설치하여 시작합니다.

큐 관리자는 IBM WebSphere MQ 설치 이름과 연관되므로 클러스터 내 모든 컴퓨터에서 IBM WebSphere MQ 설치 이름은 같아야 합니다. 설치 및 설치 제거를 참조하십시오.

큐 관리자 스스로는 큐 관리자를 작성하는 컴퓨터에만 존재해야 합니다. 장애 복구 시 MSCS는 다른 컴퓨터에서 큐 관리자를 시작합니다. 그러나 큐 관리자는 로컬 드라이브가 아닌 클러스터 공유 디스크에 로그 및 데이터 파일을 가져야 합니다. 이미 로컬 드라이브에 설치된 큐 관리자가 있는 경우 IBM WebSphere MQ에서 제공된 도구를 사용하여 이 큐 관리자를 마이그레이션할 수 있습니다. 300 페이지의 『MSCS 스토리지로 큐 관리자 이동』의 내용을 참조하십시오. MSCS에 사용하기 위해 새 큐 관리자를 작성하려면 300 페이지의 『MSCS에서 사용할 큐 관리자 작성』을 참조하십시오.

설치 및 마이그레이션 후 MSCS 클러스터 관리자를 사용하여 MSCS가 큐 관리자를 인지하도록 설정하십시오. 302 페이지의 『MSCS 제어 아래 큐 관리자 넣기』의 내용을 참조하십시오.

MSCS 제어에서 큐 관리자를 제거하기로 결정한 경우 307 페이지의 『MSCS 제어에서 큐 관리자 제거』에 설명되어 있는 프로시저를 사용하십시오.

대칭 설정

애플리케이션이 한 노드에서 다른 노드로 전환할 때 노드에 관계없이 같은 방식으로 작동해야 합니다. 이를 보장하는 최고의 방법은 환경을 동일하게 설정하는 것입니다.

가능한 경우 각 컴퓨터에서 동일한 하드웨어, 운영 체제 소프트웨어, 제품 소프트웨어 및 구성으로 클러스터를 설정하십시오. 특히, 두 컴퓨터에 설치된 모든 필수 소프트웨어가 버전, 유지보수 레벨, SupportPacs, 경로 및 엑시트 면에서 동일한지 및 299 페이지의 『MSCS 보안』에 설명된 대로 공통 네임스페이스(보안 환경)가 있는지 확인하십시오.

MSCS 보안

성공적인 MSCS 보안을 위해 다음 가이드라인을 따릅니다.

가이드라인은 다음과 같습니다.

- 클러스터 내 각 컴퓨터의 소프트웨어 설치가 동일한지 확인하십시오.
- 클러스터 전체에서 공통 네임스페이스(보안 환경)를 작성하십시오.
- 클러스터 소유자인 사용자 계정이 도메인 계정인 도메인의 MSCS 클러스터 멤버 노드를 작성하십시오.
- 클러스터의 다른 사용자 계정도 도메인 계정으로 설정하여 두 노드 모두에서 사용 가능하도록 합니다. 이미 도메인이 있거나 WebSphere MQ에 관련된 계정이 도메인 계정인 경우 이는 자동으로 이루어집니다. 현재 도메인이 없는 경우 클러스터 노드 및 관련 계정에 맞는 미니 도메인을 설정하는 것을 고려하십시오. 목적은 두 컴퓨터의 클러스터가 단일 컴퓨팅 자원처럼 보이게 하는 것입니다.

한 컴퓨터에 로컬인 계정은 다른 컴퓨터에서는 존재하지 않음에 유의하십시오. 다른 컴퓨터에서 같은 이름을 사용하여 계정을 작성하더라도 그 보안 ID(SID)는 다르므로 애플리케이션이 다른 노드로 이동하면 권한이 해당 노드에 존재하지 않습니다.

장애 복구 또는 이동 중 WebSphere MQ MSCS 지원은 큐 관리자 오브젝트가 포함된 모든 파일이 목적지 노드에서 동등한 권한을 가지고 있는지 확인합니다. 명시적으로 이 노드는 관리자 및 mqm 그룹과 시스템 SYSTEM 계정에 완전한 제어가 있는지 및 Everyone에 이전 노드에 대한 읽기 액세스가 있는지, 권한이 목적지 노드에서 추가되는지 검사합니다.

도메인 계정을 사용하여 WebSphere MQ 서비스를 실행할 수 있습니다. 이 계정이 클러스터 내 각 컴퓨터의 로컬 mqm 그룹에 존재하는지 확인하십시오.

MSCS에서 다중 큐 관리자 사용

컴퓨터에서 둘 이상의 큐 관리자를 실행하는 경우 이러한 설정 중 하나를 선택할 수 있습니다.

이 설정은 다음과 같습니다.

- 모든 큐 관리자가 단일 그룹에 있습니다. 이 구성에서 큐 관리자에 문제점이 발생하는 경우 그룹의 모든 큐 관리자는 그룹으로서 다른 컴퓨터로 장애 복구합니다.
- 단일 큐 관리자가 각 그룹에 있습니다. 이 구성에서 큐 관리자에 문제점이 발생하는 경우 이 큐 관리자는 다른 큐 관리자에 영향을 주지 않고 다른 컴퓨터로 장애 복구합니다.
- 첫 번째 두 설정의 혼합입니다.

클러스터 모드

WebSphere MQ에서 클러스터 시스템을 실행할 수 있는 두 가지 모드는 활성/수동 또는 활성/활성입니다.

참고: MSCS를 Microsoft Transaction Server(COM+)와 함께 사용 중인 경우 활성/활성 모드를 사용할 수 없습니다.

활성/수동 모드

활성/수동 모드에서는 컴퓨터 A에 실행 중인 애플리케이션이 있고 컴퓨터 B가 백업으로, MSCS가 문제점을 감지하는 경우에만 사용됩니다.

이 모드는 공유 디스크에서만 사용할 수 있지만 애플리케이션으로 인해 장애 복구가 발생하는 경우 모든 애플리케이션은 그룹으로 전송되어야 합니다(한 번에 한 컴퓨터만 공유 디스크에 액세스할 수 있기 때문).

A를 선호 컴퓨터로 MSCS를 구성할 수 있습니다. 그런 다음 컴퓨터 A가 복구되거나 바뀌고 다시 올바르게 작동하는 경우 MSCS는 이를 감지하고 자동으로 애플리케이션을 컴퓨터 A로 다시 전환합니다.

둘 이상의 큐 관리자를 실행하는 경우 각각에 대해 개별 공유 디스크를 두는 것을 고려해 보십시오. 그런 다음 MSCS에서 각 큐 관리자를 개별 그룹에 넣으십시오. 이 방법으로 큐 관리자는 다른 큐 관리자에 영향을 주지 않고 다른 컴퓨터로 장애를 복구할 수 있습니다.

활성/활성 모드

활성/활성 모드에서 컴퓨터 A 및 B 둘 다에 실행 중인 애플리케이션이 있고 각 컴퓨터의 그룹은 다른 컴퓨터를 백업으로 사용하도록 설정됩니다. 컴퓨터 A에서 실패가 감지되면 MSCS가 상태 데이터를 컴퓨터 B로 전송하고 그

위치에서 애플리케이션을 다시 시작합니다. 그러면 컴퓨터 B가 자체 애플리케이션과 A의 애플리케이션을 실행합니다.

이 설정을 위해서는 최소 둘 이상의 공유 디스크가 필요합니다. A를 A의 애플리케이션에 대한 선호 컴퓨터로, B를 B의 애플리케이션에 대한 선호 컴퓨터로 MSCS를 구성할 수 있습니다. 장애 복구 및 수리 후에 각 애플리케이션은 자동으로 자체 컴퓨터에서 종료됩니다.

WebSphere MQ의 경우 이는 예를 들어 A와 B 각 컴퓨터에서 하나씩 두 개의 큐 관리자를 각기 해당 컴퓨터의 전체 기능을 이용하면서 실행할 수 있음을 의미합니다. 컴퓨터 A에서 장애가 발생한 후 두 큐 관리자가 컴퓨터 B에서 실행됩니다. 이는 한 컴퓨터의 전원을 공유하는 것을 의미하며, 속도에서 대량의 데이터를 처리하는 능력이 감소합니다. 그러나 A에서 결함을 찾아 수리하는 동안 중요한 애플리케이션은 계속 사용 가능합니다.

MSCS에서 사용할 큐 관리자 작성

이 프로시저는 새 큐 관리자를 MSCS 제어 하에서 준비하고 배치하기에 적합한 방법으로 작성할 수 있도록 합니다.

로컬 드라이브에서 모든 해당 자원을 사용하여 큐 관리자를 작성하여 시작한 다음 로그 파일 및 데이터 파일을 공유 디스크로 마이그레이션합니다. (이 조작을 되돌릴 수 있습니다.) 공유 드라이브에서 해당 자원으로 큐 관리자를 작성하려는 시도를 하지 **마십시오**.

MSCS에서 사용할 큐 관리자를 두 가지 방법으로, 즉, 명령 프롬프트에서 작성하거나 WebSphere MQ Explorer에서 작성할 수 있습니다. 명령 프롬프트 사용의 장점은 큐 관리자가 중지됨으로 작성되고 수동 시동으로 설정되는 것이며 이는 MSCS에 대한 준비가 된 것입니다. (IBM WebSphere MQ 탐색기는 자동으로 새 큐 관리자를 시작하고 이를 작성 후 자동 시작으로 설정합니다. 이를 변경해야 합니다.)

명령 프롬프트에서 큐 관리자 작성

다음 단계에 따라 MSCS에서 사용할 큐 관리자를 명령 프롬프트에서 작성하십시오.

1. 환경 변수 MQSPREFIX가 로컬 드라이브를 참조하도록 설정(예: C:\WebSphere MQ)되었는지 확인하십시오. 이를 변경하는 경우 시스템 계정이 변경사항을 적용하도록 시스템을 다시 시작하십시오. 환경 변수를 설정하지 않으면 큐 관리자는 큐 관리자를 위한 WebSphere MQ 기본 디렉토리에 작성됩니다.
2. **crtmqm** 명령을 사용하여 큐 관리자를 작성하십시오. 예를 들어, 기본 디렉토리에서 **mscs_test**라는 큐 관리자를 작성하려면 다음을 사용하십시오.

```
crtmqm mscs_test
```

3. [300 페이지의 『MSCS 스토리지로 큐 관리자 이동』](#)으로 이동하십시오.

WebSphere MQ Explorer를 사용한 큐 관리자 작성

다음 단계에 따라 IBM WebSphere MQ 탐색기를 사용하여 MSCS에서 사용할 큐 관리자를 작성하십시오.

1. 시작 메뉴에서 IBM WebSphere MQ 탐색기를 시작하십시오.
2. Navigator 보기에서 트리 노드를 펼쳐서 Queue Managers 트리 노드를 찾으십시오.
3. Queue Managers 트리 노드를 마우스 오른쪽 단추로 클릭하고 New->Queue Manager을 선택하십시오. 큐 관리자 작성 패널이 표시됩니다.
4. 대화 상자 (1단계) 를 완료한 후 Next>을 (를) 클릭하십시오.
5. 대화 상자 (2단계) 를 완료한 후 Next>을 (를) 클릭하십시오.
6. 대화 상자 (3단계) 를 완료하여 Start Queue Manager 및 Create Server Connection Channel 이 선택되지 않았는지 확인한 후 Next>을 (를) 클릭하십시오.
7. 대화 상자 (4단계) 를 완료한 후 Finish을 (를) 클릭하십시오.
8. [300 페이지의 『MSCS 스토리지로 큐 관리자 이동』](#)으로 이동하십시오.

MSCS 스토리지로 큐 관리자 이동

이 프로시저는 기존 큐 관리자를 MSCS 제어 아래에 넣기에 적당하게 되도록 구성합니다.

이렇게 하려면 로그 파일 및 데이터 파일을 공유 디스크로 이동하여 실패가 발생할 경우 다른 컴퓨터에서 사용 가능하도록 해야 합니다. 예를 들어, 기존 큐 관리자에는 C:\WebSphere MQ\log\마십시오. 이 토픽에 설명된 대로 WebSphere MQ MSCS 지원의 일부로 제공된 유틸리티 프로그램을 사용하십시오.

이동할 큐 관리자가 SSL 연결을 사용하고 SSL 키 저장소가 로컬 시스템의 큐 관리자 데이터 디렉토리에 있는 경우 이 키 저장소는 나머지 큐 관리자와 함께 공유 디스크로 이동됩니다. 기본적으로 SSL키 저장소 위치인 SSLKEYR을 지정하는 큐 관리자 속성은 큐 관리자 데이터 디렉토리 아래에 있는 MQ_INSTALLATION_PATH\qmgrs\QMGRNAME\ssl\key로 설정됩니다. MQ_INSTALLATION_PATH WebSphere MQ 가 설치된 상위 레벨 디렉토리를 나타냅니다. hamvmqm 명령은 이 큐 관리자 속성을 수정하지 않습니다. 이 상황에서는 새 SSL키 저장소 파일을 가리키도록 IBM WebSphere MQ Explorer 또는 MQSC 명령 ALTER QMGR를 사용하여 큐 관리자 속성인 SSLKEYR을 수정해야 합니다.

이 프로시저는 다음과 같습니다.

1. 큐 관리자를 종료하고 오류가 없는지 검사하십시오.
2. 큐 관리자의 로그 파일 또는 큐 파일이 이미 공유 디스크에 저장된 경우 이 프로시저의 나머지 부분은 건너뛰고 302 페이지의 『MSCS 제어 아래 큐 관리자 넣기』로 바로 이동하십시오.
3. 큐 파일 및 로그 파일의 전체 매체 백업을 작성하고 안전한 위치에 백업을 저장하십시오(이 조치가 중요한 이유는 310 페이지의 『큐 관리자 로그 파일』 참조).
4. 이미 적합한 공유 디스크 자원이 있는 경우 6단계로 진행하십시오. 그렇지 않으면, MSCS 클러스터 관리자를 사용하여 큐 관리자 로그 파일 및 데이터(큐) 파일을 저장할 충분한 용량을 갖는 공유 디스크 유형의 자원을 작성하십시오.
5. 공유 디스크를 한 클러스터 노드에서 다른 클러스터 노드로 이동하고 다시 돌아오게 하려면 MSCS 클러스터 관리자를 사용하여 공유 디스크를 테스트하십시오.
6. 공유 디스크가 큐 관리자 로그 및 데이터 파일이 로컬로 저장되는 클러스터 노드에서 온라인인지 확인하십시오.
7. 다음과 같이 유틸리티 프로그램을 실행하여 큐 관리자를 이동하십시오.

```
hamvmqm /m qmname /dd "e:\
WebSphere MQ" /ld "e:\
WebSphere MQ\log"
```

qmname에는 큐 관리자 이름, e에는 공유 디스크 드라이브 문자, 그리고 WebSphere MQ에는 선택한 디렉토리로 대체하십시오. 디렉토리는 아직 없는 경우에 작성됩니다.

8. 큐 관리자가 작동하는지 확인하려면 IBM WebSphere MQ 탐색기를 사용하여 큐 관리자를 테스트하십시오. 예를 들면, 다음과 같습니다.
 - a. 큐 관리자 트리 노드를 마우스 오른쪽 단추로 클릭한 후 Start을 선택하십시오. 큐 관리자가 시작됩니다.
 - b. Queues 트리 노드를 마우스 오른쪽 단추로 누른 후 New->Local Queue...을 선택하고 큐에 이름을 지정하십시오.
 - c. Finish을(를) 클릭하십시오.
 - d. 큐를 마우스 오른쪽 단추로 누른 다음 Put Test Message...을 선택하십시오. 테스트 메시지 넣기 패널이 표시됩니다.
 - e. 일부 메시지 텍스트를 입력한 후 Put Test Message을 클릭하고 패널을 닫으십시오.
 - f. 큐를 마우스 오른쪽 단추로 누른 다음 Browse Messages...을 선택하십시오. 메시지 브라우저 패널이 표시됩니다.
 - g. 메시지가 큐에 있는지 확인한 다음 Close 을 (를) 클릭하십시오. 메시지 브라우저 패널이 닫힙니다.
 - h. 큐를 마우스 오른쪽 단추로 누른 다음 Clear Messages...을 선택하십시오. 큐에 있는 메시지가 지워집니다.
 - i. 큐를 마우스 오른쪽 단추로 누른 다음 Delete...을 선택하십시오. 확인 패널이 표시됩니다. OK을 (를) 클릭하십시오. 큐가 삭제됩니다.

- j. 큐 관리자 트리 노드를 마우스 오른쪽 단추로 클릭한 후 Stop... 을 선택하십시오. 큐 관리자 종료 패널이 표시됩니다.
- k. OK을(를) 클릭하십시오. 큐 관리자가 중지됩니다.

9. WebSphere MQ 관리자로서 큐 관리자의 시동 속성이 수동으로 설정되었는지 확인하십시오. IBM WebSphere MQ 탐색기의 큐 관리자 특성 패널에서 시동 필드를 manual로 설정하십시오.

10. 302 페이지의 『MSCS 제어 아래 큐 관리자 넣기』로 이동하십시오.

MSCS 제어 아래 큐 관리자 넣기

필수조건 태스크를 포함하여 큐 관리자를 MSCS의 제어 아래 배치하는 데 포함되는 태스크입니다.

MSCS 제어 아래 큐 관리자를 넣기 전에

MSCS 제어 아래 큐 관리자를 넣기 전에 다음 태스크를 수행하십시오.

1. IBM WebSphere MQ 및 해당 MSCS 지원이 클러스터 내 두 시스템에 모두 설치되어 있는지 및 298 페이지의 『MSCS 클러스터링을 위한 IBM WebSphere MQ 설정』에 설명된 대로 각 컴퓨터의 소프트웨어가 동일인지 확인하십시오.
2. **haregtyp** 유틸리티 프로그램을 사용하여 WebSphere MQ를 모든 클러스터 노드에서 MSCS 자원 유형으로 등록하십시오. 추가 정보는 311 페이지의 『IBM WebSphere MQ MSCS 지원 유틸리티 프로그램』의 내용을 참조하십시오.
3. 아직 큐 관리자를 작성하지 않은 경우 300 페이지의 『MSCS에서 사용할 큐 관리자 작성』을 참조하십시오.
4. 큐 관리자를 작성했거나 이미 있는 경우 300 페이지의 『MSCS 스토리지로 큐 관리자 이동』의 프로시저를 수행했는지 확인하십시오.
5. 명령 프롬프트 또는 IBM WebSphere MQ 탐색기를 사용하여 실행 중인 큐 관리자를 중지하십시오.
6. 이 토픽에 있는 다음의 각 Windows 프로시저를 계속 수행하기 전에 공유 드라이브의 MSCS 조작을 테스트하십시오.

윈도우 서버 2012



주의: MSCS 지원은 32비트 DLL을 사용하여 WebSphere MQ 7.5에서 제공됩니다. Windows 2012에서의 제한사항으로 인해 IBM WebSphere MQ 큐 관리자는 재시작 이후 실패 복구를 수행할 수 없습니다.

Microsoft는 Windows 2012에서 32비트 DLL을 더 이상 사용하지 않으므로 현재 이 문제에 대한 운영 체제 수정사항은 없습니다. IBM은 IBM WebSphere MQ 7.5용 64비트 라이브러리를 제공하지 않습니다.

IBM MQ 8.0의 경우 64비트 라이브러리를 사용할 수 있으므로 Windows 2012 이상에서 MSCS 전체 기능을 사용하려면 이 제품 버전을 사용해야 합니다.

Windows Server 2012의 MSCS에서 큐 관리자를 제어하도록 하려면 다음 프로시저를 사용하십시오.

1. 큐 관리자를 호스팅하는 클러스터 노드 컴퓨터에 로그인하거나 클러스터 관리 권한이 있는 사용자로 원격 워크스테이션에 로그인하고 큐 관리자를 호스팅하는 클러스터 노드에 연결하십시오.
2. 장애 복구 클러스터 관리 도구를 시작하십시오.
3. **장애 조치 클러스터 관리 > 클러스터 연결 ...** 을 마우스 오른쪽 단추로 클릭하십시오. 클러스터에 대한 연결을 열 수 있습니다.
4. 이전 버전의 Windows에서 MSCS 클러스터 관리자에 사용된 그룹 스키마와 반대로 장애 복구 클러스터 관리 도구는 서비스 및 애플리케이션의 개념을 사용합니다. 구성된 서비스 또는 애플리케이션은 클러스터될 한 애플리케이션에 필요한 모든 자원을 포함합니다. 다음과 같이 MSCS 아래에서 큐 관리자를 구성할 수 있습니다.
 - a. 클러스터에서 마우스의 오른쪽 단추를 클릭하고 **역할 구성**을 선택하여 구성 마법사를 시작하십시오.
 - b. "서비스 또는 애플리케이션 선택" 패널에서 **기타 서버**를 선택하십시오.
 - c. 적절한 IP 주소를 클라이언트 액세스 지점으로 선택하십시오.

이 주소는 클라이언트 및 다른 큐 관리자가 가상 큐 관리자에 연결하는 데 사용될 미사용 IP 주소여야 합니다. 이 IP 주소는 두 노드 중 하나의 정상(정적) 주소가 아니며 두 노드 사이에서 플로트하는 추가 주소입니다. MSCS가 이 주소의 라우팅을 핸들링하지만 주소에 도달할 수 있는지는 확인하지 않습니다.

- d. 큐 관리자가 독점적으로 사용할 수 있는 스토리지 디바이스를 지정하십시오. 이 디바이스를 지정하려면 먼저 자원 인스턴스로 이 디바이스를 작성해야 합니다.

한 개의 드라이브를 사용하여 로그 및 큐 파일 둘 다 저장하거나 이러한 파일을 여러 드라이브 전체에 분할할 수 있습니다. 둘 중 어느 경우든 각 큐 관리자에 자체 공유 디스크가 있는 경우 이 큐 관리자가 사용하는 모든 드라이브가 이 큐 관리자에 대해 독점적이어야 합니다. 즉, 그 외 어떤 것도 해당 드라이브에 종속되지 않아야 합니다. 또한 큐 관리자가 사용하는 모든 드라이브에 대해 자원 인스턴스를 작성해야 합니다.

드라이브에 대한 자원 유형은 사용하고 있는 SCSI 지원에 따라 다릅니다. SCSI 어댑터 지시사항을 참조하십시오. 이미 각 공유 드라이브에 대한 그룹 및 자원이 있을 수 있습니다. 이 경우 각 드라이브에 대해 자원 인스턴스를 작성하지 않아도 됩니다. 이를 현재 그룹에서 큐 관리자에 대해 작성된 그룹으로 이동하십시오.

각 드라이브 자원에 대해 가능한 소유자를 두 노드로 설정하십시오. 종속 자원을 없음으로 설정하십시오.

- e. "자원 유형 선택" 패널에서 **IBM MQSeries MSCS** 자원을 선택하십시오.

- f. 마법사에서 나머지 단계를 완료하십시오.

5. 자원을 온라인으로 가져오기 전에 IBM MQSeries® MSCS 자원에 추가 구성이 필요합니다.

- a. '새 IBM MQSeries MSCS'라는 자원이 포함된 새로 정의된 서비스를 선택하십시오.

- b. MQ 자원의 **특성**에서 마우스의 오른쪽 단추를 클릭하십시오.

- c. 다음 자원을 구성하십시오.

- Name; 대상이 되는 큐 관리자를 쉽게 식별할 수 있도록 이름을 선택하십시오.
- Run in a separate Resource Monitor; 더 나은 격리
- Possible owners; 두 노드 모두 설정
- Dependencies; 이 큐 관리자의 드라이브 및 IP 주소를 추가하십시오.

경고: 이러한 종속항목 추가에 실패하는 것은 IBM WebSphere MQ가 장애 복구 중 큐 관리자 상태를 잘못된 클러스터 디스크에 기록하려고 시도하는 것을 의미합니다. 다수의 프로세스가 동시에 이 디스크에 기록하려고 시도할 수 있기 때문에 일부 IBM WebSphere MQ 프로세스는 실행이 차단될 수 있습니다.

- Parameters; 다음과 같이:

- QueueManagerName(필수); 이 자원이 제어할 큐 관리자의 이름입니다. 이 큐 관리자는 로컬 컴퓨터에 있어야 합니다.
- PostOnlineCommand(선택적); 큐 관리자 자원이 그 상태를 오프라인에서 온라인으로 변경할 때마다 실행할 프로그램을 지정할 수 있습니다. 자세한 정보는 310 페이지의 『PostOnlineCommand 및 PreOfflineCommand』의 내용을 참조하십시오.
- PreOfflineCommand(선택적); 큐 관리자 자원이 그 상태를 온라인에서 오프라인으로 변경할 때마다 실행할 프로그램을 지정할 수 있습니다. 자세한 정보는 310 페이지의 『PostOnlineCommand 및 PreOfflineCommand』의 내용을 참조하십시오.

참고: looksAlive 폴 간격은 기본값 5,000밀리초로 설정됩니다. isAlive 폴 간격은 기본값 60,000밀리초로 설정됩니다. 이러한 기본값은 자원 정의를 완료한 후에만 수정할 수 있습니다. 추가 세부사항은 307 페이지의 『looksAlive 및 isAlive 폴링의 요약』의 내용을 참조하십시오.

- d. 선택적으로 선호 노드를 설정하십시오(그러나 311 페이지의 『선호 노드 사용』의 주석 참고).

- e. 장애 복구 정책은 기본적으로 실제 값으로 설정되지만, 자원 장애 복구 및 그룹 장애 복구를 제어하는 임계값 및 기간을 조정하여 큐 관리자에 놓여진 로드에서 일치시킬 수 있습니다.

6. MSCS 클러스터 관리자에서 큐 관리자를 온라인으로 설정하고 테스트 워크로드를 부과하여 큐 관리자를 테스트하십시오. 테스트 큐 관리자를 실험하고 있다면 IBM WebSphere MQ 탐색기를 사용하십시오. 예를 들면, 다음과 같습니다.

- a. Queues 트리 노드를 마우스 오른쪽 단추로 누른 후 New->Local Queue...을 선택하고 큐에 이름을 지정하십시오.

- b. Finish을(를) 클릭하십시오. 큐가 작성되고 콘텐츠 보기에 표시됩니다.

- c. 큐를 마우스 오른쪽 단추로 누른 다음 Put Test Message...을 선택하십시오. 테스트 메시지 넣기 패널이 표시됩니다.

- d. 일부 메시지 텍스트를 입력한 후 Put Test Message을 클릭하고 패널을 닫으십시오.
 - e. 큐를 마우스 오른쪽 단추로 누른 다음 Browse Messages...을 선택하십시오. 메시지 브라우저 패널이 표시됩니다.
 - f. 메시지가 큐에 있는지 확인한 후 Close 을 (를) 클릭하십시오. 메시지 브라우저 패널이 닫힙니다.
 - g. 큐를 마우스 오른쪽 단추로 누른 다음 Clear Messages...을 선택하십시오. 큐에 있는 메시지가 지워집니다.
 - h. 큐를 마우스 오른쪽 단추로 누른 다음 Delete...을 선택하십시오. 확인 패널이 표시됩니다. OK을 (를) 클릭하십시오. 큐가 삭제됩니다.
7. MSCS 클러스터 관리자를 사용하여 큐 관리자를 오프라인으로 가져오고 다시 온라인으로 되돌릴 수 있는지 테스트하십시오.
8. 장애 복구를 시뮬레이션하십시오.

MSCS 클러스터 관리자에서 큐 관리자를 포함하는 그룹에서 마우스의 오른쪽 단추를 클릭하고 Move Group 을 선택하십시오. 이를 수행하는 데에는 몇 분 정도 소요될 수 있습니다. (다른 때 큐 관리자를 다른 노드로 신속하게 이동하려고 하는 경우 300 페이지의 『MSCS 스토리지로 큐 관리자 이동』의 프로시저에 따르십시오.) 마우스 오른쪽 단추를 클릭하고 Initiate Failure를 선택할 수도 있습니다. 조치 (로컬 재시작 또는 장애 조치) 는 현재 상태 및 구성 설정에 따라 다릅니다.

윈도우 서버 2008

큐 관리자를 Windows Server 2008의 MSCS 제어 아래에 두려면 다음 프로시저를 사용하십시오.

1. 큐 관리자를 호스팅하는 클러스터 노드 컴퓨터에 로그인하거나 클러스터 관리 권한이 있는 사용자로 원격 위크스테이션에 로그인하고 큐 관리자를 호스팅하는 클러스터 노드에 연결하십시오.
2. 장애 복구 클러스터 관리 도구를 시작하십시오.
3. **장애 조치 클러스터 관리 > 클러스터 관리 ...** 를 마우스 오른쪽 단추로 클릭하십시오. 클러스터에 대한 연결을 열 수 있습니다.
4. 이전 버전의 Windows에서 MSCS 클러스터 관리자에 사용된 그룹 스키마와 반대로 장애 복구 클러스터 관리 도구는 서비스 및 애플리케이션의 개념을 사용합니다. 구성된 서비스 또는 애플리케이션은 클러스터될 한 애플리케이션에 필요한 모든 자원을 포함합니다. 다음과 같이 MSCS 아래에서 큐 관리자를 구성할 수 있습니다.
 - a. **서비스 및 애플리케이션 > 서비스 또는 애플리케이션 구성 ...** 을 마우스 오른쪽 단추로 클릭하십시오. 구성 마법사를 시작한다.
 - b. "서비스 또는 애플리케이션 선택" 패널에서 **기타 서버**를 선택하십시오.
 - c. 적절한 IP 주소를 클라이언트 액세스 지점으로 선택하십시오.

이 주소는 클라이언트 및 다른 큐 관리자가 가상 큐 관리자에 연결하는 데 사용될 미사용 IP 주소여야 합니다. 이 IP 주소는 두 노드 중 하나의 정상(정적) 주소가 아니며 두 노드 사이에서 플로트하는 추가 주소입니다. MSCS가 이 주소의 라우팅을 핸들링하지만 주소에 도달할 수 있는지는 확인하지 **않습니다**.
 - d. 큐 관리자가 독점적으로 사용할 수 있는 스토리지 디바이스를 지정하십시오. 이 디바이스를 지정하려면 먼저 자원 인스턴스로 이 디바이스를 작성해야 합니다.

한 개의 드라이브를 사용하여 로그 및 큐 파일 둘 다 저장하거나 이러한 파일을 여러 드라이브 전체에 분할할 수 있습니다. 둘 중 어느 경우든 각 큐 관리자에 자체 공유 디스크가 있는 경우 이 큐 관리자가 사용하는 모든 드라이브가 이 큐 관리자에 대해 독점적이어야 합니다. 즉, 그 외 어떤 것도 해당드라이브에 종속되지 않아야 합니다. 또한 큐 관리자가 사용하는 모든 드라이브에 대해 자원 인스턴스를 작성해야 합니다.

드라이브에 대한 자원 유형은 사용하고 있는 SCSI 지원에 따라 다릅니다. SCSI 어댑터 지시사항을 참조하십시오. 이미 각 공유 드라이브에 대한 그룹 및 자원이 있을 수 있습니다. 이 경우 각 드라이브에 대해 자원 인스턴스를 작성하지 않아도 됩니다. 이를 현재 그룹에서 큐 관리자에 대해 작성된 그룹으로 이동하십시오.

각 드라이브 자원에 대해 가능한 소유자를 두 노드로 설정하십시오. 종속 자원을 없음으로 설정하십시오.
- e. "자원 유형 선택" 패널에서 **IBM MQSeries MSCS** 자원을 선택하십시오.
- f. 마법사에서 나머지 단계를 완료하십시오.

5. 자원을 온라인으로 가져오기 전에 IBM MQSeries MSCS 자원에 추가 구성이 필요합니다.
 - a. '새 IBM MQSeries MSCS'라는 자원이 포함된 새로 정의된 서비스를 선택하십시오.
 - b. MQ 자원의 **특성**에서 마우스의 오른쪽 단추를 클릭하십시오.
 - c. 다음 자원을 구성하십시오.
 - Name; 대상이 되는 큐 관리자를 쉽게 식별할 수 있도록 이름을 선택하십시오.
 - Run in a separate Resource Monitor; 더 나은 격리
 - Possible owners; 두 노드 모두 설정
 - Dependencies; 이 큐 관리자의 드라이브 및 IP 주소를 추가하십시오.

경고: 이러한 종속 항목 추가에 대한 실패는 WebSphere MQ가 장애 복구 중 잘못된 클러스터 디스크에 큐 관리자 상태를 쓰려고 시도하는 것을 의미합니다. 다수의 프로세스가 동시에 이 디스크에 기록하려고 시도할 수 있기 때문에 일부 IBM WebSphere MQ 프로세스는 실행이 차단될 수 있습니다.

 - Parameters; 다음과 같이:
 - QueueManagerName(필수); 이 자원이 제어할 큐 관리자의 이름입니다. 이 큐 관리자는 로컬 컴퓨터에 있어야 합니다.
 - PostOnlineCommand(선택적); 큐 관리자 자원이 그 상태를 오프라인에서 온라인으로 변경할 때마다 실행할 프로그램을 지정할 수 있습니다. 자세한 정보는 [310 페이지의 『PostOnlineCommand 및 PreOfflineCommand』](#)의 내용을 참조하십시오.
 - PreOfflineCommand(선택적); 큐 관리자 자원이 그 상태를 온라인에서 오프라인으로 변경할 때마다 실행할 프로그램을 지정할 수 있습니다. 자세한 정보는 [310 페이지의 『PostOnlineCommand 및 PreOfflineCommand』](#)의 내용을 참조하십시오.

참고: looksAlive 폴 간격은 기본값 5,000밀리초로 설정됩니다. isAlive 폴 간격은 기본값 60,000밀리초로 설정됩니다. 이러한 기본값은 자원 정의를 완료한 후에만 수정할 수 있습니다. 추가 세부사항은 [307 페이지의 『looksAlive 및 isAlive 폴링의 요약』](#)의 내용을 참조하십시오.
 - d. 선택적으로 선호 노드를 설정하십시오(그러나 [311 페이지의 『선호 노드 사용』](#)의 주석 참고).
 - e. 장애 복구 정책은 기본적으로 실제 값으로 설정되지만, 자원 장애 복구 및 그룹 장애 복구를 제어하는 임계값 및 기간을 조정하여 큐 관리자에 놓여진 로드일치시킬 수 있습니다.
6. MSCS 클러스터 관리자에서 큐 관리자를 온라인으로 설정하고 테스트 워크로드를 부과하여 큐 관리자를 테스트하십시오. 테스트 큐 관리자를 실험하고 있다면 IBM WebSphere MQ 탐색기를 사용하십시오. 예를 들면, 다음과 같습니다.
 - a. Queues 트리 노드를 마우스 오른쪽 단추로 누른 후 New->Local Queue...을 선택하고 큐에 이름을 지정하십시오.
 - b. Finish을(를) 클릭하십시오. 큐가 작성되고 콘텐츠 보기에 표시됩니다.
 - c. 큐를 마우스 오른쪽 단추로 누른 다음 Put Test Message...을 선택하십시오. 테스트 메시지 넣기 패널이 표시됩니다.
 - d. 일부 메시지 텍스트를 입력한 후 Put Test Message을 클릭하고 패널을 닫으십시오.
 - e. 큐를 마우스 오른쪽 단추로 누른 다음 Browse Messages...을 선택하십시오. 메시지 브라우저 패널이 표시됩니다.
 - f. 메시지가 큐에 있는지 확인한 후 Close 을 (를) 클릭하십시오. 메시지 브라우저 패널이 닫힙니다.
 - g. 큐를 마우스 오른쪽 단추로 누른 다음 Clear Messages...을 선택하십시오. 큐에 있는 메시지가 지워집니다.
 - h. 큐를 마우스 오른쪽 단추로 누른 다음 Delete...을 선택하십시오. 확인 패널이 표시됩니다. OK을 (를) 클릭하십시오. 큐가 삭제됩니다.
7. MSCS 클러스터 관리자를 사용하여 큐 관리자를 오프라인으로 가져오고 다시 온라인으로 되돌릴 수 있는지 테스트하십시오.
8. 장애 복구를 시뮬레이션하십시오.

MSCS 클러스터 관리자에서 큐 관리자를 포함하는 그룹에서 마우스의 오른쪽 단추를 클릭하고 Move Group 을 선택하십시오. 이를 수행하는 데에는 몇 분 정도 소요될 수 있습니다. (다른 때 큐 관리자를 다른 노드로 신속하게 이동하려고 하는 경우 300 페이지의 『MSCS 스토리지로 큐 관리자 이동』의 프로시저에 따르십시오.) 마우스 오른쪽 단추를 클릭하고 Initiate Failure를 선택할 수도 있습니다. 조치 (로컬 재시작 또는 장애 조치) 는 현재 상태 및 구성 설정에 따라 다릅니다.

Windows 2003

큐 관리자를 Windows 2003의 MSCS 제어 하에 두려면 다음 프로시저를 사용하십시오.

1. 큐 관리자를 호스팅하는 클러스터 노드 컴퓨터에 로그인하거나 클러스터 관리 권한이 있는 사용자로 원격 워크스테이션에 로그인하고 큐 관리자를 호스팅하는 클러스터 노드에 연결하십시오.
2. MSCS 클러스터 관리자를 시작하십시오.
3. 클러스터에 대한 연결을 여십시오.
4. 큐 관리자 자원을 포함시키기 위해 사용될 MSCS 그룹을 작성하십시오. 이름과 관련된 큐 관리자를 명백히 알 수 있는 방식으로 그룹의 이름을 지정하십시오. 각 그룹은 299 페이지의 『MSCS에서 다중 큐 관리자 사용』에 설명된 대로 다중 큐 관리자를 포함할 수 있습니다.

남아 있는 모든 단계에서 그룹을 사용하십시오.

5. 큐 관리자가 사용하는 각각의 SCSI 논리 드라이브에 대해 자원 인스턴스를 작성하십시오.

한 개의 드라이브를 사용하여 로그 및 큐 파일 둘 다 저장하거나 이러한 파일을 여러 드라이브 전체에 분할할 수 있습니다. 둘 중 어느 경우든 각 큐 관리자에 자체 공유 디스크가 있는 경우 이 큐 관리자가 사용하는 모든 드라이브가 이 큐 관리자에 대해 독점적이어야 합니다. 즉, 그 외 어떤 것도 해당드라이브에 종속되지 않아야 합니다. 또한 큐 관리자가 사용하는 모든 드라이브에 대해 자원 인스턴스를 작성해야 합니다.

드라이브에 대한 자원 유형은 사용하고 있는 SCSI 지원에 따라 다릅니다. SCSI 어댑터 지시사항을 참조하십시오. 이미 각 공유 드라이브에 대한 그룹 및 자원이 있을 수 있습니다. 이 경우 각 드라이브에 대해 자원 인스턴스를 작성하지 않아도 됩니다. 이를 현재 그룹에서 큐 관리자에 대해 작성된 그룹으로 이동하십시오.

각 드라이브 자원에 대해 가능한 소유자를 두 노드로 설정하십시오. 종속 자원을 없음으로 설정하십시오.

6. IP 주소에 대한 자원 인스턴스를 작성하십시오.

IP 주소 자원(자원 유형 IP 주소)을 작성하십시오. 이 주소는 클라이언트 및 다른 큐 관리자가 가상 큐 관리자에 연결하는 데 사용될 미사용 IP 주소여야 합니다. 이 IP 주소는 두 노드 중 하나의 정상(정적) 주소가 아니며 두 노드 사이에서 플로트하는 추가 주소입니다. MSCS가 이 주소의 라우팅을 핸들링하지만 주소에 도달할 수 있는지는 확인하지 않습니다.

7. 큐 관리자에 대한 자원 인스턴스를 작성하십시오.

IBM WebSphere MQ MSCS 유형의 자원을 작성하십시오. 마법사가 다음을 포함한 여러 항목에 대한 프롬프트를 표시합니다.

- Name; 대상이 되는 큐 관리자를 쉽게 식별할 수 있도록 이름을 선택하십시오.
- Add to group; 작성한 그룹을 사용하십시오.
- Run in a separate Resource Monitor; 더 나은 격리
- Possible owners; 두 노드 모두 설정
- Dependencies; 이 큐 관리자의 드라이브 및 IP 주소를 추가하십시오.

경고: 이러한 종속 항목 추가에 대한 실패는 WebSphere MQ가 장애 복구 중 잘못된 클러스터 디스크에 큐 관리자 상태를 쓰려고 시도하는 것을 의미합니다. 다수의 프로세스가 동시에 이 디스크에 기록하려고 시도할 수 있기 때문에 일부 IBM WebSphere MQ 프로세스는 실행이 차단될 수 있습니다.

- Parameters; 다음과 같이:

- QueueManagerName(필수); 이 자원이 제어할 큐 관리자의 이름입니다. 이 큐 관리자는 로컬 컴퓨터에 있어야 합니다.
- PostOnlineCommand(선택적); 큐 관리자 자원이 그 상태를 오프라인에서 온라인으로 변경할 때마다 실행할 프로그램을 지정할 수 있습니다. 자세한 정보는 310 페이지의 『PostOnlineCommand 및 PreOfflineCommand』의 내용을 참조하십시오.

- PreOfflineCommand(선택적); 큐 관리자 자원이 그 상태를 온라인에서 오프라인으로 변경할 때마다 실행할 프로그램을 지정할 수 있습니다. 자세한 정보는 310 페이지의 『PostOnlineCommand 및 PreOfflineCommand』의 내용을 참조하십시오.

참고: looksAlive 폴 간격은 기본값 5,000밀리초로 설정됩니다. isAlive 폴 견각은 기본값 30,000밀리초로 설정됩니다. 이러한 기본값은 자원 정의를 완료한 후에만 수정할 수 있습니다. 추가 세부사항은 307 페이지의 『looksAlive 및 isAlive 폴링의 요약』의 내용을 참조하십시오.

- 선택적으로 선호 노드를 설정하십시오(단, 311 페이지의 『선호 노드 사용』의 주석을 참고하십시오).
- 자원 복구 정책(그룹에 대한 특성에 정의됨)은 기본적으로 실제 값으로 설정되지만, 자원 장애 복구 및 그룹 장애 복구를 제어하는 임계값 및 기간을 조정하여 큐 관리자에 놓여진 로드와 일치시킬 수 있습니다.
- MSCS 클러스터 관리자에서 큐 관리자를 온라인으로 설정하고 테스트 워크로드를 부과하여 큐 관리자를 테스트하십시오. 테스트 큐 관리자를 실험하고 있다면 IBM WebSphere MQ 탐색기를 사용하십시오. 예를 들면, 다음과 같습니다.
 - Queues 트리 노드를 마우스 오른쪽 단추로 누른 후 New->Local Queue...을 선택하고 큐에 이름을 지정하십시오.
 - Finish을(를) 클릭하십시오. 큐가 작성되고 콘텐츠 보기에 표시됩니다.
 - 큐를 마우스 오른쪽 단추로 누른 다음 Put Test Message...을 선택하십시오. 테스트 메시지 넣기 패널이 표시됩니다.
 - 일부 메시지 텍스트를 입력한 후 Put Test Message을 클릭하고 패널을 닫으십시오.
 - 큐를 마우스 오른쪽 단추로 누른 다음 Browse Messages...을 선택하십시오. 메시지 브라우저 패널이 표시됩니다.
 - 메시지가 큐에 있는지 확인한 후 Close 을(를) 클릭하십시오. 메시지 브라우저 패널이 닫힙니다.
 - 큐를 마우스 오른쪽 단추로 누른 다음 Clear Messages...을 선택하십시오. 큐에 있는 메시지가 지워집니다.
 - 큐를 마우스 오른쪽 단추로 누른 다음 Delete...을 선택하십시오. 확인 패널이 표시됩니다. OK을(를) 클릭하십시오. 큐가 삭제됩니다.
- MSCS 클러스터 관리자를 사용하여 큐 관리자를 오프라인으로 가져오고 다시 온라인으로 되돌릴 수 있는지 테스트하십시오.
- 장애 복구를 시뮬레이션하십시오.

MSCS 클러스터 관리자에서 큐 관리자를 포함하는 그룹에서 마우스의 오른쪽 단추를 클릭하고 Move Group을 선택하십시오. 이를 수행하는 데에는 몇 분 정도 소요될 수 있습니다. (다른 때에 큐 관리자를 다른 노드로 신속히 이동시키려는 경우 300 페이지의 『MSCS 스토리지로 큐 관리자 이동』의 프로시저를 따르십시오.) 마우스 오른쪽 단추를 클릭하고 Initiate Failure를 선택할 수도 있습니다. 조치 (로컬 재시작 또는 장애 조치)는 현재 상태 및 구성 설정에 따라 다릅니다.

looksAlive 및 isAlive 폴링의 요약

looksAlive 및 isAlive는 MSCS가 자원 유형 제공 라이브러리 코드로 콜백하고 자원이 그 작업 상태를 판별하기 위해 검사를 수행하도록 요청하는 간격입니다. 이는 결국 MSCS가 자원의 장애를 복구하려고 시도하는지 여부를 판별합니다.

looksAlive 간격이 경과될 때마다(기본값 5000ms) 큐 관리자 자원이 호출되어 그 상태가 만족인지 판별하기 위한 자체 검사를 수행합니다.

isAlive 간격이 경과될 때마다(기본값 30000ms) 그에 대한 큐 관리자 자원이 다시 호출되어 자원이 올바르게 작동하는지 판별하기 위한 다른 검사를 수행합니다. 이 변수를 사용하면 두 레벨의 자원 유형 검사가 가능합니다.

1. 자원이 작동하는 것으로 표시되는 지 여부를 설정하기 위한 looksAlive 상태 검사입니다.
2. 큐 관리자 자원이 활성인지 판별하는 보다 중요한 isAlive 검사입니다.

큐 관리자 자원이 활성이 아닌 것으로 판별되는 경우 MSCS는 다른 고급 MSCS 옵션에 따라 자원 및 연관된 종속 자원에 대한 장애 복구를 클러스터의 다른 노드로 트리거합니다. 자세한 정보는 MSCS 문서를 참조하십시오.

MSCS 제어에서 큐 관리자 제거

MSCS 제어에서 큐 관리자를 제거하고 이를 수동 관리로 리턴할 수 있습니다.

유지보수 작업을 위해 MSCS 제어에서 큐 관리자를 제거할 필요는 없습니다. MSCS 클러스터 관리자를 사용하여 큐 관리자를 임시로 오프라인으로 가져와서 유지보수 작업을 수행할 수 있습니다. MSCS 제어에서 큐 관리자를 제거하는 것은 좀 더 영구적인 변경이며 MSCS가 더 이상 큐 관리자의 제어를 받지 않도록 결정한 경우에만 수행하십시오.

제거 중인 큐 관리자가 SSL 연결을 사용하는 경우 WebSphere MQ Explorer 또는 MQSC 명령인 ALTER QMGR를 사용하여 큐 관리자 속성인 SSLKEYR을 로컬 디렉토리의 SSL 키 저장소 파일을 가리키도록 수정해야 합니다.

프로시저는 다음과 같습니다.

1. 308 페이지의 『MSCS에서 큐 관리자를 오프라인으로 가져오기』에 설명된 대로 MSCS 클러스터 관리자를 사용하여 큐 관리자 자원을 오프라인으로 가져오십시오.
2. 자원 인스턴스를 파기하십시오. 이는 큐 관리자를 영구 삭제하지는 않습니다.
3. 선택적으로 큐 관리자 파일을 공유 드라이브에서 로컬 드라이브로 다시 마이그레이션하십시오. 이를 수행하려면 308 페이지의 『MSCS 스토리지에서 큐 관리자 리턴』의 내용을 참조하십시오.
4. 큐 관리자를 테스트하십시오.

MSCS에서 큐 관리자를 오프라인으로 가져오기

큐 관리자를 MSCS에서 오프라인으로 가져오려면 다음 단계를 수행하십시오.

1. MSCS 클러스터 관리자를 시작하십시오.
2. 클러스터에 대한 연결을 여십시오.
3. Groups를 선택하고 이동시킬 큐 관리자를 포함하는 그룹을 여십시오.
4. 큐 관리자 자원을 선택하십시오.
5. 마우스의 오른쪽 단추를 클릭하고 Offline을 선택하십시오.
6. 완료될 때까지 대기하십시오.

MSCS 스토리지에서 큐 관리자 리턴

이 프로시저는 큐 관리자가 해당 컴퓨터의 로컬 드라이브로 돌아가도록 구성합니다. 즉 이 큐 관리자는 보통 WebSphere MQ 큐 관리자가 됩니다. 이렇게 하려면 공유 디스크에서 로그 파일 및 데이터 파일을 이동합니다. 예를 들어, 기존 큐 관리자에는 E:\WebSphere MQ\log\<QMname> 및 E:\WebSphere MQ\qmgrs\<QMname>과(와) 같은 경로가 있을 수 있습니다. 파일을 수동으로 이동하지 마십시오. WebSphere MQ MSCS 지원의 일부로 제공된 hamvmqm 유틸리티 프로그램을 사용하십시오.

1. 큐 관리자를 종료하고 오류가 없는지 검사하십시오.
2. 큐 파일 및 로그 파일의 전체 매체 백업을 작성하고 안전한 위치에 백업을 저장하십시오(이 조치가 중요한 이유는 310 페이지의 『큐 관리자 로그 파일』 참조).
3. 사용할 로컬 드라이브를 결정하고 이 드라이브가 큐 관리자 로그 파일 및 데이터(큐) 파일을 저장하기에 충분한 용량을 가지는지 확인하십시오.
4. 파일이 현재 상주하는 공유 디스크가 큐 관리자 로그 및 데이터 파일을 이동할 클러스터 노드에서 온라인인지 확인하십시오.
5. 다음과 같이 유틸리티 프로그램을 실행하여 큐 관리자를 이동하십시오.

```
hamvmqm /m qmname /dd "c:\
WebSphere MQ" /ld "c:\
WebSphere MQ\log"
```

qmname에 큐 관리자 이름을, c에 로컬 디스크 드라이브 문자로 대체하고 WebSphere MQ에 대해 선택한 디렉토리(이 디렉토리는 이미 존재하지 않는 경우에 작성됨)로 대체합니다.

6. 300 페이지의 『MSCS 스토리지로 큐 관리자 이동』에 설명된 대로 큐 관리자를 테스트하여 큐 관리자가 작동하는지 확인하십시오.

MSCS 사용의 힌트 및 팁

이 절에는 MSCS에 대해 WebSphere MQ 지원을 효과적으로 사용하는 데 도움이 되는 몇 가지 일반적인 정보가 있습니다.

이 절에는 MSCS에 대해 WebSphere MQ 지원을 효과적으로 사용하는 데 도움이 되는 몇 가지 일반적인 정보가 있습니다.

한 시스템에서 다른 시스템으로 큐 관리자의 장애를 복구하는 데 걸리는 시간은? 이는 큐 관리자에 대한 워크로드의 양 및 트래픽 혼합(예: 동기점 내에서 지속적 트래픽의 양 및 실패 전 커미트된 트래픽 양)에 따라 매우 달라집니다. IBM 테스트는 약 1분의 장애 복구 및 실패 복구 시간을 제공합니다. 이는 초경량으로 로드된 큐 관리자이며 실제 시간은 로드량에 따라 상당히 달라집니다.

MSCS가 작동하는지 확인

다음 단계에 따라 실행 중인 MSCS 클러스터가 있는지 확인합니다.

300 페이지의 『[MSCS에서 사용할 큐 관리자 작성](#)』부터 시작하는 태스크 설명에서는 자원을 작성하고 마이그레이션하며 영구 삭제할 수 있는 실행 중인 MSCS 클러스터가 있는 것으로 가정합니다. 그러한 클러스터가 있는지 확인하려면 다음을 수행하십시오.

1. MSCS 클러스터 관리자를 사용하여 그룹을 작성하십시오.
2. 해당 그룹 내에서 시스템 클럭을 지정하여 일반 애플리케이션 자원의 인스턴스를 작성하십시오(경로 이름 C:\winnt\system32\clock.exe 및 C:\의 작업 디렉토리).
3. 자원을 온라인으로 가져올 수 있는지, 이 자원이 포함된 그룹을 다른 노드로 이동할 수 있는지 및 이 자원을 오프라인으로 가져올 수 있는지 확인하십시오.

수동 시동

MSCS가 관리하는 큐 관리자의 경우 시동 속성을 수동으로 설정해야 합니다. 이렇게 하면 WebSphere MQ MSCS 지원이 큐 관리자를 즉시 시작하지 않고도 IBM MQSeries 서비스를 재시작할 수 있습니다.

WebSphere MQ MSCS 지원은 모니터링 및 제어를 수행할 수 있도록 서비스를 재시작할 수 있어야 하지만 그 자체는 계속 어떤 시스템에서 어떤 큐 관리자를 실행 중인지 관리하고 있어야 합니다. 자세한 정보는 300 페이지의 『[MSCS 스토리지로 큐 관리자 이동](#)』의 내용을 참조하십시오.

MSCS 및 큐 관리자

MSCS 사용 시 큐 관리자에 관한 고려사항

다른 노드에서 일치하는 큐 관리자 작성

클러스터링이 WebSphere MQ에 대해 작업하려면 노드 A에 있는 각 노드 B에 동일한 큐 관리자가 필요합니다. 그러나 두 번째 항목을 명시적으로 작성할 필요는 없습니다. 한 노드에서 큐 관리자를 작성하거나 준비하고, 이를 300 페이지의 『[MSCS 스토리지로 큐 관리자 이동](#)』에 설명된 대로 다른 노드로 이동하면 이 큐 관리자는 해당 노드에서 완전하게 복제됩니다.

기본 큐 관리자

MSCS 제어 아래 기본 큐 관리자를 사용하지 마십시오. 큐 관리자는 기본값으로 만드는 특성을 가지고 있지 않습니다. WebSphere MQ는 자체 분리 레코드를 보관합니다. 장애 복구 시 기본값이 되도록 설정된 큐 관리자를 다른 컴퓨터로 이동하는 경우 다른 컴퓨터에서는 기본값이 되지 않습니다. 모든 애플리케이션이 이름별로 특정 큐 관리자를 참조하게 하십시오.

큐 관리자 삭제

큐 관리자가 노드를 이동한 경우 그 세부사항은 두 컴퓨터의 레지스트리에 존재합니다. 이를 삭제하려는 경우 한 컴퓨터에서는 평소대로 삭제한 다음 311 페이지의 『[IBM WebSphere MQ MSCS 지원 유틸리티 프로그램](#)』에서 설명된 대로 유틸리티를 실행하여 다른 컴퓨터에서 레지스트리를 정리하십시오.

기존 큐 관리자에 대한 지원

큐 관리자 로그 파일 및 큐 파일을 두 시스템 간 공유 SCSI에 있는 디스크에 넣을 수 있다면 MSCS 제어 아래 기존 큐 관리자를 넣을 수 있습니다(298 페이지의 그림 62 참조). MSCS 자원이 작성되는 동안 큐 관리자를 잠시 오프라인으로 가져올 수 있습니다.

새 큐 관리자를 작성하려는 경우 MSCS와는 별도로 작성하고, 테스트한 다음 MSCS 제어 아래 넣으십시오. 다음을 참조하십시오.

- 300 페이지의 『MSCS에서 사용할 큐 관리자 작성』
- 300 페이지의 『MSCS 스토리지로 큐 관리자 이동』
- 302 페이지의 『MSCS 제어 아래 큐 관리자 넣기』

관리할 큐 관리자를 MSCS에 알림

MSCS 클러스터 관리자를 통해 MSCS 제어 아래 배치할 큐 관리자를 선택하여 그러한 큐 관리자 각각에 대한 자원 인스턴스를 작성합니다. 이 프로세스는 해당 인스턴스가 관리하게 할 큐 관리자를 선택할 수 있는 자원 목록을 제시합니다.

큐 관리자 로그 파일

MSCS 스토리지로 큐 관리자 이동할 때 해당 로그 및 데이터 파일을 공유 디스크로 이동합니다(예는 300 페이지의 『MSCS 스토리지로 큐 관리자 이동』 참조).

이동하기 전에 큐 관리자를 확실히 닫고 데이터 파일 및 로그 파일의 전체 백업을 수행하는 것이 좋습니다.

다중 큐 관리자

WebSphere MQ MSCS 지원을 사용하면 각 시스템에서 다중 큐 관리자를 실행하고 개별 큐 관리자를 MSCS 제어 아래에 둘 수 있습니다.

클러스터 관리에 항상 MSCS 사용

제어 명령 또는 IBM WebSphere MQ 탐색기를 사용하여 MSCS의 제어를 받는 큐 관리자에서 직접 조작 시작 및 중지를 수행하려고 하지 마십시오. 대신, MSCS 클러스터 관리자를 사용하거나 큐 관리자를 온라인으로 가져오거나 오프라인으로 하십시오.

MSCS 클러스터 관리자를 사용하면 실제로 MSCS의 제어 밖에서 큐 관리자를 시작한 경우 큐 관리자가 오프라인인 MSCS 보고로 인해 발생할 수 있는 혼동을 부분적으로 방지합니다. 또한 MSCS를 사용하지 않고 큐 관리자를 중지하면 MSCS에 의해 실패로 감지되어 다른 노드로의 장애 복구가 시작됩니다.

활성/활성 모드에서 작업

MSCS 클러스터에 있는 두 컴퓨터 모두 활성/활성 모드로 큐 관리자를 실행할 수 있습니다. 대기 역할을 하는 완전히 유향 상태인 시스템이 반드시 있어야 하는 것은 아닙니다(그러나 가능하며 필요한 경우에는 활성/수동 모드).

워크로드를 실행하기 위해 두 시스템 모두 사용하려는 경우 각 시스템에 전체 클러스터 워크로드를 만족스러운 수준의 성능으로 실행하기에 충분한 용량(프로세서, 메모리, 보조 스토리지)을 제공하십시오.

참고: MSCS를 Microsoft Transaction Server(COM+)와 함께 사용 중인 경우 활성/활성 모드를 사용할 수 없습니다. 이는 MSCS 및 COM+가 있는 WebSphere MQ를 사용하려면 다음 사항이 필요하기 때문입니다.

- WebSphere MQ의 COM+ 지원을 사용하는 애플리케이션 컴포넌트가 COM+의 일부인 DTC(Distributed Transaction Coordinator)와 동일한 컴퓨터에서 실행해야 합니다.
- 큐 관리자도 같은 컴퓨터에서 실행되어야 합니다.
- DTC가 MSCS 자원으로 구성되어야 하고 따라서 언제든지 클러스터 내 컴퓨터 중 하나에서만 실행될 수 있습니다.

PostOnlineCommand 및 *PreOfflineCommand*

WebSphere MQ MSCS 지원을 다른 시스템과 통합하려면 이들 명령을 사용하십시오. 이들 명령은 WebSphere MQ 명령을 실행하는 데 사용할 수 있으며 몇 가지 제한사항이 있습니다.

매개변수에 이러한 명령을 IBM WebSphere MQ MSCS 유형의 자원에 지정하십시오. 이들 명령을 WebSphere MQ MSCS 지원을 다른 시스템 또는 프로시저와 통합하는 데 사용할 수 있습니다. 예를 들어, 메일 메시지를 송신하거나, 페이지를 활성화하거나, 다른 모니터링 시스템이 캡처할 일부 다른 양식의 경보를 생성하는 프로그램의 이름을 지정할 수 있습니다.

자원이 오프라인에서 온라인으로 변경될 때 PostOnlineCommand가 호출됩니다. PreOfflineCommand는 온라인에서 오프라인으로 변경 시 호출됩니다. 이들 명령을 호출하면 기본적으로 Windows 시스템 디렉토리에서 실행됩니다. WebSphere MQ에서는 32비트 자원 모니터 프로세스를 사용하므로, Windows 64비트 시스템에서는 이것이 \Windows\system32가 아닌 \Windows\SysWOW64 디렉토리입니다. 자세한 정보는 Windows x64 환경의 파일 경로 재지정에 관한 Microsoft 문서를 참조하십시오. 두 명령은 모두 MSCS 클러스터 서비스를 실행하는 데 사용되는 사용자 계정 아래에서 실행되며 비동기적으로 호출됩니다. WebSphere MQ MSCS 지원은 이들 명령이 완료될 때까지 대기하지 않고 계속 수행합니다. 이는 추가 클러스터 조작을 차단하거나 지연시킬 수 있는 위험을 제거합니다.

또한 WebSphere MQ 명령(예: 요청자 채널 재시작)을 실행하기 위해 이들 명령을 사용할 수 있습니다. 그러나 이러한 명령은 큐 관리자의 상태가 변하는 시점에 실행되므로 장기 실행 기능을 수행하는 것을 목적으로 하지 않으며 큐 관리자의 현재 상태에 대해 가정하지 말아야 합니다. 큐 관리자가 온라인 설정된 후 즉시 관리자가 오프라인 명령을 발행하는 것이 가능할 수 있습니다.

큐 관리자의 상태에 종속되는 프로그램을 실행하려는 경우 해당 프로그램을 큐 관리자 자원과 같은 MSCS 그룹에 배치하고 큐 관리자 자원에 종속되도록 설정하여 MSCS Generic Application 자원 유형의 인스턴스를 작성하는 것을 고려해 보십시오.

선호 노드 사용

활성/활성 모드를 사용하여 각 큐 관리자에 대해 선호 노드를 구성할 때 유용할 수 있습니다. 그러나 일반적으로 선호 노드를 설정하지 않고 수동 페일백에 의존하는 것이 더 좋습니다.

상대적으로 stateless한 일부 다른 자원과 다르게 큐 관리자는 한 노드에서 다른 노드로 장애 복구(또는 페일백)하는 동안 약간의 시간이 걸릴 수 있습니다. 불필요한 정지를 피하려면 큐 관리자를 복구된 노드로 페일백하기 전에 복구된 노드를 테스트하십시오. 이는 immediate 페일백 설정이 사용되지 않도록 합니다. 하루 중 특정 시간대에 발생하도록 페일백을 구성할 수 있습니다.

가장 안전한 경로는 노드가 완전히 복구되었다고 확신할 때 큐 관리자를 필요한 노드에 수동으로 이동하는 것일 수 있습니다. 이 경우 preferred node 옵션의 사용을 피할 수 있습니다.

COM+ 오류가 애플리케이션 이벤트 로그에서 발생하는 경우

새로 설치된 MSCS 클러스터에 WebSphere MQ를 설치할 때 애플리케이션 이벤트 로그에서 소스 COM+ 및 이벤트 ID 4691의 오류가 보고된 것을 발견할 수 있습니다.

이는 마이크로소프트 MSDTC (Distributed Transaction Coordinator) 가 이러한 환경에서 실행되도록 구성되지 않은 경우 마이크로소프트 클러스터 서버 (MSCS) 환경에서 WebSphere MQ 를 실행하려고 합니다. 클러스터된 환경에서 MSDTC 구성에 대한 정보는 Microsoft 문서를 참조하십시오.

IBM WebSphere MQ MSCS 지원 유틸리티 프로그램

명령 프롬프트에서 실행할 수 있는 MSCS 유틸리티 프로그램에 대한 IBM WebSphere MQ 지원의 목록입니다.

MSCS에 대한 IBM WebSphere MQ 지원은 다음 유틸리티 프로그램을 포함합니다.

자원 유형 등록/등록 해제

haregtyp.exe

IBM WebSphere MQ MSCS 자원 유형을 등록 취소 한 후에는 더 이상 해당 유형의 자원을 작성할 수 없습니다. MSCS에서는 클러스터 내에 해당 자원 유형의 인스턴스가 아직 있을 경우 이 자원 유형을 등록 해제할 수 없습니다.

1. 308 페이지의 『MSCS에서 큐 관리자를 오프라인으로 가져오기』에 설명된 대로 MSCS 클러스터 관리자를 사용하여 MSCS 제어 하에 실행되고 있는 큐 관리자를 오프라인으로 가져와서 해당 큐 관리자를 중지하십시오.
2. MSCS 클러스터 관리자를 사용하여 자원 인스턴스를 삭제하십시오.
3. 명령 프롬프트에서 다음 명령을 입력하여 자원 유형을 등록 해제하십시오.

```
haregtyp /u
```

유형을 등록(또는 나중에 등록 해제)하려는 경우 명령 프롬프트에 다음 명령을 입력하십시오.

```
haregtyp /r
```

MSCS 라이브러리를 성공적으로 등록한 후에는 IBM WebSphere MQ 설치 후 아직 시스템을 다시 시작하지 않은 경우 시스템을 다시 시작해야 합니다.

MSCS 스토리지로 큐 관리자 이동

hamvmqm.exe

300 페이지의 『[MSCS 스토리지로 큐 관리자 이동](#)』을 참조하십시오.

노드에서 큐 관리자 삭제

hadlrmqm.exe

클러스터에 있는 큐 관리자를 한 노드에서 다른 노드로 이동했지만 이제 영구 삭제하려고 하는 경우를 고려해 보십시오. IBM WebSphere MQ Explorer를 사용하여 현재 있는 노드에서 삭제하십시오. 이 큐 관리자에 대한 레지스트리 항목이 여전히 다른 컴퓨터에 남아 있습니다. 이러한 레지스트리 항목을 삭제하려면 해당 컴퓨터의 프롬프트에 다음 명령을 입력하십시오.

```
hadlrmqm /m qmname
```

여기서, qmname은 제거할 큐 관리자의 이름입니다.

설정 세부사항 확인 및 저장

amqmsysn.exe

이 유틸리티는 IBM WebSphere MQ MSCS 지원 설정의 전체 세부사항을 표시하는 대화 상자를 제공합니다. 예를 들어, IBM 지원 센터를 호출하는 경우 요청할 수 있습니다. 세부사항을 파일에 저장하는 옵션이 있습니다.

다중 인스턴스 큐 관리자

다중 인스턴스 큐 관리자는 여러 서버에서 구성된 동일한 큐 관리자의 인스턴스입니다. 큐 관리자의 한 인스턴스는 활성 인스턴스로 정의되고 다른 인스턴스는 대기 인스턴스로 정의됩니다. 활성 인스턴스가 실패하면 다중 인스턴스 큐 관리자가 대기 서버에서 자동으로 시작됩니다.

313 페이지의 그림 63에서는 QM1의 멀티 인스턴스 구성을 보여줍니다. IBM WebSphere MQ는 두 서버에 설치되며 그 중 하나는 예비 서버입니다. 하나의 큐 관리자 QM1이 작성되었습니다. QM1의 한 인스턴스는 활성이고 한 서버에서 실행 중입니다. QM1의 다른 인스턴스는 다른 서버에서 대기 상태로 실행 중이고 활성 처리를 수행하지 않지만 활성 인스턴스가 실패할 경우 QM1의 활성 인스턴스에서 인수할 준비를 하고 있습니다.

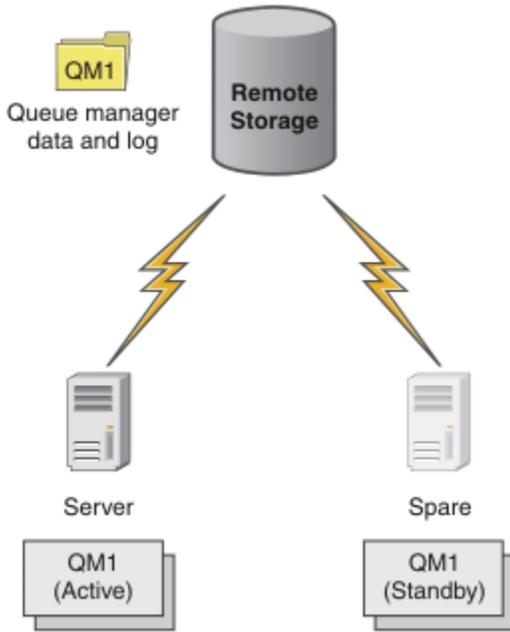


그림 63. 다중 인스턴스 큐 관리자

큐 관리자를 다중 인스턴스 큐 관리자로 사용하려고 할 때 **crtmqm** 명령을 사용하여 서버 중 하나에서 단일 큐 관리자를 작성하고 해당 큐 관리자 데이터 및 로그를 공유 네트워크 스토리지에 배치하십시오. 다른 서버에서는 큐 관리자를 다시 작성하는 대신 **addmqinf** 명령을 사용하여 네트워크 스토리지의 큐 관리자 데이터 및 로그에 대한 참조를 작성하십시오.

이제 두 서버 중 하나에서 큐 관리자를 실행할 수 있습니다. 각 서버는 동일한 큐 관리자 데이터 및 로그를 참조합니다. 큐 관리자는 한 개만 있으며 한 번에 한 서버에서만 활성화됩니다.

큐 관리자는 단일 인스턴스 큐 서버 또는 다중 인스턴스 큐 관리자로서 실행할 수 있습니다. 두 가지 경우 모두 큐 관리자의 한 인스턴스만 실행하고 요청을 처리합니다. 다른 점은 다중 인스턴스 큐 관리자로서 실행할 때 큐 관리자의 활성 인스턴스를 실행하고 있지 않은 서버가 대기 인스턴스로 실행되며 활성 서버가 실패할 경우 자동으로 활성 인스턴스에서 인수할 준비를 합니다.

두 서버에서 큐 관리자를 시작하는 순서로 어느 인스턴스가 먼저 활성화되는지에 대해 제어할 수 있습니다. 큐 관리자 데이터에 대해 읽기/쓰기 잠금을 확보하는 첫 번째 인스턴스가 활성 인스턴스가 됩니다.

활성 인스턴스가 시작된 후에는 제어를 대기로 전송하기 위한 전환 옵션을 사용하여 활성 인스턴스를 중지시켜 이 활성 인스턴스를 다른 서버로 스왑할 수 있습니다.

QM1의 활성 인스턴스는 실행 중일 때 공유 큐 관리자 데이터 및 로그 폴더에 대해 독점 액세스를 가집니다. QM1의 대기 인스턴스는 활성 인스턴스가 실패했을 때 이를 감지하고 활성 인스턴스가 됩니다. 활성 인스턴스가 남긴 상태의 QM1 데이터 및 로그를 인수하고 클라이언트 및 채널로부터의 재연결을 승인합니다.

활성 인스턴스는 대기 인수를 발생시키는 다양한 이유로 인해 실패할 수 있습니다.

- 활성 큐 관리자 인스턴스를 호스팅하는 서버의 실패
- 활성 큐 관리자 인스턴스를 호스팅하는 서버와 파일 시스템 간의 연결 실패
- 큐 관리자 프로세스의 무반응. WebSphere MQ는 이를 감지한 후 큐 관리자를 시스템 종료합니다.

큐 관리자 구성 정보를 다중 서버에 추가하고 활성/대기 쌍으로 실행할 두 개의 서버를 선택할 수 있습니다. 총 인스턴스 수는 2개로 제한됩니다. 두 개의 대기 인스턴스와 한 개의 활성 인스턴스를 가질 수 없습니다.

다중 인스턴스 큐 관리자는 고가용성 솔루션의 한 부분입니다. 유용한 고가용성 솔루션을 빌드하려면 몇 가지 추가 컴포넌트가 필요합니다.

- 활성 큐 관리자 인스턴스 실행을 인수하는 컴퓨터로 WebSphere MQ 연결을 전환하기 위한 클라이언트 및 채널 재연결.

- 올바르게 잠금을 관리하고 매체 및 파일 서버 실패에 대한 보호를 제공하는 고성능 공유 네트워크 파일 시스템 (NFS).

중요사항: NFS 드라이브에서 유지보수를 수행하려면 사용자 환경에서 실행 중인 모든 다중 인스턴스 큐 관리자를 중지해야 합니다. NFS 실패가 발생할 경우 복구하기 위한 큐 관리자 구성 백업이 있는지 확인하십시오.

- 기본 인프라에서 단일 실패 지점을 제거하기 위한 복원 네트워크 및 전원 공급 장치
- 장애 복구를 허용하는 애플리케이션. 특히 WebSphere MQ 큐를 찾아보는 애플리케이션 및 트랜잭션 애플리케이션의 작동에 세심한 주의를 기울여야 합니다.
- 활성화 및 대기 인스턴스가 실행 중인지 확인하고 실패한 활성화 인스턴스를 재시작하기 위한 활성화 및 대기 인스턴스 모니터링 및 관리. 다중 인스턴스 큐 관리자가 자동으로 재시작되더라도 대기 인스턴스가 실행 중이고 인수할 준비가 되었는지 및 실패한 인스턴스가 새 대기 인스턴스로서 온라인 상태로 되었는지 확인해야 합니다.

WebSphere MQ MQI 클라이언트 및 채널이 대기 큐 관리자가 활성화 될 때 자동으로 연결됩니다. 다시 연결 및고가용성 솔루션의 기타 컴포넌트에 대한 자세한 정보는 관련 주제에서 찾을 수 있습니다. Java용 IBM WebSphere MQ 클래스에서는 자동 클라이언트 다시 연결이 지원되지 않습니다.

지원 플랫폼

버전 7.0.1에서는 biz/OS 플랫폼에서 멀티 인스턴스 큐 관리자를 작성할 수 있습니다.

버전 7.0.1부터는 MQI 클라이언트에 대해 자동 클라이언트 재연결이 지원됩니다.

다중 인스턴스 큐 관리자 작성

한 서버에서 큐 관리자를 작성하고 다른 한 서버에서 IBM WebSphere MQ를 구성하여 다중 인스턴스 큐 관리자를 작성합니다. 다중 인스턴스 큐 관리자는 큐 관리자 데이터 및 로그를 공유합니다.

다중 인스턴스 큐 관리자를 작성하는 데 있어 대부분의 수고가 필요한 부분은 공유 큐 관리자 데이터 및 로그 파일을 설정하는 작업입니다. 네트워크 스토리지에서 공유 디렉토리를 작성해야 하고 작성한 디렉토리를 네트워크 공유를 사용하는 다른 서버가 사용할 수 있도록 설정해야 합니다. 이러한 작업은 UNIX and Linux 시스템의 루트와 같은 관리 권한을 가진 사용자에게 의해 수행되어야 합니다. 단계는 다음과 같습니다.

1. 데이터 및 로그 파일에 대한 공유를 작성하십시오.
2. 서버에서 큐 관리자를 작성하십시오.
3. 첫 번째 서버에서 **dspmqinf** 명령을 실행하여 큐 관리자 구성 데이터를 수집하고 이를 클립보드에 복사하십시오.
4. 복사한 데이터로 **addmqinf** 명령을 실행하여 두 번째 서버에서 큐 관리자 구성을 작성하십시오.

두 번째 서버에서 큐 관리자를 다시 작성하기 위해 **crtmqm**을 실행하지 않습니다.

파일 액세스 제어

다른 모든 서버에서 사용자 및 그룹 **mqm**이 공유에 액세스할 수 있음에 주의해야 합니다.

On UNIX and Linux, you need to make the uid and gid of mqm the same on all the systems. You might need to edit /etc/passwd on each system to set a common uid and gid for mqm, and then reboot your system.

Microsoft Windows에서 큐 관리자 프로세스를 실행하는 사용자 ID에는 큐 관리자 데이터 및 로그 파일을 포함하는 디렉토리에 대한 전체 제어 권한이 있어야 합니다. 이 권한을 다음 두 가지 방식으로 구성할 수 있습니다.

1. 대체 보안 프린시펄로 글로벌 그룹을 사용하여 큐 관리자를 작성하십시오. 큐 관리자 데이터 및 로그 파일이 포함된 디렉토리에 대한 전체 제어 액세스 권한을 글로벌 그룹에 부여하십시오. [339 페이지의 『Windows에서 공유 큐 관리자 데이터 및 로그 디렉토리 보안』](#)의 내용을 참조하십시오. 큐 관리자를 실행하는 사용자 ID가 글로벌 그룹의 멤버가 되도록 하십시오. 로컬 사용자를 글로벌 그룹의 멤버로 설정할 수 없으므로 큐 관리자 프로세스는 도메인 사용자 ID로 실행되어야 합니다. 도메인 사용자 ID는 로컬 그룹 **mqm**의 멤버여야 합니다. [316 페이지의 『도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성』](#) 태스크는 이 방식으로 보안 설정된 파일을 사용하여 다중 인스턴스 큐 관리자를 설정하는 방법을 보여줍니다.
2. 로컬 **mqm** 그룹이 도메인 범위 "도메인 로컬"을 가지도록 도메인 제어기에서 큐 관리자를 작성하십시오. 도메인 로컬 **mqm**으로 파일 공유를 보안 설정하고 같은 도메인 로컬 **mqm** 그룹에 있는 큐 관리자의 모든 인스턴스

에서 큐 관리자 프로세스를 실행하십시오. 330 페이지의 『도메인 제어기에 다중 인스턴스 큐 관리자 작성』 태스크는 이 방식으로 보안 설정된 파일을 사용하여 다중 인스턴스 큐 관리자를 설정하는 방법을 보여줍니다.

구성 정보

각 서버에 대한 IBM WebSphere MQ 큐 관리자 구성 정보를 수정하여 필요한 수만큼 큐 관리자 인스턴스를 구성하십시오. 각 서버에는 같은 버전의 IBM WebSphere MQ가 호환 가능한 수정사항 레벨로 설치되어 있어야 합니다. **dspmqlnf** 및 **addmqinf** 명령은 추가 큐 관리자 인스턴스를 구성하는 것을 지원합니다. 또는 **mqs.ini** 및 **qm.ini** 파일을 직접 편집할 수 있습니다. 350 페이지의 『Linux에서 다중 인스턴스 큐 관리자 작성』, 316 페이지의 『도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성』 및 330 페이지의 『도메인 제어기에 다중 인스턴스 큐 관리자 작성』 주제는 다중 인스턴스 큐 관리자를 구성하는 방법을 보여주는 예입니다.

Windows, UNIX and Linux 시스템에서 단일 **mqs.ini** 파일을 네트워크 공유에 배치하고 이 파일을 가리키도록 **AMQ_MQS_INI_LOCATION** 환경 변수를 설정하여 공유할 수 있습니다.

제한

1. 동일한 큐 관리자의 다중 인스턴스를 같은 운영 체제 아키텍처 및 엔디언을 가지는 서버에서만 구성하십시오. 예를 들어, 두 시스템 모두 32비트거나 64비트여야 합니다.
2. 모든 IBM WebSphere MQ 설치는 릴리스 레벨 7.0.1 이상이어야 합니다.
3. 일반적으로 활성화 및 대기 설치는 같은 유지보수 레벨로 유지보수됩니다. 각 업그레이드에 대해 유지보수 지시 사항을 참조하여 모든 설치를 함께 업그레이드해야 하는지 여부를 확인하십시오.
활성 및 수동 큐 관리자의 유지보수 레벨이 동일해야 함에 유의하십시오.
4. 같은 IBM WebSphere MQ 사용자, 그룹 및 액세스 제어 메커니즘으로 구성되는 큐 관리자 사이에서만 큐 관리자 데이터 및 로그를 공유합니다.
5. UNIX and Linux 시스템에서 소프트 마운트가 아닌 인터럽트 가능한 하드 마운트로 네트워크 스토리지에서 공유 파일 시스템을 구성하십시오. 인터럽트 가능한 하드 마운트는 큐 관리자가 시스템 호출에 의해 인터럽트 될 때까지 큐 관리자가 정지하도록 강제 실행합니다. 소프트 마운트는 서비스 실패 후의 데이터 일관성을 보장하지 않습니다.
6. 공유 로그 및 데이터 디렉토리는 FAT 또는 NFSv3 파일 시스템에 저장할 수 없습니다. Windows에서 다중 인스턴스 큐 관리자의 경우 네트워크 스토리지는 Windows 네트워크에서 사용되는 CIFS(Common Internet File System) 프로토콜에 의해 액세스 가능해야 합니다.

Windows 도메인 및 다중 인스턴스 큐 관리자

Windows의 다중 인스턴스 큐 관리자에서는 해당 데이터 및 로그가 공유되어야 합니다. 공유는 여러 서버 또는 워크스테이션에서 실행 중인 큐 관리자의 모든 인스턴스가 액세스할 수 있어야 합니다. 큐 관리자 및 공유를 Windows 도메인의 부분으로 구성하십시오. 큐 관리자는 도메인 워크스테이션 또는 서버 또는 도메인 제어기에서 실행할 수 있습니다.

다중 인스턴스 큐 관리자를 구성하기 전에 342 페이지의 『Windows에서 비공유 큐 관리자 데이터와 로그 디렉토리 및 파일 보안 설정』 및 339 페이지의 『Windows에서 공유 큐 관리자 데이터 및 로그 디렉토리 보안』의 내용을 읽고 큐 관리자 데이터 및 로그 파일에 대한 액세스를 제어하는 방법을 검토하십시오. Windows 도메인에서 다중 인스턴스 큐 관리자의 공유 디렉토리 설정으로 바로 이동하려는 경우 이 주제는 교육적입니다. 316 페이지의 『도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성』의 내용을 참조하십시오.

도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 실행

Version 7.1에서 다중 인스턴스 큐 관리자는 워크스테이션 또는 도메인의 멤버인 서버에서 실행됩니다. Version 7.1 이전에서는 다중 인스턴스 큐 관리자가 도메인 제어기에서만 실행되었습니다. 316 페이지의 『도메인 제어기에서 다중 인스턴스 큐 관리자 실행』의 내용을 참조하십시오. Windows에서 다중 인스턴스 큐 관리자를 실행하려면 도메인 제어기, 파일 서버 및 같은 도메인에 연결된 같은 큐 관리자를 실행하는 두 개의 워크스테이션 또는 서버가 필요합니다.

도메인 내 임의의 서버 또는 워크스테이션에서 다중 인스턴스 큐 관리자를 실행할 수 있게 한 변경은 이제 추가 보안 그룹으로 큐 관리자를 작성할 수 있다는 것입니다. 추가 보안 그룹은 **crtmqm** 명령에서 **-a** 매개변수로 전달됩니다. 이 그룹으로 큐 관리자 데이터 및 로그를 포함하는 디렉토리를 보안 설정합니다. 큐 관리자 프로세스를 실행하는 사용자 ID는 이 그룹의 멤버여야 합니다. 큐 관리자가 디렉토리에 액세스하면 Windows에서는 사용자

ID가 디렉토리에 액세스하기 위해 가지고 있는 권한을 검사합니다. 그룹 및 사용자 ID 도메인 범위를 둘 다 지정하면 큐 관리자 프로세스를 실행하는 사용자 ID는 글로벌 그룹의 신임 정보를 가집니다. 큐 관리자가 다른 서버에서 실행되고 있는 경우 큐 관리자 프로세스를 실행하는 사용자 ID는 같은 신임 정보를 가질 수 있습니다. 이 사용자 ID는 같을 필요는 없습니다. 이 사용자 ID는 로컬 mqm 그룹의 멤버일 뿐만 아니라 대체 보안 그룹의 멤버여야 합니다.

다중 인스턴스 큐 관리자를 작성하는 태스크는 한 가지 변경을 제외하고 Version 7.0.1에서와 같습니다.

crtmqm 명령의 매개변수에 추가 보안 그룹 이름을 추가해야 합니다. 이 태스크는 [316 페이지의 『도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성』](#)에 설명되어 있습니다.

도메인 및 도메인 서버와 워크스테이션을 구성하는 데에는 여러 단계가 필요합니다. Windows에서 해당 데이터 및 로그 디렉토리에 대한 큐 관리자의 액세스에 권한을 부여하는 방법에 대해 이해해야 합니다. 해당 로그 및 데이터 파일에 액세스하기 위한 권한이 큐 관리자 프로세스에 부여되는 방식에 대해 확신할 수 없으면 [342 페이지의 『Windows에서 비공유 큐 관리자 데이터와 로그 디렉토리 및 파일 보안 설정』](#) 주제를 읽어보십시오. 이 주제는 필요한 단계를 이해하는 데 유용한 두 개의 태스크를 포함합니다. 이 태스크는 [344 페이지의 『로컬 mqm 그룹에서 권한 부여된 데이터 및 로그 파일 읽기 및 쓰기』](#) 및 [346 페이지의 『대체 로컬 보안 그룹이 권한을 부여한 데이터 및 로그 파일 읽기 및 쓰기』](#)입니다. 다른 주제인 [339 페이지의 『Windows에서 공유 큐 관리자 데이터 및 로그 디렉토리 보안』](#)에서는 큐 관리자 데이터 및 로그 파일을 포함하는 공유 디렉토리를 대체 보안 그룹을 사용하여 보안 설정하는 방식을 설명합니다. 이 주제는 Windows 도메인 설정, 파일 공유 작성, IBM WebSphere MQ for Windows 설치 및 공유를 사용하도록 큐 관리자 구성의 네 개 태스크를 포함합니다. 이러한 태스크는 다음과 같습니다.

1. [319 페이지의 『IBM WebSphere MQ 에 대한 Active Directory 및 DNS 도메인 작성』](#).
2. [322 페이지의 『Windows 도메인의 서버 또는 워크스테이션에 IBM WebSphere MQ 설치』](#).
3. [325 페이지의 『큐 관리자 데이터 및 로그 파일을 위한 공유 디렉토리 작성』](#).
4. [327 페이지의 『대체 글로벌 보안 그룹이 권한을 부여한 공유 데이터 및 로그 파일 읽기 및 쓰기』](#).

도메인을 사용하여 [316 페이지의 『도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성』](#) 태스크를 수행할 수 있습니다. 이러한 태스크를 수행하여 지식을 프로덕션 도메인에 전송하기 전에 다중 인스턴스 큐 관리자 설정에 대해 알아보십시오.

도메인 제어기에서 다중 인스턴스 큐 관리자 실행

Version 7.0.1에서 다중 인스턴스 큐 관리자는 도메인 제어기에서만 실행되었습니다. 큐 관리자 데이터는 도메인 mqm 그룹을 사용하여 보안 설정할 수 있습니다. [339 페이지의 『Windows에서 공유 큐 관리자 데이터 및 로그 디렉토리 보안』](#) 주제에서 설명한 대로 워크스테이션 또는 서버에서 로컬 mqm 그룹으로 보안 설정된 디렉토리는 공유할 수 없습니다. 그러나 도메인 제어기에서는 모든 그룹 및 프린시펄에 도메인 범위가 있습니다. 도메인 제어기에 IBM WebSphere MQ for Windows를 설치하면 큐 관리자 데이터 및 로그 파일이 도메인 mqm 그룹으로 보안 설정되며 공유할 수 있습니다. [330 페이지의 『도메인 제어기에 다중 인스턴스 큐 관리자 작성』](#) 태스크의 단계에 따라 도메인 제어기에서 다중 인스턴스 큐 관리자를 구성하십시오.

관련 정보

[도메인 제어기 역할의 Windows 2000, Windows Server 2003 및 Windows Server 2008 클러스터 노트](#)

도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성

이 예에서는 Windows 도메인의 일부인 워크스테이션 또는 서버의 Windows에서 다중 인스턴스 큐 관리자를 설정하는 방법을 보여줍니다. 이 서버가 반드시 도메인 제어기일 필요는 없습니다. 이 설정에서는 프로덕션 스케일이 아닌 관련된 개념을 보여줍니다. 예는 Windows Server 2008에 기반합니다. 다른 버전의 Windows Server에서는 단계가 다를 수 있습니다.

프로덕션 스케일 구성에서는 기존 도메인에 맞게 구성을 조정해야 할 수 있습니다. 예를 들어, 여러 공유에 권한을 부여하고 큐 관리자를 실행하는 사용자 ID를 그룹화할 수 있도록 여러 도메인 그룹을 정의할 수 있습니다.

이 구성 예는 다음 세 개의 서버로 구성됩니다.

sun

Windows Server 2008 도메인 제어기. *Sun, mars* 및 *venus*가 들어 있는 *wmq.example.com* 도메인을 소유합니다. 설명을 위해 파일 서버로도 사용됩니다.

mars

첫 번째 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 *QMGR*이라고 하는 다중 인스턴스 큐 관리자의 한 인스턴스를 포함합니다.

venus

두 번째 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 *QMGR*이라고 하는 다중 인스턴스 큐 관리자의 두 번째 인스턴스를 포함합니다.

예에서 이탤릭체로 표시된 이름을 선택한 이름으로 바꾸십시오.

시작하기 전에

Windows에서는 큐 관리자 데이터와 로그 파일을 저장하려는 파일 시스템을 확인할 필요가 없습니다. 공유 파일 시스템 작동 확인 검사 프로시저는 UNIX and Linux에 적용할 수 있습니다. Windows에서는 검사가 항상 성공적입니다.

다음 태스크에 있는 단계를 수행하십시오. 이 태스크에서는 도메인 제어기 및 도메인을 작성하고 서버에 IBM WebSphere MQ for Windows를 설치하며 데이터 및 로그 파일에 대한 파일 공유를 작성합니다. 기존 도메인 제어기를 구성하면 새 Windows Server 2008에서 단계를 시도하는 데 유용할 수 있습니다. 사용자 도메인에 단계를 적용시킬 수 있습니다.

1. 319 페이지의 『IBM WebSphere MQ 에 대한 Active Directory 및 DNS 도메인 작성』.
2. 322 페이지의 『Windows 도메인의 서버 또는 워크스테이션에 IBM WebSphere MQ 설치』.
3. 325 페이지의 『큐 관리자 데이터 및 로그 파일을 위한 공유 디렉토리 작성』.
4. 327 페이지의 『대체 글로벌 보안 그룹이 권한을 부여한 공유 데이터 및 로그 파일 읽기 및 쓰기』.

이 태스크 정보

이 태스크는 큐 관리자의 인스턴스를 실행하기 위해 도메인에서 도메인 제어기 및 두 개의 서버를 구성하는 태스크 시퀀스 중 하나입니다. 이 태스크에서는 큐 관리자 *QMGR*의 또 다른 인스턴스를 실행하도록 두 번째 서버 *venus*를 구성합니다. 이 태스크의 단계에 따라 큐 관리자 *QMGR*의 두 번째 인스턴스를 작성하고 인스턴스가 작동하는지 테스트합니다.

이 태스크는 이전 절에 있는 네 개의 태스크와는 별도로입니다. 이 태스크에는 단일 인스턴스 큐 관리자를 다중 인스턴스 큐 관리자로 변환하는 단계가 포함됩니다. 다른 모든 단계는 단일 또는 다중 인스턴스 큐 관리자와 공통입니다.

프로시저

1. IBM WebSphere MQ for Windows를 실행하도록 두 번째 서버를 구성하십시오.
 - a) 322 페이지의 『Windows 도메인의 서버 또는 워크스테이션에 IBM WebSphere MQ 설치』 태스크의 단계를 수행하여 두 번째 도메인 서버를 작성하십시오. 이 태스크 순서에서 두 번째 서버를 *venus*라고 합니다.

팁: 두 서버 각각에서 IBM WebSphere MQ에 대해 동일한 설치 기본값을 사용하여 두 번째 설치를 작성하십시오. 기본값이 다르면 IBM WebSphere MQ 구성 파일 `mqs.ini`의 *QMGR QueueManager* 스탠자에 있는 `접두부` 및 `InstallationName` 변수를 조정해야 합니다. 이 변수는 각 서버에서 설치 및 큐 관리자마다 다를 수 있는 경로를 나타냅니다. 모든 서버에서 경로가 동일하게 유지되는 경우 다중 인스턴스 큐 관리자 구성이 더 간단해집니다.
2. *venus*에서 두 번째 *QMGR* 인스턴스를 작성하십시오.
 - a) *mars*의 *QMGR*이 존재하지 않는 경우 327 페이지의 『대체 글로벌 보안 그룹이 권한을 부여한 공유 데이터 및 로그 파일 읽기 및 쓰기』 태스크를 수행하십시오.
 - b) *venus*에서 `Prefix` 및 `InstallationName` 매개변수 값이 올바른지 확인하십시오.

*mars*에서 **`dspmqinf`** 명령을 실행하십시오.

```
dspmqinf QMGR
```

시스템이 다음과 같이 응답합니다.

```
QueueManager:  
  Name=QMGR  
  Directory=QMGR  
  Prefix=C:\Program Files\IBM\WebSphere MQ  
  DataPath=\\sun\wmq\data\QMGR  
  InstallationName=Installation1
```

- c) **QueueManager** 스텐자의 기계 판독 가능(machine-readable) 양식을 클립보드에 복사하십시오.

mars 에서 `-o` command 매개변수를 사용하여 **dspmqinf** 명령을 다시 실행하십시오.

```
dspmqinf -o command QMGR
```

시스템이 다음과 같이 응답합니다.

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"  
-v DataPath=\\sun\wmq\data\QMGR
```

- d) *venus*에서 클립보드의 **addmqinf** 명령을 실행하여 *venus*에 큐 관리자 인스턴스를 작성하십시오.

`Prefix` 또는 `InstallationName` 매개변수의 차이를 수용할 수 있도록 필요한 경우 명령을 조정하십시오.

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"  
-v DataPath=\\sun\wmq\data\QMGR
```

WebSphere MQ configuration information added.

3. *venus*에서 대기 인스턴스를 허용하여 큐 관리자 *QMGR*를 시작하십시오.

- a) *mars*에서 *QMGR*이 중지되었는지 확인하십시오.

*mars*에서 **dspmq** 명령을 실행하십시오.

```
dspmq -m QMGR
```

시스템 응답은 큐 관리자가 중지된 방식에 따라 다릅니다. 예를 들면, 다음과 같습니다.

```
C:\Users\Administrator>dspmq -m QMGR  
QMNAME(QMGR) STATUS(Ended immediately)
```

- b) *venus*에서 **strmqm** 명령을 실행하여 *QMGR*를 시작하고 대기를 허용하십시오.

```
strmqm -x QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation 'Installation1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

결과

다중 인스턴스 큐 관리자 스위치를 테스트하려면 다음 단계를 수행하십시오.

1. *mars*에서 **strmqm** 명령을 실행하여 *QMGR*를 시작하고 대기를 허용하십시오.

```
strmqm -x QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation 'Installation1'.  
A standby instance of queue manager 'QMGR' has been started.  
The active instance is running elsewhere.
```

2. *venus*에서 **endmqm** 명령을 실행하십시오.

```
endmqm -r -s -i QMGR
```

*venus*에 대해 시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere MQ queue manager 'QMGR' ended, permitting switchover to  
a standby instance.
```

*mars*에 대한 시스템 응답은 다음과 같습니다.

```
dspmq  
QMNAME(QMGR) STATUS(Running as standby)  
C:\Users\wmquser2>dspmq  
QMNAME(QMGR) STATUS(Running as standby)  
C:\Users\wmquser2>dspmq  
QMNAME(QMGR) STATUS(Running)
```

다음에 수행할 작업

샘플 프로그램을 사용하여 다중 인스턴스 큐 관리자를 확인하려면 337 페이지의 『Windows에서 멀티 인스턴스 큐 관리자 확인』의 내용을 참조하십시오.

*IBM WebSphere MQ*에 대한 *Active Directory* 및 *DNS* 도메인 작성

이 태스크에서는 *sun*이라고 하는 Windows 2008도메인 제어기에서 도메인 *wmq.example.com*을 작성합니다. 한 명의 사용자와 적절한 권한을 사용하여 도메인에서 Domain *mqm* 글로벌 그룹을 구성합니다.

프로덕션 스케일 구성에서는 기존 도메인에 맞게 구성을 조정해야 할 수 있습니다. 예를 들어, 여러 공유에 권한을 부여하고 큐 관리자를 실행하는 사용자 ID를 그룹화할 수 있도록 여러 도메인 그룹을 정의할 수 있습니다.

이 구성 예는 다음 세 개의 서버로 구성됩니다.

sun

Windows Server 2008 도메인 제어기. *Sun*, *mars* 및 *venus*가 들어 있는 *wmq.example.com* 도메인을 소유합니다. 설명을 위해 파일 서버로도 사용됩니다.

mars

첫 번째 *IBM WebSphere MQ* 서버로 사용되는 Windows Server 2008. 이 서버는 *QMGR*이라고 하는 다중 인스턴스 큐 관리자의 한 인스턴스를 포함합니다.

venus

두 번째 *IBM WebSphere MQ* 서버로 사용되는 Windows Server 2008. 이 서버는 *QMGR*이라고 하는 다중 인스턴스 큐 관리자의 두 번째 인스턴스를 포함합니다.

예에서 이탤릭체로 표시된 이름을 선택한 이름으로 바꾸십시오.

시작하기 전에

1. 태스크 단계는 설치되지만 역할로 구성되지 않은 Windows Server 2008과 일치합니다. 기존 도메인 제어기를 구성하면 새 Windows Server 2008에서 단계를 시도하는 데 유용할 수 있습니다. 사용자 도메인에 단계를 적용시킬 수 있습니다.

이 태스크 정보

이 태스크에서는 새 도메인 제어기에서 Active Directory 및 DNS 도메인을 작성합니다. 그런 다음 도메인에 조인한 다른 서버 및 워크스테이션에서 IBM WebSphere MQ를 설치할 준비가 되도록 구성합니다. Windows 도메인을 작성하기 위해 Active Directory를 설치하고 구성하는 것에 익숙하지 않은 경우 이 태스크를 따르십시오. 다중 인스턴스 큐 관리자 구성을 작성하기 위해서는 Windows 도메인을 작성해야 합니다. 이 태스크는 Windows 도메인을 구성하기 위한 최적의 방법을 안내하기 위한 것은 아닙니다. 프로덕션 환경에서 다중 인스턴스 큐 관리자를 배치하려면 Windows 문서를 참조해야 합니다.

태스크 중 다음 단계를 수행하십시오.

1. Active Directory 설치
2. 도메인 추가
3. DNS에 도메인 추가
4. 글로벌 그룹 Domain mqm을 작성하여 적절한 권한 부여
5. 사용자를 추가하고 글로벌 그룹 Domain mqm의 구성원으로 작성하십시오.

이 태스크는 큐 관리자 데이터 및 로그 파일 액세스를 설명하는 관련 태스크 세트 중 하나입니다. 이 태스크는 사용자가 선택한 디렉토리에 저장된 데이터 및 로그 파일을 읽고 쓸 수 있도록 권한이 부여된 큐 관리자를 작성하는 방법을 보여줍니다. 315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』 태스크를 수반합니다.

이 태스크에서 도메인 제어기 호스트 이름은 *sun*이고, 두 IBM WebSphere MQ 서버는 *mars*와 *venus*라고 합니다. 도메인은 *wmq.example.com*이라고 합니다. 태스크에서 이탤릭체로 표시된 이름은 모두 사용자가 선택한 이름으로 바꿀 수 있습니다.

프로시저

1. 로컬 또는 Workgroup 관리자로 도메인 컨트롤러 (*sun*) 에 로그인하십시오.
서버가 도메인 제어기로 이미 구성되어 있는 경우 도메인 관리자로 로그인해야 합니다.
2. Active Directory 도메인 서비스 마법사를 실행하십시오.
 - a) 시작 > 실행 ... 을 클릭하십시오. `dcpromo` 를 입력하고 **확인**을 클릭하십시오.
Active Directory 2진 파일이 아직 설치되지 않은 경우 Windows에서 파일을 자동으로 설치합니다.
3. 마법사의 첫 번째 창에서 **고급 모드 설치 사용** 선택란을 지워진 상태로 두십시오. **다음 > 다음** 을 클릭하고 **새 포리스트에서 새 도메인 작성 > 다음** 을 클릭하십시오.
4. *wmq.example.com*을 포리스트 루트 도메인의 **FQDN** 필드에 입력하십시오. **다음**을 클릭하십시오.
5. Set Forest Functional Level창에서 **Forest functional levels > Next**목록에서 **Windows Server 2003** 이상을 선택하십시오.
IBM WebSphere MQ 에서 지원하는 Windows 서버의 가장 오래된 레벨은 Windows Server 2003입니다.
6. 옵션: 도메인 기능 레벨 설정 창에서 **도메인 기능 레벨 > 다음**의 목록에서 **Windows Server 2003** 이상을 선택하십시오.
이 단계는 포리스트 기능 레벨을 **Windows Server 2003**으로 설정한 경우에만 필요합니다.
7. **DNS 서버**가 추가 옵션으로 선택된 상태로 추가 도메인 제어기 옵션 창이 열립니다. **다음** 및 **예**를 클릭하여 경고 창을 지우십시오.
팁: DNS 서버가 이미 설치된 경우에는 이 옵션이 표시되지 않습니다. 이 태스크를 정확히 수행하려면 이 도메인 제어기에서 모든 역할을 제거하고 다시 시작하십시오.
8. Database, Log Files 및 SYSVOL 디렉토리를 변경하지 않은 채로 두십시오. **다음**을 클릭하십시오.

9. 디렉토리 서비스 복원 모드 관리자 비밀번호 창에 **비밀번호 및 비밀번호 확인** 필드에 비밀번호를 입력하십시오. **다음 > 다음**을 클릭하십시오. 마지막 마법사 창에서 **완료 시 시스템 다시 시작**을 선택하십시오.
10. 도메인 제어기 시스템이 다시 시작되면 `wmq\Administrator`로 로그인하십시오.
서버 관리자가 자동으로 시작됩니다.
11. `wmq.example.com\Users` 폴더를 여십시오.
a) 서버 관리자 > 역할 > **Active Directory 도메인 서비스** > `wmq.example.com` > **사용자**를 여십시오.
12. 사용자 > 새로 작성 > 그룹을 마우스 오른쪽 단추로 클릭하십시오.
a) **그룹 이름** 필드에 그룹 이름을 입력하십시오.
참고: 선호되는 그룹 이름은 Domain mqm입니다. 표시된 그대로 입력하십시오.
 - 그룹 Domain mqm을 호출하면 도메인 워크스테이션 또는 서버에서의 "IBM WebSphere MQ 준비" 마법사 작동이 수정됩니다. 그러면 "IBM WebSphere MQ 준비" 마법사가 자동으로 Domain mqm 그룹을 도메인에 있는 새 IBM WebSphere MQ의 각 새 설치에 있는 로컬 mqm 그룹에 추가합니다.
 - Domain mqm 글로벌 그룹이 없는 도메인에 워크스테이션 또는 서버를 설치할 수 있습니다. 이를 위해서는 Domain mqm 그룹과 동일한 특성으로 그룹을 정의해야 합니다. IBM WebSphere MQ가 도메인에 설치될 때마다 로컬 mqm 그룹의 구성원인 해당 그룹 또는 구성원인 사용자를 작성해야 합니다. 여러 그룹에 도메인 사용자를 배치할 수 있습니다. 각 도메인 그룹이 별도로 관리하려는 설치 세트에 해당하는 다중 도메인 그룹을 작성하십시오. 관리하는 설치에 따라 도메인 사용자를 여러 도메인 그룹으로 분할하십시오. 각 도메인 그룹 또는 그룹을 다른 IBM WebSphere MQ 설치의 로컬 mqm 그룹에 추가하십시오. 도메인 그룹에서 특정 로컬 mqm 그룹의 구성원인 도메인 사용자만이 해당 설치를 위한 큐 관리자를 작성하고 관리하고 실행할 수 있습니다.
 - 도메인의 워크스테이션 또는 서버에 IBM WebSphere MQ를 설치할 때 사용자가 지정하는 도메인 사용자는 Domain mqm 그룹의 구성원이거나 Domain mqm 그룹과 동일한 특성을 사용하여 정의한 대체 그룹의 구성원이어야 합니다.
- b) **글로벌**을 **그룹 범위**로 클릭한 상태로 두거나 **범용**으로 변경하십시오. **보안**을 **그룹 유형**으로 클릭한 상태로 두십시오. **확인**을 클릭하십시오.
13. 권한, 허용 그룹 멤버십을 읽기 및 허용 **groupMembershipSAM**읽기 를 Domain mqm 글로벌 그룹의 권한에 추가하십시오.
a) 서버 관리자 조치 막대에서 **보기 > 고급 기능**을 클릭하십시오.
b) 서버 관리자 탐색 트리에서 **사용자**를 클릭하십시오.
c) 사용자 창에서 **도메인 mqm > 특성**을 마우스의 오른쪽 단추로 클릭하십시오.
d) **보안 > 고급 > 추가 ...**를 클릭하십시오. Domain mqm 를 입력하고 **이름 확인 > 확인**을 클릭하십시오.
이름 필드에는 Domain mqm (`domain name\Domain mqm`) 문자열이 미리 입력되어 있습니다.
e) **특성**을 클릭하십시오. **적용 대상** 목록에서 목록 맨 아래의 **중속 사용자 오브젝트**를 선택하십시오.
f) **권한** 목록에서 **그룹 멤버십 읽기** 및 **groupMembershipSAM 읽기 허용** 선택란을 선택하고 **확인 > 적용 > 확인 > 확인**을 클릭하십시오.
14. Domain mqm 글로벌 그룹에 둘 이상의 사용자를 추가하십시오.

이 예제에서 한 사용자 `wmquser1`은 IBM WebSphere MQ 서비스를 실행하고 다른 사용자인 `wmquser2`는 대화식으로 사용됩니다.

도메인 구성에서 대체 보안 그룹을 사용하는 큐 관리자를 작성하려면 도메인 사용자가 필요합니다. 관리자는 `crtmqm` 명령을 실행할 수 있는 권한이 있지만 사용자 ID가 관리자가 되는 것만으로는 충분하지 않습니다. 관리자가 될 수 있는 도메인 사용자는 로컬 mqm 그룹의 구성원은 물론 대체 보안 그룹의 구성원이어야 합니다.

이 예제에서는 Domain mqm 글로벌 그룹의 `wmquser1` 및 `wmquser2` 구성원이 되어야 합니다. "IBM WebSphere MQ 준비" 마법사는 Domain mqm을 마법사가 실행되는 로컬 mqm 그룹의 멤버로 자동 구성합니다.

단일 컴퓨터에서 각 IBM WebSphere MQ 설치에 대해 IBM WebSphere MQ 서비스를 실행하려면 다른 사용자를 제공해야 합니다. 다른 컴퓨터에서 동일한 사용자를 재사용할 수 있습니다.

- a) 서버 관리자 탐색 트리에서 **사용자 > 새로 작성 > 사용자**를 클릭하십시오.
 - b) 새 오브젝트 - 사용자 창에서 **사용자 로그인 이름** 필드에 *wmquser1*을 입력하십시오. 이름 필드에 *WebSphere*를, 성 필드에 *MQ1*을 입력하십시오. 다음을 클릭하십시오.
 - c) **비밀번호 및 비밀번호 확인** 필드에 비밀번호를 입력하고 다음 로그인 시 사용자가 비밀번호를 변경해야 함 선택란을 지우십시오. 다음 > 마침을 클릭하십시오.
 - d) 사용자 창에서 **WebSphere MQ > 그룹에 추가...**를 마우스의 오른쪽 단추로 클릭하십시오. Domain *mqm*를 입력하고 **이름 확인 > 확인 > 확인**을 클릭하십시오.
 - e) a - d 단계를 반복하여 *WebSphere MQ2*를 *wmquser2*로 추가하십시오.
15. IBM WebSphere MQ를 서비스로 실행하십시오.

IBM WebSphere MQ를 서비스로 실행한 후 도메인 사용자(도메인 관리자로부터 확보한 사용자)에게 서비스로 실행할 액세스 권한을 부여해야 하는 경우에는 다음 프로시저를 수행하십시오.

- a) **시작 > 실행...**을 클릭하십시오.
명령 *secpol.msc*를 입력하고 **확인**을 클릭하십시오.
- b) **보안 설정 > 로컬 정책 > 사용자 권한 지정**을 여십시오.
정책 목록에서 **서비스로서 로그인 > 특성**을 마우스 오른쪽 단추로 클릭하십시오.
- c) **사용자 또는 그룹 추가...**를 클릭하십시오.
도메인 관리자에서 확보한 사용자 이름을 입력하고 **이름 확인**을 클릭하십시오.
- d) Windows 보안 창으로 프롬프트되는 경우, 충분한 권한을 가진 계정 사용자 또는 관리자의 사용자 이름 및 비밀번호를 입력하고 **확인 > 적용 > 확인**을 클릭하십시오.
로컬 보안 정책 창을 닫으십시오.

참고: Windows Vista 및 Windows Server 2008에서 UAC (User Account Control) 는 기본적으로 사용 가능합니다.

UAC 기능은 사용자가 관리자 그룹의 구성원인 경우에도 특정 운영 체제 기능에 대해 사용자가 수행할 수 있는 조치를 제한합니다. 이러한 제한을 극복하려면 적절한 단계를 거쳐야 합니다.

다음에 수행할 작업

다음 322 페이지의 『Windows 도메인의 서버 또는 워크스테이션에 IBM WebSphere MQ 설치』로 이동하십시오.

Windows 도메인의 서버 또는 워크스테이션에 IBM WebSphere MQ 설치

이 태스크에서는 *wmq.example.com* Windows 도메인의 서버 또는 워크스테이션에 IBM WebSphere MQ를 설치하고 구성합니다.

프로덕션 스케일 구성에서는 기존 도메인에 맞게 구성을 조정해야 할 수 있습니다. 예를 들어, 여러 공유에 권한을 부여하고 큐 관리자를 실행하는 사용자 ID를 그룹화할 수 있도록 여러 도메인 그룹을 정의할 수 있습니다.

이 구성 예는 다음 세 개의 서버로 구성됩니다.

sun

Windows Server 2008 도메인 제어기. *Sun*, *mars* 및 *venus*가 들어 있는 *wmq.example.com* 도메인을 소유합니다. 설명을 위해 파일 서버로도 사용됩니다.

mars

첫 번째 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 QMGR이라고 하는 다중 인스턴스 큐 관리자의 한 인스턴스를 포함합니다.

venus

두 번째 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 QMGR이라고 하는 다중 인스턴스 큐 관리자의 두 번째 인스턴스를 포함합니다.

예에서 이탤릭체로 표시된 이름을 선택한 이름으로 바꾸십시오.

시작하기 전에

1. 319 페이지의 『IBM WebSphere MQ 에 대한 Active Directory 및 DNS 도메인 작성』의 단계를 수행하여 도메인 *wmq.example.com*에 대해 도메인 제어기 *sun*을 작성하십시오. 이탤릭체로 표시된 이름을 구성에 맞게 변경하십시오.
2. IBM WebSphere MQ 를 실행할 수 있는 기타 Windows 버전은 Windows 시스템의 하드웨어 및 소프트웨어 요구사항을 참조하십시오.

이 태스크 정보

이 태스크에서는 *wmq.example.com* 도메인의 멤버로 *mars*라는 Windows Server 2008을 구성합니다. IBM WebSphere MQ을(를) 설치하고 *wmq.example.com* 도메인의 멤버로 실행하도록 설치를 구성합니다.

이 태스크는 큐 관리자 데이터 및 로그 파일 액세스를 설명하는 관련 태스크 세트 중 하나입니다. 이 태스크는 사용자가 선택한 디렉토리에 저장된 데이터 및 로그 파일을 읽고 쓸 수 있도록 권한이 부여된 큐 관리자를 작성하는 방법을 보여줍니다. 315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』 태스크를 수반합니다.

이 태스크에서 도메인 제어기 호스트 이름은 *sun*이고, 두 IBM WebSphere MQ 서버는 *mars*와 *venus*라고 합니다. 도메인은 *wmq.example.com*이라고 합니다. 태스크에서 이탤릭체로 표시된 이름은 모두 사용자가 선택한 이름으로 바꿀 수 있습니다.

프로시저

1. 도메인 컨트롤러 *sun.wmq.example.com*을(를) *mars*에 DNS 서버로 추가하십시오.
 - a) *mars*에서 *mars\Administrator* 로 로그인한 후 **시작**을 클릭하십시오.
 - b) **네트워크 > 특성 > 네트워크 연결 관리**를 마우스 오른쪽 단추로 클릭하십시오.
 - c) 네트워크 어댑터에서 마우스의 오른쪽 단추를 클릭하고 **특성**을 클릭하십시오.
시스템이 로컬 영역 연결 특성 창으로 응답하고 연결에 사용되는 항목을 나열합니다.
 - d) 로컬 영역 연결 특성 창의 항목 목록에서 **Internet Protocol 버전 4** 또는 **Internet Protocol 버전 6**을 선택하십시오. **특성 > 고급 ...**을 클릭하십시오. **DNS** 탭을 클릭하십시오.
 - e) DNS 서버 주소 아래에서 **추가...**를 클릭하십시오.
 - f) DNS 서버이기도 한 도메인 제어기의 IP 주소를 입력하고 **추가**를 클릭하십시오.
 - g) **DNS 접미부 추가 > 추가...**를 클릭하십시오.
 - h) *wmq.example.com*을(를) 입력하고 **추가**를 클릭하십시오.
 - i) 이 연결에 대한 **DNS 접미어** 필드에 *wmq.example.com*을(를) 입력하십시오.
 - j) **DNS에 이 연결의 주소 등록 및 DNS 등록에 이 연결의 접미부 사용**을 선택하십시오. **확인 > 확인 > 닫기**를 누르십시오.
 - k) 명령 창을 열고 **ipconfig /all** 명령을 입력하여 TCP/IP 설정을 검토하십시오.
2. *mars*에서 *wmq.example.com* 도메인에 컴퓨터를 추가하십시오.
 - a) **시작**을 클릭하십시오.
 - b) **컴퓨터 > 특성**을 마우스 오른쪽 단추로 클릭하십시오. 컴퓨터 이름, 도메인 및 작업 그룹 설정 부분에서 **설정 변경**을 클릭하십시오.
 - c) 시스템 특성 창에서 **변경...**을 클릭하십시오.
 - d) 도메인을 클릭하고 *wmq.example.com*을(를) 입력한 후 **확인**을 클릭하십시오.
 - e) 컴퓨터가 도메인에 조인하도록 허용하는 권한을 가진 도메인 제어기 관리자의 **사용자 이름** 및 **비밀번호**를 입력하고 **확인**을 클릭하십시오.
 - f) "*wmq.example.com* 도메인 시작" 메시지에 대한 응답으로 **확인 > 확인 > 닫기 > 지금 다시 시작**을 클릭하십시오.
3. 컴퓨터가 *wmq.example.com* 도메인의 멤버인지 확인하십시오.
 - a) *sun*에서 도메인 컨트롤러에 *wmq\Administrator(으)*로 로그인하십시오.

- b) 서버 관리자 > **Active Directory** 도메인 서비스 > *wmq.example.com* > 컴퓨터 를 열고 *mars* 가 컴퓨터 창에 올바르게 나열되어 있는지 확인하십시오.
4. *mars*에 IBM WebSphere MQ for Windows을 설치하십시오.
- IBM WebSphere MQ for Windows 설치 마법사 실행에 대한 자세한 정보는 [Windows 에 IBM WebSphere MQ 서버 설치](#) 를 참조하십시오.
- a) *mars*에서 로컬 관리자(*mars\Administrator*)로 로그인하십시오.
- b) IBM WebSphere MQ for Windows 설치 매체에서 **Setup** 명령을 실행하십시오.
- IBM WebSphere MQ 런치패드 애플리케이션이 시작합니다.
- c) **소프트웨어 요구사항**을 클릭하여 필수 소프트웨어가 설치되는지 확인하십시오.
- d) **네트워크 구성 > 예**를 클릭하여 도메인 사용자 ID를 구성하십시오.
- 319 페이지의 『IBM WebSphere MQ 에 대한 Active Directory 및 DNS 도메인 작성』 태스크에서는 이 태스크 세트에 대한 도메인 ID를 구성합니다.
- e) **WebSphere MQ 설치**를 클릭하고 설치 언어를 선택한 후 IBM WebSphere MQ 설치 프로그램을 시작을 클릭하십시오.
- f) 라이선스 계약을 확인하고 **다음 > 다음 > 설치**를 클릭하여 기본 구성을 승인하십시오. 설치가 완료될 때까지 대기한 후 **마침**을 클릭하십시오.
- 설치 이름을 변경하거나, 다른 컴포넌트를 설치하거나, 큐 관리자 데이터 및 로그를 위한 다른 디렉토리를 구성하거나, 다른 디렉토리에 설치하려고 할 수 있습니다. 이 경우 **일반 설치**가 아닌 **사용자 설치**를 클릭하십시오.
- IBM WebSphere MQ가 설치되었으며 설치 프로그램이 "IBM WebSphere MQ 준비" 마법사를 시작합니다.
- 중요사항:** 아직 마법사를 실행하지 마십시오.
5. **서비스로 실행** 권한으로 IBM WebSphere MQ 서비스를 실행할 사용자를 구성하십시오.
- 로컬 *mqm* 그룹, Domain *mqm* 그룹 또는 IBM WebSphere MQ 서비스를 실행할 사용자를 올바르게 구성할지 여부를 선택하십시오. 이 예에서는 사용자에게 권한을 제공합니다.
- a) **시작 > 실행 ...**을 클릭하십시오. **secpol.msc** 명령을 입력하고 **확인**을 클릭하십시오.
- b) **보안 설정 > 로컬 정책 > 사용자 권한 지정**을 여십시오. 정책 목록에서 **서비스로 로그인 > 특성**을 마우스 오른쪽 단추로 클릭하십시오.
- c) **사용자 또는 그룹 추가...**를 클릭하십시오. 및 *wmquser1* 를 입력하고 **이름 확인** 을 클릭하십시오.
- d) 도메인 관리자의 사용자 이름과 비밀번호를 입력하고 *wmq\Administrator*를 클릭하고 **확인 > 적용 > 확인**을 클릭하십시오. 로컬 보안 정책 창을 닫으십시오.
6. "IBM WebSphere MQ 준비" 마법사를 실행하십시오.
- "IBM WebSphere MQ 준비" 마법사 실행에 대한 자세한 정보는 [준비 WebSphere MQ 마법사를 사용하여 WebSphere MQ 구성](#) 의 내용을 참조하십시오.
- a) IBM WebSphere MQ 설치 프로그램이 "IBM WebSphere MQ 준비"를 자동으로 실행합니다.
- 마법사를 수동으로 시작하려면 **시작 > 모든 프로그램 > IBM WebSphere MQ** 폴더에서 "IBM WebSphere MQ 준비" 의 바로 가기를 찾으십시오. 다중 설치 구성에서 IBM WebSphere MQ의 설치에 해당하는 바로 가기를 선택하십시오.
- b) **다음**을 클릭하고 "네트워크에 Windows 2000 이상의 도메인 제어기가 있는지 식별"이라는 질문에 대한 응답으로 **예**를 클릭한 상태로 두십시오.
- c) 첫 번째 Windows 도메인 사용자를 위한 IBM WebSphere MQ for Windows 구성 창에서 **예 > 다음**을 클릭하십시오.
- d) 두 번째 Windows 도메인 사용자를 위한 IBM WebSphere MQ for Windows 구성 창의 **도메인** 필드에 *wmq*를 입력하십시오. **사용자 이름** 필드에 *wmquser1*을 입력하고 **비밀번호** 필드에 비밀번호(설정한 경우)를 입력하십시오. **다음**을 클릭하십시오.
- 마법사는 *wmquser1*을 사용하여 IBM WebSphere MQ 를 구성하고 시작합니다.

e) 마지막 마법사 페이지에서 필요에 따라 선택란을 선택하거나 선택 취소하고 **마침**을 클릭하십시오.

다음에 수행할 작업

1. 설치 및 구성이 올바르게 작동하는지 확인하려면 344 페이지의 『로컬 mqm 그룹에서 권한 부여된 데이터 및 로그 파일 읽기 및 쓰기』 태스크를 수행하십시오.
2. 다중 인스턴스 큐 관리자의 로그 파일 및 데이터를 저장하도록 파일 공유를 구성하려면 325 페이지의 『큐 관리자 데이터 및 로그 파일을 위한 공유 디렉토리 작성』 태스크를 수행하십시오.

관련 개념

WebSphere MQ Windows 서비스에 필요한 사용자 권한

큐 관리자 데이터 및 로그 파일을 위한 공유 디렉토리 작성

이 태스크는 큐 관리자 데이터 및 로그 파일 액세스를 설명하는 관련 태스크 세트 중 하나입니다. 이 태스크는 사용자가 선택한 디렉토리에 저장된 데이터 및 로그 파일을 읽고 쓸 수 있도록 권한이 부여된 큐 관리자를 작성하는 방법을 보여줍니다.

프로덕션 스케일 구성에서는 기존 도메인에 맞게 구성을 조정해야 할 수 있습니다. 예를 들어, 여러 공유에 권한을 부여하고 큐 관리자를 실행하는 사용자 ID를 그룹화할 수 있도록 여러 도메인 그룹을 정의할 수 있습니다.

이 구성 예는 다음 세 개의 서버로 구성됩니다.

sun

Windows Server 2008 도메인 제어기. *Sun, mars* 및 *venus*가 들어 있는 *wmq.example.com* 도메인을 소유합니다. 설명을 위해 파일 서버로도 사용됩니다.

mars

첫 번째 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 *QMGR*이라고 하는 다중 인스턴스 큐 관리자의 한 인스턴스를 포함합니다.

venus

두 번째 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 *QMGR*이라고 하는 다중 인스턴스 큐 관리자의 두 번째 인스턴스를 포함합니다.

예에서 이탤릭체로 표시된 이름을 선택한 이름으로 바꾸십시오.

시작하기 전에

1. 문서에 설명된 그대로 이 태스크를 수행하려면 319 페이지의 『IBM WebSphere MQ에 대한 Active Directory 및 DNS 도메인 작성』 태스크의 단계를 수행하여 도메인 제어기 *sun*에서 도메인 *sun.wmq.example.com*을 작성합니다. 이탤릭체로 표시된 이름을 구성에 맞게 변경하십시오.

이 태스크 정보

이 태스크는 큐 관리자 데이터 및 로그 파일 액세스를 설명하는 관련 태스크 세트 중 하나입니다. 이 태스크는 사용자가 선택한 디렉토리에 저장된 데이터 및 로그 파일을 읽고 쓸 수 있도록 권한이 부여된 큐 관리자를 작성하는 방법을 보여줍니다. 315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』 태스크를 수반합니다.

이 태스크에서 데이터 및 로그 디렉토리를 포함하는 공유 및 이 공유에 액세스 권한을 부여하기 위한 글로벌 그룹을 작성합니다. 공유에 권한을 부여하는 글로벌 그룹의 이름을 해당 -a 매개변수로 *crtmqm* 명령에 전달합니다. 글로벌 그룹은 이 공유의 사용자를 다른 공유 사용자로부터 분리할 수 있는 유연성을 제공합니다. 이 유연성이 필요하지 않은 경우 새 글로벌 그룹을 작성하는 대신, *Domain mqm* 그룹으로 공유에 대한 권한을 부여합니다.

이 태스크를 공유하는 데 사용되는 글로벌 그룹은 *wmqha*라고 하며, 공유는 *wmq*라고 합니다. 이들은 Windows 도메인 *wmq.example.com*의 도메인 제어기 *sun*에서 정의됩니다. 공유에는 글로벌 그룹 *wmqha*에 대한 전체 제어 권한이 있습니다. 태스크에서 이탤릭체로 표시된 이름을 선택한 이름으로 바꾸십시오.

이 태스크에서는 도메인 제어기가 파일 서버와 같은 서버입니다. 실제 애플리케이션에서는 성능 및 가용성을 위해 서로 다른 서버 사이에서 디렉토리 및 파일 서비스를 분리하십시오.

큐 관리자 실행에 사용되는 사용자 ID를 두 그룹의 멤버가 되도록 구성해야 합니다. 로컬 *mqm* 그룹의 멤버(IBM WebSphere MQ 서버의 경우) 및 *wmqha* 글로벌 그룹의 멤버여야 합니다.

이 태스크 세트에서 큐 관리자가 서비스로 실행되면 사용자 ID *wmquser1*로 실행됩니다. 따라서 *wmquser1*은 *wmqha*의 멤버여야 합니다. 큐 관리자가 대화식으로 실행되는 경우 사용자 ID *wmquser2*로 실행됩니다. 따라서 *wmquser2*는 *wmqha*의 멤버여야 합니다. *wmquser1* 및 *wmquser2*는 모두 글로벌 그룹 Domain *mqm*의 멤버입니다. Domain *mqm* 은 *mars* 및 *venus* IBM WebSphere MQ 서버의 로컬 *mqm* 그룹의 구성원입니다. 따라서 *wmquser1* 및 *wmquser2*는 두 IBM WebSphere MQ 서버에서 로컬 *mqm* 그룹의 멤버입니다.

프로시저

1. 도메인 제어기 *sun.wmq.example.com*에 도메인 관리자로 로그인하십시오.
2. 글로벌 그룹 *wmqha*를 작성하십시오.
 - a) 서버 관리자 > 역할 > **Active Directory 도메인 서비스** > *wmq.example.com* > 사용자를 여십시오.
 - b) *wmq.example.com\Users* 폴더를 여십시오.
 - c) 사용자 > 새로 작성 > 그룹을 마우스 오른쪽 단추로 클릭하십시오.
 - d) 그룹 이름 필드에 *wmqha*를 입력하십시오.
 - e) 그룹 범위로 글로벌을 클릭된 상태로 두고 그룹 유형으로 보안을 클릭된 상태로 두십시오. 확인을 클릭하십시오.
3. 도메인 사용자 *wmquser1* 및 *wmquser2*를 글로벌 그룹 *wmqha*에 추가하십시오.
 - a) 서버 관리자 탐색 트리에서 사용자를 클릭한 후 사용자 목록에서 *wmqha* > 특성을 마우스 오른쪽 단추로 클릭하십시오.
 - b) *wmqha* 특성 창에서 멤버 탭을 클릭하십시오.
 - c) 추가 ...를 클릭하십시오. *wmquser1*; *wmquser2* 를 입력하고 이름 확인 > 확인 > 적용 > 확인을 클릭하십시오.
4. 큐 관리자 데이터 및 로그 파일이 포함될 디렉토리 트리를 작성하십시오.
 - a) 명령 프롬프트를 여십시오.
 - b) 명령을 입력하십시오.

```
md c:\wmq\data , c:\wmq\logs
```

5. *c:\wmq* 디렉토리 및 공유에 대한 전체 제어 권한을 보유하도록 글로벌 그룹 *wmqha*에 권한을 부여하십시오.
 - a) Windows 탐색기의 *c:\wmq* > 특성에서 마우스의 오른쪽 단추를 클릭하십시오.
 - b) 보안 탭을 클릭하고 고급 > 편집 ...을 클릭하십시오.
 - c) 이 오브젝트의 소유자로부터 상속 가능한 권한을 포함함 선택란을 지우십시오. Windows 보안 창에서 복사를 클릭하십시오.
 - d) 권한 항목 목록에서 사용자에게 대한 행을 선택하고 제거를 클릭하십시오. 권한 항목의 목록에서 SYSTEM, 관리자 및 CREATOR OWNER에 대한 행은 그대로 두십시오.
 - e) 추가 ...를 클릭하십시오. 글로벌 그룹 *wmqha*의 이름을 입력하십시오. 이름 확인 > 확인을 클릭하십시오.
 - f) *wmq* 창에 있는 권한 항목의 권한 목록에서 전체 제어를 선택하십시오.
 - g) 확인 > 적용 > 확인 > 확인 > 확인 을 클릭하십시오.
 - h) Windows 탐색기의 *c:\wmq* > 공유...에서 마우스의 오른쪽 단추를 클릭하십시오.
 - i) 고급 공유 ... 를 클릭하십시오. 이 폴더 공유 선택란을 선택하십시오. 공유 이름은 *wmq*로 두십시오.
 - j) 권한 > 추가 ...를 클릭하십시오. 글로벌 그룹 *wmqha*의 이름을 입력하십시오. 이름 확인 > 확인을 클릭하십시오.
 - k) 그룹 또는 사용자 이름 목록에서 *wmqha*를 선택하십시오. *wmqha* 에 대한 권한목록에서 전체 제어 선택란을 선택하고 적용을 클릭하십시오.
 - l) 그룹 또는 사용자 이름 목록에서 *Administrators*를 선택하십시오. 관리자 에 대한 권한목록에서 전체 제어 선택란을 선택하고 적용 > 알았다 > 알았다 > 닫기를 클릭하십시오.

다음에 수행할 작업

각 IBM WebSphere MQ 서버로부터 공유 디렉토리에 파일을 작성하고 읽을 수 있는지 확인하십시오. IBM WebSphere MQ 서비스 사용자 ID, *wmquser1* 및 대화식 사용자 ID, *wmquser2*를 확인하십시오.

- 원격 데스크탑을 사용하는 경우 *wmq\wmquser1* 및 *wmquser2*를 *mars*의 로컬 그룹 Remote Desktop Users에 추가해야 합니다.
 - mars*에 *wmq\Administrator*로서 로그인하십시오.
 - lusrmgr.msc** 명령을 실행하여 로컬 사용자 및 그룹 창을 여십시오.
 - 그룹을 클릭하십시오. 원격 데스크탑 사용자 > 특성 > 추가 ...를 마우스 오른쪽 단추로 클릭하십시오. *wmquser1*; *wmquser2*를 입력하고 이름 검사를 클릭하십시오.
 - 도메인 관리자의 사용자 이름과 비밀번호를 입력하고 *wmq\Administrator*를 입력하고 확인 > 적용 > 확인을 클릭하십시오.
 - 로컬 사용자 및 그룹 창을 닫으십시오.
- wmq\wmquser1*으로 *mars*에 로그인하십시오.
 - Windows 탐색기 창을 열고 `\\sun\wmq`에 입력하십시오.
시스템은 *sun.wmq.example.com*에서 *wmq* 공유를 열어 응답하고 데이터 및 로그 디렉토리를 나열합니다.
 - 데이터 서브디렉토리에서 파일을 작성하고, 일부 콘텐츠를 추가하고, 읽은 다음 삭제하여 *wmquser1*의 권한을 검사하십시오.
- wmq\wmquser2*로 *mars*에 로그인하고 검사를 반복하십시오.
- 공유 데이터 및 로그 디렉토리를 사용하기 위한 큐 관리자를 작성하려면 다음 태스크를 수행하십시오. 327 페이지의 『[대체 글로벌 보안 그룹이 권한을 부여한 공유 데이터 및 로그 파일 읽기 및 쓰기](#)』의 내용을 참조하십시오.

대체 글로벌 보안 그룹이 권한을 부여한 공유 데이터 및 로그 파일 읽기 및 쓰기

이 태스크에서는 **crtmqm** 명령에서 -a 플래그를 사용하는 방법을 보여줍니다. -a 플래그는 대체 보안 그룹을 사용하여 원격 파일 공유에 있는 해당 로그 및 데이터 파일에 대한 액세스 권한을 큐 관리자에게 제공합니다.

프로덕션 스케일 구성에서는 기존 도메인에 맞게 구성을 조정해야 할 수 있습니다. 예를 들어, 여러 공유에 권한을 부여하고 큐 관리자를 실행하는 사용자 ID를 그룹화할 수 있도록 여러 도메인 그룹을 정의할 수 있습니다.

이 구성 예는 다음 세 개의 서버로 구성됩니다.

sun

Windows Server 2008 도메인 제어기. *Sun*, *mars* 및 *venus*가 들어 있는 *wmq.example.com* 도메인을 소유합니다. 설명을 위해 파일 서버로도 사용됩니다.

mars

첫 번째 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 *QMGR*이라고 하는 다중 인스턴스 큐 관리자의 한 인스턴스를 포함합니다.

venus

두 번째 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 *QMGR*이라고 하는 다중 인스턴스 큐 관리자의 두 번째 인스턴스를 포함합니다.

예에서 이탤릭체로 표시된 이름을 선택한 이름으로 바꾸십시오.

시작하기 전에

다음 태스크에 있는 단계를 수행하십시오. 이 태스크에서는 도메인 제어기 및 도메인을 작성하고 서버에 IBM WebSphere MQ for Windows를 설치하며 데이터 및 로그 파일에 대한 파일 공유를 작성합니다. 기존 도메인 제어기를 구성하면 새 Windows Server 2008에서 단계를 시도하는 데 유용할 수 있습니다. 사용자 도메인에 단계를 적용시킬 수 있습니다.

- 319 페이지의 『IBM WebSphere MQ 에 대한 Active Directory 및 DNS 도메인 작성』.
- 322 페이지의 『Windows 도메인의 서버 또는 워크스테이션에 IBM WebSphere MQ 설치』.

3. 325 페이지의 『큐 관리자 데이터 및 로그 파일을 위한 공유 디렉토리 작성』.

이 태스크 정보

이 태스크는 큐 관리자 데이터 및 로그 파일 액세스를 설명하는 관련 태스크 세트 중 하나입니다. 이 태스크는 사용자가 선택한 디렉토리에 저장된 데이터 및 로그 파일을 읽고 쓸 수 있도록 권한이 부여된 큐 관리자를 작성하는 방법을 보여줍니다. 315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』 태스크를 수반합니다.

이 태스크에서는 이 태스크에서는 파일 서버의 원격 디렉토리에 해당 데이터 및 로그를 저장하는 큐 관리자를 작성합니다. 이 예에서는 파일 서버가 도메인 제어기와 동일한 서버입니다. 데이터 및 로그 폴더가 들어 있는 디렉토리는 글로벌 그룹 `wmqha`에 부여된 전체 제어 권한과 공유됩니다.

프로시저

1. 로컬 관리자(`mars\Administrator`)로 도메인 서버(`mars`)에 로그인하십시오.
2. 명령 창을 여십시오.
3. IBM IBM WebSphere MQ 서비스를 재시작하십시오.

서비스를 실행하는 사용자 ID가 그에 대해 구성한 추가 보안 신임 정보를 획득할 수 있도록 서비스를 재시작해야 합니다.

다음 명령을 입력하십시오.

```
endmqsvc  
strmqsvc
```

시스템이 다음과 같이 응답합니다.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

및

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' started successfully.
```

4. 큐 관리자를 작성하십시오.

```
crtmqm -a wmq\wmqha -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\sun\wmq\data -ld \\sun\wmq\logs  
QMGR
```

글로벌 그룹 "`wmq\wmqha`"의 전체 도메인 이름을 지정하여 대체 보안 그룹 `wmqha`의 도메인 `wmq`를 지정해야 합니다.

공유 `\\sun\wmq`의 UNC(Universal Naming Convention)를 전체 철자 양식으로 하고 맵핑된 드라이브 참조를 사용해서는 안 됩니다.

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager created.  
Directory '\\sun\wmq\data\QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
Default objects statistics : 74 created. 0 replaced.  
Completing setup.  
Setup completed.
```

다음에 수행할 작업

메시지를 큐에 넣거나 가져와서 큐 관리자를 테스트하십시오.

1. 큐 관리자를 시작하십시오.

```
strmqm QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. 테스트 큐를 작성하십시오.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

시스템이 다음과 같이 응답합니다.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: WebSphere MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. 샘플 프로그램 **amqsput**을 사용하여 테스트 메시지를 넣으십시오.

```
echo 'A test message' | amqsput QTEST QMGR
```

시스템이 다음과 같이 응답합니다.

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. 샘플 프로그램 **amqsget**을 사용하여 테스트 메시지를 가져오십시오.

```
amqsget QTEST QMGR
```

시스템이 다음과 같이 응답합니다.

```
Sample AMQSGET0 start  
message <A test message>  
Wait 15 seconds ...  
no more messages  
Sample AMQSGET0 end
```

5. 큐 관리자를 중지합니다.

```
endmqm -i QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere MQ queue manager 'QMGR' ended.
```

6. 큐 관리자를 삭제합니다.

```
dltmqm QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' deleted.
```

7. 작성한 디렉토리를 삭제하십시오.

팁: 명령이 각 파일 또는 디렉토리를 삭제하도록 프롬프트하지 않도록 명령에 /Q 옵션을 추가하십시오.

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

도메인 제어기에 다중 인스턴스 큐 관리자 작성

이 예에서는 도메인 제어기의 Windows에서 다중 인스턴스 큐 관리자를 설정하는 방법을 보여줍니다. 이 설정에서는 프로덕션 스케일이 아닌 관련된 개념을 보여줍니다. 이 예제는 Windows Server 2008을 기반으로 합니다. 다른 버전의 Windows Server에서는 단계가 다를 수 있습니다.

구성은 미니 도메인 또는 "domainlet"의 개념을 사용합니다. 도메인 제어기 역할의 Windows 2000, Windows Server 2003 및 Windows Server 2008 클러스터 노트를 참조하십시오. 기존 도메인에 다중 인스턴스 큐 관리자를 추가하려면 316 페이지의 『도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성』의 내용을 참조하십시오.

이 구성 예는 다음 세 개의 서버로 구성됩니다.

sun

첫 번째 도메인 제어기로 사용되는 Windows Server 2008 서버입니다. 이는 *sun*, *earth* 및 *mars*를 포함하는 *wmq.example.com* 도메인을 정의합니다. 이 서버는 QMGR이라고 하는 다중 인스턴스 큐 관리자의 한 인스턴스를 포함합니다.

earth

두 번째 도메인 컨트롤러 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 QMGR이라고 하는 다중 인스턴스 큐 관리자의 두 번째 인스턴스를 포함합니다.

mars

파일 서버로 사용되는 Windows Server 2008입니다.

예에서 이탤릭체로 표시된 이름을 선택한 이름으로 바꾸십시오.

시작하기 전에

1. Windows에서는 큐 관리자 데이터와 로그 파일을 저장하려는 파일 시스템을 확인할 필요가 없습니다. 공유 파일 시스템 작동 확인 검사 프로시저는 UNIX and Linux에 적용할 수 있습니다. Windows에서는 검사가 항상 성공적입니다.
2. 319 페이지의 『IBM WebSphere MQ 에 대한 Active Directory 및 DNS 도메인 작성』의 단계를 수행하여 첫 번째 도메인 제어기를 작성하십시오.
3. 333 페이지의 『wmq.example.com 도메인에 두 번째 도메인 제어기 추가』의 단계를 수행하여 두 번째 도메인 제어기를 추가하고 두 도메인 제어기 모두에 IBM WebSphere MQ for Windows를 설치하고 설치를 확인하십시오.
4. 335 페이지의 『wmq.example.com 도메인에 도메인 컨트롤러에 IBM WebSphere MQ 설치』의 단계를 수행하여 두 도메인 제어기에 IBM WebSphere MQ를 설치하십시오.

이 태스크 정보

같은 도메인의 파일 서버에서 큐 관리자 로그 및 데이터 디렉토리에 대한 공유를 작성하십시오. 그 다음에는 도메인 제어기 중 하나에서 파일 공유를 사용하는 다중 인스턴스 큐 관리자의 첫 번째 인스턴스를 작성하십시오. 다른 도메인 제어기에서 다른 인스턴스를 작성하고 마지막으로 구성을 확인하십시오. 도메인 제어기에서 파일 공유를 작성할 수 있습니다.

샘플에서 *sun*은 첫 번째 도메인 제어기이고 *earth*는 두 번째 도메인 제어기이고 *mars*는 파일 서버입니다.

프로시저

1. 큐 관리자 데이터 및 로그 파일을 포함할 디렉토리를 작성하십시오.
 - a) *mars*에서 다음 명령을 입력하십시오.

```
md c:\wmq\data , c:\wmq\logs
```

2. 큐 관리자 데이터 및 로그 파일을 포함할 디렉토리를 공유하십시오.

큐 관리자를 작성하는 데 사용하는 사용자 ID 및 도메인 로컬 그룹 *wmq*에 대한 전체 제어 액세스를 허용해야 합니다. 이 예에서 Domain Administrators의 멤버인 사용자 ID에는 큐 관리자를 작성하기 위한 권한이 있습니다.

파일 공유는 도메인 제어기와 같은 도메인에 있는 서버에 있어야 합니다. 이 예에서 서버 *mars*는 도메인 제어기와 같은 도메인에 있습니다.

- a) Windows 탐색기의 **c:\wmq > 특성**에서 마우스의 오른쪽 단추를 클릭하십시오.
- b) **보안** 탭을 클릭하고 **고급 > 편집 ...**을 클릭하십시오.
- c) 이 오브젝트의 소유자로부터 상속 가능한 권한을 포함함 선택란을 지우십시오. Windows 보안 창에서 **복사**를 클릭하십시오.
- d) **권한 항목** 목록에서 사용자에게 대한 행을 선택하고 **제거**를 클릭하십시오. **권한 항목**의 목록에서 SYSTEM, 관리자 및 CREATOR OWNER에 대한 행은 그대로 두십시오.
- e) **추가 ...**를 클릭하십시오. 도메인 로컬 그룹 *wmq*의 이름을 입력하십시오. **이름 확인**을 클릭하십시오.
- f) Windows 보안 창에 대한 응답으로 Domain Administrator의 이름 및 비밀번호를 입력하고 **확인 > 확인**을 클릭하십시오.
- g) *wmq* 창에 있는 권한 항목의 **권한** 목록에서 **전체 제어**를 선택하십시오.
- h) **확인 > 적용 > 확인 > 확인 > 확인** 을 클릭하십시오.
- i) **e - h** 단계를 반복하여 Domain Administrators를 추가하십시오.
- j) Windows 탐색기의 **c:\wmq > 공유...**에서 마우스의 오른쪽 단추를 클릭하십시오.
- k) **고급 공유 ...** 를 클릭하십시오. 이 **폴더 공유** 선택란을 선택하십시오. 공유 이름은 *wmq*로 두십시오.
- l) **권한 > 추가 ...**를 클릭하십시오. 도메인 로컬 그룹 *wmq*; Domain Administrators의 이름을 입력하십시오. **이름 확인**을 클릭하십시오.
- m) Windows 보안 창에 대한 응답으로 Domain Administrator의 이름 및 비밀번호를 입력하고 **확인 > 확인**을 클릭하십시오.

3. 첫 번째 도메인 제어기 *sun*에서 큐 관리자 *QMGR*를 작성하십시오.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\mars\wmq\data -ld \\mars\wmq\logs QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager created.  
Directory '\\mars\wmq\data\QMGR' created.  
The queue manager is associated with installation 'Installation1'.  
Creating or replacing default objects for queue manager 'QMGR'.  
Default objects statistics : 74 created. 0 replaced. 0 failed.  
Completing setup.  
Setup completed.
```

4. 대기 인스턴스를 허용하며, *sun*에서 큐 관리자를 시작하십시오.

```
strmqm -x QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation 'Installation1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

5. *earth*에서 *QMGR*의 두 번째 인스턴스를 작성하십시오.

- a) Prefix 및 InstallationName 매개변수의 값이 *earth*에 대해 올바른지 확인하십시오.

*sun*에서 **dspmqinf** 명령을 실행하십시오.

```
dspmqinf QMGR
```

시스템이 다음과 같이 응답합니다.

```
QueueManager:
  Name=QMGR
  Directory=QMGR
  Prefix=C:\Program Files\IBM\WebSphere MQ
  DataPath=\\mars\wmq\data\QMGR
  InstallationName=Installation1
```

b) **QueueManager** 스탠자의 기계 판독 가능(machine-readable) 양식을 클립보드에 복사하십시오.

sun 에서 -o command 매개변수를 사용하여 **dspmqinf** 명령을 다시 실행하십시오.

```
dspmqinf -o command QMGR
```

시스템이 다음과 같이 응답합니다.

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

c) *earth*에서 클립보드로부터 **addmqinf** 명령을 실행하여 *earth*에서 큐 관리자의 인스턴스를 작성하십시오.

Prefix 또는 InstallationName 매개변수의 차이를 수용할 수 있도록 필요한 경우 명령을 조정하십시오.

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

WebSphere MQ configuration information added.

6. *earth*에서 큐 관리자의 대기 인스턴스를 시작하십시오.

```
strmqm -x QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started. The active
instance is running elsewhere.
```

결과

*sun*에서 *earth*로 큐 관리자가 전환되었는지 확인하십시오.

1. *sun*에서 다음 명령을 실행하십시오.

```
endmqm -i -r -s QMGR
```

시스템이 *sun*에 대해 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' ending.
```

WebSphere MQ queue manager 'QMGR' ended, permitting switchover to a standby instance.

2. *earth*에서 다음 명령을 반복적으로 입력하십시오.

```
dspmqr
```

시스템이 다음과 같이 응답합니다.

```
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running)
```

다음에 수행할 작업

샘플 프로그램을 사용하여 다중 인스턴스 큐 관리자를 확인하려면 337 페이지의 『Windows에서 멀티 인스턴스 큐 관리자 확인』의 내용을 참조하십시오.

관련 태스크

333 페이지의 『wmq.example.com 도메인에 두 번째 도메인 제어기 추가』

335 페이지의 『wmq.example.com 도메인에 도메인 컨트롤러에 IBM WebSphere MQ 설치』

관련 정보

도메인 제어기 역할의 Windows 2000, Windows Server 2003 및 Windows Server 2008 클러스터 노트

wmq.example.com 도메인에 두 번째 도메인 제어기 추가

두 번째 도메인 제어기를 *wmq.example.com* 도메인에 추가하여 도메인 제어기 및 파일 서버에서 다중 인스턴스 큐 관리자를 실행하기 위한 Windows 도메인을 구성합니다.

이 구성 예는 다음 세 개의 서버로 구성됩니다.

sun

첫 번째 도메인 제어기로 사용되는 Windows Server 2008 서버입니다. 이는 *sun*, *earth* 및 *mars*를 포함하는 *wmq.example.com* 도메인을 정의합니다. 이 서버는 QMGR이라고 하는 다중 인스턴스 큐 관리자의 한 인스턴스를 포함합니다.

earth

두 번째 도메인 컨트롤러 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 QMGR이라고 하는 다중 인스턴스 큐 관리자의 두 번째 인스턴스를 포함합니다.

mars

파일 서버로 사용되는 Windows Server 2008입니다.

예에서 이탤릭체로 표시된 이름을 선택한 이름으로 바꾸십시오.

시작하기 전에

- 319 페이지의 『IBM WebSphere MQ에 대한 Active Directory 및 DNS 도메인 작성』의 단계를 수행하여 도메인 *wmq.example.com*에 대해 도메인 제어기 *sun*을 작성하십시오. 이탤릭체로 표시된 이름을 구성에 맞게 변경하십시오.
- Windows Server 2008을 서버의 기본 작업 그룹, WORKGROUP에 설치합니다. 예를 들어, 서버의 이름은 *earth*로 지정됩니다.

이 태스크 정보

이 태스크에서는 Windows Server 2008(*earth*라고 함)을 *wmq.example.com* 도메인에서 두 번째 도메인 제어기로 구성합니다.

이 태스크는 큐 관리자 데이터 및 로그 파일 액세스를 설명하는 관련 태스크 세트 중 하나입니다. 이 태스크는 사용자가 선택한 디렉토리에 저장된 데이터 및 로그 파일을 읽고 쓸 수 있도록 권한이 부여된 큐 관리자를 작성하는 방법을 보여줍니다. 315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』 태스크를 수반합니다.

프로시저

- 도메인 제어기 *sun.wmq.example.com*을 *earth*에 DNS 서버로 추가하십시오.
 - earth*에서 *earth\Administrator* 로 로그인한 후 **시작**을 클릭하십시오.
 - 네트워크 > 특성 > 네트워크 연결 관리**를 마우스 오른쪽 단추로 클릭하십시오.
 - 네트워크 어댑터에서 마우스의 오른쪽 단추를 클릭하고 **특성**을 클릭하십시오.
시스템이 로컬 영역 연결 특성 창으로 응답하고 연결에 사용되는 항목을 나열합니다.
 - 로컬 영역 연결 특성 창의 항목 목록에서 **Internet Protocol 버전 4** 또는 **Internet Protocol 버전 6**을 선택하십시오. **특성 > 고급 ...**을 클릭하십시오. **DNS** 탭을 클릭하십시오.
 - DNS 서버 주소 아래에서 **추가...**를 클릭하십시오.
 - DNS 서버이기도 한 도메인 제어기의 IP 주소를 입력하고 **추가**를 클릭하십시오.
 - DNS 접미부 추가 > 추가...**를 클릭하십시오.
 - wmq.example.com*을(를) 입력하고 **추가**를 클릭하십시오.
 - 이 연결에 대한 **DNS 접미어** 필드에 *wmq.example.com*을(를) 입력하십시오.
 - DNS에 이 연결의 주소 등록 및 DNS 등록에 이 연결의 접미부 사용**을 선택하십시오. **확인 > 확인 > 닫기**를 누르십시오.
 - 명령 창을 열고 **ipconfig /all** 명령을 입력하여 TCP/IP 설정을 검토하십시오.
- 로컬 또는 Workgroup 관리자로 도메인 컨트롤러 (*sun*)에 로그인하십시오.
서버가 도메인 제어기로 이미 구성되어 있는 경우 도메인 관리자로 로그인해야 합니다.
- Active Directory 도메인 서비스 마법사를 실행하십시오.
 - 시작 > 실행 ...**을 클릭하십시오. *dcpromo*를 입력하고 **확인**을 클릭하십시오.
Active Directory 2진 파일이 아직 설치되지 않은 경우 Windows에서 파일을 자동으로 설치합니다.
- earth*를 *wmq.example.com* 도메인의 두 번째 도메인 제어기로 구성하십시오.
 - 마법사의 첫 번째 창에서 **고급 모드 설치 사용** 선택란을 지워진 상태로 두십시오. **다음 > 다음**을 클릭하고 **기존 도메인에 도메인 제어기 추가 > 다음**를 클릭하십시오.
 - 이 포리스트에 있는 도메인의 이름 입력 ...에 *wmq*를 입력하십시오. 필드. **대체 신임 정보** 단일 선택 단추가 클릭되고 **설정...**을 클릭하십시오. 도메인 관리자의 이름 및 암호를 입력하고 **확인 > 다음 > 다음 > 다음**을 누르십시오.
 - 추가 도메인 컨트롤러 옵션 창에서 선택한 **DNS 서버 및 글로벌 카탈로그** 옵션을 승인하십시오. **다음 > 다음**을 클릭하십시오.
 - Directory Services 복원 모드 관리자 암호에서 **암호 및 암호 확인**을 입력하고 **다음 > 다음**을 누르십시오.
 - 네트워크 신임 정보**에 대해 프롬프트 표시되면 도메인 관리자의 비밀번호를 입력하십시오. 마지막 마법사 창에서 **완료 시 시스템 다시 시작**을 선택하십시오.
 - 잠시 후에 DNS 위임에 관한 **DCPromo** 오류가 포함된 창이 열릴 수 있습니다. **확인**을 클릭하십시오. 서버가 다시 시작됩니다.

결과

*earth*가 다시 시작되면 도메인 관리자로 로그인하십시오. *wmq.example.com* 도메인이 *earth*로 복제되었는지 확인하십시오.

다음에 수행할 작업

IBM WebSphere MQ 설치를 계속하십시오. 335 페이지의 『[wmq.example.com 도메인에 도메인 컨트롤러에 IBM WebSphere MQ 설치](#)』의 내용을 참조하십시오.

관련 태스크

319 페이지의 『[IBM WebSphere MQ에 대한 Active Directory 및 DNS 도메인 작성](#)』

wmq.example.com 도메인에 도메인 컨트롤러에 IBM WebSphere MQ 설치

wmq.example.com 도메인에 있는 두 도메인 컨트롤러에 IBM WebSphere MQ 설치를 설치하고 구성하십시오.

여기에 첫 번째 단락과 요약에 사용되는 간단한 설명을 적으십시오.

이 구성 예는 다음 세 개의 서버로 구성됩니다.

sun

첫 번째 도메인 제어기로 사용되는 Windows Server 2008 서버입니다. 이는 *sun*, *earth* 및 *mars*를 포함하는 *wmq.example.com* 도메인을 정의합니다. 이 서버는 QMGR이라고 하는 다중 인스턴스 큐 관리자의 한 인스턴스를 포함합니다.

earth

두 번째 도메인 컨트롤러 IBM WebSphere MQ 서버로 사용되는 Windows Server 2008. 이 서버는 QMGR이라고 하는 다중 인스턴스 큐 관리자의 두 번째 인스턴스를 포함합니다.

mars

파일 서버로 사용되는 Windows Server 2008입니다.

예에서 이탤릭체로 표시된 이름을 선택한 이름으로 바꾸십시오.

시작하기 전에

1. 319 페이지의 『IBM WebSphere MQ 에 대한 Active Directory 및 DNS 도메인 작성』의 단계를 수행하여 도메인 *wmq.example.com*에 대해 도메인 제어기 *sun*을 작성하십시오. 이탤릭체로 표시된 이름을 구성에 맞게 변경하십시오.
2. 333 페이지의 『*wmq.example.com* 도메인에 두 번째 도메인 제어기 추가』의 단계를 수행하여 도메인 *wmq.example.com*에 대해 두 번째 도메인 제어기 *earth*를 작성하십시오. 이탤릭체로 표시된 이름을 구성에 맞게 변경하십시오.
3. IBM WebSphere MQ 를 실행할 수 있는 기타 Windows 버전은 [Windows 시스템의 하드웨어 및 소프트웨어 요구사항](#)을 참조하십시오.

이 태스크 정보

wmq.example.com 도메인에 있는 두 도메인 컨트롤러에 IBM WebSphere MQ 설치를 설치하고 구성하십시오.

프로시저

1. *sun* 및 *earth*에서 IBM WebSphere MQ 설치.

IBM WebSphere MQ for Windows 설치 마법사 실행에 대한 자세한 정보는 [Windows 에 IBM WebSphere MQ 서버 설치](#)를 참조하십시오.

a) *sun* 및 *earth*에서 둘 다 도메인 관리자로 로그인하십시오. *wmq\Administrator*

b) IBM WebSphere MQ for Windows 설치 매체에서 **Setup** 명령을 실행하십시오.

IBM WebSphere MQ 런치패드 애플리케이션이 시작합니다.

c) **소프트웨어 요구사항**을 클릭하여 필수 소프트웨어가 설치되는지 확인하십시오.

d) **네트워크 구성 > 아니오**를 클릭하십시오.

이 설치에 대해 도메인 사용자 ID를 구성하거나 구성하지 않을 수 있습니다. 작성되는 사용자 ID는 도메인 로컬 사용자 ID입니다.

e) **WebSphere MQ 설치**를 클릭하고 설치 언어를 선택한 후 IBM WebSphere MQ 설치 프로그램을 시작을 클릭하십시오.

f) 라이선스 계약을 확인하고 **다음 > 다음 > 설치**를 클릭하여 기본 구성을 승인하십시오. 설치가 완료될 때까지 대기한 후 **마침**을 클릭하십시오.

설치 이름을 변경하거나, 다른 컴포넌트를 설치하거나, 큐 관리자 데이터 및 로그를 위한 다른 디렉토리를 구성하거나, 다른 디렉토리에 설치하려는 경우 **일반 설치**가 아닌 **사용자 설치**를 클릭하십시오.

IBM WebSphere MQ가 설치되었으며 설치 프로그램이 "IBM WebSphere MQ 준비" 마법사를 시작합니다.

IBM WebSphere MQ for Windows 설치하는 도메인 로컬 그룹 `mqm` 및 도메인 그룹 `Domain mqm`을 구성합니다. `Domain mqm`을(를) `mqm`의 멤버로 만듭니다. 동일한 도메인에 있는 후속 도메인 컨트롤러는 `mqm` 및 `Domain mqm` 그룹을 공유합니다.

2. `earth` 및 `sun` 모두에서 "IBM WebSphere MQ 준비" 마법사를 실행하십시오.

"IBM WebSphere MQ 준비" 마법사 실행에 대한 자세한 정보는 [준비 WebSphere MQ 마법사를 사용하여 WebSphere MQ 구성](#)의 내용을 참조하십시오.

- a) IBM WebSphere MQ 설치 프로그램이 "IBM WebSphere MQ 준비"를 자동으로 실행합니다.

마법사를 수동으로 시작하려면 시작 > 모든 프로그램 > **IBM WebSphere MQ** 폴더에서 "IBM WebSphere MQ 준비"의 바로 가기를 찾으십시오. 다중 설치 구성에서 IBM WebSphere MQ의 설치에 해당하는 바로 가기를 선택하십시오.

- b) 다음 을 클릭하고 질문에 대한 응답으로 **아니오** 를 클릭하십시오. "네트워크에 Windows 2000이상 도메인 컨트롤러가 있는지 식별"¹.

- c) 마지막 마법사 페이지에서 필요에 따라 선택란을 선택하거나 선택 취소하고 **마침**을 클릭하십시오.

"IBM WebSphere MQ 준비" 마법사는 첫 번째 도메인 제어기에서는 도메인 로컬 사용자 `MUSR_MQADMIN`을 작성하고, 두 번째 도메인 제어기에서는 다른 도메인 로컬 사용자 `MUSR_MQADMIN1`을 작성합니다. 마법사는 서비스에 로그인하는 사용자로 `MUSR_MQADMIN` 또는 `MUSR_MQADMIN1` 를 사용하여 각 제어기에 IBM WebSphere MQ 서비스를 작성합니다.

3. 큐 관리자 작성 권한이 있는 사용자를 정의하십시오.

사용자는 로컬로 로그인할 수 있는 권한을 가지고 있어야 하며 도메인 로컬 `mqm` 그룹의 구성원이어야 합니다. 도메인 제어기에서 도메인 사용자는 로컬로 로그인할 수 있는 권한을 가지지 않지만 관리자에게는 해당 권한이 있습니다. 기본적으로 이러한 두 속성 모두 가지는 사용자는 없습니다. 이 태스크에서 도메인 관리자를 도메인 로컬 `mqm` 그룹에 추가하십시오.

- a) 서버 관리자 > 역할 > **Active Directory** 도메인 서비스 > `wmq.example.com` > 사용자를 여십시오.

- b) 도메인 관리 > 그룹에 추가 ... 를 마우스 오른쪽 단추로 클릭하십시오. 및 `mqm`을 입력하십시오. **확인 이름** > **확인** 을 클릭하십시오.

결과

1. "IBM WebSphere MQ 준비"가 도메인 사용자 `MUSR_MQADMIN`를 작성했는지 확인하십시오.

- a) 서버 관리자 > 역할 > **Active Directory** 도메인 서비스 > `wmq.example.com` > 사용자를 여십시오.

- b) 마우스 오른쪽 단추 누르기 **MUSR_MQADMIN** > 특성 ... > 구성원 및 Domain users 및 `mqm`의 구성원임을 확인하십시오.

2. `MUSR_MQADMIN`에 서비스로 실행할 수 있는 권한이 있는지 확인하십시오.

- a) 시작 > 실행 ...을 클릭하십시오. `secpol.msc` 명령을 입력하고 **확인**을 클릭하십시오.을 클릭하십시오.

- b) 보안 설정 > 로컬 정책 > 사용자 권한 지정을 여십시오. 정책 목록에서 서비스로 로그인 > 특성을 수행하고 `MUSR_MQADMIN`이 서비스로 로그인할 수 있는 권한을 가진 것으로 나열되는지 확인하십시오. **확인**을 누르십시오.

다음에 수행할 작업

1. 설치 및 구성이 올바르게 작동하는지 확인하려면 344 페이지의 『로컬 `mqm` 그룹에서 권한 부여된 데이터 및 로그 파일 읽기 및 쓰기』 태스크를 수행하십시오.

¹ 도메인에 대한 설치를 구성할 수 있습니다. 도메인 제어기의 모든 사용자 및 그룹에는 도메인 범위가 있기 때문에 이 옵션은 별다른 차이를 만들지 않습니다. 도메인에 없는 것처럼 IBM WebSphere MQ(를) 설치하는 것이 더 간단합니다.

2. 330 페이지의 『도메인 제어기에 다중 인스턴스 큐 관리자 작성』 태스크로 돌아가서 도메인 제어기에서 다중 인스턴스 큐 관리자를 구성하는 태스크를 완료하십시오.

관련 개념

[WebSphere MQ Windows 서비스에 필요한 사용자 권한](#)

Windows에서 멀티 인스턴스 큐 관리자 확인

샘플 프로그램 **amqsgnac**, **amqspnac** 및 **amqsmnac**를 사용하여 다중 인스턴스 큐 관리자 구성을 확인합니다. 이 토픽은 Windows Server 2003의 멀티 인스턴스 큐 관리자 구성을 확인하기 위한 예제 구성을 제공합니다.

고가용성 샘플 프로그램은 자동 클라이언트 다시 연결을 사용합니다. 연결된 큐 관리자가 실패하면 클라이언트가 같은 큐 관리자 그룹에 있는 큐 관리자에 다시 연결하려고 시도합니다. 샘플 **고가용성 샘플 프로그램**에 대한 설명에서는 단순성을 위해 단일 인스턴스 큐 관리자를 사용한 클라이언트 다시 연결을 보여줍니다. 동일한 샘플을 다중 인스턴스 큐 관리자와 사용하여 다중 인스턴스 큐 관리자 구성을 확인할 수 있습니다.

이 예에서는 330 페이지의 『도메인 제어기에 다중 인스턴스 큐 관리자 작성』에 설명된 다중 인스턴스 구성을 사용합니다. 이 구성을 사용하여 다중 인스턴스 큐 관리자가 대기 인스턴스로 전환하는지 확인하십시오.

endmqm 명령을 사용하여 큐 관리자를 중지하고 전환 옵션인 **-s**를 사용하십시오. 클라이언트 프로그램이 새 큐 관리자 인스턴스에 다시 연결하여 잠깐의 지연 후 새 인스턴스에 대한 작업을 계속합니다.

클라이언트는 XP 서비스 팩 2를 실행 중인 400MB VMware 이미지에 설치됩니다. 보안상의 이유로 멀티 인스턴스 큐 관리자를 실행 중인 도메인 서버와 동일한 VMware 호스트 전용 네트워크에 연결됩니다. 이 클라이언트는 구성을 단순화하기 위해 클라이언트 연결 테이블을 포함하는 /MQHA 폴더를 공유합니다.

WebSphere MQ Explorer를 사용한 장애 복구 확인

샘플 애플리케이션을 사용하여 장애 복구를 확인하기 전에 각 서버에서 WebSphere MQ Explorer를 실행하십시오. **리모트 큐 관리자 추가 > 멀티 인스턴스 큐 관리자에 직접 연결** 마법사를 사용하여 각 탐색기에 큐 관리자 인스턴스 모듈을 추가하십시오. 대기를 허용하여 두 인스턴스 모두 실행되고 있는지 확인하십시오. 활성 인스턴스로 VMware 이미지를 실행하는 창을 닫아서 가상으로 서버의 전원을 끄거나 활성 인스턴스를 중지하여 대기 인스턴스로의 전환을 허용하고 다시 연결 가능한 클라이언트가 다시 연결할 수 있게 하십시오.

참고: 서버의 전원을 끌 경우에는 해당 서버가 MQHA 폴더를 호스팅하는 서버가 아니어야 합니다!

참고: 대기 인스턴스로 전환 허용 옵션은 큐 관리자 중지 대화 상자에서 사용할 수 없습니다. 이 옵션은 큐 관리자가 단일 인스턴스 큐 관리자로 실행되고 있기 때문에 누락되었습니다. **대기 인스턴스 허용** 옵션 없이 큐 관리자를 시작했습니다. 큐 관리자 중지 요청이 거부된 경우 **자세한 내용** 창을 확인하십시오. 실행 중인 대기 인스턴스가 없을 수 있습니다.

샘플 프로그램을 사용하여 장애 복구 확인

활성 인스턴스를 실행할 서버 선택

서버 중 MQHA 디렉토리 또는 파일 시스템을 호스팅할 서버를 선택할 수 있습니다. 활성 서버를 실행 중인 VMware 창을 닫아 장애 복구를 테스트하려는 경우 MQHA를 호스트하고 있는 서버가 아닌지 확인하십시오.

활성 큐 관리자 인스턴스를 실행하는 서버에서

1. **ipaddr1** 및 **ipaddr2**를 수정하고 다음 명령을 N:\hasample.tst.에 저장하십시오.

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('ipaddr1(1414),ipaddr2(1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

참고: **MCAUSER** 매개변수를 공백으로 둬으로써 클라이언트 사용자 ID가 서버로 송신됩니다. 이 클라이언트 사용자 ID에는 서버에 대한 올바른 권한이 있어야 합니다. 대안은 서버에서 구성한 사용자 ID에 대한 SVRCONN 채널에서 **MCAUSER** 매개변수를 설정하는 것입니다.

2. N:\ 경로에서 명령 프롬프트를 열고 다음 명령을 실행하십시오.

```
runmqsc -m QM1 < hasample.tst
```

3. **runmqsc** 명령의 출력을 검사하여 리스너가 실행 중이고 큐 관리자 제어를 가지는지 확인하십시오.

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)  
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

또는 WebSphere MQ 탐색기를 사용하여, TCPIP 리스너가 실행 중이고 Control = Queue Manager가 있습니다.

클라이언트에서

1. 서버의 공유 디렉토리 C:\MQHA를 클라이언트의 N:\ 로 맵핑하십시오.
2. N:\ 경로를 사용하여 명령 프롬프트를 여십시오. 서버의 클라이언트 채널 정의 테이블(CCDT)을 가리키도록 환경 변수 MQCHLLIB를 다음과 같이 설정하십시오.

```
SET MQCHLLIB=N:\data\QM1\@ipcc
```

3. 명령 프롬프트에 다음 명령을 입력하십시오.

```
start amqsghac TARGET QM1  
start amqsmhac -s SOURCE -t TARGET -m QM1  
start amqsphac SOURCE QM1
```

참고: 문제점이 있는 경우 이유 코드가 콘솔에 인쇄되도록 명령 프롬프트에서 애플리케이션을 시작하거나 N:\data\QM1\errors 폴더에서 AMQERR01.LOG 파일을 보십시오.

활성 큐 관리자 인스턴스를 실행하는 서버에서

1. 다음 중 하나입니다.
 - 활성 서버 인스턴스로 VMware 이미지를 실행하는 창을 닫으십시오.
 - WebSphere MQ Explorer를 사용하여 활성 큐 관리자 인스턴스를 중지하고 대기 인스턴스로 전환할 수 있게 하고 재연결 가능한 클라이언트가 재연결하도록 지시하십시오.
2. 세 클라이언트는 결국 연결이 끊어진 것을 감지한 후 다시 연결합니다. 이 구성에서 서버 창을 닫은 경우 세 개의 연결이 모두 재설정되는 데 7분 정도 걸립니다. 일부 연결은 다른 연결 전에 정상적으로 재설정됩니다.

결과

```
N:\>amqsphac SOURCE QM1  
Sample AMQSPHAC start  
target queue is SOURCE  
message <Message 1>  
message <Message 2>  
message <Message 3>  
message <Message 4>  
message <Message 5>  
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)  
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)  
17:05:52 : EVENT : Connection Reconnected  
message <Message 6>  
message <Message 7>  
message <Message 8>  
message <Message 9>
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1  
Sample AMQSMHA0 start  
  
17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)  
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)  
17:05:53 : EVENT : Connection Reconnected
```

```

N:\>amqsgnac TARGET QM1
Sample AMQSGHAC start
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message <Message 6>
message <Message 7>
message <Message 8>
message <Message 9>

```

Windows에서 공유 큐 관리자 데이터 및 로그 디렉토리 보안

이 주제에서는 글로벌 대체 보안 그룹을 사용하여 큐 관리자 데이터 및 로그 파일의 공유 위치를 보안 설정하는 방법을 설명합니다. 서로 다른 서버에서 실행되는 여러 큐 관리자 인스턴스 사이에서 위치를 공유할 수 있습니다.

일반적으로 큐 관리자 데이터 및 로그 파일에 대한 공유 위치는 설정하지 않습니다. IBM WebSphere MQ for Windows를 설치하면 이 설치 프로그램은 해당 서버에서 작성되는 큐 관리자에 대해 사용자가 선택하는 홈 디렉토리를 작성합니다. 로컬 mqm 그룹으로 디렉토리를 보안하고 IBM WebSphere MQ 서비스에 대한 사용자 ID가 디렉토리에 액세스하도록 구성합니다.

보안 그룹으로 공유 폴더를 보안 설정하는 경우 폴더에 대한 액세스가 허용된 사용자에게는 그룹의 신임 정보가 있어야 합니다. 원격 파일 서버의 폴더가 mars 서버의 로컬 mqm 그룹으로 보안되는 것으로 가정합니다. 큐 관리자 프로세스를 실행하는 사용자를 mars의 로컬 mqm 그룹 멤버로 설정하십시오. 사용자에게는 원격 파일 서버에 있는 폴더의 신임 정보와 일치하는 신임 정보가 있습니다. 이러한 신임 정보를 사용하여 큐 관리자는 폴더에 있는 해당 데이터 및 로그 파일에 액세스할 수 있습니다. 다른 서버에서 큐 관리자 프로세스를 실행하는 사용자는 일치하는 신임 정보가 없는 다른 로컬 mqm 그룹의 멤버입니다. 큐 관리자가 다른 서버에서 mars로 실행되면 mars에서 실행될 때 작성된 데이터 및 로그 파일에 액세스할 수 없습니다. 사용자를 도메인 사용자로 설정하더라도 이 사용자는 다른 신임 정보를 가집니다. 이 사용자가 mars의 로컬 mqm 그룹에서 신임 정보를 획득하고 다른 서버에서는 신임 정보 획득을 수행할 수 없기 때문입니다.

큐 관리자에 글로벌 대체 보안 그룹을 제공하면 문제점이 해결됩니다. 340 페이지의 그림 64의 내용을 참조하십시오. 글로벌 그룹으로 원격 폴더를 보안 설정하십시오. mars에서 작성할 때 글로벌 그룹의 이름을 큐 관리자에 전달하십시오. **crtmqm** 명령에 **-a[r]** 매개변수를 사용하여 글로벌 그룹 이름을 대체 보안 그룹으로 전달하십시오. 다른 서버에서 실행하기 위해 큐 관리자를 전송하는 경우 보안 그룹의 이름이 함께 전송됩니다. 이름은 **qm.ini** 파일의 **AccessMode** 스탠자에서 SecurityGroup으로 전송됩니다. 예를 들면, 다음과 같습니다.

```

AccessMode:
  SecurityGroup=wmq\wmq

```

또한 **qm.ini** 내의 **AccessMode** 스탠자에 **RemoveMQMAccess**가 포함됩니다. 예를 들어 다음과 같습니다.

```

AccessMode:
  RemoveMQMAccess=<true\false>

```

이 속성이 true 값으로 지정되고 액세스 그룹도 제공된 경우 로컬 mqm 그룹에는 큐 관리자 데이터 파일에 대한 액세스 권한이 부여되지 않습니다.

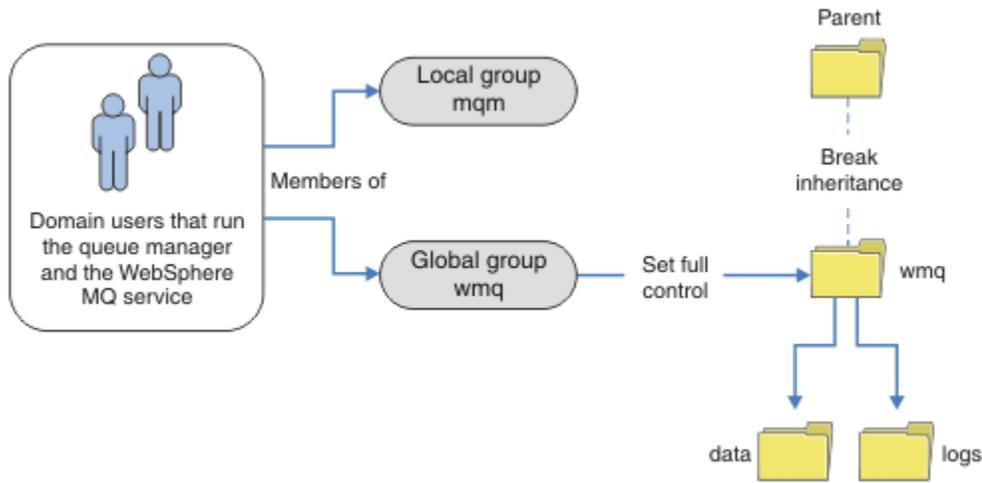


그림 64. 대체 글로벌 보안 그룹을 사용하여 큐 관리자 데이터 및 로그 보안 설정(1)

큐 관리자 프로세스 실행에 사용하는 사용자 ID가 글로벌 보안 그룹의 신임 정보와 일치하는 신임 정보를 가지도록 하려면 이 사용자 ID에도 글로벌 범위가 있어야 합니다. 로컬 그룹 또는 프린시펄을 글로벌 그룹의 멤버로 설정할 수 없습니다. 340 페이지의 그림 64에 큐 관리자 프로세스를 실행하는 사용자가 도메인 사용자로 표시되어 있습니다.

많은 IBM WebSphere MQ 서버를 배치한 경우 340 페이지의 그림 64의 사용자 그룹은 편리하지 않습니다. 모든 IBM WebSphere MQ 서버에 대해 사용자를 로컬 그룹에 추가하는 프로세스를 반복해야 합니다. 대신 도메인 컨트롤러에서 Domain mqm 글로벌 그룹을 작성하고 Domain mqm 그룹의 IBM WebSphere MQ 멤버를 실행하는 사용자를 작성하십시오 (340 페이지의 그림 65 참조). IBM WebSphere MQ를 도메인 설치로 설치하면 "IBM WebSphere MQ 준비" 마법사가 자동으로 Domain mqm 그룹을 로컬 mqm 그룹의 멤버로 설정합니다. 동일한 사용자가 글로벌 그룹 Domain mqm 및 wmq 모두에 있습니다.

팁: 동일한 사용자가 여러 서버에서 IBM WebSphere MQ를 실행할 수 있지만 개별 서버에서는 IBM WebSphere MQ를 서비스로 실행하고 대화식으로 실행하기 위해 다른 사용자가 있어야 합니다. 서버의 모든 설치에 대해서도 다른 사용자가 있어야 합니다. 일반적으로 Domain mqm에는 여러 사용자가 포함되어 있습니다.

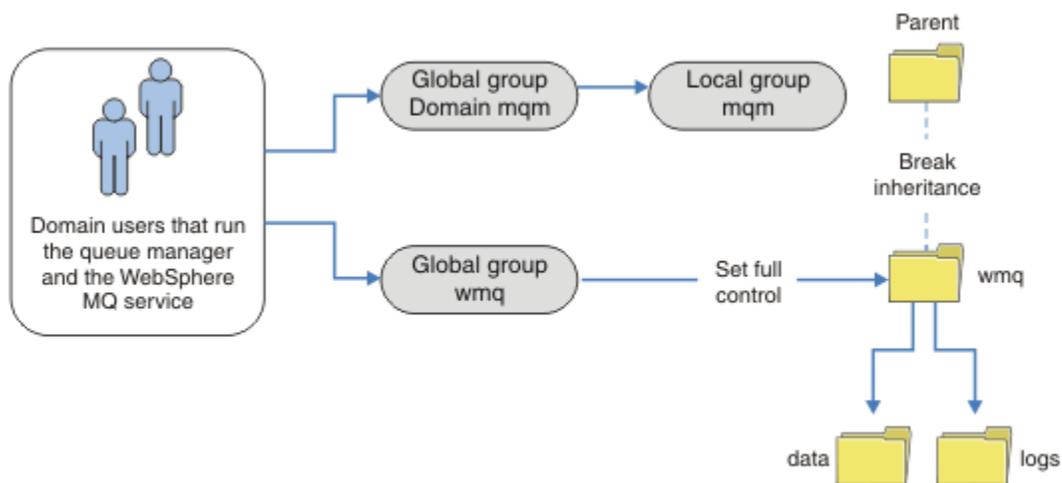


그림 65. 대체 글로벌 보안 그룹을 사용하여 큐 관리자 데이터 및 로그 보안 설정(2)

340 페이지의 그림 65에서 조직은 현재 상태 그대로 불필요하게 복잡합니다. 이 배열에는 동일한 멤버를 포함하는 두 개의 글로벌 그룹이 있습니다. 조직을 단순화하고 하나의 글로벌 그룹만 정의할 수 있습니다(341 페이지의 그림 66 참조).

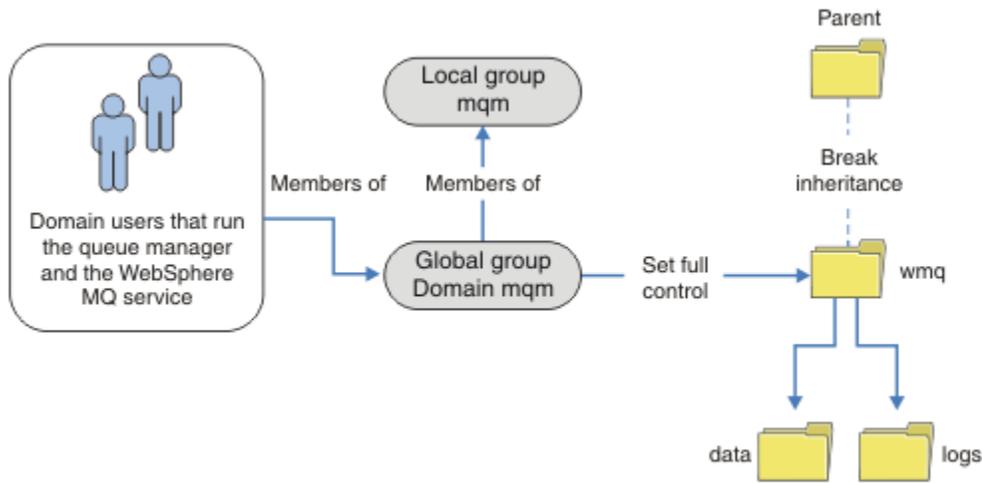


그림 66. 대체 글로벌 보안 그룹을 사용하여 큐 관리자 데이터 및 로그 보안 설정(3)

또는 폴더에 따라 다른 큐 관리자가 액세스할 수 있도록 제한되는 더 세분화된 수준의 액세스 제어가 필요할 수 있습니다. 342 페이지의 그림 67의 내용을 참조하십시오. 342 페이지의 그림 67에서 다른 큐 관리자 로그 및 데이터 파일을 보안 설정하기 위해 두 그룹의 도메인 사용자가 개별 글로벌 그룹으로 정의됩니다. 서로 다른 두 개의 로컬 mqm 그룹이 표시되며, 이는 다른 IBM WebSphere MQ 서버에 있어야 합니다. 이 예에서 큐 관리자는 두 세트에 다른 사용자가 할당되는 두 개의 세트로 파티션됩니다. 이 두 세트는 테스트 및 프로덕션 큐 관리자일 수 있습니다. 대체 보안 그룹은 wmq1 및 wmq2라고 합니다. 글로벌 그룹 wmq1 및 wmq2 를 테스트 또는 프로덕션 부서에 있는지에 따라 올바른 큐 관리자에 수동으로 추가해야 합니다. 구성은 두 그룹의 사용자가 있기 때문에 IBM WebSphere MQ 설치가 341 페이지의 그림 66에서와 같이 로컬 mqm 그룹에 Domain mqm 전파되는 것을 이용할 수 없습니다.

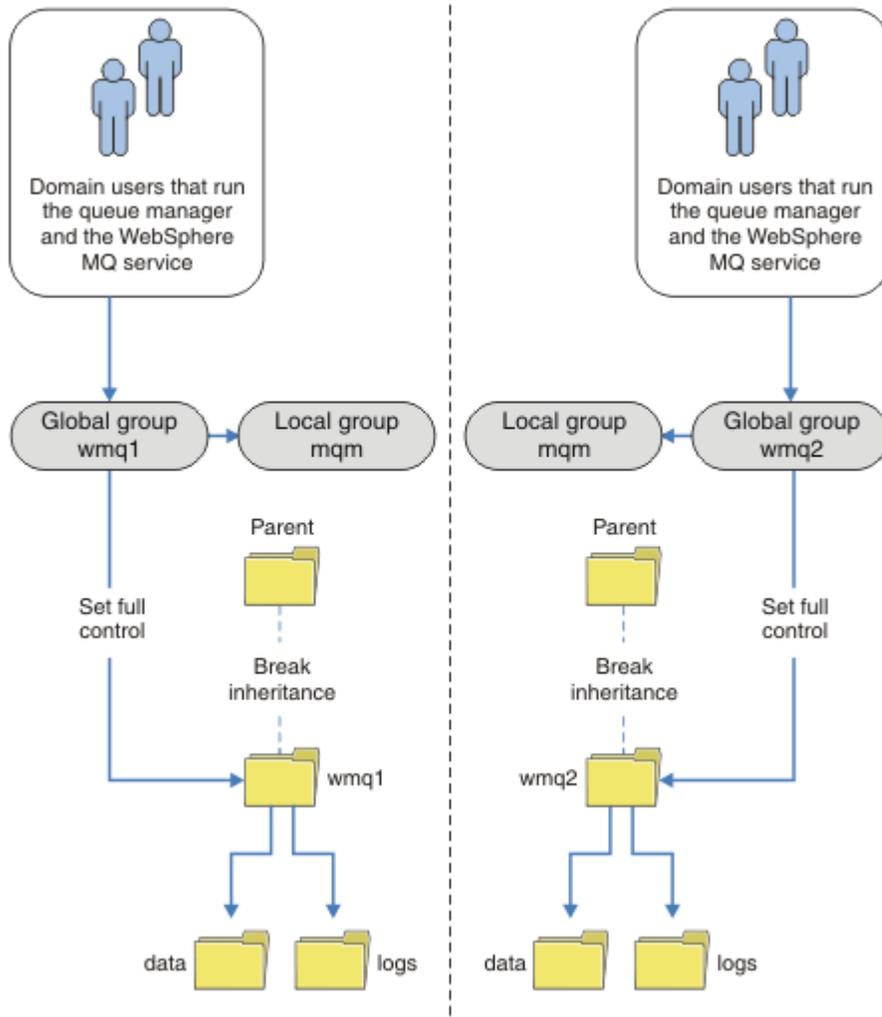


그림 67. 대체 글로벌 보안 그룹을 사용하여 큐 관리자 데이터 및 로그 보안 설정(4)

두 구획으로 파티션하는 것에 대한 대체 방식은 이를 두 개의 창 도메인에 배치하는 것입니다. 이 경우에 [341 페이지의 그림 66](#)에 표시된 보다 단순한 모델 사용으로 돌아갈 수 있습니다.

Windows에서 비공유 큐 관리자 데이터와 로그 디렉토리 및 파일 보안 설정

이 주제에서는 로컬 mqm 그룹 및 대체 보안 그룹 둘 다를 사용하여 큐 관리자 데이터 및 로그 파일의 대체 위치를 보안 설정하는 방법을 설명합니다.

일반적으로 큐 관리자 데이터 및 로그 파일에 대한 대체 위치는 설정하지 않습니다. IBM WebSphere MQ for Windows를 설치하면 이 설치 프로그램은 작성되는 큐 관리자에 대해 사용자가 선택하는 홈 디렉토리를 작성합니다. 로컬 mqm 그룹으로 디렉토리를 보안하고 IBM WebSphere MQ 서비스에 대한 사용자 ID가 디렉토리에 액세스하도록 구성합니다.

두 가지 예에서 IBM WebSphere MQ에 대한 액세스 제어를 구성하는 방법을 보여줍니다. 이러한 예는 설치 시 작성된 데이터 및 로그 경로에 없는 디렉토리의 해당 데이터 및 로그를 사용하여 큐 관리자를 작성하는 방법을 보여줍니다. 첫 번째 예 [344 페이지의 『로컬 mqm 그룹에서 권한 부여된 데이터 및 로그 파일 읽기 및 쓰기』](#)에서는 로컬 mqm 그룹에 의해 권한을 부여하여 큐 및 로그 디렉토리에 대한 액세스를 허용합니다. 두 번째 예 [346 페이지의 『대체 로컬 보안 그룹이 권한을 부여한 데이터 및 로그 파일 읽기 및 쓰기』](#)는 디렉토리에 대한 액세스 권한이 대체 보안 그룹에 의해 부여된다는 점에서 다릅니다. 한 서버에서만 실행되는 큐 관리자가 디렉토리에 액세스하는 경우 대체 보안 그룹으로 데이터 및 로그 파일을 보안 설정하면 다른 로컬 그룹 또는 프린시펄을 사용하여 다른 큐 관리자를 보안 설정하는 방법을 선택할 수 있습니다. 다중 인스턴스 큐 관리자와 같이 여러 서버에서 실행되는 큐 관리자가 디렉토리에 액세스하는 경우 대체 보안 그룹으로 데이터 및 로그 파일을 보안 설정하는 방

범만 선택할 수 있습니다. 339 페이지의 『Windows에서 공유 큐 관리자 데이터 및 로그 디렉토리 보안』의 내용을 참조하십시오.

큐 관리자 데이터 및 로그 파일의 보안 권한 구성은 Windows에서 공통 구성이 아닙니다. IBM WebSphere MQ for Windows를 설치할 때 큐 관리자 데이터 및 로그에 대한 디렉토리를 지정하거나 기본 디렉토리를 승인합니다. 설치 프로그램은 로컬 mqm 그룹에 전체 제어 권한을 제공하여 이 그룹으로 이러한 디렉토리를 자동으로 보안 설정합니다. 설치 프로세스는 큐 관리자를 실행하는 사용자 ID가 로컬 mqm 그룹의 멤버인지 확인합니다. 액세스 요구사항을 충족시키기 위해 디렉토리에 대한 다른 액세스 제어를 수정할 수 있습니다.

데이터 및 로그 파일 디렉토리를 새 위치로 이동하는 경우 새 위치의 보안을 구성해야 합니다. 큐 관리자를 백업하고 다른 컴퓨터로 복원하는 경우 또는 큐 관리자가 다중 인스턴스 큐 관리자가 되도록 변경하는 경우 디렉토리의 위치를 변경할 수 있습니다. 새 위치에서 큐 관리자 데이터 및 로그 디렉토리를 보안 설정하기 위해 선택할 수 있는 두 가지 방법이 있습니다. 로컬 mqm 그룹에 대한 액세스를 제한하여 디렉토리를 보안 설정하거나 선택한 보안 그룹에 대한 액세스를 제한할 수 있습니다.

로컬 mqm 그룹을 사용하여 디렉토리를 보안 설정하는 방법에 가장 적은 수의 단계가 소요됩니다. 데이터 및 로그 디렉토리에 대한 권한을 설정하여 로컬 mqm 그룹 전체 제어를 허용하십시오. 일반적인 접근법은 기존 권한 세트를 복사하고 상위에서 상속을 제거하는 것입니다. 그런 다음 다른 프린시펄의 권한을 제거하거나 제한할 수 있습니다.

IBM WebSphere MQ 준비 마법사에서 설정된 서비스에 대해 다른 사용자 ID로 큐 관리자를 실행하는 경우 해당 사용자 ID는 로컬 mqm 그룹의 멤버여야 합니다. 이 344 페이지의 『로컬 mqm 그룹에서 권한 부여된 데이터 및 로그 파일 읽기 및 쓰기』 태스크는 단계를 통해 안내합니다.

대체 보안 그룹을 사용하여 큐 관리자 데이터 및 로그 파일을 보안 설정할 수도 있습니다. 대체 보안 그룹으로 큐 관리자 데이터 및 로그 파일을 보안 설정하는 프로세스에는 343 페이지의 그림 68의 내용을 참조하는 다수의 단계가 포함됩니다. 로컬 그룹 wmq는 대체 보안 그룹의 예입니다.

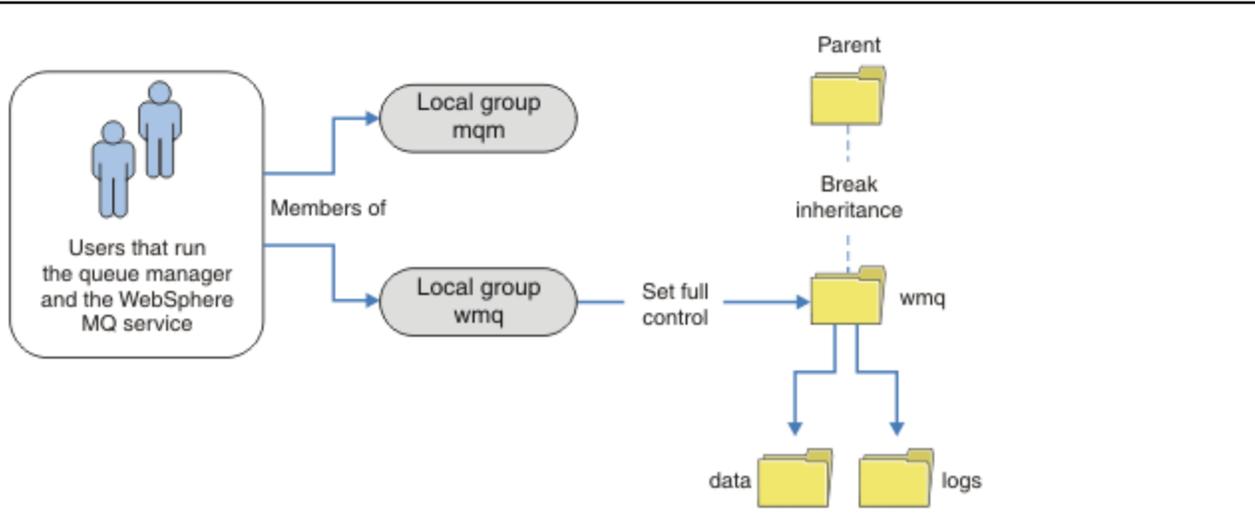


그림 68. 대체 로컬 보안 그룹, wmq를 사용하여 큐 관리자 데이터 및 로그 보안

1. 큐 관리자 데이터 및 로그에 대한 개별 디렉토리, 공용 디렉토리 또는 공용 상위 디렉토리를 작성하십시오.
2. 디렉토리에 대한 기존 상속 권한 세트 또는 상위 디렉토리를 복사하고 필요에 따라 수정하십시오.
3. 대체 그룹 wmq에 디렉토리에 대한 전체 제어 권한을 제공하여 큐 관리자 및 로그를 포함할 디렉토리를 보안 설정하십시오.
4. 큐 관리자 프로세스를 실행하는 모든 사용자 ID에 대체 보안 그룹의 신임 정보 또는 프린시펄을 제공하십시오.
 - a. 사용자를 대체 보안 프린시펄로 정의하는 경우 이 사용자는 큐 관리자를 실행할 관리자와 같은 관리자여야 합니다. 이 사용자는 로컬 mqm 그룹의 멤버여야 합니다.

- b. 로컬 그룹을 대체 보안 그룹으로 정의하는 경우 큐 관리자를 실행할 사용자를 대체 그룹에 추가하십시오. 이 사용자는 또한 로컬 `mqm` 그룹의 멤버여야 합니다.
 - c. 글로벌 그룹을 대체 보안 그룹으로 정의하는 경우 339 페이지의 『Windows에서 공유 큐 관리자 데이터 및 로그 디렉토리 보안』의 내용을 참조하십시오.
5. -a 매개변수로 `crtmqm` 명령에서 대체 보안 그룹 또는 프린시펄을 지정하여 큐 관리자를 작성하십시오.

로컬 `mqm` 그룹에서 권한 부여된 데이터 및 로그 파일 읽기 및 쓰기

이 태스크에서는 선택한 디렉토리에 저장된 해당 데이터 및 로그 파일을 사용하여 큐 관리자를 작성하는 방법을 보여줍니다. 파일에 대한 액세스는 로컬 `mqm` 그룹에 의해 보안됩니다. 이 디렉토리는 공유하지 않습니다.

시작하기 전에

1. 기본 설치로 IBM WebSphere MQ for Windows를 설치하십시오.
2. "IBM WebSphere MQ 준비" 마법사를 실행하십시오. 이 태스크에서는 로컬 사용자 ID로 실행하거나 또는 도메인 사용자 ID로 실행하도록 설치를 구성하십시오. 결국 315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』의 모든 태스크를 완료하기 위해 설치하는 도메인에 대해 구성되어야 합니다.
3. 관리자 권한으로 로그인하여 태스크의 첫 번째 부분을 수행하십시오.

이 태스크 정보

이 태스크는 큐 관리자 데이터 및 로그 파일 액세스를 설명하는 관련 태스크 세트 중 하나입니다. 이 태스크는 사용자가 선택한 디렉토리에 저장된 데이터 및 로그 파일을 읽고 쓸 수 있도록 권한이 부여된 큐 관리자를 작성하는 방법을 보여줍니다. 315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』 태스크를 수반합니다.

Windows의 경우 선택한 모든 디렉토리에서 IBM WebSphere MQ for Windows에 대한 기본 데이터 및 로그 경로를 작성할 수 있습니다. 설치 및 구성 마법사는 자동으로 로컬 `mqm` 그룹 및 큐 관리자 프로세스를 실행 중인 사용자 ID, 디렉토리에 대한 액세스를 제공합니다. 큐 관리자 데이터 및 로그 파일에 대해 다른 디렉토리를 지정하는 큐 관리자를 작성하는 경우 이 디렉토리에 대한 전체 제어 권한을 구성해야 합니다.

이 예제에서는 로컬 `mqm` 그룹 권한을 `c:\wmq` 디렉토리에 부여하여 큐 관리자가 데이터 및 로그 파일을 완전히 제어할 수 있도록 합니다.

`crtmqm` 명령은 IBM WebSphere MQ 서비스를 사용하여 워크스테이션을 시작할 때 자동으로 시작되는 큐 관리자를 작성합니다.

이 태스크는 설명용이며 변경할 수 있는 특정 값을 사용합니다. 변경할 수 있는 값은 이탤릭체로 표시됩니다. 태스크 종료 시에 사용자가 작성한 모든 변경사항을 제거하려면 지시사항을 따르십시오.

프로시저

1. 명령 프롬프트를 여십시오.
2. 명령을 입력하십시오.

```
md c:\wmq\data , c:\wmq\logs
```

3. 로컬 `mqm` 그룹 읽기 및 쓰기 액세스를 허용하도록 디렉토리에 대한 권한을 설정하십시오.

```
cacls c:\wmq /T /E /G mqm:F
```

시스템이 다음과 같이 응답합니다.

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

4. 옵션: 로컬 `mqm` 그룹의 구성원인 사용자 ID로 전환하십시오.

관리자로서 계속할 수 있지만 실제 프로덕션 구성에서는 보다 제한적인 권한을 가지는 사용자 ID로 계속할 수 있습니다. 사용자 ID는 최소한 로컬 `mqm` 그룹의 구성원이어야 합니다.

IBM WebSphere MQ 설치가 도메인의 일부로 구성된 경우 사용자 ID를 Domain mqm 그룹의 구성원으로 설정하십시오. "IBM WebSphere MQ 준비" 마법사에서는 Domain mqm 글로벌 그룹이 로컬 mqm 그룹의 멤버가 되도록 하므로, 사용자 ID를 로컬 mqm 그룹의 멤버로 만들 필요가 없습니다.

5. 큐 관리자를 작성하십시오.

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager created.  
Directory 'c:\wmq\data\QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
Default objects statistics : 74 created. 0 replaced.  
Completing setup.  
Setup completed.
```

6. 큐 관리자가 작성한 디렉토리가 c:\wmq 디렉토리에 있는지 확인하십시오.

```
dir c:\wmq /D /B /S
```

7. 파일에 읽기 및 쓰기가 있는지 확인하거나 로컬 mqm 그룹에 대한 전체 제어 권한이 있는지 확인하십시오.

```
cacls c:\wmq\*.*
```

다음에 수행할 작업

메시지를 큐에 넣거나 가져와서 큐 관리자를 테스트하십시오.

1. 큐 관리자를 시작하십시오.

```
strmqm QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. 테스트 큐를 작성하십시오.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

시스템이 다음과 같이 응답합니다.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: WebSphere MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. 샘플 프로그램 **amqsput**을 사용하여 테스트 메시지를 넣으십시오.

```
echo 'A test message' | amqsput QTEST QMGR
```

시스템이 다음과 같이 응답합니다.

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. 샘플 프로그램 **amqsget**을 사용하여 테스트 메시지를 가져오십시오.

```
amqsget QTEST QMGR
```

시스템이 다음과 같이 응답합니다.

```
Sample AMQSGET0 start
message <A test message>
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. 큐 관리자를 중지합니다.

```
endmqm -i QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ended.
```

6. 큐 관리자를 삭제합니다.

```
dltmqm QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' deleted.
```

7. 작성한 디렉토리를 삭제하십시오.

팁: 명령이 각 파일 또는 디렉토리를 삭제하도록 프롬프트하지 않도록 명령에 /Q 옵션을 추가하십시오.

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

관련 개념

[315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』](#)

Windows의 다중 인스턴스 큐 관리자에서는 해당 데이터 및 로그가 공유되어야 합니다. 공유는 여러 서버 또는 워크스테이션에서 실행 중인 큐 관리자의 모든 인스턴스가 액세스할 수 있어야 합니다. 큐 관리자 및 공유를 Windows 도메인의 부분으로 구성하십시오. 큐 관리자는 도메인 워크스테이션 또는 서버 또는 도메인 제어기에서 실행할 수 있습니다.

관련 태스크

[346 페이지의 『대체 로컬 보안 그룹이 권한을 부여한 데이터 및 로그 파일 읽기 및 쓰기』](#)

이 태스크에서는 **crtmqm** 명령에서 -a 플래그를 사용하는 방법을 보여줍니다. 이 플래그는 큐 관리자에게 대체 로컬 보안 그룹을 제공하여 해당 로그 및 데이터 파일에 대한 액세스 권한을 제공합니다.

[327 페이지의 『대체 글로벌 보안 그룹이 권한을 부여한 공유 데이터 및 로그 파일 읽기 및 쓰기』](#)

[316 페이지의 『도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성』](#)

대체 로컬 보안 그룹이 권한을 부여한 데이터 및 로그 파일 읽기 및 쓰기

이 태스크에서는 **crtmqm** 명령에서 -a 플래그를 사용하는 방법을 보여줍니다. 이 플래그는 큐 관리자에게 대체 로컬 보안 그룹을 제공하여 해당 로그 및 데이터 파일에 대한 액세스 권한을 제공합니다.

시작하기 전에

1. 기본 설치로 IBM WebSphere MQ for Windows를 설치하십시오.

2. "IBM WebSphere MQ 준비" 마법사를 실행하십시오. 이 태스크에서는 로컬 사용자 ID로 실행하거나 또는 도메인 사용자 ID로 실행하도록 설치를 구성하십시오. 결국 315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』의 모든 태스크를 완료하기 위해 설치하는 도메인에 대해 구성되어야 합니다.
3. 관리자 권한으로 로그인하여 태스크의 첫 번째 부분을 수행하십시오.

이 태스크 정보

이 태스크는 큐 관리자 데이터 및 로그 파일 액세스를 설명하는 관련 태스크 세트 중 하나입니다. 이 태스크는 사용자가 선택한 디렉토리에 저장된 데이터 및 로그 파일을 읽고 쓸 수 있도록 권한이 부여된 큐 관리자를 작성하는 방법을 보여줍니다. 315 페이지의 『Windows 도메인 및 다중 인스턴스 큐 관리자』 태스크를 수반합니다.

Windows의 경우 선택한 모든 디렉토리에서 IBM WebSphere MQ for Windows에 대한 기본 데이터 및 로그 경로를 작성할 수 있습니다. 설치 및 구성 마법사는 자동으로 로컬 mqm 그룹 및 큐 관리자 프로세스를 실행 중인 사용자 ID, 디렉토리에 대한 액세스를 제공합니다. 큐 관리자 데이터 및 로그 파일에 대해 다른 디렉토리를 지정하는 큐 관리자를 작성하는 경우 이 디렉토리에 대한 전체 제어 권한을 구성해야 합니다.

이 예에서는 큐 관리자에게 디렉토리에 대한 전체 제어 권한이 있는 대체 보안 로컬 그룹을 제공합니다. 이 대체 보안 그룹은 큐 관리자에게 디렉토리에 있는 파일을 관리할 수 있는 권한을 제공합니다. 대체 보안 그룹의 기본 용도는 대체 보안 글로벌 그룹에 권한을 부여하는 것입니다. 대체 보안 글로벌 그룹을 사용하여 다중 인스턴스 큐 관리자를 설정하십시오. 이 예에서는 사용자가 도메인에 IBM WebSphere MQ를 설치하지 않고도 대체 보안 그룹의 사용을 익힐 수 있도록 로컬 그룹을 구성합니다. 로컬 그룹을 대체 보안 그룹으로 구성하는 것은 일반적이지 않습니다.

crtmqm 명령은 IBM IBM WebSphere MQ 서비스를 사용하여 워크스테이션을 시작할 때 자동으로 시작되는 큐 관리자를 작성합니다.

이 태스크는 설명용이며 변경할 수 있는 특정 값을 사용합니다. 변경할 수 있는 값은 이탤릭체로 표시됩니다. 태스크 종료 시에 사용자가 작성한 모든 변경사항을 제거하려면 지시사항을 따르십시오.

프로시저

1. 대체 보안 그룹을 설정하십시오.

대체 보안 그룹은 일반적으로 도메인 그룹입니다. 이 예에서는 로컬 대체 보안 그룹을 사용하는 큐 관리자를 작성합니다. 로컬 대체 보안 그룹을 사용하여 도메인의 일부가 아닌 IBM WebSphere MQ 설치에 대한 태스크를 수행할 수 있습니다.

- a) **lusrmgr.msc** 명령을 실행하여 로컬 사용자 및 그룹 창을 여십시오.
- b) 다음에서 마우스의 오른쪽 단추를 클릭하십시오. **그룹 > 새 그룹...**
- c) **그룹 이름** 필드에 **altmqm** 를 입력하고 **작성 > 닫기**를 클릭하십시오.
- d) IBM IBM WebSphere MQ 서비스를 실행하는 사용자 ID를 식별하십시오.
 - i) **시작 > 실행 ...**을 클릭하십시오. **services.msc** 를 입력하고 **확인**을 클릭하십시오.
 - ii) 서비스 목록에서 IBM IBM WebSphere MQ 서비스를 클릭하고 로그인 탭을 클릭하십시오.
 - iii) 이 사용자 ID를 기억해두고 Services Explorer를 닫으십시오.
- e) IBM IBM WebSphere MQ 서비스를 실행하는 사용자 ID를 **altmqm** 그룹에 추가하십시오. 또한 로그인할 때 사용하는 사용자 ID를 추가하여 큐 관리자를 작성하고 대화식으로 실행하십시오.

Windows는 큐 관리자 프로세스를 실행하는 사용자 ID의 권한을 확인하여 데이터 및 로그 디렉토리에 액세스하기 위한 큐 관리자의 권한을 확인합니다. 사용자 ID는 디렉토리를 권한 부여한 **altmqm** 그룹의 글로벌 그룹을 통해 직접 또는 간접적으로 구성원이어야 합니다.

IBM WebSphere MQ를 도메인의 일부로 설치하고 316 페이지의 『도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성』의 태스크를 수행하려는 경우 319 페이지의 『IBM WebSphere MQ에 대한 Active Directory 및 DNS 도메인 작성』에서 작성된 도메인 사용자 ID는 **wmquser1** 및 **wmquser2**입니다.

도메인의 일부로 큐 관리자를 설치하지 않은 경우 IBM IBM WebSphere MQ 서비스를 실행하는 기본 로컬 사용자 ID는 **MUSR_MQADMIN**입니다. 관리자 권한 없이 이 태스크를 수행하려고 하는 경우 로컬 **mqm** 그룹의 멤버인 사용자를 작성하십시오.

다음 단계에 따라 *altnmqm*에 *wmquser1* 및 *wmquser2*을(를) 추가하십시오. 구성이 다른 경우 사용자 ID 및 그룹을 자신의 이름으로 대체하십시오.

- i) 그룹 목록에서 **altnmqm > 특성 > 추가 ...**를 마우스 오른쪽 단추로 클릭하십시오.
- ii) 사용자, 컴퓨터 또는 그룹 선택 창에서 *wmquser1*; *wmquser2*를 입력하고 **이름 검사**를 클릭하십시오.
- iii) Windows 보안 창에서 도메인 관리자의 이름 및 암호를 입력한 후 **확인 > 확인 > 적용 > 확인**을 누르십시오.

2. 명령 프롬프트를 여십시오.

3. IBM IBM WebSphere MQ 서비스를 재시작하십시오.

서비스를 실행하는 사용자 ID가 그에 대해 구성한 추가 보안 신임 정보를 획득할 수 있도록 서비스를 재시작해야 합니다.

다음 명령을 입력하십시오.

```
endmqsvc  
strmqsvc
```

시스템이 다음과 같이 응답합니다.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

및

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' started successfully.
```

4. 명령을 입력하십시오.

```
md c:\wmq\data , c:\wmq\logs
```

5. 로컬 사용자 *user* 읽기 및 쓰기 액세스를 허용하도록 디렉토리에 대한 권한을 설정하십시오.

```
cacls c:\wmq /T /E /G altnmqm:F
```

시스템이 다음과 같이 응답합니다.

```
processed dir: c:\wmq  
processed dir: c:\wmq\data  
processed dir: c:\wmq\logs
```

6. 옵션: 로컬 mqm 그룹의 구성원인 사용자 ID로 전환하십시오.

관리자로서 계속할 수 있지만 실제 프로덕션 구성에서는 보다 제한적인 권한을 가지는 사용자 ID로 계속할 수 있습니다. 사용자 ID는 최소한 로컬 mqm 그룹의 구성원이어야 합니다.

IBM WebSphere MQ 설치가 도메인의 일부로 구성된 경우 사용자 ID를 Domain mqm 그룹의 구성원으로 설정하십시오. "IBM WebSphere MQ 준비" 마법사에서는 Domain mqm 글로벌 그룹이 로컬 mqm 그룹의 멤버가 되도록 하므로, 사용자 ID를 로컬 mqm 그룹의 멤버로 만들 필요가 없습니다.

7. 큐 관리자를 작성하십시오.

```
crtmqm -a altnmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager created.  
Directory 'c:\wmq1\data\QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
Default objects statistics : 74 created. 0 replaced.  
Completing setup.  
Setup completed.
```

8. 큐 관리자가 작성한 디렉토리가 `c:\wmq` 디렉토리에 있는지 확인하십시오.

```
dir c:\wmq /D /B /S
```

9. 파일에 읽기 및 쓰기가 있는지 확인하거나 로컬 mqm 그룹에 대한 전체 제어 권한이 있는지 확인하십시오.

```
cacls c:\wmq\*.*
```

다음에 수행할 작업

메시지를 큐에 넣거나 가져와서 큐 관리자를 테스트하십시오.

1. 큐 관리자를 시작하십시오.

```
strmqm QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. 테스트 큐를 작성하십시오.

```
echo define qlocal(QTEST) | runmqsc QMGR
```

시스템이 다음과 같이 응답합니다.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: WebSphere MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. 샘플 프로그램 **amqsput**을 사용하여 테스트 메시지를 넣으십시오.

```
echo 'A test message' | amqsput QTEST QMGR
```

시스템이 다음과 같이 응답합니다.

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. 샘플 프로그램 **amqsget**을 사용하여 테스트 메시지를 가져오십시오.

```
amqsget QTEST QMGR
```

시스템이 다음과 같이 응답합니다.

```
Sample AMQSGET0 start  
message <A test message>  
Wait 15 seconds ...  
no more messages  
Sample AMQSGET0 end
```

5. 큐 관리자를 중지합니다.

```
endmqm -i QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ended.
```

6. 큐 관리자를 삭제합니다.

```
dltmqm QMGR
```

시스템이 다음과 같이 응답합니다.

```
WebSphere MQ queue manager 'QMGR' deleted.
```

7. 작성한 디렉토리를 삭제하십시오.

팁: 명령이 각 파일 또는 디렉토리를 삭제하도록 프롬프트하지 않도록 명령에 /Q 옵션을 추가하십시오.

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

Linux에서 다중 인스턴스 큐 관리자 작성

이 예는 Linux에서 다중 인스턴스 큐 관리자를 설정하는 방법을 보여줍니다. 이 설정은 포함된 개념을 설명하기에는 작습니다. 예제는 Linux Red Hat Enterprise 5를 기반으로 합니다. 이 단계는 다른 플랫폼 따라 다릅니다.

이 예제는 XP 서비스 팩 2를 실행하는 3GB RAM이 있는 2GHz 노트북 컴퓨터에 설정됩니다. 두 개의 VMware 가상 머신(Server1 및 Server2)이 640MB 이미지에서 Linux Red Hat Enterprise 5를 실행합니다. Server1은 네트워크 파일 시스템(NFS), 큐 관리자 로그 및 HA 인스턴스를 호스팅합니다. NFS 서버가 큐 관리자 인스턴스 중 하나를 호스팅하는 것은 일반적인 사례가 아닙니다. 이는 예를 단순화하기 위한 것입니다. Server2는 대기 인스턴스로 Server1의 큐 관리자 로그를 마운트합니다. WebSphere MQ MQI 클라이언트는 Windows XP 서비스 팩 2를 실행하고 샘플 고가용성 애플리케이션을 실행하는 추가 400MB VMware 이미지에 설치되었습니다. 모든 가상 시스템은 보안상의 이유로 VMware 호스트 전용 네트워크의 일부로 구성됩니다.

참고: 큐 관리자 데이터만 NFS 서버에 배치해야 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.

예

표 30. Linux 의 예시적인 멀티 인스턴스 큐 관리자 구성	
Server1	Server2
루트로 로그인	

표 30. Linux 의 예시적인 멀티 인스턴스 큐 관리자 구성 (계속)	
Server1	Server2
<p>IBM WebSphere MQ 설치의 지시사항에 따라 WebSphere MQ를 설치하고 mqm 사용자와 그룹이 없으면 이를 작성하고 /var/mqm을 정의하십시오.</p> <p>Check what uid and gid in /etc/passwd on the first machine displays for mqm; for example,</p> <pre>mqm:x:501:100:MQ User:/var/mqm:/bin/bash</pre> <p>두 번째 시스템의 /etc/passwd 에 있는 mqm의 uid 및 gid 와 일치하여 이러한 값이 동일한지 확인하십시오. 값을 변경해야 하는 경우 시스템을 다시 시작하십시오.</p>	
<p>공유 파일 시스템 작동 확인 태스크를 완료하여 파일 시스템이 다중 인스턴스 큐 관리자를 지원하는지 확인하십시오.</p>	
<p>공유될 공용 폴더 /MQHA에서 로그 및 데이터 디렉토리를 작성하십시오. 예를 들면, 다음과 같습니다.</p> <ol style="list-style-type: none"> mkdir /MQHA mkdir /MQHA/logs mkdir /MQHA/qmgrs 	<p>공유 파일 시스템을 마운트할 /MQHA 폴더를 작성하십시오. 경로를 서버 1에서와 동일하게 유지하십시오. 예를 들어 다음과 같습니다.</p> <ol style="list-style-type: none"> mkdir /MQHA
<p>사용자 및 그룹 mqm이 MQHA 디렉토리를 소유하고 사용자 및 그룹에 대해 액세스 권한이 rwx 로 설정되어 있는지 확인하십시오. 예를 들어, ls -al 가 표시됩니다.</p> <pre>drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA</pre> <ol style="list-style-type: none"> chown -R mqm:mqm /MQHA chmod -R ug+rwx /MQHA 	
<p>큐 관리자를 작성하십시오.</p> <pre>crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM1</pre>	
<p>추가²/MQHA *(rw, sync, no_wdelay, fsid=0) 을 /etc/exports로 지정하십시오.</p>	
<p>NFS 디먼 시작: /etc/init.d/nfs start</p>	<p>내보내진 파일 시스템 /MQHA 마운트:</p> <pre>mount -t nfs4 -o hard,intr Server1:/ /MQHA</pre>
<p>Server1의 큐 관리자 구성 세부사항을 복사하십시오.</p> <pre>dspmqlnf -o command QM1</pre> <p>그리고 그 결과를 클립보드에 복사하십시오.</p> <pre>addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1</pre>	<p>큐 관리자 구성 명령을 Server2에 붙여넣으십시오.</p> <pre>addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1</pre>

² '*'는 읽기/쓰기를 위해 이 마운트 /MQHA에 도달할 수 있는 모든 시스템을 허용합니다. 프로덕션 시스템에 대한 액세스를 제한합니다.

표 30. Linux 의 예시적인 멀티 인스턴스 큐 관리자 구성 (계속)	
Server1	Server2
<p>큐 관리자 인스턴스를 -x 매개변수를 사용하여 어떤 순서로든 시작하십시오. strmqm -x QM1</p> <p>큐 관리자 인스턴스를 시작하는 데 사용되는 명령은 addmqinf 명령과 동일한 IBM WebSphere MQ 설치에서 실행해야 합니다. 다른 설치에서 큐 관리자를 시작하고 중지하려면 먼저 setmqm 명령을 사용하여 큐 관리자와 연관된 설치를 설정해야 합니다. 자세한 정보는 setmqm을 참조하십시오.</p>	

Linux에서 다중 인스턴스 큐 관리자 확인

샘플 프로그램 **amqsgshac**, **amqspshac** 및 **amqsmhac** 를 사용하여 다중 인스턴스 큐 관리자 구성을 확인하십시오. 이 주제에서는 Linux Red Hat Enterprise 5에서 다중 인스턴스 큐 관리자 구성을 확인하는 구성 예를 제공합니다.

고가용성 샘플 프로그램은 자동 클라이언트 다시 연결을 사용합니다. 연결된 큐 관리자가 실패하면 클라이언트가 같은 큐 관리자 그룹에 있는 큐 관리자에 다시 연결하려고 시도합니다. 샘플 고가용성 샘플 프로그램에 대한 설명에서는 단순성을 위해 단일 인스턴스 큐 관리자를 사용한 클라이언트 다시 연결을 보여줍니다. 동일한 샘플을 다중 인스턴스 큐 관리자와 사용하여 다중 인스턴스 큐 관리자 구성을 확인할 수 있습니다.

이 예에서는 350 페이지의 『Linux에서 다중 인스턴스 큐 관리자 작성』에 설명된 다중 인스턴스 구성을 사용합니다. 이 구성을 사용하여 다중 인스턴스 큐 관리자가 대기 인스턴스로 전환하는지 확인하십시오. **endmqm** 명령을 사용하여 큐 관리자를 중지하고 전환 옵션인 **-s**를 사용하십시오. 클라이언트 프로그램이 새 큐 관리자 인스턴스에 다시 연결하여 잠깐의 지연 후 새 인스턴스에 대한 작업을 계속합니다.

예제에서 클라이언트는 Windows XP 서비스 팩 2 시스템에서 실행하고 있습니다. 이 시스템은 다중 인스턴스 큐 관리자를 실행하는 두 VMware Linux 서버를 호스팅합니다.

WebSphere MQ Explorer를 사용한 장애 복구 확인

샘플 애플리케이션을 사용하여 장애 복구를 확인하기 전에 각 서버에서 WebSphere MQ Explorer를 실행하십시오. **리모트 큐 관리자 추가 > 멀티 인스턴스 큐 관리자에 직접 연결** 마법사를 사용하여 각 탐색기에 큐 관리자 인스턴스 모두를 추가하십시오. 대기를 허용하여 두 인스턴스 모두 실행되고 있는지 확인하십시오. 활성 인스턴스로 VMware 이미지를 실행하는 창을 닫아서 가상으로 서버의 전원을 끄거나 활성 인스턴스를 중지하여 대기 인스턴스로의 전환을 허용하십시오.

참고: 서버의 전원을 끄는 경우 /MQHA를 호스트하고 있는 서버가 아닌지 확인하십시오.

참고: 대기 인스턴스로 전환 허용 옵션은 큐 관리자 중지 대화 상자에서 사용할 수 없습니다. 이 옵션은 큐 관리자가 단일 인스턴스 큐 관리자로 실행되고 있기 때문에 누락되었습니다. **대기 인스턴스 허용** 옵션 없이 큐 관리자를 시작했습니다. 큐 관리자 중지 요청이 거부되면 **자세한 내용** 창을 보십시오. 실행 중인 대기 인스턴스가 없기 때문일 수 있습니다.

샘플 프로그램을 사용하여 장애 복구 확인

활성 인스턴스를 실행할 서버 선택

MQHA 디렉토리 또는 파일 시스템을 호스트할 서버 중 하나를 선택했을 수 있습니다. 활성 서버를 실행 중인 VMware 창을 닫아 장애 복구를 테스트하려는 경우 MQHA를 호스트하고 있는 서버가 아닌지 확인하십시오.

활성 큐 관리자 인스턴스를 실행하는 서버에서

참고: MCAUSER 를 mqm로 설정하여 SVRCONN 채널을 실행하면 예제의 구성 단계 수를 줄일 수 있습니다. 다른 사용자 ID를 선택하고 시스템이 예에서 사용된 것과 다르게 설정된 경우 액세스 권한 문제점을 경험할 수 있습니다. 노출된 시스템에서 mqm을 MCAUSER로 사용하지 마십시오. 상당한 보안성 저하를 초래할 수 있습니다.

1. **ipaddr1** 및 **ipaddr2**를 수정하고 다음 명령을 /MQHA/hasamples.tst.에 저장하십시오.

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER('mqm') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME('ipaddr1(1414),ipaddr2
```

```
(1414)' ) QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
START LISTENER(LISTENER.TCP)
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

- 경로 /MQHA로 터미널 창을 열고 다음 명령을 실행하십시오.

```
runmqsc -m QM1 < hasamples.tst
```

- runmqsc** 명령의 출력을 검사하여 리스너가 실행 중이고 큐 관리자 제어를 가지는지 확인하십시오.

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

또는 WebSphere MQ 탐색기를 사용하여, TCP/IP 리스너가 실행 중이고 Control = Queue Manager가 있습니다.

클라이언트에서

- 클라이언트 연결 테이블 AMQCLCHL.TAB를 서버의 /MQHA/qmgrs/QM1.000/@ipcc에서 클라이언트의 C:\ 로 복사하십시오.
- 경로 C:\로 명령 프롬프트를 열고 클라이언트 채널 정의 테이블(CCDT)을 가리키도록 환경 변수 MQCHLLIB를 설정하십시오.

```
SET MQCHLLIB=C:\
```

- 명령 프롬프트에 다음 명령을 입력하십시오.

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

활성 큐 관리자 인스턴스를 실행하는 서버에서

- 다음 중 하나입니다.
 - 활성 서버 인스턴스로 VMware 이미지를 실행하는 창을 닫으십시오.
 - WebSphere MQ Explorer를 사용하여 활성 큐 관리자 인스턴스를 중지하고 대기 인스턴스로 전환할 수 있게 하고 재연결 가능한 클라이언트가 재연결하도록 지시하십시오.
- 세 클라이언트는 결국 연결이 끊어진 것을 감지한 후 다시 연결합니다. 이 구성에서 서버 창을 닫은 경우 세 개의 연결이 모두 재설정되는 데 7분 정도 걸립니다. 일부 연결은 다른 연결 전에 정상적으로 재설정됩니다.

결과

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message <Message 6>
message <Message 7>
message <Message 8>
message <Message 9>
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start
```

```
17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsgghac TARGET QM1
Sample AMQSGHAC start
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message <Message 6>
message <Message 7>
message <Message 8>
message <Message 9>
```

다중 인스턴스 큐 관리자 삭제

다중 인스턴스 큐 관리자를 완전하게 삭제하려면 **dltmqm** 명령을 사용하여 큐 관리자를 삭제한 다음 **rmvmqinf** 또는 **dltmqm** 명령을 사용하여 다른 서버에서 인스턴스를 제거해야 합니다.

dltmqm 명령을 실행하여 다른 서버에서 정의된 인스턴스가 있는 큐 관리자를 이 큐 관리자가 정의되는 서버에서 삭제하십시오. 큐 관리자를 작성한 서버와 같은 서버에서 **dltmqm** 명령을 실행할 필요는 없습니다. 그런 다음 큐 관리자의 정의가 있는 다른 모든 서버에서 **rmvmqinf** 또는 **dltmqm** 명령을 실행하십시오.

중지된 큐 관리자만 삭제할 수 있습니다. 큐 관리자를 삭제할 때 실행 중인 인스턴스가 없고 큐 관리자가 단일 인스턴스 큐 관리자 또는 다중 인스턴스 큐 관리자가 아니며 단순히 원격 공유에 해당 큐 관리자 데이터 및 로그가 있는 큐 관리자입니다. 큐 관리자를 삭제하면 해당 큐 관리자 데이터 및 로그가 삭제되고 **dltmqm** 명령을 발행한 서버의 **mq5.ini** 파일에서 큐 관리자 스탠자가 제거됩니다. 큐 관리자를 삭제할 때 큐 관리자 데이터 및 로그가 포함된 네트워크 공유에 대한 액세스 권한이 있어야 합니다.

이전에 큐 관리자의 인스턴스를 작성한 다른 서버의 **mq5.ini** 파일에도 여러 항목이 있습니다. 각 서버에 순서대로 방문하고 **rmvmqinf Queue manager stanza name** 명령을 실행하여 큐 관리자 스탠자를 제거해야 합니다.

UNIX and Linux 시스템에서 네트워크 스토리지에 공용 **mq5.ini** 파일을 배치했고 각 서버에서 **AMQ_MQS_INI_LOCATION** 환경 변수를 설정하여 모든 서버에서 이 파일을 참조한 경우 업데이트할 한 개의 **mq5.ini** 파일만 있기 때문에 해당 서버 중 한 서버에서만 큐 관리자를 삭제해야 합니다.

예

첫 번째 서버
dltmqm QM1

인스턴스가 정의되는 다른 서버
rmvmqinf QM1, or
dltmqm QM1

다중 인스턴스 큐 관리자 시작 및 중지

단일 인스턴스 또는 다중 인스턴스 큐 관리자로 구성된 큐 관리자를 시작 및 중지합니다.

한 쌍의 서버에서 다중 인스턴스 큐 관리자를 정의한 경우 두 서버 중 하나에서 단일 인스턴스 큐 관리자 또는 다중 인스턴스 큐 관리자로 이 큐 관리자를 실행할 수 있습니다.

다중 인스턴스 큐 관리자를 실행하려면 **strmqm -x QM1** 명령을 사용하여 서버 중 하나에서 큐 관리자를 시작하십시오. **-x** 옵션은 인스턴스가 장애를 복구하도록 합니다. 이는 활성 인스턴스가 됩니다. 같은 **strmqm -x QM1** 명령을 사용하여 다른 서버에서 대기 인스턴스를 시작하십시오. **-x** 옵션은 인스턴스가 대기로 시작할 수 있게 합니다.

큐 관리자는 모든 요청을 처리하는 한 개의 활성 인스턴스와 활성 인스턴스가 실패하면 인수할 준비가 되어 있는 대기 인스턴스로 실행되고 있습니다. 활성 인스턴스에는 큐 관리자 데이터 및 로그에 대한 독점 액세스 권한이 부여됩니다. 대기는 큐 관리자 데이터 및 로그에 대한 독점 액세스 권한이 부여될 때까지 대기합니다. 대기에 독점 액세스 권한이 부여되는 경우 활성 인스턴스가 됩니다.

활성 인스턴스에서 **endmqm -s** 명령을 발행하여 수동으로 제어를 대기 인스턴스로 전환할 수도 있습니다.

endmqm -s 명령은 대기를 종료하지 않고 활성 인스턴스를 종료합니다. 큐 관리자 데이터 및 로그에 대한 독점 액세스 잠금이 해제되고 이를 대기가 인수합니다.

여러 서버에서 다중 인스턴스로 구성된 큐 관리자를 단일 인스턴스 큐 관리자로 시작하고 중지할 수도 있습니다. **strmqm** 명령에 **-x** 옵션을 사용하지 않고 큐 관리자를 시작하면 다른 시스템에서 구성된 큐 관리자의 인스턴스가 대기 인스턴스로 시작되지 않습니다. 다른 인스턴스를 시작하려고 시도하면 큐 관리자 인스턴스가 대기로 실행되는 것이 허용되지 않는다는 응답을 수신합니다.

-s 옵션 없이 **endmqm** 명령을 사용하여 다중 인스턴스 큐 관리자의 활성 인스턴스를 중지하는 경우 활성 및 대기 인스턴스 둘 다 중지됩니다. **-x** 옵션과 함께 **endmqm** 명령을 사용하여 대기 인스턴스를 중지하면 대기 인스턴스 상태가 중지되고 활성 인스턴스는 실행을 계속합니다. 대기에서 **-x** 옵션 없이 **endmqm**을 발행할 수 없습니다.

두 개의 큐 관리자 인스턴스만 동시에 실행될 수 있으며 이 중 하나는 활성 인스턴스이고 다른 하나는 대기 인스턴스입니다. 두 개의 인스턴스를 동시에 시작한 경우 WebSphere MQ는 어떤 인스턴스가 활성 인스턴스가 되는지에 대해 제어를 가지고 있지 않습니다. 이는 네트워크 파일 시스템이 판별합니다. 큐 관리자 데이터에 대한 독점 액세스를 획득하는 첫 번째 인스턴스가 활성 인스턴스가 됩니다.

참고: 실패한 큐 관리자를 재시작하기 전에 큐 관리자의 해당 인스턴스로부터 애플리케이션의 연결을 끊어야 합니다. 이를 수행하지 않으면 큐 관리자가 올바르게 재시작하지 않을 수 있습니다.

공유 파일 시스템

다중 인스턴스 큐 관리자는 네트워크 파일 시스템을 사용하여 큐 관리자 인스턴스를 관리합니다.

다중 인스턴스 큐 관리자는 파일 시스템 잠금과 공유 큐 관리자 데이터 및 로그의 결합을 사용하여 장애 복구를 자동화합니다. 큐 관리자의 한 개의 인스턴스만 공유 큐 관리자 데이터 및 로그에 대한 독점 액세스 권한을 가질 수 있습니다. 액세스하게 되면 활동 인스턴스가 됩니다. 독점 액세스 권한을 가져오는 데 성공하지 못한 다른 인스턴스는 큐 관리자 데이터 및 로그가 사용 가능하게 될 때까지 대기 인스턴스로 대기합니다.

네트워크 파일 시스템은 활성 큐 관리자 인스턴스에 대해 보유하고 있는 잠금 해제를 담당합니다. 활성 인스턴스가 어떤 식으로든 실패하면 네트워크 파일 시스템이 활성 인스턴스에 대해 보유하고 있는 잠금을 해제합니다. 독점 잠금이 해제되자마자 잠금 대기 중인 대기 큐 관리자가 이를 획득하려고 시도합니다. 획득에 성공하면 활성 인스턴스가 되고 공유 큐 시스템에 있는 큐 관리자 데이터 및 로그에 대한 독점 액세스 권한을 가지게 됩니다. 그러면 시작을 계속합니다.

관련 주제 [파일 시스템 지원 계획](#)에서는 파일 시스템이 다중 인스턴스 큐 관리자를 지원하도록 설정하고 확인하는 방법을 설명합니다.

다중 인스턴스 큐 관리자는 파일 시스템에서의 실패에 대해 보호하지 않습니다. 데이터를 보호하기 위한 다양한 방법이 있습니다.

- 중복 디스크 어레이(RAID)와 같은 신뢰할 수 있는 스토리지에 투자하고 이를 네트워크 복원성이 있는 네트워크 파일 시스템에 포함시키십시오.
- WebSphere MQ 선행 로그를 대체 매체에 백업하고 1차 로그 매체가 실패하면 대체 매체에 있는 로그를 사용하여 복구하십시오. 백업 큐 관리자를 사용하여 이 프로세스를 관리할 수 있습니다.

다중 큐 관리자 인스턴스

다중 인스턴스 큐 관리자는 실패 후 큐 관리자 가용성을 복원하는 데 대기 큐 관리자 인스턴스를 사용하기 때문에 복원성이 있습니다.

큐 관리자 인스턴스 복제는 큐 관리자 프로세스의 가용성을 향상시키기 위한 매우 효과적인 방법입니다. 설명만을 위해 단순 가용성 모델을 사용할 때 한 큐 관리자 인스턴스의 신뢰도가 99%(1년 동안, 누적 정지 시간은 3.65 일임)인 경우 큐 관리자의 다른 인스턴스를 추가하면 가용성이 99.99%로 증가합니다(1년 동안, 누적 정지 시간은 약 1시간임).

이는 가용성의 실제 숫자 추정치를 제시하기에는 너무 단순한 모델입니다. 가용성을 현실적으로 모델링하려면 평균 실패 간격(MTBF) 및 평균 복구 시간(MTTR), 실패와 복구 시간 사이의 시간 확률 분배에 대한 통계를 수집해야 합니다.

용어 다중 인스턴스 큐 관리자는 큐 관리자 데이터 및 로그를 공유하는 큐 관리자의 활성 인스턴스와 대기 인스턴스의 결합을 나타냅니다. 다중 인스턴스 큐 관리자는 한 서버에 한 개의 활성인 큐 관리자 인스턴스를 두고 활성 인스턴스가 실패할 경우 자동으로 인계받을 준비가 되어 있는 대기 상태의 다른 큐 관리자 인스턴스를 다른 서버에 둬으로써 큐 관리자 프로세스의 실패에 대해 보호합니다.

장애 복구 또는 전환

대기 큐 관리자 인스턴스는 요청 시(전환) 또는 활성 인스턴스가 실패하는 경우(장애 복구) 활성 인스턴스로부터 인수합니다.

- 전환은 활성 큐 관리자 인스턴스로 발행되는 **endmqm -s** 명령에 대한 응답으로 대기 인스턴스가 시작될 때 발생합니다. **endmqm** 매개변수 **-c**, **-i** 또는 **-p**를 지정하여 큐 관리자가 갑자기 중지되는 방법을 제어할 수 있습니다.

참고: 전환은 대기 큐 관리자 인스턴스가 이미 시작된 경우에만 발생합니다. **endmqm -s** 명령은 활성 큐 관리자 잠금을 해제하고 전환을 허용합니다. 이 명령은 대기 큐 관리자 인스턴스를 시작하지 않습니다.

- 장애 복구는 활성 인스턴스가 보유하는 큐 관리자 데이터에 대한 잠금이 해제되면 발생합니다. 이 인스턴스가 예상치 못하게 중지되는 것으로 표시되었기 때문입니다(즉 **endmqm** 명령이 발행되지 않음)

대기 인스턴스가 활성 인스턴스로 인수하면 큐 관리자 오류 로그에 메시지를 기록합니다.

다시 연결 가능한 클라이언트는 큐 관리자가 실패하거나 전환할 때 자동으로 다시 연결됩니다. 클라이언트 재연결을 요청하기 위해 **endmqm** 명령에 **-r** 플래그를 포함시킬 필요가 없습니다. 자동 클라이언트 재연결이 Java용 WebSphere MQ 클래스에서 지원되지 않습니다.

실패한 인스턴스를 재시작할 수 없으면 장애 복구가 발생하고 대기 인스턴스가 활성화된 경우에도 실패한 인스턴스에서 실패한 인스턴스에 로컬로 연결된 애플리케이션의 연결이 끊어졌는지 확인하십시오. 실패한 인스턴스를 재시작할 수 있도록 로컬로 연결된 애플리케이션이 종료되거나 실패한 큐 관리자 인스턴스로부터 연결이 끊깁니다. 공유 바인딩을 사용하고(기본 설정임) 실패한 인스턴스에 대한 연결을 계속 보유하는 로컬로 연결된 애플리케이션은 인스턴스가 재시작되는 것을 막습니다. 로컬로 연결된 애플리케이션을 종료하거나 로컬 큐 관리자 인스턴스가 실패할 경우 연결을 끊는 것이 불가능한 경우 격리 바인딩 사용을 고려해 보십시오. 격리 바인딩을 사용하는 로컬로 연결된 애플리케이션은 연결을 끊지 않더라도 로컬 큐 관리자가 재시작되는 것을 막지 않습니다.

채널 및 클라이언트 다시 연결

채널 및 클라이언트 다시 연결은 대기 큐 관리자 인스턴스가 활성이 된 후의 메시지 처리 복원에서 필수적인 부분입니다.

다중 인스턴스 큐 관리자 인스턴스는 여러 네트워크 주소로 서버에서 설치됩니다. 모든 큐 관리자 인스턴스에 대한 연결 정보를 사용하여 IBM WebSphere MQ 채널 및 클라이언트를 구성해야 합니다. 대기가 인수를 하면 클라이언트 및 채널이 새 네트워크 주소에서 새로 활성이 된 큐 관리자 인스턴스에 자동으로 다시 연결됩니다. 자동 클라이언트 재연결이 Java용 WebSphere MQ 클래스에서 지원되지 않습니다.

이 설계는 HA-CMP 작업과 같은 고가용성 환경 방식과는 다릅니다. HA-CMP는 클러스터에 대한 가상 IP 주소를 제공하고 이 주소를 활성 서버에 전송합니다. WebSphere MQ 재연결은 IP 주소를 변경하거나 리라우트하지 않습니다. 이는 채널 정의 및 클라이언트 연결에서 정의한 네트워크 주소를 사용한 다시 연결을 통해 작동합니다. 관리자로서 채널 정의 및 다중 인스턴스 큐 관리자의 모든 인스턴스에 대한 연결에서 네트워크 주소를 정의해야 합니다. 다중 인스턴스 큐 관리자에 대해 네트워크 주소를 구성하는 최고의 방법은 연결에 따라 다릅니다.

큐 관리자 채널

채널의 CONNAME 속성은 연결 이름의 심표로 구분된 목록입니다 (예: CONNAME('127.0.0.1(1234), 192.0.2.0(4321)')). 연결은 연결이 성공적으로 설정될 때까지 연결에서 지정된 순서대로 시도됩니다. 연결에 실패하면 채널이 다시 연결을 시도합니다.

클러스터 채널

일반적으로 다중 인스턴스 큐 관리자가 클러스터에서 작동하게 하기 위해서 필요한 추가 구성은 없습니다.

큐 관리자가 저장소 큐 관리자에 연결하는 경우 저장소는 큐 관리자의 네트워크 주소를 검색합니다. 큐 관리자에서 CLUSRCVR 채널의 CONNAME을 참조합니다. TCP/IP에서 사용자가 CONNAME을 생략하거나 공백으로 구성한 경우 큐 관리자는 자동으로 이 옵션을 설정합니다. 대기 인스턴스가 인수하면 해당 IP 주소가 이전 활성 인스턴스의 IP 주소를 CONNAME으로 바꿉니다.

필요한 경우 큐 관리자 인스턴스의 네트워크 주소 목록으로 CONNAME을 수동으로 구성할 수 있습니다.

클라이언트 연결

클라이언트 연결에서는 연결 목록 또는 큐 관리자 그룹을 사용하여 대체 연결을 선택할 수 있습니다. 멀티 인스턴스 큐 관리자의 클라이언트 재연결에 대한 자세한 정보는 282 페이지의 『자동 클라이언트 다시 연결』의 내용을 참조하십시오. 클라이언트는 WebSphere MQ 버전 7.0.1 클라이언트 라이브러리 이상으로 실행하도록 컴파일되어야 합니다. 최소한 버전 7.0.1 큐 관리자로 연결되어야 합니다.

장애 복구가 발생하면 다시 연결에 시간이 걸립니다. 대기 큐 관리자는 그 시동을 완료해야 합니다. 실패한 큐 관리자와 연결된 클라이언트는 연결 실패를 감지하고 새 클라이언트 연결을 시작해야 합니다. 새 클라이언트 연결이 새로 활성이 된 대기 큐 관리자를 선택하면 클라이언트가 동일한 큐 관리자에 다시 연결됩니다.

클라이언트가 다시 연결 중 MQI 호출 중간에 있게 되면 호출 완료까지 길어진 대기를 허용해야 합니다.

메시지 채널에서 배치 전송 중에 실패가 발생하면 배치가 롤백되고 재시작됩니다.

전환이 장애 복구보다 빠르며 큐 관리자의 한 인스턴스를 중지하고 다른 인스턴스를 시작하는 시간만 소요됩니다. 재실행할 로그 레코드 수가 적은 큐 관리자의 경우 전환은 길어봐야 몇 초 정도 걸릴 수 있습니다. 장애 복구에 걸리는 시간을 추정하려면 실패가 감지되는 데 걸리는 시간을 추가해야 합니다. 감지에는 길어봐야 10초 정도 걸리지만 네트워크 및 파일 시스템에 따라 몇 분이 될 수도 있습니다.

애플리케이션 복구

애플리케이션 복구는 장애 복구 후 자동화된 애플리케이션 연속 처리입니다. 장애 복구 후 애플리케이션 복구에는 주의 깊은 설계가 필요합니다. 일부 애플리케이션은 장애 복구가 발생했음을 인지해야 합니다.

애플리케이션 복구의 목적은 애플리케이션이 잠깐 동안의 지연 후에 처리를 계속하게 하는 것입니다. 새 처리를 계속하기 전에 애플리케이션은 백아웃하고 실패 중 처리하고 있었던 작업 단위를 다시 제출해야 합니다.

응용프로그램 복구에 대한 문제점은 WebSphere MQ MQI 클라이언트와 큐 관리자 사이에 공유되고 큐 관리자에 저장되는 컨텍스트를 상실하고 있습니다. WebSphere MQ MQI 클라이언트는 컨텍스트의 대부분을 복원하지만 컨텍스트에서 정확히 복원될 수 없는 부분이 있습니다. 다음 절에서는 애플리케이션 복구의 일부 특성 및 그러한 특성이 다중 인스턴스 큐 관리자에 연결된 애플리케이션의 복구에 영향을 주는 방식에 대해 설명합니다.

트랜잭션 가능한 메시지

메시지를 전달하는 관점에서 볼 때 장애 복구는 WebSphere MQ 메시지의 지속 특성을 변경하지 않습니다. 메시지가 지속적이고 작업 단위 내에서 올바르게 관리되는 경우 장애 복구 중 메시지가 손실되지 않습니다.

트랜잭션 처리의 관점에서 트랜잭션은 백아웃되거나 장애 복구 후 커밋됩니다.

커밋되지 않은 트랜잭션은 롤백됩니다. 장애 복구 후 재연결 가능한 애플리케이션은 트랜잭션이 실패했음을 표시하는 MQRC_BACKED_OUT 이유 코드를 수신하며 MQBACK을 실행하여 트랜잭션을 롤백해야 합니다. 그런 다음 트랜잭션을 다시 재시작해야 합니다.

커밋된 트랜잭션은 2단계 커밋 중 두 번째 단계에 도달한 트랜잭션이거나 MQCMIT를 시작한 단일 단계(메시지만 포함) 트랜잭션입니다.

큐 관리자가 트랜잭션 코디네이터이고 MQCMIT가 실패 전에 두 번째 단계 커밋의 두 번째 단계를 시작한 경우, 트랜잭션이 성공적으로 완료됩니다. 완료는 큐 관리자의 제어를 받으며 큐 관리자가 다시 실행될 때 계속됩니다. 재연결 가능한 애플리케이션에서 MQCMIT 호출은 정상적으로 완료됩니다.

메시지만 포함되는 1단계 커밋에서 커밋 처리를 시작한 트랜잭션은 일단 다시 실행되면 큐 관리자의 제어 하에 정상적으로 완료됩니다. 재연결 가능한 애플리케이션에서 MQCMIT는 정상적으로 완료됩니다.

다시 연결 가능한 클라이언트는 큐 관리자의 제어 하에 1단계 트랜잭션을 트랜잭션 코디네이터로 사용할 수 있습니다. 확장된 트랜잭션 클라이언트는 다시 연결을 지원하지 않습니다. 트랜잭션 가능한 클라이언트가 연결할 때 다시 연결이 요청되면 연결에 성공하지만 다시 연결되는 기능은 없습니다. 이 연결은 다시 연결 불가능한 것처럼 작동합니다.

애플리케이션 재시작 또는 계속

장애 복구는 애플리케이션을 인터럽트합니다. 실패 후 애플리케이션은 처음부터 재시작하거나 인터럽트 후 처리를 계속할 수 있습니다. 후자를 자동 클라이언트 다시 연결이라고 합니다. 자동 클라이언트 재연결이 Java용 WebSphere MQ 클래스에서 지원되지 않습니다.

WebSphere MQ MQI 클라이언트 애플리케이션을 사용하여 클라이언트를 자동으로 다시 연결하도록 연결 옵션을 설정할 수 있습니다. 이들 옵션은 MQCNO_RECONNECT 또는 MQCNO_RECONNECT_Q_MGR입니다. 옵션을 설정하지 않으면 클라이언트가 자동으로 다시 연결을 시도하지 않으며 큐 관리자 실패는 MQRC_CONNECTION_BROKEN을 클라이언트에 리턴합니다. 새 MQCONN 또는 MQCONNX 호출을 실행하여 새 연결을 시작하도록 클라이언트를 설계할 수 있습니다.

서버 프로그램은 재시작되어야 하고 큐 관리자 또는 서버가 실패했을 때 처리하고 있었던 시점에서 큐 관리자에 의해 자동으로 다시 연결되어야 합니다. WebSphere MQ 서버 프로그램은 일반적으로 멀티 인스턴스 큐 관리자 인스턴스가 실패하면 대기 큐 관리자 인스턴스에서 재시작되지 않습니다.

다음 두 가지 방법으로 WebSphere MQ 서버 프로그램이 대기 서버에서 재시작하도록 자동화할 수 있습니다.

1. 서버 애플리케이션을 큐 관리자 서비스로 패키징하십시오. 서버 애플리케이션은 대기 큐 관리자를 재시작할 때 재시작됩니다.
2. 대기 큐 관리자 인스턴스가 시작할 때 작성한 장애 복구 로그 메시지 등으로 트리거되는 자체 장애 복구 논리를 작성하십시오. 그러면 애플리케이션 인스턴스가 시작 후 MQCONN 또는 MQCONNX를 호출하여 큐 관리자에 대한 연결을 작성해야 합니다.

장애 복구 감지

일부 애플리케이션은 장애 복구를 인지해야 하고 나머지 애플리케이션을 인지하지 않아도 됩니다. 다음 두 가지 예를 고려하십시오.

1. 메시징 채널을 통해 메시지를 가져오거나 수신하는 메시징 애플리케이션에서는 일반적으로 채널의 다른 측 큐 관리자가 실행되도록 요구하지 않습니다. 이 애플리케이션은 채널의 다른 측에 있는 큐 관리자가 대기 인스턴스에서 재시작되는 경우 영향을 받을 가능성이 거의 없습니다.
2. A WebSphere MQ MQI client application processes persistent message input from one queue and puts persistent message responses onto another queue as part of a single unit of work: if it handles an MQRC_BACKED_OUT reason code from MQPUT, MQGET or MQCMIT within sync point by issuing MQBACK and restarting the unit of work, then no messages are lost. 또한 이 애플리케이션은 연결 실패를 처리하기 위한 특수 처리를 수행하지 않아도 됩니다.

그러나 두 번째 예제에서 애플리케이션이 MQGET 옵션인 MQGMO_MSG_UNDER_CURSOR를 사용하여 처리할 메시지를 선택하기 위해 큐를 찾아보는 것으로 가정합니다. 다시 연결은 찾아보기 커서를 재설정하고 MQGET 호출은 올바른 메시지를 리턴하지 않습니다. 이 예에서 애플리케이션은 장애 복구가 발생했음을 인지해야 합니다. 또한 커서 아래 메시지에 대해 다른 MQGET을 발행하기 전에 이 애플리케이션은 이 브라우저 커서를 복원해야 합니다.

브라우저 커서 손실은 애플리케이션 컨텍스트가 다음 다시 연결을 변경하는 방식의 한 가지 예입니다. 다른 경우는 359 페이지의 『자동으로 다시 연결된 클라이언트의 복구』에 설명되어 있습니다.

장애 복구에 따라 WebSphere MQ MQI 클라이언트 애플리케이션에 대해 세 가지 대체 설계 패턴이 제공됩니다. 이들 중 하나만 장애 복구를 감지하지 않아도 됩니다.

다시 연결 안함

이 패턴에서는 연결이 끊긴 경우 애플리케이션이 현재 연결에 대한 모든 처리를 중지합니다. 처리를 계속하려면 이 애플리케이션은 큐 관리자와의 새 연결을 설정해야 합니다. 이 애플리케이션은 이 애플리케이션이 새 연결에 대한 처리를 계속하기 위해 필요로 하는 상태 정보 전송에 전적으로 책임이 있습니다. 연결이 끊긴 후 큐 관리자와 다시 연결되는 기존 클라이언트 애플리케이션은 이 방식으로 작성됩니다.

클라이언트는 연결이 끊어진 후 그 다음 MQI 호출로부터 MQRC_CONNECTION_BROKEN 또는 MQRC_Q_MGR_NOT_AVAILABLE와 같은 이유 코드를 수신합니다. 애플리케이션은 모든 해당 WebSphere MQ 상태 정보(예: 큐 핸들)를 제거하고 새 MQCONN 또는 MQCONNX 호출을 발행하여 새 연결을 설정한 후 처리하는 데 필요한 WebSphere MQ 오브젝트를 다시 엽니다.

기본 MQI 작동은 큐 관리자와의 연결이 끊기면 큐 관리자 연결 핸들이 사용할 수 없게 되는 것입니다. 기본값은 장애 복구 후 애플리케이션 재연결을 방지하기 위해 MQCONNX 에서 MQCNO_RECONNECT_DISABLED 옵션을 설정하는 것과 같습니다.

장애 복구 허용

장애 복구의 영향을 받지 않도록 애플리케이션을 작성하십시오. 주의하여 오류를 핸들링하는 것으로 장애 복구 처리에 충분한 경우가 있습니다.

다시 연결 인지

MQCBT_EVENT_HANDLER 이벤트 핸들러를 큐 관리자에 등록하십시오. 클라이언트가 서버에 다시 연결을 시도하기 시작하면 MQRC_RECONNECTING 및 성공적인 재연결 후에 MQRC_RECONNECTED 와 함께 이벤트 핸들러가 게시됩니다. 그런 다음 클라이언트 애플리케이션이 처리를 계속할 수 있도록 예상 가능한 상태를 재설정하는 루틴을 시작할 수 있습니다.

자동으로 다시 연결된 클라이언트의 복구

장애 복구는 예상치 못한 이벤트이고 자동으로 다시 연결된 클라이언트가 설계한대로 작동하도록 다시 연결의 결과가 예상 가능해야 합니다.

예상치 못한 실패를 예상 가능하고 신뢰할 수 있는 복구로 전환하는 주요 요소는 트랜잭션의 사용입니다.

이전 절에서 예제인 358 페이지의 『2』는 MQGET 및 MQPUT를 조정하기 위해 로컬 트랜잭션을 사용하는 WebSphere MQ MQI 클라이언트에 제공되었습니다. 클라이언트는 MQRC_BACKED_OUT 오류에 대한 응답으로 MQCMIT 또는 MQBACK 호출을 발행한 후 백아웃된 트랜잭션을 다시 제출합니다. 큐 관리자 실패로 인해 트랜잭션은 백아웃되고 클라이언트 애플리케이션의 작동은 트랜잭션 및 메시지가 손실되지 않도록 합니다.

모든 프로그램 상태가 트랜잭션의 일부로 관리되는 것은 아니기 때문에 다시 연결의 결과가 더 이해하기 어려워 집니다. 클라이언트 애플리케이션이 큐 관리자 장애 복구에서 존속하도록 설계하려면 재연결이 WebSphere MQ MQI 클라이언트의 상태를 어떻게 변경하는지를 알아야 합니다.

특수 장애 복구 코드 없이 다른 오류와 같은 논리를 사용하여 다시 연결 오류를 핸들링하는 애플리케이션을 설계하도록 결정할 수 있습니다. 또는 재연결에 특별한 오류 처리가 필요한 것을 인식하고, 장애 복구를 처리하기 위한 루틴을 실행하기 위해 WebSphere MQ에 이벤트 핸들러를 등록하도록 선택할 수 있습니다. 이 루틴은 다시 연결 처리 그 자체를 핸들링하거나 처리를 재개할 때 복구 처리를 수행해야 하는 기본 프로그램 스프레드를 표시하는 플래그를 설정할 수 있습니다.

WebSphere MQ MQI 클라이언트 환경은 장애 복구 자체를 인식합니다. 그리고 WebSphere MQ 상태를 복원하기 위해 클라이언트 애플리케이션 대신 추가 MQI 호출을 발행하고 클라이언트에 일부 상태 정보를 저장하여 재연결된 후 가능한 많은 컨텍스트를 복원합니다. 예를 들어, 실패 지점에서 열린 오브젝트에 대한 핸들이 복원되고 임시 동적 큐가 동일한 이름으로 열립니다. 그러나 불가피한 변경사항이 있고 설계가 이러한 변경사항을 처리해야 합니다. 이러한 변경사항은 다섯 가지 종류로 분류할 수 있습니다.

1. 일치하는 새 컨텍스트 상태가 애플리케이션 프로그램에 의해 복원될 때까지 새 오류 또는 이전에 진단되지 않은 오류가 MQI 호출에서 리턴됩니다.

재연결 전에 컨텍스트를 저장한 후 컨텍스트를 전달하려고 할 때 새 오류를 수신하는 예제는 리턴 코드 MQRC_CONTEXT_NOT_AVAILABLE 입니다. 이 컨텍스트는 다시 연결 후에는 복원할 수 없습니다. 보안 컨텍스트는 권한이 없는 클라이언트 프로그램에 전달되지 않기 때문입니다. 전달될 경우 악성 애플리케이션 프로그램이 보안 컨텍스트를 확보할 수 있습니다.

일반적으로 애플리케이션은 주의하여 설계된 방식으로 공통 오류 및 예상 가능한 오류를 핸들링하고 비공통 오류는 일반 오류 핸들러로 격하시킵니다. 오류 핸들러는 WebSphere MQ와 연결이 끊어졌다가 다시 연결되거나 프로그램을 완전히 중지시킬 수도 있습니다. 지속성을 개선하려면 일부 오류를 다른 방법으로 처리해야 하는 수가 있습니다.

2. 비지속 메시지는 손실될 수 있습니다.
3. 트랜잭션은 롤백됩니다.
4. 동기점 외부에서 사용된 MQGET 또는 MQPUT 호출은 가능한 메시지 손실로 인터럽트될 수 있습니다.
5. MQI 호출에서의 연장된 대기 시간으로 인해 발생한 타이밍 오류입니다.

손실된 컨텍스트에 대한 일부 세부사항이 다음 절에 나열됩니다.

- NPMCLASS(HIGH) 옵션을 사용하여 큐에 넣지 않거나 큐 관리자 실패가 종료 시 비지속 메시지 저장 옵션을 인터럽트한 경우 비지속 메시지는 버려집니다.
- 연결이 끊기면 지속 불가능 구독이 손실됩니다. 다시 연결 시 이 구독은 재설정됩니다. 지속 가능 구독의 사용을 고려해 보십시오.

- 가져오기-대기 간격이 다시 계산되고 그 한계가 초과되면 MQRC_NO_MSG_AVAILABLE을 리턴합니다. 마찬가지로 구독 만기가 다시 계산되어 동일한 전체 만기 시간을 제공합니다.
- 큐에서 찾아보기 커서의 위치가 손실되고 이는 일반적으로 첫 번째 메시지 전에 재설정됩니다.
 - MQGMO_BROWSE_MSG_UNDER_CURSOR 또는 MQGMO_MSG_UNDER_CURSOR를 지정하는 MQGET 호출은 이유 코드 MQRC_NO_MSG_AVAILABLE(으)로 실패합니다.
 - 찾아보기에 대해 잠겨진 메시지의 잠금이 해제됩니다.
 - 핸들 범위가 있는 찾아보기 표시 메시지는 표시가 취소되고 다시 찾아볼 수 있습니다.
 - 협업 찾아보기 표시 메시지는 대부분의 경우 표시가 취소됩니다.
- 보안 컨텍스트는 손실됩니다. MQPMO_PASS_ALL_CONTEXT로 메시지 넣기가 MQRC_CONTEXT_NOT_AVAILABLE로 실패함과 같이 저장된 메시지 컨텍스트를 사용하려고 시도합니다.
- 메시지 토큰이 손실됩니다. 메시지 토큰을 사용한 MQGET은 이유 코드 MQRC_NO_MSG_AVAILABLE을 리턴합니다.

참고: 메시지의 일부인 *MsgId* 및 *CorrelId*는 장애 복구 중에 메시지와 함께 보존되므로 *MsgId* 또는 *CorrelId*를 사용하는 MQGET은 예상대로 작동합니다.

- 커밋되지 않은 트랜잭션에서 동기점 아래 큐에 넣은 메시지는 더 이상 사용 가능하지 않습니다.
- 논리 순서로 또는 메시지 그룹으로 메시지를 처리하면 다시 연결 후 리턴 코드 MQRC_RECONNECT_INCOMPATIBLE이 발생합니다.
- MQI 호출은 클라이언트가 요즘 일반적으로 수신하는 보다 일반적인 MQRC_CONNECTION_BROKEN이 아닌 MQRC_RECONNECT_FAILED를 리턴할 수 있습니다.
- 동기점 밖에서 MQPUT 호출이 수행되는 동안 재연결한 경우, WebSphere MQ MQI 클라이언트가 메시지가 큐 관리자에게 성공적으로 전달되었는지 알지 못하면 MQRC_CALL_INTERRUPTED가 리턴됩니다. MQCMIT 중 다시 연결도 비슷하게 작동합니다.
- WebSphere MQ MQI 클라이언트가 큐 관리자에게서 실패 또는 성공을 표시하는 응답을 받지 않은 경우 재연결에 성공한 후에 MQRC_CALL_INTERRUPTED가 리턴됩니다.
 - 동기점 외부 MQPUT 호출을 사용한 지속 메시지의 전달
 - 동기점 외부 MQPUT1 호출을 사용한 지속 메시지 또는 기본값 지속성을 가진 메시지의 전달
 - MQCMIT 호출을 사용하는 트랜잭션의 커밋. 응답은 다시 연결에 성공한 후에만 리턴됩니다.
- 채널이 새 인스턴스(다른 채널일 수도 있음)로 재시작되어 채널 엑시트 상태가 보유됩니다.
- 임시 동적 큐가 열린 다시 연결 가능한 클라이언트 처리의 일부로 임시 동적 큐가 복원됩니다. 임시 동적 큐의 어떤 메시지도 복원되지 않지만 큐가 열려 있거나 큐의 이름을 기억하는 애플리케이션은 처리를 계속할 수 있습니다.

큐를 작성한 애플리케이션 외의 애플리케이션에서 해당 큐가 사용되고 있는 경우 이 큐는 다음에 참조될 때 표시될 정도로 빠르게 복원되지 못할 가능성이 있습니다. 예를 들어, 클라이언트가 응답 대상 큐로 임시 동적 큐를 작성하고 응답 메시지가 채널에 의해 큐에 배치되어야 하는 경우 이 큐는 제 시간에 복구되지 않을 수 있습니다. 이 경우 일반적으로 채널은 응답 대상 메시지를 데드-레터 큐에 넣습니다.

다시 연결 가능한 클라이언트 애플리케이션이 이름에 따라 임시 동적 큐를 여는 경우(다른 애플리케이션이 이미 작성했으므로) 다시 연결이 발생할 때 WebSphere MQ MQI 클라이언트에는 작성에 사용할 모델이 없으므로 임시 동적 큐를 다시 작성할 수 없습니다. MQI에서는 한 개의 애플리케이션만 모델에 의해 임시 동적 큐를 열 수 있습니다. 임시 동적 큐를 사용하려고 하는 다른 애플리케이션은 MQPUT1 또는 서버 바인딩을 사용하거나 실패할 경우 다시 연결을 다시 시도할 수 있어야 합니다.

비지속 메시지만 임시 동적 큐에 넣을 수 있으며 이러한 메시지는 장애 복구 중에 손실됩니다. 이 손실은 다시 연결 중 MQPUT1을 사용하여 임시 동적 큐에 넣는 메시지에 대해 True입니다. MQPUT1 중에 장애 복구가 발생하면 MQPUT1이 성공하더라도 이 메시지를 넣을 수 없습니다. 이 문제점에 대한 임시 해결책 하나는 영구 동적 큐를 사용하는 것입니다. 서버 바인딩 애플리케이션은 다시 연결 가능하지 않기 때문에 임시 동적 큐를 이름으로 열 수 있습니다.

데이터 복구 및 고가용성

다중 인스턴스 큐 관리자를 사용한 고가용성 솔루션에는 스토리지 실패 후 데이터를 복구하는 메커니즘이 포함 되어야 합니다.

다중 인스턴스 큐 관리자는 큐 관리자 프로세스의 가용성을 높이지만 파일 시스템과 같이 큐 관리자가 메시지 및 기타 정보를 저장하는 데 사용하는 다른 컴포넌트의 가용성은 높이지 않습니다.

데이터의 가용성을 높이는 한 가지 방법은 네트워크 복원성 데이터 스토리지를 사용하는 것입니다. 네트워크 파일 시스템 및 복원성 데이터 스토리지를 사용하여 자체 솔루션을 빌드하거나 통합 솔루션을 구매할 수 있습니다. 복원성과 재해 복구를 결합하려는 경우 수십 또는 수백 킬로미터가 넘는 디스크 복제를 허용하는 비동기 디스크 복제가 사용 가능합니다.

매체를 최상으로 사용하기 위해 여러 WebSphere MQ 디렉토리가 스토리지 매체에 맵핑되는 방법을 구성할 수 있습니다. 멀티 인스턴스 큐 관리자의 경우 두 가지 유형의 WebSphere MQ 디렉토리 및 파일 사이에 중요한 차이점이 있습니다.

큐 관리자의 인스턴스 사이에 공유되어야 하는 디렉토리입니다.

큐 관리자의 여러 인스턴스 사이에서 공유되어야 하는 정보는 `qmgrs` 및 `logs` 두 디렉토리에 있습니다. 이러한 디렉토리는 공유된 네트워크 파일 시스템에 있어야 합니다. 데이터는 메시지가 작성되고 삭제될 때마다 계속해서 변경되기 때문에 지속적 고가용성과 우수한 성능을 제공하는 스토리지 매체를 사용하도록 권장됩니다.

큐 관리자의 인스턴스 사이에서 공유되지 않아도 되는 디렉토리 및 파일입니다.

일부 다른 디렉토리는 큐 관리자의 여러 인스턴스 사이에서 공유되지 않아도 되며 미러링된 파일 시스템 외의 수단으로 빠르게 복원됩니다.

- WebSphere MQ 실행 가능 파일 및 도구 디렉토리. 재설치 또는 백업된 파일 아카이브에서 백업 및 복원하여 바꾸십시오.
- 설치를 위해 전체적으로 수정되는 구성 정보입니다. 구성 정보는 Windows, UNIX and Linux 시스템의 `mqsc.ini` 파일과 같이 WebSphere MQ에 의해 관리되거나 **MQSC** 구성 스크립트와 같은 사용자 자신의 일부 구성 관리를 통해 관리됩니다. 파일 아카이브를 사용하여 백업하고 복원하십시오.
- 추적, 오류 로그 및 FFDC 파일과 같은 설치 전체 출력입니다. 파일은 `errors` 및 기본 데이터 디렉토리의 `trace` 서브디렉토리에 저장됩니다. UNIX and Linux 시스템의 기본 데이터 디렉토리는 `/var/mqm`입니다. Windows에서 기본 데이터 디렉토리가 WebSphere MQ 설치 디렉토리입니다.

백업 큐 관리자를 사용하여 선형 로깅을 사용하는 다중 인스턴스 큐 관리자의 정기적인 매체 백업을 수행할 수 있습니다. 백업 큐 관리자는 미러링된 파일 시스템만큼 빠른 복구를 제공하고 마지막 백업 이후 변경사항은 복구하지 않습니다. 백업 큐 관리자 메커니즘은 로컬화된 스토리지 실패 후 큐 관리자 복구보다는 외부 재해 복구 시나리오에서의 사용에 더 적절합니다.

IBM WebSphere MQ 가용성 솔루션 결합

애플리케이션은 가용성을 향상시키기 위해 다른 IBM WebSphere MQ 기능을 사용합니다. 다중 인스턴스 큐 관리자는 다른 고가용성 기능을 보완합니다.

IBM WebSphere MQ 클러스터는 큐 가용성을 증가시킵니다

클러스터 내 각 관리자에 있는 모든 큐 중 하나까지 클러스터 큐의 다중 정의를 작성하여 큐 가용성을 높일 수 있습니다.

클러스터의 한 멤버가 실패한 후 새 메시지가 클러스터 큐에 송신된다고 가정하십시오. 메시지가 실패한 큐 관리자에게 이동해야 하는 경우가 아니면 이 메시지는 큐의 정의가 있는 클러스터 내 다른 실행 큐 관리자로 송신됩니다.

클러스터가 가용성을 상당히 높이긴 하지만 메시지가 지연되는 결과를 가져올 수 있는 두 개의 관련 실패 시나리오가 있습니다. 다중 인스턴스 큐 관리자로 클러스터를 빌드하면 메시지가 지연될 가능성이 줄어듭니다.

고립된 메시지

클러스터의 큐 관리자가 실패하면 클러스터 내 다른 큐 관리자에게 라우트될 수 있는 메시지가 실패한 큐 관리자로 더 이상 라우트되지 않습니다. 이미 송신된 메시지는 실패한 큐 관리자가 재시작될 때까지 고립됩니다.

연관관계

연관관계는 둘 사이에 공유되거나 그렇지 않으면 개별 계산 사이에서 공유되는 정보를 설명하는 데 사용되는 용어입니다. 예를 들어, 연관관계는 요청 메시지를 서버에 송신하는 애플리케이션과 응답을 처리할 것으로 예상되는 샘플 애플리케이션 사이에 존재합니다. 또 다른 예는 이전 메시지에 따른 각 메시지의 처리인 메시지의 순서입니다.

메시지를 클러스터된 큐에 송신하는 경우 연관관계를 고려해야 합니다. 동일한 큐 관리자에게 연속 메시지를 송신해야 하나? 또는 각 메시지는 클러스터의 임의 멤버로 이동할 수 있습니까?

메시지를 클러스터 내 동일한 큐 관리자에게 송신해야 하고 이 송신이 실패할 경우 메시지는 실패한 클러스터 큐 관리자가 다시 실행될 때까지 송신자의 전송 큐에서 대기합니다.

클러스터가 다중 인스턴스 큐 관리자로 구성되는 경우 실패한 큐 관리자가 재시작될 때까지 대기하는 지연은 1분 정도로 제한되거나 잠시 동안 대기가 인수합니다. 대기가 실행되면 고립된 메시지는 처리를 계속하고 새로 활성화된 큐 관리자 인스턴스에 대한 채널이 시작되며 전송 큐에서 대기하고 있었던 메시지가 플로우를 시작합니다.

실패한 큐 관리자에 의해 메시지가 지연되는 것을 극복하도록 클러스터를 구성하는 가능한 방법은 서로 다른 두 개의 큐 관리자를 클러스터 내 각 서버에 배치하고 그 중 하나는 활성으로, 다른 하나는 대기 인스턴스로 배열하는 것입니다. 이는 활성 대기 구성이고 클러스터의 가용성을 높입니다.

관리가 줄고 확장성이 증가하는 이점 외에도 클러스터는 계속해서 추가 가용성 요소를 제공하여 다중 인스턴스 큐 관리자를 보완합니다. 클러스터는 큐 관리자의 활성 및 대기 인스턴스 둘 다에 영향을 미치는 다른 유형의 실패에 대해 보호합니다.

인터럽트되지 않는 서비스

클러스터는 인터럽트되지 않는 서비스를 제공합니다. 클러스터에 의해 수신되는 새 메시지는 처리되기 위해 활성 큐 관리자로 송신됩니다. 인터럽트되지 않는 서비스를 제공하기 위해 다중 인스턴스 큐 관리자에만 의존하지 마십시오. 대기 큐 관리자가 실패를 감지하고 그 시동을 완료하고, 해당 채널이 다시 연결되며, 실패한 메시지 배치가 다시 제출되는 데에는 시간이 걸리기 때문입니다.

로컬화된 가동 중단

활성, 대기 및 파일 시스템 서버는 허용 가능한 성능을 제공하면서 밀리초 속도로 상호작용하므로 그 거리에 대한 실제적 제한이 있습니다.

클러스터된 큐 관리자는 수초 정도의 속도의 상호작용을 요구하며 지리적으로는 전 세계 어디로든 분산될 수 있습니다.

조작 오류

가용성을 높이기 위해 두 개의 서로 다른 메커니즘을 사용하면 인적 오류와 같은 조작상의 오류가 가용성을 높이기 위한 노력을 반감시킬 가능성이 줄어듭니다.

큐 공유 그룹은 메시지 처리 가용성을 높임

z/OS에만 제공되는 큐 공유 그룹은 일단의 큐 관리자가 큐 서비스를 공유할 수 있도록 합니다. 한 큐 관리자가 실패하면 다른 큐 관리자가 큐에 있는 모든 메시지를 계속해서 처리합니다. 멀티 인스턴스 큐 관리자는 z/OS에서 지원되지 않고 광범위한 메시징 아키텍처의 일부로서만 큐 공유 그룹을 보완합니다.

WebSphere MQ 클라이언트가 애플리케이션 사용 가능성을 증가시킴

WebSphere MQ MQI 클라이언트 프로그램은 큐 관리자 사용 가능성, 연결 가중치 및 연관관계를 기본으로 하여 큐 관리자 그룹의 여러 큐 관리자에 연결할 수 있습니다. 애플리케이션이 연결된 큐 관리자 인스턴스가 실패할 경우 애플리케이션을 다시 연결할 수 있는 방법이 있는 한 큐 관리자가 실행되고 있는 시스템과 다른 시스템에서 애플리케이션을 실행하여 솔루션의 전체 가용성을 향상시킬 수 있습니다.

큐 관리자 그룹은 클라이언트를 중지된 큐 관리자로부터 분리하고 IP 스프레이어와 같이 큐 관리자 그룹 전체에 클라이언트 연결을 로드 밸런싱하여 클라이언트 가용성을 높이는 데 사용됩니다. 이 클라이언트 애플리케이션에는 실패한 큐 관리자와의 연관관계(예: 특정 큐에 대한 종속 항목)가 없어야 합니다. 그렇지 않으면 처리를 계속할 수 없습니다.

자동 클라이언트 다시 연결과 다중 인스턴스 큐 관리자는 일부 연관관계 문제점을 해결하여 클라이언트 가용성을 높입니다. 자동 클라이언트 재연결이 Java용 WebSphere MQ 클래스에서 지원되지 않습니다.

MQCNO 옵션 MQCNO_RECONNECT_Q_MGR을 설정하여 클라이언트가 동일한 큐 관리자에 다시 연결하도록 강제 실행할 수 있습니다.

1. 이전에 연결된 단일 인스턴스 큐 관리자가 실행되지 않을 경우 이 큐 관리자가 다시 실행될 때까지 연결이 다시 시도됩니다.
 2. 큐 관리자가 다중 인스턴스 큐 관리자로 구성되면 클라이언트가 활성인 인스턴스에 다시 연결합니다.
- 동일한 큐 관리자에 자동으로 다시 연결하면 큐 관리자가 클라이언트를 대신해서 보유하고 있었던 대부분의 상태 정보(예: 클라이언트가 열었던 큐 및 구독했던 토픽)가 복원됩니다. 클라이언트가 요청에 대한 응답을 수신하기 위해 동적 응답 대상 큐를 연 경우 이 응답 대상 큐에 대한 연결도 복원됩니다.

메시지가 유실되지 않았는지 확인(로깅)

WebSphere MQ는 큐 관리자 실패로부터 복구하는 데 필요한 모든 정보를 기록합니다.

WebSphere MQ는 큐 관리자가 제어하는 데이터에 대한 모든 중요한 변경사항을 복구 로그에 기록합니다.

여기에는 오브젝트 작성 및 삭제, 지속 메시지 업데이트, 트랜잭션 상태, 오브젝트 속성에 대한 변경사항 및 채널 활동이 포함됩니다. 이 로그에는 다음을 수행하여 메시지 큐에 대한 모든 업데이트를 복구하는 데 필요한 정보가 들어 있습니다.

- 큐 관리자 변경사항의 레코드 보관
- 재시작 프로세스에 사용할 큐 업데이트의 레코드 보관
- 하드웨어 및 소프트웨어 실패 후 데이터를 복원할 수 있게 함

그러나 WebSphere MQ도 파일을 호스트하는 디스크 시스템에 의존합니다. 디스크 시스템 자체가 신뢰할 수 없는 경우 로그 정보를 포함한 정보가 계속 유실될 수 있습니다.

로그의 표시

로그는 기본 및 보조 파일과 제어 파일로 구성됩니다. 로그 파일의 수 및 크기와 파일 시스템에 저장되는 위치를 정의합니다.

WebSphere MQ 로그는 다음의 두 컴포넌트로 구성됩니다.

1. 하나 이상의 로그 데이터 파일
2. 로그 제어 파일

로그 데이터 파일은 로그 익스텐트라고도 합니다.

기록되는 데이터가 포함되는 로그 파일은 여러 개 있습니다. 수와 크기(380 페이지의 『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』에 설명된 대로)를 정의하거나 시스템 기본값인 세 개의 파일을 사용할 수 있습니다.

In WebSphere MQ for 윈도우, each of the three files defaults to 1 MB. UNIX and Linux 시스템의 WebSphere MQ 에서 각 세 파일의 기본값은 4MB입니다.

큐 관리자를 작성할 때 사용자가 정의한 로그 파일의 수는 할당된 1차 로그 파일의 수입니다. 숫자를 지정하지 않을 경우 기본값이 사용됩니다.

In WebSphere MQ for 윈도우, if you have not changed the log path, log files are created under the directory:

```
C:\Program Files\IBM\WebSphere MQ\log\<QMgrName>
```

In WebSphere MQ for UNIX and Linux systems, if you have not changed the log path, log files are created under the directory:

```
/var/mqm/log/<QMgrName>
```

WebSphere MQ는 이러한 1차 로그 파일로 시작하지만 1차 로그 공간이 부족하면 보조 로그 파일을 할당합니다. 이를 동적으로 수행하며 로그 공간에 대한 수요가 감소되면 보조 로그 파일을 제거합니다. 기본적으로 최대 두 개의 2차 로그 파일이 할당될 수 있습니다. 380 페이지의 『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』에서 설명한 대로 이 기본 할당을 변경할 수 있습니다.

로그 제어 파일

로그 제어 파일에는 로그 파일의 사용을 제어하는 데 필요한 정보가 포함됩니다(예: 로그 파일의 크기와 위치 및 그다음 사용 가능한 파일의 이름).

로그 제어 파일은 내부 큐 관리자 전용입니다.

큐 관리자는 로그 제어 파일에 복구 로그의 상태와 연관된 제어 데이터를 보관하고 로그 제어 파일의 콘텐츠를 수정해서는 안 됩니다.

참고: 큐 관리자를 시작할 때 작성된 로그가 사용 중인 애플리케이션이 핸들링할 메시지의 크기 및 볼륨을 충분히 수용할 수 있는 크기인지 확인하십시오. 요구사항을 충족하기 위해 기본 로그 수 및 크기를 변경할 필요가 있을 수 있습니다. 자세한 정보는 [367 페이지의 『로그의 크기 계산』](#)의 내용을 참조하십시오.

로깅 유형

WebSphere MQ에서 로깅에 필요한 파일 수는 파일 크기, 수신한 메시지 수 및 메시지 길이에 따라 다릅니다. 큐 관리자 활동의 레코드를 유지보수하는 두 가지 방법은 순환 로깅 및 선형 로깅입니다.

순환 로깅

재시작 복구만 필요한 경우 로그를 사용하여 시스템이 중지될 때 진행 중이었던 트랜잭션을 롤백하는 순환 로깅을 사용하십시오.

순환 로깅은 모든 재시작 데이터를 로그 파일의 링에 보관합니다. 로깅은 링에서 첫 번째 파일을 채운 후 다음 단계로 이동하며 모든 파일이 가득 찰 때까지 계속됩니다. 그런 다음 링에서 첫 번째 파일로 돌아와서 다시 시작합니다. 이 순환은 제품을 사용 중인 한 계속되며 로그 파일 부족이 발생하지 않는 장점이 있습니다.

WebSphere MQ는 큐 관리자 데이터 복구를 더 이상 확인할 필요가 없을 때까지 데이터 손실 없이 큐 관리자를 재시작하는 데 필요한 로그 입력 항목을 보관합니다. 재사용을 위해 로그 파일을 릴리스하기 위한 메커니즘은 [365 페이지의 『체크포인트를 사용한 전체 복구 보장』](#)에 설명되어 있습니다.

선형 로깅

재시작 복구 및 매체 복구를 둘 다 원하는 경우 선형 로깅을 사용하십시오(로그의 콘텐츠를 재실행하여 손실되거나 손상된 데이터를 다시 작성). 선형 로깅은 지속적인 순차 파일에 로그 데이터를 보관합니다. 공간이 재사용되지 않으므로 삭제되지 않은 모든 로그 익스텐트에 로그된 모든 레코드를 항상 검색할 수 있습니다.

디스크 공간이 한정되므로 아카이빙의 양식을 고려해야 합니다. 필요에 따라 기존 공간을 재사용하거나 확장하면서 로그의 디스크 공간을 관리하는 것은 관리 태스크입니다.

선형 로깅에 사용되는 로그 파일 수는 메시지 플로우 및 큐 관리자의 사용 연한에 따라 매우 많아질 수 있습니다. 그러나 활성으로 불리는 파일이 여러 개 있습니다. 활성 파일에는 큐 관리자를 재시작하는 데 필요한 로그 입력 항목이 포함됩니다. 전체적으로 활성 로그 파일은 활성 로그라고 합니다. 활성 로그 파일의 수는 일반적으로 구성 파일에 정의된 1차 로그 파일의 수보다 적습니다. (이 활성 로그 파일 수 정의에 대한 정보는 [367 페이지의 『로그의 크기 계산』](#)의 내용을 참조하십시오.)

로그 파일을 활성이라고 하는지 여부를 제어하는 키 이벤트가 체크포인트입니다. WebSphere MQ 체크포인트는 복구 로그 및 오브젝트 파일 사이에서 일치되는 지점입니다. 체크포인트는 재시작 복구를 수행하는 데 필요한 로그 파일 세트를 판별합니다. 활성이 아닌 로그 파일은 재시작 복구에 필요하지 않으며 비활성으로 불립니다. 경우에 따라서는 비활성 로그 파일이 매체 복원에 필요합니다. (체크포인트에 관한 자세한 정보는 [365 페이지의 『체크포인트를 사용한 전체 복구 보장』](#)을 참조하십시오.)

비활성 로그 파일은 재시작 복구에 필요하지 않으므로 아카이브될 수 있습니다. 매체 복구에 필요하지 않은 비활성 로그 파일은 불필요한 로그 파일로 간주할 수 있습니다. 더 이상 조작에 이용되지 않는 불필요한 로그 파일은 삭제할 수 있습니다. 로그 파일의 속성 지정에 대한 추가 정보는 [368 페이지의 『로그 관리』](#)의 내용을 참조하십시오.

새 체크포인트가 두 번째 또는 나중의 1차 로그 파일에 기록된 경우, 첫 번째 파일이 비활성으로 되고 새 1차 파일이 형식화되어 1차 풀의 맨 끝에 추가되며 로깅에 사용 가능한 1차 파일 수를 복원합니다. 이러한 방법으로 1차 로그 파일 풀이 로그 파일의 계속 확장되는 목록에서 현재 파일 세트로 표시될 수 있습니다. 사용자 조작의 요구사항에 따라 비활성 파일을 관리하는 것은 관리 태스크입니다.

선형 로깅에 대해 보조 로그 파일이 정의되더라도 이러한 파일은 정상 조작에서는 사용되지 않습니다. 재시작을 위해 계속해서 필요할 수 있으므로 파일을 활성 풀로부터 비울 수 없는 상황이 발생하면(아마도 장기 트랜잭션으로 인해) 보조 파일이 형식화되고 활성 로그 파일 풀에 추가됩니다.

사용 가능한 2차 파일 수를 모두 사용한 경우, 로그 활동이 필요한 추가 조작에 대한 요청이 거부되며 애플리케이션으로 MQRC_RESOURCE_PROBLEM 리턴 코드가 리턴됩니다.

두 가지 유형의 로깅 모두 하드웨어 실패가 없다는 가정 하에 예상치 못한 전원 손실에 대처할 수 있습니다.

체크포인트를 사용한 전체 복구 보장

체크포인트는 큐 관리자 데이터와 로그 파일을 동기화하고 로그 레코드를 제거할 수 있는 일관성 지점을 표시합니다. 체크포인트를 자주 수행하면 복구 속도가 더 신속해집니다.

메시지 큐에 대한 지속 업데이트는 두 단계로 발생합니다. 첫 번째, 업데이트를 나타내는 레코드가 로그에 기록된 후 큐 파일이 업데이트됩니다. 그러면 로그 파일이 큐 파일 보다 더 최신으로 될 수 있습니다. 재시작 처리가 일관성 지점에서 시작되도록 하기 위해 WebSphere MQ는 체크포인트를 사용합니다. 체크포인트는 로그에 설명된 레코드가 큐에 있는 레코드와 동일한 시점입니다. 체크포인트 자체는 큐 관리자를 재시작하는 데 필요한 일련의 로그 레코드로 구성됩니다(예: 체크포인트 시점에서 활성인 모든 트랜잭션(작업 단위)의 상태).

WebSphere MQ가 자동으로 체크포인트를 생성합니다. 큐 관리자가 시작될 때, 시스템 종료 시, 로깅 공간이 느리게 실행 중일 때 및 10,000 조작이 로그될 때마다 수행됩니다.

큐는 추가 메시지를 핸들링하기 때문에 체크포인트 레코드는 큐의 현재 상태와 불일치하게 됩니다.

WebSphere MQ는 재시작할 때 로그에서 최신 체크포인트 레코드를 찾습니다. 이 정보는 모든 체크포인트의 끝에서 업데이트되는 체크포인트 파일에 있습니다. 체크포인트 레코드는 로그와 데이터 사이의 최신 일관성 지점을 나타냅니다. 체크포인트 이후 발생한 모든 조작은 앞으로 재실행됩니다. 이를 재실행 단계라고 합니다. 재실행 단계는 큐를 시스템 실패 또는 종료 전 논리 상태로 되돌립니다. 재실행 단계 중에 시스템 실패 또는 종료가 발생하기 전에 인플라이트였던 트랜잭션의 목록이 작성됩니다. 메시지 AMQ7229 및 AMQ7230은 재실행 단계의 진행 상태를 표시하기 위해 발행됩니다.

백아웃 또는 커밋할 조작을 알아내기 위해 WebSphere MQ가 불완전한 트랜잭션과 연관된 각 활성 로그 레코드에 액세스합니다. 이를 복구 단계라고 합니다. 메시지 AMQ7231, AMQ7232 및 AMQ7234는 복구 단계의 진행 상태를 표시하기 위해 발행됩니다.

복구 단계 중 필요한 모든 로그 레코드에 액세스하면 각 활성 트랜잭션이 순서대로 해석되고 트랜잭션과 연관된 각 조작은 백아웃되거나 커밋됩니다. 이를 해석 단계라고 합니다. 메시지 AMQ7233은 해석 단계의 진행 상태를 표시하기 위해 발행됩니다.

WebSphere MQ는 로그의 헤드 및 후미에 대한 내부 포인터를 유지보수합니다. 헤드 포인터를 복구 중인 메시지 데이터와 일치하는 가장 최근의 체크포인트로 이동시킵니다.

체크포인트는 복구를 가장 효율적으로 만들고 1차 및 2차 로그 파일의 재사용을 제어하는 데 사용됩니다.

366 페이지의 그림 69에서 최신 체크포인트 앞에 있는 모든 레코드는 더 이상 WebSphere MQ에 필요하지 않습니다. 큐는 체크포인트 정보 및 이후 로그 입력 항목으로부터 복구할 수 있습니다. 순환 로깅에서는 체크포인트 이전 비워진 파일을 재사용할 수 있습니다. 선형 로깅에서는 비워진 로그 파일이 정상 조작을 위해 더 이상 액세스되지 않으며 비활성이 됩니다. 이 예제에서, 큐 헤드 포인터는 마지막 체크포인트인 체크포인트 2를 가리키도록 이동되며, 체크포인트 2는 새로운 큐 헤드인 헤드 2가 됩니다. 이제 로그 파일 1을 재사용할 수 있습니다.

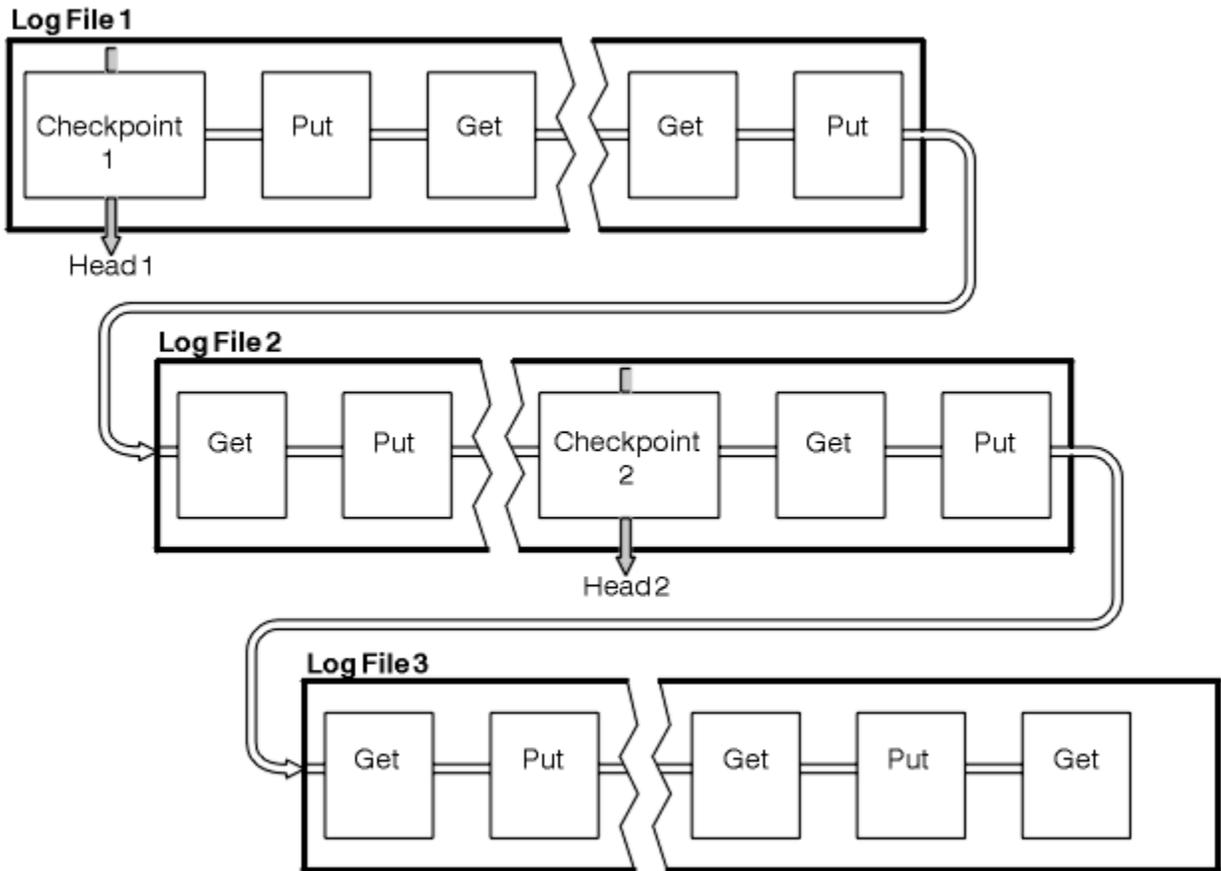


그림 69. 체크포인트링

장기 실행 트랜잭션 체크포인트

장기 실행 트랜잭션이 로그 파일의 재사용에 영향을 미치는 방식입니다.

367 페이지의 그림 70에서는 장기 실행 트랜잭션이 로그 파일의 재사용에 영향을 미치는 방식을 보여줍니다. 이 예에서 장기 실행 트랜잭션은 표시된 첫 번째 체크포인트 뒤에 LR1으로 표시되는 로그에 대한 입력 항목을 작성했습니다. 이 트랜잭션은 세 번째 체크포인트를 지나서까지(포인트 LR2) 완료되지 않습니다. LR1 이후의 모든 로그 정보는 해당 트랜잭션의 복구를 위해 보존되며 필요한 경우 그러한 복구가 완료될 때까지 보존됩니다.

장기 실행 트랜잭션이 완료된 후 LR 2에서 로그의 헤드는 최근에 로그된 체크포인트인 체크포인트 3으로 이동됩니다. 체크포인트 3, 헤드 2 전의 로그 레코드를 포함하는 파일은 더 이상 필요하지 않습니다. 순환 로깅을 사용하는 경우 공간을 재사용할 수 있습니다.

장기 실행 트랜잭션이 완료되기 전에 1차 로그 파일이 완전히 채워지면 2차 로그 파일을 사용하여 로그가 가득 채워지지 않게 합니다.

로그 헤드가 이동하고 순환 로깅을 사용하는 경우 1차 로그 파일이 재사용에 적합해질 수 있고 현재 파일을 채운 후에 로그는 사용 가능한 첫 번째 1차 파일을 재사용합니다. 선형 로깅을 사용하는 경우 로그 헤드가 여전히 아래로 이동하면 활성 풀 및 첫 번째 파일이 비활성이 됩니다. 이후 로깅 활동을 준비하기 위해 새 1차 파일이 형식화되어 풀의 맨 아래에 추가됩니다.

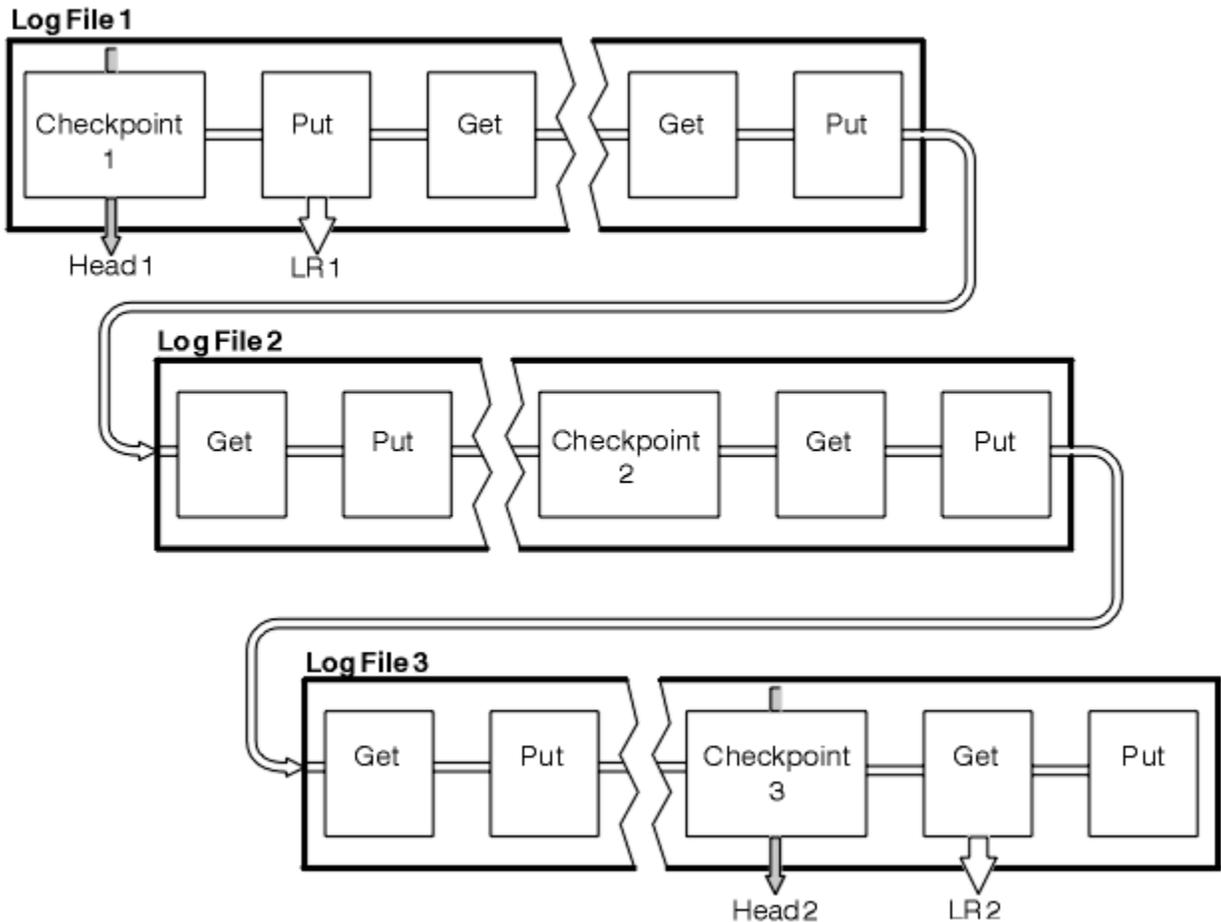


그림 70. 장기 실행 트랜잭션 체크포인트

로그의 크기 계산

큐 관리자가 필요로 하는 로그의 크기를 계산합니다.

큐 관리자가 순환 또는 선형 로깅을 사용하는지 여부를 결정한 후에 큐 관리자가 필요로 하는 로그의 크기를 계산해야 합니다. 로그의 크기는 다음 로그 구성 매개변수에 의해 판별됩니다.

LogFilePages

각 1차 및 보조 로그 파일의 크기(4K 페이지 단위)

LogPrimaryFiles

사전 할당된 1차 로그 파일의 수

LogSecondaryFiles

1차 로그 파일이 다 차면 사용하기 위해 작성될 수 있는 2차 로그 파일의 수

367 페이지의 표 31는 큐 관리자가 다양한 조작을 위해 로그하는 데이터의 양을 표시합니다. 대부분의 큐 관리자 조작에는 최소한의 양의 로그 공간이 필요합니다. 하지만 지속 메시지를 큐에 넣을 때 모든 메시지 데이터는 로그에 작성되어 메시지 복구가 가능하도록 해야 합니다. 로그의 크기는 일반적으로 큐 관리자가 핸들링해야 하는 지속 메시지의 수 및 크기에 따라 달라집니다.

표 31. 로그 항목 크기(모든 값은 근사값임)	
Operation	크기
지속 메시지를 넣음	750바이트 + 메시지 길이 메시지가 클 경우 이는 261844 바이트의 세그먼트로 나뉘지며 각 세그먼트에는 추가로 300바이트가 추가됩니다.

표 31. 로그 항목 크기(모든 값은 근사값임) (계속)	
Operation	크기
메시지 가져오기	260바이트
동기점, 커밋	750바이트
동기점, 롤백	롤백될 각 Get 또는 Put에 대해 1000바이트 + 12바이트
오브젝트 작성	1500바이트
오브젝트 삭제	300바이트
속성 대체	1024바이트
매체 이미지 기록	800바이트 + 이미지 이미지는 260 000바이트 세그먼트로 나뉘지며 각 세그먼트에는 추가로 300바이트가 추가됩니다.
체크포인트	750바이트 + 각각의 활성화 작업 단위에 대해 200바이트 + 각각의 클러스터-송신자 채널에 대해 380바이트(큐 관리자마다 여러 개의 클러스터 전송 큐를 사용 중인 경우). 추가 데이터가 성능상의 이유로 버퍼된 커밋되지 않은 Put 및 Get에 대해 로그될 수 있습니다. 클러스터 송신자 채널이 있는 경우, 각 체크포인트에서 클러스터 송신자 채널당 추가 380바이트가 로그에 기록됩니다.

참고:

1. 큐 관리자가 시작할 때마다 1차 및 2차 로그 파일의 수를 변경할 수 있습니다.
2. 로그 파일의 크기를 변경할 수 없으며 큐 관리자를 작성하기 **전에** 로그 파일 크기를 판별해야 합니다.
3. 1차 로그 파일의 수 및 로그 파일 크기는 큐 관리자가 작성될 때 사전 할당되는 로그 공간의 양을 판별합니다.
4. 1차 및 보조 로그 파일의 총 갯수는 UNIX and Linux 시스템에서는 511개, Windows에서는 255개를 초과할 수 없고, 장기 실행 트랜잭션이 있으면 재시작 복구를 위해 큐 관리자가 사용할 수 있는 로그 공간의 최대 양을 제한합니다. 큐 관리자가 매체 복구에 필요로 할 수 있는 로그 공간의 양은 이러한 제한을 공유하지 않습니다.
5. 순환 로깅을 사용 중일 때 큐 관리자는 1차 로그 공간을 재사용합니다. 이는 큐 관리자의 로그가 큐 관리자가 로그에 필요로 할 것으로 사용자가 계산한 데이터의 양보다 작을 수 있음을 의미합니다. 로그 파일이 다 차고 순서 상으로 다음인 1차 로그 파일이 사용 불가능하면 큐 관리자는 보조 로그 파일을 최대로 할당합니다.
6. 체크포인트 중에 1차 로그 파일을 재사용할 수 있게 됩니다. 로그 공간의 양이 줄고 있으므로 큐 관리자는 체크포인트를 사용하기 전에 1차 및 보조 로그 공간을 모두 고려합니다.

1차 로그 파일을 2차 로그 파일보다 크지 않게 정의한 경우 큐 관리자는 체크포인트를 사용하기 전에 2차 로그 파일을 할당할 수 있습니다. 이렇게 하면 1차 로그 파일을 재사용에 사용할 수 있습니다.

로그 관리

로그는 거의 자체 관리되고 있지만 때때로 공간 문제 해결을 위해 관리가 필요합니다.

시간이 지날수록 기록된 로그 레코드는 큐 관리자를 재시작하는 데 불필요해집니다. 순환 로깅을 사용 중인 경우 큐 관리자는 로그 파일에서 비워진 공간을 요구합니다. 이 활동은 사용자에게 분명하게 드러나지 않으며 할당된 공간이 신속히 재사용되기 때문에 사용된 디스크 공간량이 줄어드는 것은 보통 표시되지 않습니다.

로그 레코드 중 마지막 완료 체크포인트 이후에 기록된 로그 레코드와 활성화 트랜잭션에 의해 기록된 로그 레코드만 큐 관리자 재시작에 필요합니다. 따라서 체크포인트가 오래동안 수행되지 않았거나 장기 실행 트랜잭션이 오래 전에 로그 레코드를 기록한 경우 로그가 채워질 수 있습니다. 큐 관리자는 첫 번째 문제점을 피하기에 충분히 자주 체크포인트를 사용하려고 시도합니다.

장기 실행 트랜잭션이 로그를 채울 경우, 로그 레코드 작성 시도에 실패하고 일부 MQI 호출이 MQRC_RESOURCE_PROBLEM을 리턴합니다. (공간은 모든 인플라이트 트랜잭션을 커밋하거나 롤백하도록 예약되어 있으므로 MQCMIT 또는 MQBACK 는 실패하지 않아야 합니다.)

큐 관리자는 너무 많은 로그 공간을 사용하는 트랜잭션을 롤백합니다. 트랜잭션이 있는 애플리케이션이 이 방식으로 롤백되고 같은 트랜잭션에서 동기점을 지정하는 후속 MQPUT 또는 MQGET 조작을 수행할 수 없습니다. 이 상태에서 동기점 아래에서 메시지를 넣거나 가져오려고 시도하면 MQRC_BACKED_OUT이 리턴됩니다. 그러면 애플리케이션이 MQRC_BACKED_OUT을 리턴하는 MQCMIT 또는 MQBACK을 발행하고 새 트랜잭션을 시작할 수 있습니다. 로그 공간을 너무 많이 사용하는 트랜잭션이 롤백될 때 해당 로그 공간이 릴리스되고 큐 관리자는 정상적으로 계속 작동합니다.

로그가 채워지면 AMQ7463 메시지가 발행됩니다. 또한 장기 실행 트랜잭션이 공간이 릴리스되지 못하게 했기 때문에 로그가 채워진 경우 AMQ7465 메시지가 발행됩니다.

마지막으로, 레코드가 로그 처리보다 더 빠르게 로그에 기록되는 경우 AMQ7466 메시지가 표시됩니다. 이 메시지가 표시되면 로그 파일 수를 늘리거나 큐 관리자가 처리하는 데이터의 양을 줄이십시오.

디스크가 가득 차면 발생하는 상황

큐 관리자 로깅 컴포넌트는 가득 찬 디스크 및 가득 찬 로그 파일을 처리할 수 있습니다. 로그가 포함된 디스크가 가득 차면 큐 관리자가 AMQ6708 메시지를 발행하고 오류 레코드가 생성됩니다.

로그 파일은 로그 레코드가 기록됨에 따라 확장되지 않고 최대 크기로 작성됩니다. 이는 WebSphere MQ가 새 파일을 작성 중일 때만 디스크 공간을 다 소모할 수 있으며 로그에 레코드를 기록 중일 때는 공간을 다 소모할 수 없다는 것을 의미합니다. WebSphere MQ는 기존 로그 파일에서 사용 가능한 공간의 크기를 항상 알고 있으며 이에 따라 파일 내의 공간을 관리합니다.

로그 파일이 포함된 드라이브를 채우는 경우 일부 디스크 공간을 비울 수 있습니다. 선형 로그를 사용하는 경우 로그 디렉토리에 일부 비활성 로그 파일이 있을 수 있으며 이러한 파일을 다른 드라이브 또는 디바이스에 복사할 수 있습니다. 여전히 공간이 부족하면 큐 관리자 구성 파일의 로그 구성이 올바른지 확인하십시오. 로그가 사용 가능한 공간보다 커지지 않도록 1차 또는 보조 로그 파일의 수를 줄일 수 있습니다. 기존 큐 관리자의 로그 파일 크기는 대체할 수 없습니다. 큐 관리자는 모든 로그 파일을 동일한 크기로 간주합니다.

로그 파일 관리

로그 파일에 충분한 공간을 할당합니다. 선형 로깅에서는 더 이상 필요하지 않은 오래된 로그 파일을 삭제할 수 있습니다.

선형 로깅을 사용하는 경우 시스템 구성 시 로그 파일을 보유하기에 충분한 공간이 있는지 확인하십시오(388 페이지의 『IBM WebSphere MQ에 대한 로그 기본값』 및 395 페이지의 『큐 관리자 로그』 참조). 필요 시 작성될 보조 파일을 위한 공간을 포함하여 로그가 사용하는 디스크 공간의 양은 구성된 크기를 넘게 증가하지 않습니다.

선형 로그를 사용하는 경우 데이터가 로깅될 때마다 로그 파일이 계속해서 추가되고 사용되는 디스크 공간의 양은 시간의 경과에 따라 증가합니다. 로깅되는 데이터의 비율이 높은 경우 새 로그 파일이 빠르게 디스크 공간을 사용합니다.

시간의 경과에 따라 선형 로그의 오래된 로그 파일이 큐 관리자를 재시작하거나 손상된 오브젝트의 매체 복구를 수행하는 데 더 이상 필요하지 않게 됩니다. 다음은 계속 필요한 로그 파일을 판별하기 위한 메소드입니다.

로거 이벤트 메시지

사용 가능한 경우, 큐 관리자가 새 로그 파일에 로그 레코드를 쓰기 시작할 때 로그 프로그램 이벤트 메시지가 생성됩니다. 로거 이벤트 메시지의 콘텐츠는 큐 관리자 재시작 및 매체 복구를 위해 여전히 필요한 로그 파일을 지정합니다. 로거 이벤트 메시지에 대한 자세한 정보는 로거 이벤트를 참조하십시오.

큐 관리자 상태

MQSC 명령인 DISPLAY QMSTATUS 또는 PCF 명령인 Inquire Queue Manager Status를 실행하면 필요한 로그 파일의 세부사항을 포함한 큐 관리자 정보를 리턴합니다. MQSC 명령에 대한 자세한 정보는 스크립트 (MQSC) 명령을 참조하고, PCF 명령에 대한 자세한 정보는 관리 태스크 자동화를 참조하십시오.

큐 관리자 메시지

주기적으로 큐 관리자는 필요한 로그 파일을 표시하도록 한 쌍의 메시지를 발행합니다.

- 메시지 AMQ7467은 큐 관리자를 재시작하는 데 필요한 가장 오래된 로그 파일의 이름을 제공합니다. 이 로그 파일 및 모든 새 로그 파일은 큐 관리자 재시작 중 사용 가능해야 합니다.

- 메시지 AMQ7468은 매체 복구에 필요한 가장 오래된 로그 파일의 이름을 제공합니다.

큐 관리자 재시작에 필요한 로그 파일 및 활성 로그 파일만 온라인이어야 합니다. 비활성 로그 파일은 재해 복구용 테이프와 같은 아카이브 매체에 복사하고 로그 디렉토리에서는 제거할 수 있습니다. 매체 복구에 필요하지 않은 비활성 로그 파일은 불필요한 로그 파일로 간주할 수 있습니다. 더 이상 조작에 이용되지 않는 불필요한 로그 파일은 삭제할 수 있습니다.

"오래된" 및 "새로운" 로그 파일을 판별하려면 파일 시스템에서 적용된 수정 시간 보다는 로그 파일 번호를 사용하십시오.

필요한 로그 파일을 찾을 수 없는 경우 운영자 메시지 AMQ6767이 발행됩니다. 로그 파일 및 모든 후속 로그 파일을 큐 관리자가 사용할 수 있게 하고 조작을 다시 시도하십시오.

참고: 매체 복구를 수행할 때 필요한 모든 로그 파일은 로그 파일 디렉토리에서 동시에 사용 가능해야 합니다. 필요한 모든 로그 파일을 보유하기 위해 디스크 공간이 부족해지는 것을 피하려면 복구가 필요할 수도 있는 오브젝트의 정기적 매체 이미지를 제거해야 합니다. 큐 관리자에 있는 모든 오브젝트의 매체 이미지를 가져오려면 다음 예제에 표시된 대로 **rcdmqimg** 명령을 실행하십시오.

Windows

```
rcdmqimg -m QMNAME -t all *
```

UNIX and Linux에서

```
rcdmqimg -m QMNAME -t all "*"
```

rcdmqimg를 실행하면 매체 로그 순서 번호(LSN)가 앞으로 이동합니다. 로그 순서 번호에 대한 자세한 내용은 373 페이지의 『**dmpmqlog** 명령을 사용하여 로그의 콘텐츠 덤프』의 내용을 참조하십시오. **rcdmqimg**는 자동으로 실행되지 않으므로 수동으로 실행되거나 작성한 자동 태스크로부터 실행되어야 합니다. 이 명령에 대한 자세한 정보는 **rcdmqimg** 및 **dmpmqlog**를 참조하십시오.

참고: 메시지 AMQ7467 및 AMQ7468은 **rcdmqimg** 명령 실행 시 발행될 수도 있습니다.

여분의 로그 파일 판별

선형 로그 파일을 관리할 때 삭제하거나 아카이브할 수 있는 파일을 확인하는 것이 중요합니다. 이 정보는 이 결정을 내리는 데 도움을 줍니다.

"오래된" 로그 파일을 판별하는 데 파일 시스템의 수정 시간을 사용하지 마십시오. 로그 파일 번호만 사용하십시오. 큐 관리자의 로그 파일 사용은 필요 전 로그 파일 사전 할당 및 형식화와 같은 복잡한 규칙을 따릅니다. 수정 시간을 사용하여 상대적 연령을 판별하려고 시도할 경우 잘못 유도할 수 있는 수정 시간이 포함된 로그 파일이 표시될 수 있습니다.

큐 관리자를 재시작하는 데 필요한 가장 오래된 로그 파일을 판별하려면 **DISPLAY QMSTATUS RECLOG** 명령을 실행하십시오.

매체 복원을 수행하는 데 필요한 가장 오래된 로그 파일을 판별하려면 **DISPLAY QMSTATUS MEDIALOG** 명령을 실행하십시오.

일반적으로 파일 번호가 낮을 수록 오래된 로그를 의미합니다. 10년 동안 매일 약 3,000개의 로그 파일을 교차하는 것처럼 상당한 로그 파일 회전율을 갖지 않은 한, 9 999 999에서 랩핑되는 번호에 맞추지 않아도 됩니다. 이 경우 RECLOG 값 미만의 번호를 가지는 로그 파일을 아카이브할 수 있고 RECLOG 및 MEDIALOG 값 미만의 번호를 가지는 로그 파일을 삭제할 수 있습니다.

그러나 상당한 로그 파일 회전율을 가지고 있거나 일반적인 경우로 자신있게 복사하려는 경우 다음 알고리즘을 사용할 수 있습니다.

S == 로그 파일 번호 재시작

(DISPLAY QMSTATUS RECLOG에서).

Let M == media recovery log file number

(from DISPLAY QMSTATUS MEDIALOG).

Let L == a log file number with eligibility for deletion or archiving that needs to be determined.

```
function minlog (a, b) {
  if (abs (a - b) < 5000000)
    return min (a, b); # Not wrapped.
  else
    최대 리턴 (a, b); # 랩핑됨.}
```

A log file L can be deleted if
(L != S && L != M && minlog (L, minlog (S, M)) == L).

A log file L can be archived if
(L != S && minlog (L, S) == L).

로그 파일 위치

로그 파일 위치를 선택할 때 디스크 공간이 부족하기 때문에 WebSphere MQ가 새 로그를 형식화하는 데 실패하면 해당 조치가 심각한 영향을 받는다는 것을 기억하십시오.

순환 로깅을 사용하는 경우 적어도 구성된 1차 로그 파일에 대해서만이라도 드라이브에 충분한 공간이 있는지 확인하십시오. 로그의 크기가 커져야 할 경우 필요한 보조 로그 파일 1개에 대한 공간만이라도 남겨두십시오.

선형 로그를 사용하는 경우 상당히 많은 공간을 허용하십시오. 로그가 이용하는 공간은 데이터가 로그됨에 따라 지속적으로 늘어납니다.

이상적으로는 로그 파일을 큐 관리자 데이터와 분리된 디스크 드라이브에 두십시오. 이렇게 하면 성능면에서 유익합니다. 로그 파일을 미러링된 배열의 다중 디스크 드라이브에 배치할 수도 있습니다. 이는 로그를 포함하는 드라이브의 실패로부터 보호합니다. 미러링을 하지 않으면 WebSphere MQ 시스템의 최신 백업으로 강제로 돌아가게 될 수 있습니다.

복구에 로그 사용

실패에서 복구하는 데 로그를 사용합니다.

데이터가 손상될 수 있는 여러 상황이 있습니다. WebSphere MQ는 다음에서 복구하는 데 도움을 줍니다.

- 손상된 데이터 오브젝트
- 시스템에서 전원 손실
- 통신 실패

이 절에서는 이러한 문제점으로부터 복구하는 데 로그가 사용되는 방식을 알아봅니다.

전원 유실 또는 통신 장애로부터 복구

WebSphere MQ는 통신 장애 및 정전 모두에서 복구할 수 있습니다. 또한 파일의 의도치 않은 삭제와 같은 다른 유형의 문제점으로부터도 복구할 수 있습니다.

통신 장애의 경우, 메시지는 수신 애플리케이션이 제거할 때까지 큐에 남아 있습니다. 메시지가 전송되고 있는 경우 성공적으로 전송될 때까지 이 메시지는 전송 큐에 남아 있습니다. 통신 실패로부터 복구하기 위해 일반적으로 실패한 링크를 사용하여 채널을 재시작할 수 있습니다.

정전된 경우, 큐 관리자가 재시작될 때 WebSphere MQ는 해당 큐를 장애 발생 시의 커밋된 상태로 복원합니다. 이는 지속 메시지가 손실되지 않도록 합니다. 비지속 메시지는 제거됩니다. WebSphere MQ가 갑자기 중지되면 이들 메시지는 유지되지 않습니다.

손상된 오브젝트 복구

IBM WebSphere MQ 오브젝트가 의도치 않은 손상 등으로 인해 사용할 수 없게 되는 상황이 있습니다. 그러면 전체 시스템 또는 그 일부분을 복구해야 합니다. 필요한 조치는 손상이 감지되는 시간, 선택한 로그 메소드가 매체 복구를 지원하는지 여부 및 손상된 오브젝트에 따라 다릅니다.

매체 복구

매체 복구는 선형 로그에 기록된 정보로부터 오브젝트를 다시 작성합니다. 예를 들어, 오브젝트 파일이 의도치 않게 삭제되거나 다른 이유로 사용할 수 없게 되는 경우 매체 복구가 이 파일을 다시 작성할 수 있습니다. 오브젝트의 매체 복구에 필요한 로그의 정보는 *media image*라고 합니다.

매체 이미지는 오브젝트 자체를 다시 작성하는 데 사용할 수 있는 오브젝트 이미지가 포함된 일련의 로그 레코드입니다.

오브젝트 재작성에 필요한 첫 번째 로그 레코드를 매체 복원 레코드라고 하며, 오브젝트의 최신 매체 이미지가 시작됩니다. 각 오브젝트의 매체 복구 레코드는 체크포인트 중에 기록되는 정보 조각 중 하나입니다.

오브젝트가 그 매체 이미지로부터 다시 작성될 때 마지막 이미지를 제거한 후 오브젝트에 대해 수행된 업데이트를 설명하는 로그 레코드를 재실행하는 데에도 필요합니다.

예를 들어, 지속 메시지를 큐에 넣기 전에 제거된 큐 오브젝트의 이미지가 있는 로컬 큐를 고려해 보십시오. 오브젝트의 최근 이미지를 다시 작성하기 위해 이미지 자체를 재실행하는 것 외에도 큐에 메시지를 넣은 것을 기록하는 로그 입력 항목을 재실행할 필요가 있습니다.

오브젝트가 작성될 때 기록된 로그 레코드는 오브젝트를 완전히 다시 작성하기 위한 충분한 정보를 포함합니다. 이러한 레코드는 오브젝트의 첫 번째 매체 이미지를 구성합니다. 그런 다음 종료될 때마다 큐 관리자는 다음과 같이 매체 이미지를 자동으로 기록합니다.

- 로컬이 아닌 모든 프로세스 오브젝트와 큐의 이미지
- 비어 있는 로컬 큐의 이미지

또한 `rcdmqimg`에 설명된 `rcdmqimg` 명령을 사용하여 수동으로 매체 이미지를 기록할 수 있습니다. 이 명령은 IBM WebSphere MQ 오브젝트의 매체 이미지를 기록합니다. 매체 이미지 작성이 완료되면 매체 이미지를 보유하는 로그와 이 시간 후에 작성된 모든 로그만 손상된 오브젝트를 다시 작성한 데 필요합니다. 매체 이미지를 작성하는 것의 이점은 사용 가능한 여유 스토리지의 양 및 로그 파일이 작성되는 속도와 같은 요소에 따라 다릅니다.

매체 이미지로부터 복구

큐 관리자는 큐 관리자를 시작하는 동안 매체 이미지에서 일부 오브젝트를 자동으로 복구합니다. 또한 큐 관리자가 마지막으로 종료될 때 완료되지 않은 트랜잭션과 관련되어 있고 재시작 처리 중에 손상된 경우 큐를 자동으로 복구합니다.

`rcrmqobj` 사용하여 다른 오브젝트도 수동으로 복구해야 합니다. 이 명령은 로그에 있는 레코드를 재실행하여 IBM WebSphere MQ 오브젝트를 재작성합니다. 이미지가 저장된 시간과 다시 작성 명령이 발행된 시간 사이에 적용 가능한 모든 로그 이벤트와 함께 로그에서 찾은 최근 이미지로부터 이 오브젝트가 다시 작성됩니다. IBM WebSphere MQ 오브젝트가 손상된 경우, 수행할 수 있는 올바른 조치는 이를 삭제하거나 이 메소드로 다시 작성하는 것입니다. 이 방식으로는 비지속 메시지를 복구할 수 없습니다.

`rcrmqobj` 명령의 추가 세부사항은 `rcrmqobj`를 참조하십시오.

오브젝트의 매체 복구를 시도할 때 매체 복구 레코드 및 모든 후속 로그 파일을 포함하는 로그 파일이 로그 파일 디렉토리에서 사용 가능해야 합니다. 필요한 파일을 찾을 수 없는 경우 운영자 메시지 AMQ6767이 발행되고 매체 복구 조작이 실패합니다. 다시 작성하려는 오브젝트의 정기적 매체 이미지를 제거하지 않을 경우 오브젝트를 다시 작성하는 데 필요한 모든 로그 파일을 보유하기에 디스크 공간이 충분하지 않을 수 있습니다.

시동 중 손상된 오브젝트 복구

시동 중 큐 관리자가 손상된 오브젝트를 발견하는 경우에 큐 관리자가 취하는 조치는 오브젝트의 유형 및 큐 관리자가 매체 복구를 지원하도록 구성되는지 여부에 따라 다릅니다.

큐 관리자 오브젝트가 손상되는 경우 오브젝트를 복구할 수 없는 한 큐 관리자는 시작할 수 없습니다. 큐 관리자가 선행 로그로 구성되어 매체 복구를 지원하는 경우 IBM WebSphere MQ는 자동으로 해당 매체 이미지로부터 큐 관리자를 다시 작성하려고 시도합니다. 선택한 로그 메소드가 매체 복구를 지원하지 않는 경우 큐 관리자의 백업을 복원하거나 큐 관리자를 삭제할 수 있습니다.

큐 관리자가 중지될 때 활성인 트랜잭션이 있었다면 이러한 트랜잭션 내에 넣거나 가져온 지속적이고 커밋되지 않은 메시지가 포함된 로컬 큐도 큐 관리자를 성공적으로 시작하기 위해 필요합니다. 로컬 큐 중 손상된 것으로 확인되는 로컬 큐가 있고 큐 관리자가 매체 복구를 지원하는 경우 이 큐 관리자는 해당 매체 이미지로부터 손상된 로컬 큐를 다시 작성하려고 시도합니다. 큐 중 복구할 수 없는 큐가 있을 경우 IBM WebSphere MQ를 시작할 수 없습니다.

매체 복구를 지원하지 않는 큐 관리자에서 시동 처리 중 커밋되지 않은 메시지가 포함된 손상된 로컬 큐가 발견되면 큐가 손상된 오브젝트로 표시되고 이 큐에 있는 커밋되지 않은 메시지는 무시됩니다. 이 상황은 그러한 큐

관리자에서 손상된 오브젝트의 매체 복구가 불가능하기 때문이고 남은 조치는 해당 오브젝트를 삭제하는 것뿐입니다. 손상을 보고하기 위해 메시지 AMQ7472가 발행됩니다.

다른 때의 손상된 오브젝트 복구

오브젝트의 매체 복구는 시동 중에만 자동입니다. 다른 때에는 오브젝트 손상이 감지되면 메시지 AMQ7472이 발행되고 이 오브젝트를 사용하는 대부분의 조작은 실패합니다. 큐 관리자가 시작된 후 언제든지 큐 관리자 오브젝트가 손상되면 큐 관리자가 선제 종료룰 수행합니다. 오브젝트가 손상된 경우 이를 삭제할 수 있습니다. 또는 큐 관리자가 선형 로그를 사용하는 경우 `rcrmqobj` 명령을 사용하여 해당 매체 이미지로부터 이 오브젝트 복구를 시도하십시오(추가 세부사항은 `rcrmqobj` 참조).

IBM WebSphere MQ 로그 파일 보호

큐 관리자가 실행되고 있을 때에는 로그 파일을 건드리지 않습니다. 복구가 불가능할 수 있습니다. `superuser` 또는 `mqm` 권한을 사용하여 의도치 않게 수정하지 않도록 로그 파일을 보호하십시오.

IBM WebSphere MQ 큐 관리자가 실행 중인 경우 활성 로그 파일을 수동으로 제거하지 마십시오. 큐 관리자가 다시 시작하는 데 필요한 로그 파일을 사용자가 실수로 삭제하면, IBM WebSphere MQ는 오류를 발행하지 않고 지속적 메시지를 포함하는 데이터를 계속 처리합니다. 큐 관리자는 정상적으로 종료되지만 재시작에 실패할 수 있습니다. 그러면 메시지의 복구가 불가능해집니다.

활성 큐 관리자가 사용하고 있는 로그를 삭제할 수 있는 권한이 있는 사용자는 다른 중요한 큐 관리자 자원(예: 큐 파일, 오브젝트 카탈로그 및 IBM WebSphere MQ 실행 파일)을 삭제할 수 있는 권한도 가집니다. 따라서 이러한 사용자는 미숙함 등으로 인해 IBM WebSphere MQ가 자체 보호할 수 없는 방식으로 실행 중이거나 중단된 큐 관리자에게 손상을 입힐 수 있습니다.

`superuser` 또는 `mqm` 권한을 부여할 때 주의하십시오.

dmpmqlog 명령을 사용하여 로그의 콘텐츠 덤프

`dmpmqlog` 명령을 사용하여 큐 관리자 로그의 콘텐츠를 덤프하는 방법입니다.

`dmpmqlog` 명령을 사용하여 큐 관리자 로그의 콘텐츠를 덤프하십시오. 기본적으로 모든 활성 로그 레코드가 덤프됩니다. 즉, 이 명령은 로그의 헤드(일반적으로 마지막으로 완료된 체크포인트의 시작)부터 덤프를 시작합니다.

이 로그는 일반적으로 큐 관리자가 실행되고 있지 않을 때에만 덤프할 수 있습니다. 큐 관리자는 종료 중 체크포인트를 사용하기 때문에 로그의 활성 부분은 일반적으로 적은 수의 로그 레코드를 포함합니다. 그러나 `dmpmqlog` 명령을 사용하면 다음 옵션 중 하나를 사용하여 덤프의 시작 위치를 변경함으로써 보다 많은 로그 레코드를 덤프할 수 있습니다.

- 로그의 베이스부터 덤프를 시작하십시오. 로그의 베이스는 로그의 헤드를 포함하는 로그 파일에서 첫 번째 로그 레코드입니다. 이 경우에 덤프되는 추가 데이터의 양은 로그 파일에서의 로그 헤드 위치에 따라 다릅니다. 로그 헤드가 로그 파일의 시작부 가까이 있으면 적은 양의 추가 데이터만 덤프됩니다. 이 헤드가 로그 파일의 끝부분에 가까이 있으면 훨씬 더 많은 데이터가 덤프됩니다.
- 덤프의 시작 위치를 개별 로그 레코드로 지정하십시오. 각 로그 레코드는 고유 로그 순서 번호(LSN)로 식별됩니다. 순환 로깅의 경우 이 시작 로그 레코드는 로그의 베이스 앞에 있을 수 없습니다. 이 제한은 선형 로깅에는 적용되지 않습니다. 명령을 실행하기 전에 비활성 로그 파일을 복원해야 될 수도 있습니다. 이전 `dmpmqlog` 출력에서 가져온 유효한 LSN을 시작 위치로 지정해야 합니다.

예를 들어, 선형 로깅을 사용하면 마지막 `dmpmqlog` 출력에서 `nextlsn` 를 지정할 수 있습니다. `nextlsn` 은 Log File Header에 표시되고 기록될 다음 로그 레코드의 LSN을 표시합니다. 이를 로그가 마지막으로 덤프된 이후 기록된 모든 로그 레코드를 형식화하기 위한 시작 위치로 사용하십시오.

- **선형 로깅에만 해당됨**, 임의의 제공된 로그 파일 익스텐트에서 로그 레코드 형식화를 시작하도록 `dmpmqlog` 를 지시할 수 있습니다. 이 경우 `dmpmqlog` 는 이 로그 파일 및 활성 로그 파일과 같은 디렉토리에 있는 각각의 연속 로그 파일을 찾을 것으로 예상합니다. 이 옵션은 `dmpmqlog` 가 로그의 베이스 앞 로그 레코드에 액세스할 수 없는 순환 로그에는 적용되지 않습니다.

`dmpmqlog` 명령의 출력은 Log File Header 및 일련의 형식화된 로그 레코드입니다. 큐 관리자는 여러 로그 레코드를 사용하여 해당 데이터에 변경사항을 기록합니다.

형식화되는 정보 중 일부는 내부 전용입니다. 다음 목록에는 가장 유용한 로그 레코드가 들어 있습니다.

로그 파일 헤더

각 로그에는 단일 로그 파일 헤더가 있으며 이 헤더는 항상 `dmpmqlog` 명령에 의해 처음으로 형식화됩니다. 이 헤더에는 다음 필드가 포함됩니다.

로그 활동	기본 로그 익스텐트의 수입입니다.
로그 비활성	보조 로그 익스텐트의 수입입니다.
로그 크기	익스텐트당 4KB 페이지의 수입입니다.
기준선	로그의 헤드가 포함된 로그 익스텐트에서 첫 번째 LSN입니다.
<i>nextlsn</i>	기록될 다음 로그 레코드의 LSN입니다.
<i>headlsn</i>	로그의 헤드에 있는 로그 레코드의 LSN입니다.
테일런	로그 끝의 위치를 식별하는 LSN입니다.
<i>hflag1</i>	로그가 CIRCULAR 또는 LOG RETAIN(선형)인지 여부입니다.
<i>HeadExtentID</i>	로그의 헤드가 포함된 로그 익스텐트입니다.

로그 레코드 헤더

로그 내 각 로그 레코드에는 다음 정보를 포함하는 고정된 헤더가 있습니다.

<i>LSN</i>	로그 순서 번호입니다.
<i>LogRecd</i> 유형	로그 레코드의 유형입니다.
<i>XTranid</i>	이 로그 레코드와 연관된 트랜잭션 ID입니다(있는 경우). MQI의 <i>TranType</i> 은 WebSphere MQ 전용 트랜잭션을 표시합니다. <i>TranType</i> 이 XA이면 다른 자원 관리자와 관련됩니다. 같은 작업 단위 내에 포함된 업데이트는 같은 <i>XTranid</i> 를 가집니다.
<i>QueueName</i>	이 로그 레코드와 연관된 큐입니다(있는 경우).
<i>Qid</i>	큐의 고유 내부 ID입니다.
<i>PrevLSN</i>	같은 트랜잭션 내 이전 로그 레코드의 LSN입니다(있는 경우).

큐 관리자 시작

큐 관리자가 시작되었음을 로깅합니다.

<i>StartDate</i>	큐 관리자가 시작된 날짜입니다.
<i>StartTime</i>	큐 관리자가 시작된 시간입니다.

큐 관리자 중지

큐 관리자가 중지되었음을 로깅합니다.

<i>StopDate</i>	큐 관리자가 중지된 날짜입니다.
<i>StopTime</i>	큐 관리자가 중지된 시간입니다..
<i>ForceFlag</i>	사용된 시스템 종료의 유형입니다.

체크포인트 시작

큐 관리자 체크포인트의 시작을 나타냅니다.

체크포인트 종료

큐 관리자 체크포인트의 종료를 나타냅니다.

<i>ChkPtLSN</i>	이 체크포인트를 시작한 로그 레코드의 LSN입니다.
-----------------	------------------------------

메시지 넣기

큐에 넣은 지속 메시지를 로깅합니다. 메시지를 동기점 아래 넣은 경우 로그 레코드 헤더는 널이 아닌 *XTranid*를 포함합니다. 레코드의 나머지는 다음을 포함합니다.

<i>MapIndex</i>	큐에 있는 메시지의 ID입니다. 이는 큐에서 이 메시지를 가져오는 데 사용된 해당 MQGET을 일치시키는 데 사용할 수 있습니다. 이 경우 후속 <i>Get Message</i> 로그 레코드가 같은 <i>QueueName</i> 및 <i>MapIndex</i> 를 포함하는 것으로 확인될 수 있습니다. 이 때 <i>MapIndex</i> ID는 해당 큐에 대한 후속 넣기 메시지에 재사용할 수 있습니다.
데이터	이 로그 레코드의 16진수 덤프에 포함된 데이터는 다양한 내부 데이터로 뒤에 메시지 디스크립터(eyecatcher MD) 및 메시지 데이터 자체가 따라옵니다.

부분 넣기

단일 로그 레코드에 대해 너무 큰 지속적 메시지는 뒤에 하나의 *Put Message* 레코드가 오는 여러 *Put Part* 로그 레코드로 기록됩니다. *Put Part* 레코드가 있는 경우 *PrevLSN* 필드는 *Put Part* 레코드와 최종 *Put Message* 레코드를 함께 연결합니다.

데이터	이전 로그 레코드가 중단된 메시지 데이터가 계속됩니다.
-----	--------------------------------

메시지 가져오기

지속 메시지의 가져오기만 로깅됩니다. 동기점 아래로 메시지를 가져온 경우 로그 레코드 헤더는 널이 아닌 *XTranid*를 포함합니다. 레코드의 나머지는 다음을 포함합니다.

<i>MapIndex</i>	큐에서 검색된 메시지를 식별합니다. 동일한 <i>QueueName</i> 및 <i>MapIndex</i> 를 포함하는 가장 최근의 <i>Put Message</i> 로그 레코드가 검색된 메시지를 식별합니다.
<i>QPriority</i>	큐에서 검색되는 메시지의 우선순위입니다.

트랜잭션 시작

새 트랜잭션의 시작을 표시합니다. MQI의 *TranType*은 WebSphere MQ 전용 트랜잭션을 표시합니다. *TranType*이 XA이면 다른 자원 관리자가 관련된 트랜잭션을 나타냅니다. 이 트랜잭션이 작성한 모든 업데이트는 같은 *XTranid*를 가집니다.

트랜잭션 준비

큐 관리자가 지정된 *XTranid*와 연관된 업데이트를 커밋할 준비가 되었음을 표시합니다. 이 로그 레코드는 다른 자원 관리자도 관련되는 2단계 커밋의 일부로 기록됩니다.

트랜잭션 커밋

큐 관리자가 트랜잭션에서 작성된 모든 업데이트를 커밋했음을 표시합니다.

트랜잭션 롤백

트랜잭션을 롤백하려는 큐 관리자의 의도를 나타냅니다.

트랜잭션 종료

롤백된 트랜잭션의 종료를 나타냅니다.

트랜잭션 테이블

이 레코드는 동기점 중에 기록됩니다. 이 레코드는 지속적 업데이트를 해온 각 트랜잭션의 상태를 기록합니다. 각 트랜잭션에 대해 다음 정보가 기록됩니다.

<i>XTranid</i>	트랜잭션 ID입니다.
<i>FirstLSN</i>	트랜잭션과 연관된 첫 번째 로그 레코드의 LSN입니다.
<i>LastLSN</i>	트랜잭션과 연관된 마지막 로그 레코드의 LSN입니다.

트랜잭션 참가자

이 로그 레코드는 큐 관리자의 XA 트랜잭션 관리자에 의해 기록됩니다. 이 레코드는 트랜잭션에 참가하는 외부 자원 관리자를 기록합니다. 각 참가자에 대해 다음이 기록됩니다.

<i>RMName</i>	자원 관리자의 이름입니다.
<i>RMID</i>	자원 관리자 ID입니다. 자원 관리자가 참여하는 글로벌 트랜잭션을 기록하는 후속 <i>Transaction Prepared</i> 로그 레코드에도 로그됩니다.
<i>SwitchFile</i>	이 자원 관리자에 대한 스위치 로드 파일입니다.
<i>XAOpenString</i>	이 자원 관리자에 대한 XA 열기 문자열입니다.
<i>XACloseString</i>	이 자원 관리자에 대한 XA 닫기 문자열입니다.

트랜잭션 준비됨

이 로그 레코드는 큐 관리자의 XA 트랜잭션 관리자에 의해 기록됩니다. 이 레코드는 지정된 글로벌 트랜잭션이 성공적으로 준비되었음을 표시합니다. 참가하는 자원 관리자 각각은 커밋하도록 지시를 받습니다. 준비된 자원 관리자 각각의 *RMID*가 로그 레코드에 기록됩니다. 큐 관리자 자체가 트랜잭션에 참여하는 경우 *RMID*가 0인 *Participant Entry*가 존재합니다.

트랜잭션 정보 제거(forget)

이 로그 레코드는 큐 관리자의 XA 트랜잭션 관리자에 의해 기록됩니다. 커밋 결정이 각 참여자에게 전달된 경우 *Transaction Prepared* 로그 레코드를 따릅니다.

큐 영구 제거

MQSC 명령 CLEAR QUEUE 등을 사용하여 큐의 모든 메시지가 영구 되었다는 사실을 로깅합니다.

큐 속성

큐의 속성 초기화 또는 변경을 로깅합니다.

오브젝트 작성

WebSphere MQ 오브젝트의 작성을 로깅합니다.

<i>ObjName</i>	작성된 오브젝트의 이름입니다.
<i>UserId</i>	작성을 수행하는 사용자 ID입니다.

오브젝트 삭제

WebSphere MQ 오브젝트의 삭제를 로깅합니다.

<i>ObjName</i>	삭제된 오브젝트의 이름입니다.
----------------	------------------

IBM WebSphere MQ 큐 관리자 데이터 백업 및 복원

큐 관리자 및 큐 관리자 데이터를 백업합니다.

하드웨어 실패로 인해 발생할 수 있는 손상으로부터 큐 관리자를 보호하기 위한 조치를 주기적으로 수행할 수 있습니다. 큐 관리자를 보호하는 세 가지 방법이 있습니다.

큐 관리자 데이터 백업

하드웨어가 실패하는 경우 큐 관리자 중지가 강제 실행될 수 있습니다. 큐 관리자 로그 데이터가 하드웨어 실패로 인해 손실된 경우 큐 관리자를 재시작하지 못할 수 있습니다. 큐 관리자 데이터를 백업하는 경우 손실된 큐 관리자 데이터의 일부 또는 전체를 복구할 수도 있습니다.

일반적으로 큐 관리자 데이터를 백업하는 빈도가 높을수록 복구 로그의 무결성 손실의 원인이 되는 하드웨어 실패 시 손실되는 데이터가 줄어듭니다.

큐 관리자 데이터를 백업하려면 큐 관리자가 실행 중이면 안됩니다.

큐 관리자 데이터를 백업 및 복원하려면 다음을 참조하십시오.

- 377 페이지의 『[큐 관리자 데이터 백업](#)』.
- 377 페이지의 『[큐 관리자 데이터 복원](#)』.

백업 큐 관리자 사용

하드웨어 실패가 심각한 경우 큐 관리자는 복구 불가능할 수 있습니다. 이러한 상황에서 복구 불가능한 큐 관리자에 전용 백업 큐 관리자가 있을 경우 복구 불가능한 큐 관리자를 대신하여 이 백업 큐 관리자가 활성화될 수 있습니다. 정기적으로 업데이트된 경우 백업 큐 관리자 로그는 복구 불가능한 큐 관리자의 마지막 완료 로그가 포함된 로그 데이터를 포함할 수 있습니다.

기존 큐 관리자가 실행 중인 동안에도 백업 큐 관리자를 업데이트할 수 있습니다.

백업 큐 관리자를 작성하고 활성화하려면 다음을 참조하십시오.

- [378 페이지의 『백업 큐 관리자 작성』](#).
- [379 페이지의 『백업 큐 관리자 시작』](#).

큐 관리자 구성만 백업

하드웨어가 실패하는 경우 큐 관리자 중지가 강제 실행될 수 있습니다. 큐 관리자 구성 및 로그 데이터가 둘 다 하드웨어 실패로 인해 손실되는 경우 큐 관리자를 재시작하거나 로그에서 복구할 수 없습니다. 큐 관리자 구성을 백업하는 경우 저장된 정의에서 큐 관리자 및 그 오브젝트 모두를 다시 작성할 수 있습니다.

큐 관리자 구성을 백업하려면 큐 관리자가 실행 중이어야 합니다.

큐 관리자 구성을 백업 및 복원하려면 다음을 참조하십시오.

- [380 페이지의 『큐 관리자 구성 백업』](#)
- [380 페이지의 『큐 관리자 구성 복원』](#)

큐 관리자 데이터 백업

큐 관리자 데이터를 백업하면 하드웨어 오류로 인한 데이터 손실에 대해 보호하는 데 유용할 수 있습니다.

시작하기 전에

큐 관리자가 실행되고 있지 않은지 확인하십시오. 실행 중인 큐 관리자의 백업을 수행하려고 시도하면 파일이 복사될 때 진행 중인 업데이트로 인해 백업이 일치하지 않을 수 있습니다. 가능한 경우, `endmqm -w` 명령 (대기 종료) 을 실행하여 큐 관리자를 중지하십시오. 실패하는 경우에만 `endmqm -i` 명령 (즉시 종료) 을 사용하십시오.

이 태스크 정보

큐 관리자의 데이터 백업 사본을 작성하려면 다음 태스크를 완료하십시오.

1. 구성 파일의 정보를 사용하여 큐 관리자가 해당 데이터 및 해당 로그 파일을 매치한 디렉토리를 검색하십시오. 자세한 정보는 [380 페이지의 『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』](#)의 내용을 참조하십시오.
참고: 디렉토리에 표시된 이름을 이해하는 데 어려움이 있을 수 있습니다. 이러한 이름은 WebSphere MQ를 사용하는 플랫폼과 호환 가능하도록 변환됩니다. 이름 변환에 대한 자세한 정보는 [WebSphere MQ 파일 이름 이해](#)를 참조하십시오.
2. 모든 서브디렉토리를 포함하여 모든 큐 관리자의 데이터 및 로그 파일 디렉토리의 사본을 작성하십시오.
[363 페이지의 『로그의 표시』](#)에 설명된 모든 파일, 특히 로그 제어 파일과 [63 페이지의 『초기화 및 구성 파일』](#)에 설명된 구성 파일이 누락되지 않았는지 확인하십시오. 디렉토리 중 일부는 비어 있을 수 있지만 나중에 백업을 복원하려면 해당 디렉토리 모두 필요합니다.
3. 파일의 소유권을 보존하십시오. UNIX and Linux 시스템용 WebSphere MQ의 경우, `tar` 명령을 사용하여 이를 수행할 수 있습니다. 2GB를 초과하는 크기의 큐가 있는 경우 `tar` 명령을 사용할 수 없습니다. 자세한 정보는 [큰 큐를 사용으로 설정](#)을 참조하십시오.
참고: WebSphere MQ 버전 7.5 이상으로 업그레이드하는 경우, `.ini` 파일 및 레지스트리 입력 항목의 백업을 작성해 두십시오. 큐 관리자 정보는 `.ini` 파일에 저장되며 WebSphere MQ의 이전 버전으로 복귀할 때 사용할 수 있습니다.

큐 관리자 데이터 복원

다음 단계에 따라 큐 관리자 데이터의 백업을 복원합니다.

시작하기 전에

큐 관리자가 실행되고 있지 않은지 확인하십시오.

이 태스크 정보

큐 관리자의 데이터 백업을 복원하려면 다음을 수행하십시오.

1. 구성 파일의 정보를 사용하여 큐 관리자가 해당 데이터 및 해당 로그 파일을 매치한 디렉토리를 찾으십시오.
2. 백업된 데이터를 배치할 디렉토리를 비우십시오.
3. 백업된 큐 관리자 데이터 및 로그 파일을 올바른 위치에 복사하십시오.
4. 구성 정보 파일을 업데이트하십시오.

결과로 생성된 디렉토리 구조를 확인하여 필요한 모든 디렉토리가 있는지 확인하십시오.

IBM WebSphere MQ 디렉토리 및 서브디렉토리에 대한 자세한 정보는 [Windows 시스템에서 디렉토리 구조 및 UNIX and Linux 시스템에서 디렉토리 콘텐츠를 참조하십시오.](#)

로그 파일과 함께 로그 제어 파일이 있는지 확인하십시오. 또한 IBM WebSphere MQ 및 큐 관리자 구성 파일이 일치하는지 확인하여 WebSphere MQ가 올바른 위치에서 복원된 데이터를 찾을 수 있는지 확인하십시오.

순환 로깅의 경우 큐 관리자 데이터 및 로그의 일치 세트를 복원할 수 있도록 큐 관리자 데이터 및 로그 파일 디렉토리를 동시에 백업하십시오.

선형 로깅의 경우 큐 관리자 데이터 및 로그 파일 디렉토리를 동시에 백업하십시오. 로그 파일의 해당 완료 순서가 사용 가능한 경우 큐 관리자 데이터 파일만 복원할 수 있습니다.

참고: WebSphere MQ 버전 7.5 이상으로 업그레이드하는 경우, **.ini** 파일 및 레지스트리 입력 항목의 백업을 작성해 두십시오. 큐 관리자 정보는 **.ini** 파일에 저장되며 WebSphere MQ의 이전 버전으로 복귀할 때 사용할 수 있습니다.

결과

데이터가 올바르게 백업 및 복원된 경우 큐 관리자가 시작됩니다.

백업 큐 관리자 사용

기존 큐 관리자는 전용 백업 큐 관리자를 가질 수 있습니다.

백업 큐 관리자는 기존 큐 관리자의 비활성 사본입니다. 기존 큐 관리자가 심각한 하드웨어 실패로 인해 복구 불가능하게 되면 백업 큐 관리자가 온라인 상태가 되어 복구 불가능한 큐 관리자를 대체할 수 있습니다.

기존 큐 관리자 로그 파일은 정기적으로 백업 큐 관리자에 복사되어 백업 큐 관리자가 재해 복구를 위해 효과적인 메소드로 남을 수 있도록 해야 합니다. 로그 파일을 복사하기 위해 기존 큐 관리자를 반드시 중지해야 하는 것은 아닙니다. 그러나 큐 관리자가 로그 파일에 기록을 완료한 경우에만 로그 파일을 복사해야 합니다. 기존 큐 관리자 로그는 계속해서 업데이트되기 때문에 기존 큐 관리자 로그와 백업 큐 관리자 로그에 복사되는 로그 데이터 사이에는 항상 약간의 불일치가 있습니다. 백업 큐 관리자에 대한 정기적 업데이트는 이 두 로그 간 불일치를 최소화합니다.

백업 큐 관리자를 온라인으로 가져와야 하는 경우 백업 큐 관리자를 활성화한 다음 시작해야 합니다. 백업 큐 관리자가 시작되기 전에 활성화하는 요구사항은 백업 큐 관리자가 갑자기 시작되지 않도록 보호하기 위한 예방 조치입니다. 백업 큐 관리자가 활성화되면 더 이상 업데이트할 수 없습니다.

백업 큐 관리자를 작성하고 업데이트하고 시작하는 방법에 대한 정보는 다음 주제를 참조하십시오.

- [378 페이지의 『백업 큐 관리자 작성』](#)
- [379 페이지의 『백업 큐 관리자 업데이트』](#)
- [379 페이지의 『백업 큐 관리자 시작』](#)

백업 큐 관리자 작성

선형 로깅을 사용할 때에만 백업 큐 관리자를 사용할 수 있습니다.

기존 큐 관리자의 백업 큐 관리자를 작성하려면 다음을 수행하십시오.

1. 제어 명령 `crtmqm`을 사용하여 기존 큐 관리자의 백업 큐 관리자를 작성하십시오. 백업 큐 관리자에 대해 다음과 같은 요구사항이 있습니다.

- 기존 큐 관리자와 같은 속성(예: 큐 관리자 이름, 로깅 유형 및 로그 파일 크기)을 가져야 합니다.
 - 기존 큐 관리자와 같은 플랫폼에 있어야 합니다.
 - 기존 큐 관리자 이상의 코드 레벨에 있어야 합니다.
2. 377 페이지의 『큐 관리자 데이터 백업』에서 설명된 대로 기존 큐 관리자의 모든 데이터 및 모든 서브디렉토리를 포함한 로그 파일 디렉토리 사본을 작성하십시오.
 3. 모든 서브디렉토리를 포함하여 백업 큐 관리자의 데이터 및 로그 파일 디렉토리를 기존 큐 관리자에서 작성한 사본으로 덮어쓰십시오.
 4. 백업 큐 관리자에서 다음 제어 명령을 실행하십시오.

```
strmqm -r BackupQMName
```

이 명령은 WebSphere MQ에서 큐 관리자를 백업 큐 관리자로 플래그 지정하며 복사된 모든 로그 익스텐트를 재실행하여 기존 큐 관리자와 보조를 맞춰 백업 큐 관리자를 가져옵니다.

백업 큐 관리자 업데이트

백업 큐 관리자가 재해 복구에 효과적인 방법으로 사용되게 하려면 정기적으로 업데이트해야 합니다.

정기적 업데이트는 백업 큐 관리자 로그와 현재 큐 관리자 로그 간 불일치를 줄입니다. 백업할 큐 관리자를 중지하지 않아도 됩니다.

백업 큐 관리자를 업데이트하려면 다음을 수행하십시오.

1. 백업할 큐 관리자에서 다음 스크립트(MQSC) 명령을 발행하십시오.

```
RESET QMGR TYPE(ADVANCELOG)
```

이는 현재 로그에 대한 기록을 중지한 다음 큐 관리자 로깅을 다음 로그 익스텐트로 진행시킵니다. 이렇게 하면 현재까지 로깅된 모든 정보를 백업합니다.

2. 백업할 큐 관리자에서 다음 스크립트(MQSC) 명령을 발행하여 (새) 현재 활성 로그 익스텐트를 확보하십시오.

```
DIS QMSTATUS CURRLOG
```

3. 현재 큐 관리자 로그 디렉토리에서 백업 큐 관리자 로그 디렉토리로 업데이트된 로그 익스텐트 파일을 복사하십시오. 마지막 업데이트 이후의 모든 로그 익스텐트를 복사하고 2단계에서 언급한 현재 익스텐트까지 (포함하지는 않음). "S..."로 시작하는 로그 범위 파일만 복사하십시오.
4. 백업 큐 관리자에서 다음 제어 명령을 발행하십시오.

```
strmqm -r BackupQMName
```

이 명령은 복사된 모든 로그 익스텐트를 재실행하고 백업 큐 관리자를 큐 관리자가 있는 단계로 가져옵니다. 재실행이 완료되면 재시작 복구에 필요한 모든 로그 익스텐트와 매체 복구에 필요한 모든 로그 익스텐트를 식별하는 메시지를 수신합니다.

경고: 로그의 non-contiguous 세트를 백업 큐 관리자에 복사하면 첫 번째 누락된 로그를 찾은 지점까지의 로그만 재실행됩니다.

백업 큐 관리자 시작

복구 불가능한 큐 관리자를 백업 큐 관리자가 대신할 수 있습니다.

이를 수행하려면 다음 단계를 수행하십시오.

1. 다음 제어 명령을 실행하여 백업 큐 관리자를 활성화하십시오.

```
strmqm -a BackupQMName
```

백업 큐 관리자가 활성화됩니다. 현재 활성인 백업 큐 관리자는 더 이상 업데이트할 수 없습니다.

2. 다음 제어 명령을 실행하여 백업 큐 관리자를 시작하십시오.

```
strmqm BackupQMName
```

WebSphere MQ는 이를 재시작 복구로 보고 백업 큐 관리자의 로그를 이용합니다. 백업 큐 관리자에 대한 마지막 업데이트 중에 재실행이 발생할 수 있습니다. 따라서 마지막으로 기록된 체크포인트의 활성 트랜잭션만 롤백됩니다.

복구 불가능한 큐 관리자를 백업 큐 관리자가 대체하는 경우 복구 불가능한 큐 관리자의 일부 큐 관리자 데이터가 손실될 수 있습니다. 손실되는 데이터의 양은 백업 큐 관리자가 얼마나 최근에 마지막으로 업데이트되었는지에 따라 다릅니다. 마지막 업데이트가 최근일수록 큐 관리자 데이터 손실이 적어집니다.

3. 모든 채널을 재시작하십시오.

결과로 생성된 디렉토리 구조를 확인하여 필요한 모든 디렉토리가 있는지 확인하십시오.

WebSphere MQ 디렉토리 및 서브디렉토리에 대한 자세한 정보는 [파일 시스템 지원 계획](#) 를 참조하십시오.

로그 파일과 함께 로그 제어 파일이 있는지 확인하십시오. WebSphere MQ 및 큐 관리자 구성 파일이 일치하는지도 확인하여 WebSphere MQ가 올바른 위치에서 복원된 데이터를 찾을 수 있도록 하십시오.

데이터가 올바르게 백업 및 복원된 경우 큐 관리자가 시작됩니다.

참고: 큐 관리자 데이터 및 로그 파일이 서로 다른 디렉토리에 보유되더라도 이러한 디렉토리를 동시에 백업 및 복원하십시오. 큐 관리자 데이터 및 로그 파일이 서로 연령이 다른 경우 큐 관리자가 유효한 상태가 아니고 시작되지 않을 수 있습니다. 시작되지 않으면 데이터가 손상될 가능성이 높습니다.

큐 관리자 구성 백업

큐 관리자 구성 백업은 큐 관리자 정의에서 큐 관리자를 다시 빌드하는 데 유용할 수 있습니다.

큐 관리자 구성의 백업 사본을 작성하려면 다음을 수행하십시오.

1. 큐 관리자가 실행 중인지 확인하십시오.
2. a. AIX, HP-UX, Linux, Solaris 또는 의: (- f mqsc) MQSC 및 모든 속성 (-a) 의 디폴트 형식화 옵션을 사용하여 덤프 MQ Configuration 명령 (dmpmqcfg) 을 실행하고, 다음과 같이 표준 출력 경로 재지정을 사용하여 정의를 파일에 저장하십시오.

```
dmpmqcfg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

큐 관리자 구성 복원

다음 단계에 따라 큐 관리자 구성의 백업을 복원합니다.

큐 관리자 구성의 백업을 복원하려면 다음을 수행하십시오.

1. 큐 관리자가 실행 중인지 확인하십시오. 데이터 및 로그에 대한 손상을 다른 방법으로 복구할 수 없는 경우 큐 관리자가 다시 작성되었을 수도 있음을 참고하십시오.
2. 플랫폼에 따라 다음 명령 중 하나를 실행하십시오.
 - a. AIX, HP-UX, Linux, Solaris 또는 의 경우: 관리자에 대해 runmqsc를 실행하고, 표준 입력 경로 재지정을 사용하여 덤프 MQ Configuration (dmpmqcfg) 명령으로 생성된 스크립트 파일에서 정의를 복원하십시오. 예를 들면 다음과 같습니다.

```
runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
```

관련 참조

[dmpmqcfg](#)

IBM WebSphere MQ 및 큐 관리자 구성 정보 변경

설치 요구에 맞게 IBM WebSphere MQ의 작동 또는 개별 큐 관리자를 변경합니다.

IBM WebSphere MQ를 제어하는 구성 속성(또는 매개변수)의 세트에 지정된 값을 변경하여 IBM WebSphere MQ 구성 정보를 변경할 수 있습니다.

IBM WebSphere MQ 구성 파일을 편집하여 속성 정보를 변경하십시오. Windows 및 Linux (x86 및 x86-64 플랫폼)의 IBM WebSphere MQ에서 IBM WebSphere MQ Explorer를 사용하여 IBM WebSphere MQ 구성 파일을 편집할 수 있습니다.

Windows 시스템에서 `amqmdain`에 설명된 대로 `amqmdain`을 사용하여 구성 정보를 변경할 수도 있습니다.

사용자 플랫폼에 대한 IBM WebSphere MQ 및 큐 관리자 구성에 대한 추가 정보는 다음 하위 주제를 참조하십시오.

관련 개념

[5 페이지의 『구성』](#)

하나 이상의 컴퓨터에서 하나 이상의 큐 관리자를 작성하고 개발, 테스트, 프로덕션 시스템에서 해당 큐 관리자가 비즈니스 데이터를 포함하는 메시지를 처리하도록 구성합니다.

관련 태스크

[계획 중](#)

[WebSphere MQ 관리](#)

UNIX, Linux, and Windows 시스템에서 구성 정보 변경

구성 속성은 큐 관리자 및 노드 레벨의 구성 파일에 보유됩니다.

Windows, UNIX and Linux 플랫폼에서는 다음에 있는 IBM WebSphere MQ 구성 속성을 변경할 수 있습니다.

- 노드 전체에서 IBM WebSphere MQ에 대한 변경사항을 적용하기 위한 IBM WebSphere MQ 구성 파일 (`mqs.ini`). 각 노드에 대해 하나의 `mqs.ini` 파일이 있습니다.
- 특정 큐 관리자에 변경사항이 적용되도록 하는 큐 관리자 구성 파일(`qm.ini`)입니다. 노드에 있는 각 큐 관리자에 대해 하나의 `qm.ini` 파일이 있습니다.

클라이언트 구성 옵션은 클라이언트 구성 파일에 별도로 보유됩니다.

구성 파일(또는 스탠자 파일)에는 하나 이상의 스탠자가 포함되며, 이들 스탠자는 `.ini` 파일에서 모두 공통 기능을 갖거나 시스템의 부분(예: 로그 기능, 채널 기능 및 설치 가능한 서비스)을 정의하는 행 그룹입니다.

IBM WebSphere MQ 구성 파일은 큐 관리자와 연관된 데이터를 찾는 데 사용되므로 존재하지 않거나 잘못된 구성 파일이 일부 또는 모든 MQSC 명령을 실패하게 할 수 있습니다. 또한 애플리케이션은 IBM WebSphere MQ 구성 파일에 정의되지 않은 큐 관리자에 연결할 수 없습니다.

구성 파일에 작성하는 모든 변경사항은 일반적으로 큐 관리자가 다음에 다시 시작될 때까지 적용되지 않습니다.

Windows 및 Linux(x86 및 플랫폼) 시스템에서는 IBM WebSphere MQ Explorer에서 구성 정보를 편집할 수 있습니다.

Windows 시스템에서 `amqmdain` 명령을 사용하여 구성 파일을 편집할 수도 있습니다.

Windows, UNIX and Linux 시스템에서의 구성 옵션에 대한 자세한 정보는 다음 하위 주제를 참조하십시오.

관련 개념

[5 페이지의 『구성』](#)

하나 이상의 컴퓨터에서 하나 이상의 큐 관리자를 작성하고 개발, 테스트, 프로덕션 시스템에서 해당 큐 관리자가 비즈니스 데이터를 포함하는 메시지를 처리하도록 구성합니다.

[380 페이지의 『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』](#)

설치 요구에 맞게 IBM WebSphere MQ의 작동 또는 개별 큐 관리자를 변경합니다.

관련 태스크

[계획 중](#)

[WebSphere MQ 관리](#)

관련 참조

[386 페이지의 『IBM WebSphere MQ 구성 정보를 변경하기 위한 속성』](#)

On IBM WebSphere MQ for Windows systems and on IBM WebSphere MQ for Linux (x86 and x86-64 platforms) systems, modify configuration information using the IBM WebSphere MQ Explorer. 기타 시스템에서는 mqs.ini 구성 파일을 편집하여 이 정보를 수정합니다.

392 페이지의 『큐 관리자 구성 정보 변경』

여기에 설명된 속성은 개별 큐 관리자의 구성을 수정합니다. WebSphere MQ의 모든 속성을 대체합니다.

구성 파일 편집

명령 또는 표준 문서 편집기를 사용하여 구성 파일을 편집합니다.

구성 파일을 편집하기 전에, 필요에 따라 되돌릴 수 있도록 백업 사본을 만들어 두십시오.

다음과 같은 방법으로 구성 파일을 편집할 수 있습니다.

- 노드의 큐 관리자 구성을 변경하는 명령을 사용하여 자동으로 편집
- 표준 텍스트 편집기를 사용하여 수동으로 편집

설치 후 WebSphere MQ 구성 파일의 기본값을 편집할 수 있습니다.

구성 파일 속성에 올바르지 않은 값을 설정한 경우, 이 값을 무시하고 문제점을 표시하는 운영자 메시지가 발행됩니다. (결과는 속성 전체를 누락한 것과 동일합니다.)

새 큐 관리자를 작성할 때 다음을 수행하십시오.

- WebSphere MQ 구성 파일을 백업하십시오.
- 새 큐 관리자 구성 파일 백업

주석 텍스트 앞에 ";" 또는 "#" 문자를 추가하여 주석을 구성 파일에 포함시킬 수 있습니다. 주석을 나타내는 문자 없이 ";" 또는 "#" 문자를 사용하려는 경우 이 문자에 "\" 문자를 접두부로 지정할 수 있습니다. 이는 구성 데이터의 일부로 사용됩니다.

구성 파일을 언제 편집해야 합니까?

백업에서 복구하거나 큐 관리자를 이동, 기본 큐 관리자를 변경하거나 IBM 지원을 도와주기 위해 구성 파일을 편집합니다.

다음과 같은 경우에 구성 파일을 편집해야 할 수도 있습니다.

- 구성 파일이 손실되었습니다. (가능한 경우 백업에서 복구하십시오.)
- 하나 이상의 큐 관리자를 새 디렉토리로 이동해야 합니다.
- 기본 큐 관리자를 변경해야 합니다. 이는 기존 큐 관리자를 실수로 삭제한 경우에 발생할 수 있습니다.
- IBM 지원 센터에서 권유하는 경우

구성 파일 우선순위

속성의 값이 여러 위치에서 정의됩니다. 명령에서 설정된 속성이 구성 파일의 속성에 우선합니다.

구성 파일의 속성 값은 다음 우선순위에 따라 설정됩니다.

- 명령행에 입력된 매개변수가 구성 파일에 정의된 값보다 우선합니다.
- qm.ini 파일에서 정의된 값이 mqs.ini 파일에서 정의된 값에 우선합니다.

IBM WebSphere MQ 구성 파일, mqs.ini

IBM WebSphere MQ 구성 파일 mqs.ini는 노드에 있는 모든 큐 관리자에 대한 정보를 포함합니다. 이 파일은 설치 중 자동으로 작성됩니다.

UNIX and Linux 제품의 경우 IBM WebSphere MQ 에서 데이터 디렉토리 및 로그 디렉토리는 항상 /var/mqm 및 /var/mqm/log 입니다.

Windows 시스템에서 데이터 디렉토리 mqs.ini의 위치 및 로그 디렉토리의 위치는 다를 수 있으므로 레지스트리에 저장됩니다.

또한 Windows 시스템에서 Windows에 mqinst.ini 파일이 없으므로 설치 구성 정보 (UNIX and Linux 시스템의 IBM WebSphere MQ 에 있는 mqinst.ini 에 포함됨) 가 레지스트리에 있습니다.

Windows 시스템의 mqs.ini 파일은 HKLM\SOFTWARE\IBM\WepSphere MQ 키에 지정된 WorkPath 에 의해 제공됩니다. 이 파일은 다음을 포함합니다.

- 큐 관리자의 이름
- 기본 큐 관리자의 이름
- 파일 각각에 연관된 파일의 위치

새 IBM WebSphere MQ 설치의 제공된 LogDefaults 스탠자에 속성에 대한 명시 값이 포함되어 있지 않습니다. 속성 부족은 새 큐 관리자 작성 시 이 값에 대한 기본값이 사용됨을 의미합니다. 383 페이지의 그림 71의 LogDefaults 스탠자에 대해 기본값이 표시됩니다. LogBufferPages 속성의 값이 0이면 512를 의미합니다.

비기본값이 필요한 경우 LogDefaults 스탠자에 그 값을 명시적으로 지정해야 합니다.

```
#####  
#* Module Name: mqs.ini                                     *#  
#* Type       : WebSphere MQ Machine-wide Configuration File   *#  
#* Function   : Define WebSphere MQ resources for an entire machine *#  
#####  
#* Notes      :                                               *#  
#* 1) This is the installation time default configuration      *#  
#*                                                     *#  
#####  
AllQueueManagers:  
#####  
#* The path to the qmgrs directory, below which queue manager data *#  
#* is stored                                                    *#  
#####  
DefaultPrefix=/var/mqm  
  
LogDefaults:  
  LogPrimaryFiles=3  
  LogSecondaryFiles=2  
  LogFilePages=4096  
  LogType=CIRCULAR  
  LogBufferPages=0  
  LogDefaultPath=/var/mqm/log  
  
QueueManager:  
  Name=saturn.queue.manager  
  Prefix=/var/mqm  
  Directory=saturn!queue!manager  
  InstallationName=Installation1  
  
QueueManager:  
  Name=pluto.queue.manager  
  Prefix=/var/mqm  
  Directory=pluto!queue!manager  
  InstallationName=Installation2  
  
DefaultQueueManager:  
  Name=saturn.queue.manager  
  
ApiExitTemplate:  
  Name=OurPayrollQueueAuditor  
  Sequence=2  
  Function=EntryPoint  
  Module=/usr/ABC/auditor  
  Data=123  
  
ApiExitCommon:  
  Name=MQPoliceman  
  Sequence=1  
  Function=EntryPoint  
  Module=/usr/MQPolice/tmqp  
  Data=CheckEverything
```

그림 71. UNIX 시스템에 대한 IBM WebSphere MQ 구성 파일의 예제

큐 관리자 구성 파일, qm.ini

큐 관리자 구성 파일인 qm.ini에는 특정 큐 관리자에 관한 정보가 있습니다.

큐 관리자마다 한 개의 큐 관리자 구성 파일이 있습니다. qm.ini 파일은 연관된 큐 관리자가 작성될 때 자동으로 작성됩니다.

V7.5.0.9 IBM WebSphere MQ Version 7.5.0, 수정팩 9에서는 **strmqm** 명령이 큐 관리자를 완전히 시작하기 전에 qm.ini 파일에서 CHANNELS 및 SSL 스탠자의 구문을 확인하므로 **strmqm**에서 qm.ini 파일이 오류를 포함하고 있음을 발견하는 경우 문제점을 더 쉽게 확인하고 이를 빠르게 정정할 수 있습니다. 자세한 정보는 [strmqm](#)을 참조하십시오.

qm.ini 파일의 위치



UNIX and Linux 시스템에서 qm.ini 파일은 큐 관리자가 점유한 디렉토리 트리의 루트에 보유됩니다. 예를 들어, QMNAME이라는 큐 관리자의 구성 파일에 대한 경로 및 이름은 다음과 같습니다.

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

Windows 시스템에서 qm.ini 파일의 위치는 HKLM\SOFTWARE\IBM\WebSphere MQ 키에 지정된 WorkPath에 의해 제공됩니다. 예를 들어, QMNAME이라는 큐 관리자의 구성 파일에 대한 경로 및 이름은 다음과 같습니다.

```
C:\Program Files\IBM\WebSphere MQ\qmgrs\QMNAME\qm.ini
```

큐 관리자 이름의 최대 길이는 48자입니다. 그러나 이것은 이름이 올바르거나 고유하다는 것을 보장하지 않습니다. 따라서 디렉토리 이름은 큐 관리자 이름을 기반으로 생성됩니다. 이 프로세스를 이름 변환이라고 합니다. 설명은 [WebSphere MQ 파일 이름 이해](#)를 참조하십시오.

qm.ini 파일 예



다음 예제는 UNIX and Linux 시스템에 대한 IBM WebSphere MQ의 큐 관리자 구성 파일에 속성 그룹이 배열되는 방법을 보여줍니다.

```
## Module Name: qm.ini                ##
## Type       : WebSphere MQ queue manager configuration file  ##
## Function    : Define the configuration of a single queue manager ##
##           ##
##*****#
## Notes      :                ##
## 1) This file defines the configuration of the queue manager  ##
##           ##
##*****#

ExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

Service:
  Name=AuthorizationService
  EntryPoints=13

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module=opt/mqm/bin/amqzfu
  ComponentDataSize=0

Log:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=01
```

```

LogPath=/var/mqm/log/saturn!queue!manager/

AccessMode:
  SecurityGroup=wmq\wmq

XAResourceManager:
  Name=DB2 Resource Manager Bank
  SwitchFile=/usr/bin/db2swit
  XAOpenString=MQBankDB
  XACloseString=
  ThreadOfControl=THREAD

Channels: 2
  MaxChannels=200
  MaxActiveChannels=100
  MQIBindType=STANDARD

AccessMode:
  SecurityGroup=wmq\wmq
TCP:
  KeepAlive = Yes
  SvrSndBuffSize=32768
  SvrRcvBuffSize=32768
  Connect_Timeout=0

QMErrorLog:
  ErrorLogSize=262144
  ExcludeMessage=7234
  SuppressMessage=9001,9002,9202
  SuppressInterval=30

ApiExitLocal:
  Name=ClientApplicationAPIChecker
  Sequence=3
  Function=EntryPoint
  Module=/usr/Dev/ClientAppChecker
  Data=9.20.176.20

```

참고사항:

1. LogBufferPages의 값 0은 값 512를 제공합니다.
2. 채널 스탠자에 대한 자세한 정보는 63 페이지의 『초기화 및 구성 파일』을 참조하십시오.
3. XAResourceManager 스탠자의 최대 길이는 255로 제한됩니다. 그러나 트랜잭션 성능 저하를 피하려면 적은 수의 스탠자만 사용해야 합니다.

Unix에서 WebSphere MQ는 확장자가 .ini인 구성 파일(예: qm.ini)을 사용합니다. WebSphereMQ에는 파일의 임시 백업 사본을 작성할 **setmqm**과 같은 유틸리티가 있습니다. 예를 들어 qm.ini 파일은 qm.ini.bak 백업 사본을 작성합니다. 유틸리티는 qm.ini 파일을 수정하고, 업데이트된 파일을 저장한 후 qm.ini.bak 파일을 삭제합니다. 유틸리티가 qm.ini 파일을 저장할 수 없으면, 백업 파일 qm.ini.bak에서 qm.ini의 콘텐츠를 복원한 후 qm.ini.bak 파일을 삭제합니다.

기존의 qm.ini.bak 파일이 있으면, 유틸리티는 qm.ini.bak의 콘텐츠로 qm.ini 파일을 복원한 후 qm.ini.bak 파일을 삭제합니다. 이러한 백업 파일은 WebSphere MQ 유틸리티에서 삭제될 수 있으므로 .bak 파일 확장자를 사용하여 *.ini 파일의 백업 사본을 작성해서는 안 됩니다.

변경사항 변경 시기에 대한 정보는 381 페이지의 『UNIX, Linux, and Windows 시스템에서 구성 정보 변경』의 내용을 참조하십시오.

설치 구성 파일, mqinst.ini

UNIX and Linux 시스템

설치 구성 파일 mqinst.ini에는 UNIX 또는 Linux 시스템의 모든 IBM WebSphere MQ 설치에 대한 정보가 포함되어 있습니다.

mqinst.ini 파일은 UNIX and Linux 시스템의 /etc/opt/mqm 디렉토리에 있습니다. 여기에는 어떤 설치(있는 경우)가 기본 설치인지에 대한 정보는 물론, 각 설치에 대한 다음과 같은 정보가 포함되어 있습니다.

- 설치 이름
- 설치 설명

- 설치 ID
- 설치 경로

이 파일의 형식은 고정되지 않고 변경이 가능하므로 해당 파일을 직접 편집하거나 참조해서는 안됩니다. 대신 mqinst.ini 파일의 값을 작성, 삭제, 조회 및 수정하려면 다음 명령을 사용하십시오.

입력 항목을 작성하려는 경우 [crtmqinst](#)
 입력 항목을 삭제하려는 경우 [dlmqinst](#)
 입력 항목을 표시하려는 경우 [dspmqinst](#)
 입력 항목을 설정하려는 경우 [setmqinst](#)

내부 전용의 설치 ID는 자동으로 설정되므로 변경해서는 안됩니다.

Windows 시스템

설치 구성 정보는 Windows 시스템의 다음 키에 보유됩니다.

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\

```

이 키의 형식은 고정되지 않고 변경이 가능하므로 해당 키를 직접 편집하거나 참조해서는 안됩니다. 대신 레지스트리의 값을 조회 및 수정하려면 다음 명령을 사용하십시오.

입력 항목을 표시하려는 경우 [dspmqinst](#)
 입력 항목을 설정하려는 경우 [setmqinst](#)

Windows에서는 **crtmqinst** 및 **dlmqinst** 명령을 사용할 수 없습니다. 설치 및 설치 제거 프로세스는 필요한 레지스트리 입력 항목의 작성과 삭제를 핸들링합니다.

IBM WebSphere MQ 구성 정보를 변경하기 위한 속성

On IBM WebSphere MQ for Windows systems and on IBM WebSphere MQ for Linux (x86 and x86-64 platforms) systems, modify configuration information using the IBM WebSphere MQ Explorer. 기타 시스템에서는 mqs.ini 구성 파일을 편집하여 이 정보를 수정합니다.

특정 컴포넌트의 속성에 대해서는 다음 하위 주제를 참조하십시오.

관련 개념

[5 페이지의 『구성』](#)

하나 이상의 컴퓨터에서 하나 이상의 큐 관리자를 작성하고 개발, 테스트, 프로덕션 시스템에서 해당 큐 관리자가 비즈니스 데이터를 포함하는 메시지를 처리하도록 구성합니다.

[380 페이지의 『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』](#)

설치 요구에 맞게 IBM WebSphere MQ의 작동 또는 개별 큐 관리자를 변경합니다.

관련 태스크

[계획 중](#)

[WebSphere MQ 관리](#)

관련 참조

[392 페이지의 『큐 관리자 구성 정보 변경』](#)

여기에 설명된 속성은 개별 큐 관리자의 구성을 수정합니다. WebSphere MQ의 모든 속성을 대체합니다.

모든 큐 관리자

Use the General and Extended WebSphere MQ properties page from the IBM WebSphere MQ Explorer, or the AllQueueManagers stanza in the mqs.ini file to specify the following information about all queue managers.

DefaultPrefix=directory_name

이 속성은 큐 관리자 데이터가 보관되는 qmgrs 디렉토리에 대한 경로를 지정합니다.

큐 관리자의 기본 접두부를 변경하는 경우 설치 시 작성된 디렉토리 구조를 복제하십시오.

특히 qmgrs 구조를 작성해야 합니다. 기본 접두부를 변경하기 전에 WebSphere MQ를 중지한 후, 해당 구조를 새 위치로 이동시키고 기본 접두부를 변경한 후에만 WebSphere MQ를 재시작하십시오.

참고: UNIX and Linux 시스템의 /var/mqm/errors 디렉토리 또는 시스템 \errors 디렉토리를 삭제하지 마십시오.

기본 접두부 변경에 대한 대안으로 환경 변수 MQSPREFIX를 사용하여 crtmqm 명령의 DefaultPrefix를 대체할 수 있습니다.

운영 체제 제한이 있으므로 경로 길이 및 큐 관리자 이름의 합계가 최대 70자 길이가 되도록 제공된 경로의 길이를 짧게 유지하십시오.

ConvEBCDICNewline=NL_TO_LF|TABLE|ISO

ASCII 코드 페이지에서 지원되지 않는 줄 바꾸기(NL) 문자가 EBCDIC 코드 페이지에 있습니다(ASCII의 일부 ISO 변형에는 대등한 문자가 포함됨).

WebSphere MQ가 EBCDIC NL 문자를 ASCII 형식으로 변환하는 방법을 지정하려면 ConvEBCDICNewline 속성을 사용하십시오.

NL_TO_LF

EBCDIC를 모두 ASCII로 변환하려면 EBCDIC NL 문자(X'15')를 ASCII 줄 바꾸기 문자 LF(X'0A')로 변환하십시오.

NL_TO_LF가 기본값입니다.

TABLE

EBCDIC를 모두 ASCII로 변환하려면 사용자의 플랫폼에 사용되는 변환 테이블에 따라 EBCDIC NL 문자를 변환하십시오.

이 변환 유형의 결과는 플랫폼 및 언어에 따라 다르며 서로 다른 CCSID를 사용하는 경우 동일한 플랫폼에서도 다르게 작동합니다.

ISO

다음을 변환합니다.

- TABLE 메소드를 사용하여 ISO CCSID를 변환합니다.
- 기타 모든 CCSID는 NL_TO_CF 메소드를 사용하여 변환합니다.

가능한 ISO CCSID는 387 페이지의 표 32에 표시되어 있습니다.

표 32. 가능한 ISO CCSID 목록	
CCSID	코드 세트
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

ASCII CCSID가 ISO 서브세트가 아닌 경우 ConvEBCDICNewline은 NL_TO_LF로 기본값 설정됩니다.

기본 큐 관리자

Use the General WebSphere MQ properties page from the IBM WebSphere MQ Explorer, or the DefaultQueueManager stanza in the mq.ini file to specify the default queue manager.

Name=default_queue_manager

기본 큐 관리자는 큐 관리자 이름을 명시적으로 지정하지 않은 명령을 처리합니다. 새 기본 큐 관리자를 작성하면 DefaultQueueManager 속성이 자동으로 업데이트됩니다. 의도치 않게 기본 큐 관리자를 작성한 후 다시 원래의 기본 큐 관리자로 되돌리려는 경우 수동으로 DefaultQueueManager 속성을 대체하십시오.

엑시트 특성

큐 관리자 엑시트 프로그램에서 사용되는 구성 옵션을 지정하려면 IBM WebSphere MQ Explorer에서 Extended IBM WebSphere MQ 특성 페이지를 사용하거나 mqs.ini 파일의 ExitProperties 스탠자를 사용하십시오.

CLWLMode=SAFE|FAST

클러스터 워크로드 (CLWL) 종료를 사용하면 MQI 호출에 대한 응답으로 열리는 클러스터의 클러스터 큐 (예: MQOPEN, MQPUT) 를 지정할 수 있습니다. CLWL 종료는 CLWLMode 속성에 지정한 값에 따라 FAST 모드 또는 SAFE 모드에서 실행됩니다. CLWLMode 속성을 생략하면 클러스터 워크로드 엑시트가 SAFE 모드로 실행됩니다.

SAFE

큐 관리자와는 별도의 프로세스로 CLWL 엑시트를 실행하십시오. 기본값입니다.

SAFE 모드로 실행할 때 사용자 작성 CLWL 엑시트에서 문제점이 발생하면 다음이 발생합니다.

- CLWL 서버 프로세스(amqzlw0)가 실패합니다.
- 큐 관리자가 CLWL 서버 프로세스를 재시작합니다.
- 오류 로그에 오류가 보고됩니다. MQI 호출이 진행 중인 경우 리턴 코드 양식으로 된 알림이 수신됩니다.

큐 관리자의 무결성은 보존됩니다.

참고: 별도의 프로세스로 CLWL 엑시트를 실행하면 성능에 영향을 줄 수 있습니다.

FAST

큐 관리자 프로세스에서 클러스터 엑시트 인라인을 실행합니다.

이 옵션을 지정하면 SAFE 모드 실행과 연관된 프로세스 전환 비용이 발생하는 것을 방지하여 성능을 향상시키지만 큐 관리자 무결성은 손상됩니다. CLWL 엑시트에 문제점이 **없음**을 확인하고 성능에 대해 특히 걱정되는 경우에만 CLWL 엑시트를 FAST 모드로 실행해야 합니다.

CLWL 엑시트를 FAST 모드로 실행하는 동안 문제점이 발생하면 큐 관리자는 실패하고 큐 관리자의 무결성이 손상될 위험이 있습니다.

IBM WebSphere MQ 에 대한 로그 기본값

Use the Default log settings IBM WebSphere MQ properties page from the IBM WebSphere MQ Explorer, or the LogDefaults stanza in the mqs.ini file to specify information about log defaults for all queue managers.

스탠자가 없는 경우 MQ 기본값이 사용됩니다. 큐 관리자를 작성할 때 기본값으로 로그 기본값이 사용되지만 **crtmqm** 명령에서 로그 속성을 지정하는 경우에는 대체될 수 있습니다. 이 명령에 대한 자세한 내용은 **crtmqm** 을 참조하십시오.

큐 관리자가 작성되고 나면 해당 큐 관리자에 대한 로그 속성은 [395 페이지의 『큐 관리자 로그』](#) 에서 설명된 설정에서 가져옵니다.

특정 큐 관리자([395 페이지의 『큐 관리자 로그』](#) 에 지정됨)에 대해 지정된 기본 접두부([386 페이지의 『모든 큐 관리자』](#) 에 지정됨) 및 로그 경로는 큐 관리자와 로그를 서로 다른 물리적 드라이브에 있게 할 수 있습니다. 이는 기본적으로 같은 드라이브에 있더라도 권장되는 메소드입니다.

로그 크기 계산에 대한 정보는 [367 페이지의 『로그의 크기 계산』](#) 의 내용을 참조하십시오.

참고: 다음 매개변수 목록에 제공된 한계는 WebSphere MQ에 의해 설정되었습니다. 운영 체제 한계는 가능한 최대 로그 크기를 감소시킬 수 있습니다.

LogPrimaryFiles=3|2-254 (윈도우) |2-510 (UNIX and Linux 시스템)

로그 파일은 큐 관리자가 작성될 때 할당됩니다.

사용할 수 있는 최소 기본 로그 파일 수는 두 개이며 최대 기본 로그 파일 수는 Windows의 경우 254개이고 UNIX and Linux 시스템의 경우 510개입니다. 기본은 3입니다.

전체 기본 및 보조 로그 파일 수는 Windows의 경우 255개, UNIX and Linux 시스템의 경우 511개를 초과해서는 안되며 3개 미만이어서도 안됩니다.

큐 관리자가 작성되거나 시작될 때 값을 검토합니다. 큐 관리자가 작성된 후에는 이 값을 변경할 수 있습니다. 그러나 변경된 값은 큐 관리자가 재시작될 때까지 적용되지 않으며 즉시 적용되지 않을 수도 있습니다.

LogSecondaryFiles=2|1-253 (윈도우) |1-509 (UNIX and Linux 시스템)

로그 파일은 1차 파일이 모두 사용될 때 할당됩니다.

최소 보조 로그 파일 수는 한 개이며 최대 보조 로그 파일 수는 Windows의 경우 253개이고 UNIX and Linux 시스템의 경우 509개입니다. 기본 수는 2입니다.

전체 기본 및 보조 로그 파일 수는 Windows의 경우 255개, UNIX and Linux 시스템의 경우 511개를 초과해서는 안되며 3개 미만이어서도 안됩니다.

이 값은 큐 관리자가 시작될 때 조사됩니다. 이 값을 변경할 수 있지만 변경사항은 큐 관리자가 재시작될 때까지 적용되지 않으며 즉시 적용되지 않을 수도 있습니다.

LogFilePages=number

로그 데이터는 로그 파일이라는 일련의 파일에 보유됩니다. 로그 파일 크기는 4KB 페이지 단위로 지정됩니다.

로그 파일 기본 페이지 수는 4096이고 로그 파일 크기 16MB를 지정합니다.

UNIX and Linux 시스템에서 로그 파일 페이지의 최소 수는 64이며, Windows에서 로그 파일 페이지의 최소 수는 32입니다. 모두에서 최대 수는 65 535입니다.

참고: 큐 관리자 작성 중 지정된 로그 파일 크기는 큐 관리자에 대해 변경할 수 없습니다.

LogType=CIRCULAR|LINEAR

사용할 로그의 유형입니다. 기본값은 CIRCULAR입니다.

CIRCULAR

로그를 사용한 재시작 복구를 시작하여 시스템이 중지되었을 때 진행 중이었던 트랜잭션을 롤백합니다.

순환 로깅에 대한 자세한 설명은 [364 페이지의 『로깅 유형』](#)의 내용을 참조하십시오.

LINEAR

재시작 복구 및 매체나 정방향 복구(로그의 콘텐츠를 재실행하여 손실되거나 손상된 데이터 작성)에 사용됩니다.

선형 로깅에 대한 자세한 설명은 [364 페이지의 『로깅 유형』](#)의 내용을 참조하십시오.

기본값을 변경하려는 경우 LogType 속성을 편집하거나 `crtmqm` 명령을 사용하여 선형 로깅을 지정할 수 있습니다. 큐 관리자가 작성된 후에는 로깅 메소드를 변경할 수 없습니다.

LogBufferPages=0|0-4096

버퍼의 크기를 4KB 페이지 단위로 작성하고 지정하기 위해 버퍼 레코드에 할당되는 메모리의 양입니다.

최소 버퍼 페이지 수는 18이고 최대는 4096입니다. 버퍼가 클수록 처리량이 늘어나며 특히 메시지가 큰 경우가 이에 해당됩니다.

0(기본값)을 지정하면 큐 관리자가 크기를 선택합니다. WebSphere MQ 버전 7.0에서 이 값은 512(2048KB)입니다.

1 - 17 범위의 숫자를 지정하는 경우 큐 관리자는 18(72KB)로 기본값 설정됩니다. 18 - 4096 범위의 숫자를 지정하는 경우 큐 관리자는 지정된 숫자를 사용하여 할당되는 메모리를 설정합니다.

LogDefaultPath=directory_name

큐 관리자의 로그 파일이 상주하는 디렉토리입니다. 이 디렉토리는 가급적이면 메시지 큐와는 다른 드라이브에서 큐 관리자가 기록할 수 있는 로컬 디바이스에 상주합니다. 다른 드라이브를 지정하면 시스템이 실패할 경우에 추가 보호를 제공합니다.

기본값은 다음과 같습니다.

- <DefaultPrefix>\log for WebSphere MQ for 윈도우 where <DefaultPrefix> is the value specified on the DefaultPrefix attribute on the All Queue Managers WebSphere MQ properties page. 이 값은 설치 시 설정됩니다.
- /var/mqm/log for WebSphere MQ for UNIX and Linux systems

또는 -ld 플래그를 사용하여 crtmqm 명령에 디렉토리의 이름을 지정할 수 있습니다. 큐 관리자가 작성될 때 디렉토리도 큐 관리자 디렉토리 아래에서 작성되며 이 디렉토리는 로그 파일을 보유하는 데 사용됩니다. 이 디렉토리의 이름은 큐 관리자 이름을 기반으로 합니다. 이는 로그 파일 경로가 고유하도록 보장하고 디렉토리 이름 길이에 대한 제한을 준수하도록 합니다.

crtmqm 명령에 -ld를 지정하지 않으면 mqs.ini 파일의 LogDefaultPath 속성 값이 사용됩니다.

다중 큐 관리자가 서로 다른 로그 디렉토리를 사용하도록 큐 관리자 이름이 디렉토리 이름에 추가됩니다.

큐 관리자가 작성될 때 구성 정보의 로그 속성에서 LogPath 값이 작성되어 큐 관리자의 로그에 대한 전체 디렉토리 이름을 제공합니다. 이 값은 큐 관리자가 시작되거나 삭제될 때 로그를 찾는 데 사용됩니다.

LogWrite무결성 =**SingleWrite|DoubleWrite|TripleWrite**

로거가 신뢰할 수 있는 로그 레코드를 쓰기 위해 사용하는 메소드입니다.

TripleWrite

이는 기본 메소드입니다.

DoubleWrite를 선택할 수 있지만, 선택하면 시스템이 이를 **TripleWrite**로 해석한다는 점에 유의하십시오.

SingleWrite

WebSphere MQ 복구 로그를 호스트하는 파일 시스템 또는 디바이스가 명시적으로 4KB의 쓰기를 보장하는 경우에만 **SingleWrite**를 사용해야 합니다.

즉, 어떤 이유로 4KB 페이지를 쓰는 데 실패하는 경우에는 두 가지 상태(사전 이미지 또는 사후 이미지)만 가능합니다. 중간 상태는 발생하지 않아야 합니다.

고급 구성 및 전원 인터페이스(ACPI)

Use the ACPI WebSphere MQ properties page from the IBM WebSphere MQ Explorer, to specify how WebSphere MQ is to behave when the system receives a suspend request.

Windows는 ACPI(Advanced Configuration and Power Interface) 표준을 지원합니다. 이는 ACPI 사용 하드웨어를 사용하는 Windows 사용자가 시스템이 일시중단 모드에 들어가고 재개할 때 채널을 중지하고 재시작할 수 있도록 합니다.

ACPI WebSphere MQ 특성 페이지에 지정된 설정값은 경보 모니터가 실행 중일 때만 적용되는 점을 참고하십시오. 경보 모니터 아이콘은 경보 모니터가 실행 중인 경우 작업 표시줄에 표시됩니다.

DoDialog=Y | N

일시중단 요청 시 대화 상자에 표시됩니다.

DenySuspend=Y | N

일시중단 요청을 거부합니다. DoDialog=N이거나 DoDialog=Y인 경우에 사용되며 노트북 덮개가 닫힘 등으로 인해 대화 상자를 표시할 수 없습니다.

CheckChannelsRunning=Y | N

채널이 실행 중인지 여부를 검사합니다. 이 결과는 다른 설정의 결과를 판별할 수 있습니다.

다음 표에는 이러한 매개변수의 각 결합 효과가 요약되어 있습니다.

DoDialog	DenySuspend	CheckChannels 실행	Action
N	N	N	일시중단 요청을 승인합니다.
N	N	Y	일시중단 요청을 승인합니다.
N	Y	N	일시중단 요청을 거부합니다.

N	Y	Y	실행 중인 채널이 있는 경우 일시중단 요청을 거부하고 그렇지 않은 경우 요청을 승인합니다.
Y	N	N	대화 상자를 표시합니다(참고 참조. 일시중단 요청을 승인합니다). 기본값입니다.
Y	N	Y	실행 중인 채널이 없는 경우 일시중단 요청을 승인합니다. 있는 경우 대화 상자를 표시합니다(참고 참조. 요청을 승인합니다).
Y	Y	N	대화 상자를 표시합니다(참고 참조. 일시중단 요청을 거부합니다).
Y	Y	Y	실행 중인 채널이 없는 경우 일시중단 요청을 승인합니다. 있는 경우 대화 상자를 표시합니다(참고 참조. 요청을 거부합니다).

참고: 조치가 대화 상자를 표시하는 것인 경우에, 대화 상자를 표시할 수 없으면(예를 들어, 노트북 덮개가 닫힘으로 인해) DenySuspend 옵션이 일시중단 요청이 승인되는지 또는 거부되는지 판별하는 데 사용됩니다.

API 엑시트

IBM WebSphere MQ Explorer 또는 amqmdain 명령을 사용하여 API 엑시트에 대한 입력 항목을 변경합니다.

Use the Exits IBM WebSphere MQ properties page from the IBM WebSphere MQ Explorer, or the ApiExitTemplate and ApiExitCommon stanza in the mq.ini file to identify API exit routines for all queue managers. Windows 시스템에서 amqmdain 명령을 사용하여 API 엑시트에 대한 입력 항목을 변경할 수도 있습니다. (개별 큐 관리자에 대한 API 엑시트를 식별하려면 404 페이지의 『API 엑시트』에 설명된 대로 ApiExitLocal 스탠자를 사용합니다.)

이러한 스탠자의 속성에 대한 전체 설명은 [API 엑시트 구성](#)을 참조하십시오.

큐 관리자

큐 관리자마다 한 개의 QueueManager 스탠자가 있습니다. 스탠자를 사용하여 큐 관리자 디렉토리의 위치를 지정합니다.

Windows, UNIX and Linux 시스템에는 큐 관리자마다 한 개의 QueueManager 스탠자가 있습니다. 이러한 속성은 해당 큐 관리자와 연관된 파일이 들어 있는 디렉토리의 이름 및 큐 관리자 이름을 지정합니다. 이 디렉토리의 이름은 큐 관리자 이름을 기반으로 하지만 큐 관리자 이름이 유효한 파일 이름이 아닌 경우 변환됩니다. 이름 변환에 대한 자세한 정보는 [WebSphere MQ 파일 이름 이해](#)를 참조하십시오.

Name=queue_manager_name

대기열 관리자의 이름입니다.

Prefix=prefix

큐 관리자 파일이 저장되는 위치입니다. 기본적으로 이 값은 모든 큐 관리자 정보의 DefaultPrefix 속성에서 지정된 값과 같습니다.

Directory=name

큐 관리자 파일이 저장된 <prefix>\QMGRS 디렉토리 아래의 서브디렉토리 이름입니다. 이 이름은 큐 관리자 이름을 기반으로 하지만 중복 이름이 있거나 큐 관리자 이름이 유효한 파일 이름이 아닌 경우 변환될 수 있습니다.

DataPath=path

큐 관리자가 작성될 때 제공된 명시적 데이터 경로이며 큐 관리자 데이터에 대한 경로로서 접두부 및 디렉토리를 대체합니다.

InstallationName=name

이 큐 관리자와 연관된 WebSphere MQ 설치의 이름. 이 설치의 명령은 이 큐 관리자와 상호작용할 때 사용되어야 합니다. InstallationName 값이 없으면 버전 7.1 이전의 WebSphere MQ 설치에 큐 관리자가 연관됩니다.

관련 개념

15 페이지의 『큐 관리자를 설치와 연관』

큐 관리자를 작성하면 **crtmqm** 명령을 실행하는 설치와 자동으로 연관됩니다. UNIX, Linux, and Windows에서 **setmqm** 명령을 사용하여 큐 관리자와 연관된 설치를 변경할 수 있습니다.

보안

qm.ini 파일의 Security 스탠자를 사용하여 오브젝트 권한 관리자(OAM)에 대한 옵션을 지정합니다.

ClusterQueueAccessControl=RQMName|Xmitq

클러스터 큐 관리자에서 호스팅되는 완전한 큐 또는 클러스터 큐의 액세스 제어를 검사하기 위해 이 속성을 설정합니다.

RQMName

원격으로 호스팅되는 큐의 액세스 제어에 대해 검사된 프로파일을 큐 또는 큐 관리자 프로파일이라고 합니다.

Xmitq

원격으로 호스팅되는 큐의 액세스 제어에 대해 검사된 프로파일이 SYSTEM.CLUSTER.TRANSMIT.QUEUE로 해석됩니다.

Xmitq가 기본값입니다.

GroupModel=GlobalGroups

이 속성은 Windows에서 사용자의 그룹 멤버십을 판별할 때 OAM이 글로벌 그룹을 검사하는지 여부를 판별합니다.

기본값은 글로벌 그룹을 검사하지 않는 것입니다.

GlobalGroups

OAM이 글로벌 그룹을 검사합니다.

GlobalGroups가 설정되어 있으면, 권한 부여 명령 **setmqaut**, **dspmqaut** 및 **dmpmqaut**는 글로벌 그룹 이름을 승인합니다. **setmqaut -g** 매개변수를 참조하십시오.

참고: ClusterQueueAccessControl=RQMName을 설정하고 MQZAS_VERSION_6 이전 권한 서비스의 사용자 정의 구현이 있으면 큐 관리자가 시작되지 않습니다. 이 경우 ClusterQueueAccessControl=Xmitq를 설정하거나 사용자 정의 권한 서비스를 MQZAS_VERSION_6 이상으로 업그레이드하십시오.

큐 관리자 구성 정보 변경

여기에 설명된 속성은 개별 큐 관리자의 구성을 수정합니다. WebSphere MQ의 모든 속성을 대체합니다.

UNIX and Linux 시스템에서 qm.ini 구성 파일을 편집하여 큐 관리자 구성 정보를 수정합니다. qm.ini에 스탠자를 정의할 때 각 항목을 새로운 줄에서 시작할 필요는 없습니다. 세미콜론(;)이나 해시 문자(#)를 사용해서 주석을 나타낼 수 있습니다.

On Windows and Linux (x86 and x86-64 platforms) systems, you can modify some configuration information by using the IBM WebSphere MQ Explorer. 그러나 설치 가능 서비스와 해당 컴포넌트를 변경하는 것은 중대한 의미를 가지므로 설치 가능 서비스는 IBM WebSphere MQ Explorer에서 읽기 전용입니다. 따라서 Windows에서 **regedit** 를 사용하고 UNIX and Linux에서 qm.ini 파일을 편집하여 설치 가능한 서비스를 변경해야 합니다.

큐 관리자 구성 정보 변경에 대한 자세한 정보는 다음 하위 주제를 참조하십시오.

관련 개념

5 페이지의 『구성』

하나 이상의 컴퓨터에서 하나 이상의 큐 관리자를 작성하고 개발, 테스트, 프로덕션 시스템에서 해당 큐 관리자가 비즈니스 데이터를 포함하는 메시지를 처리하도록 구성합니다.

380 페이지의 『IBM WebSphere MQ 및 큐 관리자 구성 정보 변경』

설치 요구에 맞게 IBM WebSphere MQ의 작동 또는 개별 큐 관리자를 변경합니다.

관련 태스크

[계획 중](#)

[WebSphere MQ 관리](#)

관련 참조

[386 페이지의 『IBM WebSphere MQ 구성 정보를 변경하기 위한 속성』](#)

On IBM WebSphere MQ for Windows systems and on IBM WebSphere MQ for Linux (x86 and x86-64 platforms) systems, modify configuration information using the IBM WebSphere MQ Explorer. 기타 시스템에서는 mqs.ini 구성 파일을 편집하여 이 정보를 수정합니다.

액세스 모드

Access Mode는 Windows Server에만 적용됩니다. AccessMode 스탠자는 **crtmqm** 명령의 **-a [r]** 옵션에 의해 설정됩니다. 큐 관리자가 작성된 후에는 AccessMode 스탠자를 변경하지 마십시오.

어떤 Windows 보안 그룹 또는 구성원에게 모든 큐 관리자 데이터 파일에 대한 전체 액세스 권한이 부여되는지 지정하려면 **crtmqm** 명령의 액세스 그룹(**-a [r]**) 옵션을 사용하십시오. 그룹은 사용하는 구문에 따라서 로컬 또는 글로벌 그룹일 수 있습니다. 그룹 이름의 올바른 구문은 다음과 같습니다.

LocalGroup

Domain name \ GlobalGroup name

GlobalGroup name@Domain name

-a [r] 옵션과 함께 **crtmqm** 명령을 실행하기 전에 추가 액세스 그룹을 정의해야 합니다.

-a 대신 **-ar** 을 사용하여 그룹을 지정하는 경우, 로컬 **mqm** 그룹에는 큐 관리자 데이터 파일에 대한 액세스가 부여되지 않습니다. 큐 관리자 데이터 파일을 호스팅하는 파일 시스템이 로컬로 정의된 그룹에 대한 액세스 제어 입력 항목을 지원하지 않는 경우 이 옵션을 사용하십시오.

이 그룹은 공유 큐 관리자 데이터 및 로그 폴더에 대한 액세스 권한과 함께 다중 인스턴스 큐 관리자를 제공하는 데 사용되는 일반적인 글로벌 보안 그룹입니다. 폴더에 대해 읽기 및 쓰기 권한을 설정하거나 큐 관리자 데이터 및 로그 파일이 포함된 폴더를 공유하려면 추가 보안 액세스 그룹을 사용하십시오.

추가 보안 액세스 그룹은 **mqm(이)**라는 로컬 그룹을 사용하여 큐 관리자 데이터 및 로그를 포함하는 폴더에 대한 권한을 설정하는 것에 대한 대안입니다. 로컬 그룹 **mqm**과(와)는 달리, 추가 보안 액세스 그룹을 로컬 또는 전역 그룹으로 만들 수 있습니다. 다중 인스턴스 큐 관리자에 사용되는 데이터 및 로그 파일이 포함된 공유 폴더에 대한 권한을 설정하려면 글로벌 그룹이어야 합니다.

Windows 운영 체제는 큐 관리자 데이터 및 로그 파일을 읽고 쓰기 위한 액세스 권한을 확인합니다. 큐 관리자 프로세스를 실행하는 사용자 ID의 권한을 검사합니다. 확인되는 사용자 ID는 큐 관리자를 서비스로 시작했는지, 아니면 대화식으로 시작했는지에 따라 달라집니다. 큐 관리자를 서비스로 시작한 경우, Windows 시스템에서 선택한 사용자 ID는 **준비 IBM WebSphere MQ 마법사로** 구성된 사용자 ID입니다. 큐 관리자를 대화식으로 시작한 경우 Windows 시스템에서 검사하는 사용자 ID는 **strmqm** 명령을 실행한 사용자 ID입니다.

큐 관리자를 시작하려면 사용자 ID가 로컬 **mqm** 그룹의 구성원이어야 합니다. 사용자 ID가 추가 보안 액세스 그룹의 멤버일 경우 큐 관리자가 해당 그룹을 사용하여 권한이 부여된 파일을 읽고 쓸 수 있습니다.

제한사항: Windows 운영 체제에서만 추가적인 보안 액세스 그룹을 지정할 수 있습니다. 다른 운영 체제에서 추가적인 보안 액세스 그룹을 지정하는 경우 **crtmqm** 명령이 오류를 리턴합니다.

관련 개념

[342 페이지의 『Windows에서 비공유 큐 관리자 데이터와 로그 디렉토리 및 파일 보안 설정』](#)

[339 페이지의 『Windows에서 공유 큐 관리자 데이터 및 로그 디렉토리 보안』](#)

관련 태스크

[316 페이지의 『도메인 워크스테이션 또는 서버에서 다중 인스턴스 큐 관리자 작성』](#)

관련 참조

[crtmqm](#)

설치 가능 서비스

You change installable services on Windows by using **regedit**, and on UNIX and Linux by using the Service stanza in the `qm.ini` file.

참고: 설치 가능 서비스 및 해당 컴포넌트를 변경하는 것은 중요한 의미가 있습니다. 따라서 WebSphere MQ Explorer에서 설치 가능 서비스는 읽기 전용입니다.

Windows 시스템에서 설치 가능 서비스를 변경하려면 **regedit**를 사용하거나 UNIX and Linux 시스템에서 `qm.ini` 파일의 Service 스탠자를 사용하십시오. 서비스 내 각 컴포넌트에 대해 해당 컴포넌트의 코드를 포함하는 모듈의 이름 및 경로도 지정해야 합니다. UNIX and Linux 시스템에서는 `ServiceComponent` 스탠자를 사용하십시오.

Name=AuthorizationService|NameService

필요한 서비스의 이름입니다.

AuthorizationService

WebSphere MQ에서 권한 서비스 컴포넌트는 오브젝트 권한 관리자 또는 OAM입니다.

큐 관리자가 작성될 때 `AuthorizationService` 스탠자 및 그에 연관된 `ServiceComponent` 스탠자가 자동으로 추가됩니다. 다른 `ServiceComponent` 스탠자는 수동으로 추가하십시오.

NameService

기본적으로 이름 서비스는 제공되지 않습니다. 이름 서비스가 필요한 경우 `NameService` 스탠자를 수동으로 추가해야 합니다.

EntryPoints=number-of-entries

서비스에 대해 정의된 시작점의 수입니다. 여기에는 초기화 및 종료 시작점이 포함됩니다.

SecurityPolicy=Default|NTSIDsRequired (WebSphere MQ for 윈도우 only)

`SecurityPolicy` 속성은 지정된 서비스가 기본 권한 서비스, 즉, OAM인 경우에만 적용됩니다. `SecurityPolicy` 속성을 사용하면 각 큐 관리자에 대한 보안 정책을 지정할 수 있습니다. 가능한 값은 다음과 같습니다.

Default

적용할 기본 보안 정책을 사용합니다. Windows 보안 ID(NT SID)가 특정 사용자 ID의 OAM으로 전달되지 않을 경우 관련된 보안 데이터베이스를 검색하여 적절한 SID를 확보하기 위한 시도가 수행됩니다.

NTSIDsRequired

보안 검사를 수행할 때 NT SID를 OAM에 전달합니다.

자세한 정보는 [Windows 보안 ID\(SID\)](#)를 참조하십시오.

SharedBindingsUserId=user-type

`SharedBindingsUserId` 속성은 지정된 서비스가 기본 권한 서비스 즉, OAM인 경우에만 적용됩니다.

`SharedBindingsUserId` 속성은 공유 바인딩과 관련해서만 사용됩니다. 이 값을 사용하면

`MQZ_AUTHENTICATE_USER` 함수에서 `IdentityContext` 구조의 `UserIdentifier` 필드가 유효한 사용자 ID이거나 실제 사용자 ID입니다. `MQZ_AUTHENTICATE_USER` 함수에 대한 정보는 [MQZ_AUTHENTICATE_USER-인증 사용자](#)를 참조하십시오. 가능한 값은 다음과 같습니다.

Default

`UserIdentifier` 필드의 값이 실제 사용자 ID로 설정됩니다.

실제

`UserIdentifier` 필드의 값이 실제 사용자 ID로 설정됩니다.

Effective

`UserIdentifier` 필드의 값이 유효 사용자 ID로 설정됩니다.

FastpathBindingsUserId=user-type

`FastpathBindingsUserId` 속성은 지정된 서비스가 기본 권한 서비스 즉, OAM인 경우에만 적용됩니다.

`FastpathBindingsUserId` 속성은 빠른 경로 바인딩과 관련해서만 사용됩니다. 이 값을 사용하면

`MQZ_AUTHENTICATE_USER` 함수에서 `IdentityContext` 구조의 `UserIdentifier` 필드가 유효한 사용자 ID이거나 실제 사용자 ID입니다. `MQZ_AUTHENTICATE_USER` 함수에 대한 정보는 [MQZ_AUTHENTICATE_USER-인증 사용자](#)를 참조하십시오. 가능한 값은 다음과 같습니다.

Default

`UserIdentifier` 필드의 값이 실제 사용자 ID로 설정됩니다.

실제

UserIdentifier 필드의 값이 실제 사용자 ID로 설정됩니다.

Effective

UserIdentifier 필드의 값이 유효 사용자 ID로 설정됩니다.

IsolatedBindingsUserId =user-type

IsolatedBindingsUserId 속성은 지정된 서비스가 기본 권한 서비스, 즉, OAM인 경우에만 적용됩니다.

IsolatedBindingsUserId 속성은 분리된 바인딩에만 관련되어 사용됩니다. 이 값을 사용하면 MQZ_AUTHENTICATE_USER 함수에서 *IdentityContext* 구조의 *UserIdentifier* 필드가 유효한 사용자 ID이거나 실제 사용자 ID입니다. MQZ_AUTHENTICATE_USER 함수에 대한 정보는 [MQZ_AUTHENTICATE_USER-인증 사용자](#)를 참조하십시오. 가능한 값은 다음과 같습니다.

Default

UserIdentifier 필드의 값이 유효 사용자 ID로 설정됩니다.

실제

UserIdentifier 필드의 값이 실제 사용자 ID로 설정됩니다.

Effective

UserIdentifier 필드의 값이 유효 사용자 ID로 설정됩니다.

설치 가능 서비스 및 구성요소에 대한 자세한 정보는 [UNIX, Linux 및 Windows의 설치 가능 서비스 및 구성요소](#)를 참조하십시오.

일반적으로 보안 서비스에 대한 자세한 정보는 [Windows, UNIX and Linux 시스템에서 보안 설정을 참조하십시오](#).

관련 참조

[설치 가능 서비스 참조 정보](#)

서비스 컴포넌트

새 설치 가능 서비스를 설치할 때 서비스 컴포넌트 정보를 지정해야 합니다. Windows 시스템의 경우 `regedit`를 사용하고, UNIX and Linux 시스템의 경우 `qm.ini` 파일의 `ServiceComponent` 스탠자를 사용하십시오. 권한 서비스 스탠자는 기본적으로 표시되고 연관된 컴포넌트 OAM은 활성화됩니다.

다음과 같이 서비스 컴포넌트를 지정하십시오.

Service=service_name

필요한 서비스의 이름입니다. 이는 서비스 구성 정보의 `Name` 속성에서 지정된 값과 일치해야 합니다.

Name=component_name

서비스 컴포넌트의 설명 이름입니다. 이 이름은 고유해야 하며 WebSphere MQ 오브젝트의 이름(예: 큐 이름)에 올바른 문자만 포함해야 합니다. 이 이름은 서비스에서 생성되는 운영자 메시지에 나타납니다. 이 이름은 회사 상표 또는 이와 유사하게 구별되는 문자열로 시작하는 것이 좋습니다.

Module=module_name

이 컴포넌트에 대한 코드가 포함된 모듈의 이름입니다. 이 이름은 전체 경로 이름이어야 합니다.

ComponentDataSize=size

각 호출 시 컴포넌트에 전달된 컴포넌트 데이터 영역의 크기(바이트)입니다. 컴포넌트 데이터가 필요하지 않으면 0을 지정하십시오.

설치 가능 서비스 및 구성요소에 대한 자세한 정보는 [UNIX, Linux 및 Windows의 설치 가능 서비스 및 구성요소](#)를 참조하십시오.

큐 관리자 로그

IBM WebSphere MQ 탐색기의 Log 큐 관리자 특성 페이지 또는 `qm.ini` 파일의 Log 스탠자를 사용하여 큐 관리자에 대한 로깅에 대한 정보를 지정하십시오.

기본적으로 이러한 설정은 큐 관리자에 대한 기본 로그 설정에 대해 지정된 설정에서 상속됩니다(388 페이지의 『IBM WebSphere MQ에 대한 로그 기본값』에 설명됨). 이 큐 관리자를 다른 방식으로 구성하려는 경우에만 이러한 설정을 변경하십시오.

로그 크기 계산에 대한 정보는 367 페이지의 『로그의 크기 계산』의 내용을 참조하십시오.

참고: 다음 매개변수 목록에 제공된 한계는 WebSphere MQ에 의해 설정됩니다. 운영 체제 한계는 가능한 최대 로그 크기를 감소시킬 수 있습니다.

LogPrimary파일 =3 |2-254 (윈도우) |2-510 (UNIX and Linux 시스템)

로그 파일은 큐 관리자가 작성될 때 할당됩니다.

사용할 수 있는 최소 기본 로그 파일 수는 두 개이며 최대 기본 로그 파일 수는 Windows의 경우 254개이고 UNIX and Linux 시스템의 경우 510개입니다. 기본은 3입니다.

전체 기본 및 보조 로그 파일 수는 Windows의 경우 255개, UNIX and Linux 시스템의 경우 511개를 초과해서는 안되며 3개 미만이어도 안됩니다.

큐 관리자가 작성되거나 시작될 때 값을 검토합니다. 큐 관리자가 작성된 후에는 이 값을 변경할 수 있습니다. 그러나 변경된 값은 큐 관리자가 재시작될 때까지 적용되지 않으며 즉시 적용되지 않을 수도 있습니다.

LogSecondary파일 =2 |1-253 (윈도우) |1-509 (UNIX and Linux 시스템)

로그 파일은 1차 파일이 모두 사용될 때 할당됩니다.

최소 보조 로그 파일 수는 한 개이며 최대 보조 로그 파일 수는 Windows의 경우 253개이고 UNIX and Linux 시스템의 경우 509개입니다. 기본 수는 2입니다.

전체 기본 및 보조 로그 파일 수는 Windows의 경우 255개, UNIX and Linux 시스템의 경우 511개를 초과해서는 안되며 3개 미만이어도 안됩니다.

이 값은 큐 관리자가 시작될 때 조사됩니다. 이 값을 변경할 수 있지만 변경사항은 큐 관리자가 재시작될 때까지 적용되지 않으며 즉시 적용되지 않을 수도 있습니다.

LogFilePages=number

로그 데이터는 로그 파일이라는 일련의 파일에 보유됩니다. 로그 파일 크기는 4KB 페이지 단위로 지정됩니다.

로그 파일 기본 페이지 수는 4096이고 로그 파일 크기 16MB를 지정합니다.

UNIX and Linux 시스템에서 로그 파일 페이지의 최소 수는 64이며, Windows에서 로그 파일 페이지의 최소 수는 32입니다. 모두에서 최대 수는 65 535입니다.

참고: 큐 관리자 작성 중 지정된 로그 파일 크기는 큐 관리자에 대해 변경할 수 없습니다.

LogType=CIRCULAR|LINEAR

큐 관리자가 사용할 로깅 유형입니다. 큐 관리자가 작성되고 나면 사용할 로깅 유형을 변경할 수 없습니다. 필요한 로깅 유형으로 큐 관리자를 작성하는 것에 대한 정보는 388 페이지의 『IBM WebSphere MQ에 대한 로그 기본값』의 LogType 속성에 대한 설명을 참조하십시오.

CIRCULAR

로그를 사용한 재시작 복구를 시작하여 시스템이 중지되었을 때 진행 중이었던 트랜잭션을 롤백합니다.

순환 로깅에 대한 자세한 설명은 364 페이지의 『로깅 유형』의 내용을 참조하십시오.

LINEAR

재시작 복구 및 매체나 정방향 복구(로그의 콘텐츠를 재실행하여 손실되거나 손상된 데이터 작성)에 사용됩니다.

선형 로깅에 대한 자세한 설명은 364 페이지의 『로깅 유형』의 내용을 참조하십시오.

LogBufferPages=0|0-4096

버퍼의 크기를 4KB 페이지 단위로 작성하고 지정하기 위해 버퍼 레코드에 할당되는 메모리의 양입니다.

최소 버퍼 페이지 수는 18이고 최대는 4096입니다. 버퍼가 클수록 처리량이 늘어나며 특히 메시지가 큰 경우가 이에 해당됩니다.

0(기본값)을 지정하면 큐 관리자가 크기를 선택합니다. WebSphere MQ 버전 7.0에서 이 값은 512(2048KB)입니다.

1 - 17 범위의 숫자를 지정하는 경우 큐 관리자는 18(72KB)로 기본값 설정됩니다. 18 - 4096 범위의 숫자를 지정하는 경우 큐 관리자는 지정된 숫자를 사용하여 할당되는 메모리를 설정합니다.

이 값은 큐 관리자가 시작될 때 조사됩니다. 이 값은 명시된 한계 내에서 증가되거나 감소될 수 있습니다. 그러나 값의 변경은 다음에 큐 관리자가 시작될 때까지 적용되지 않습니다.

LogPath=directory_name

큐 관리자의 로그 파일이 상주하는 디렉토리입니다. 이 디렉토리는 가급적이면 메시지 큐와는 다른 드라이브에서 큐 관리자가 기록할 수 있는 로컬 디바이스에 존재해야 합니다. 다른 드라이브를 지정하면 시스템이 실패할 경우에 추가 보호를 제공합니다.

기본값은 다음과 같습니다.

- C:\Program Files\IBM\WebSphere MQ\log in WebSphere MQ for 윈도우.
- /var/mqm/log in WebSphere MQ for UNIX and Linux systems.

-ld 플래그를 사용하여 `crtmqm` 명령에 디렉토리의 이름을 지정할 수 있습니다. 큐 관리자가 작성될 때 디렉토리도 큐 관리자 디렉토리 아래에서 작성되며 이 디렉토리는 로그 파일을 보유하는 데 사용됩니다. 이 디렉토리의 이름은 큐 관리자 이름을 기반으로 합니다. 이는 로그 파일 경로가 고유하도록 보장하고 디렉토리 이름 길이에 대한 제한을 준수하도록 합니다.

`crtmqm` 명령에 -ld를 지정하지 않으면 `LogDefaultPath` 속성의 값이 사용됩니다.

UNIX and Linux 시스템의 WebSphere MQ 에서 사용자 ID `mqm` 및 그룹 `mqm`은 로그 파일에 대한 전체 권한을 가지고 있어야 합니다. 이들 파일의 위치를 변경할 경우, 해당 권한을 직접 부여해야 합니다. 이는 로그 파일이 제품에서 제공된 기본 위치에 있는 경우 필요하지 않습니다.

LogWrite무결성 =SingleWrite|DoubleWrite| TripleWrite

로거가 신뢰할 수 있는 로그 레코드를 쓰기 위해 사용하는 메소드입니다.

TripleWrite

이는 기본 메소드입니다.

DoubleWrite를 선택할 수 있지만, 선택하면 시스템이 이를 **TripleWrite**로 해석한다는 점에 유의하십시오.

SingleWrite

WebSphere MQ 복구 로그를 호스트하는 파일 시스템 또는 디바이스가 명시적으로 4KB의 쓰기를 보장하는 경우에만 **SingleWrite**를 사용해야 합니다.

즉, 어떤 이유로 4KB 페이지를 쓰는 데 실패하는 경우에는 두 가지 상태(사전 이미지 또는 사후 이미지)만 가능합니다. 중간 상태는 발생하지 않아야 합니다.

제한 모드

이 옵션은 UNIX and Linux 시스템에만 적용됩니다. `RestrictedMode` 스탠자는 `crtmqm` 명령의 -g 옵션에 의해 설정됩니다. 큐 관리자가 작성된 후에는 이 스탠자를 변경하지 마십시오. -g 옵션을 사용하지 않으면 스탠자가 `qm.ini` 파일에 작성되지 않습니다.

IBM WebSphere MQ 애플리케이션이 큐 관리자 데이터 디렉토리 내에서 큐 관리자에 연결되는 파일을 작성하는 몇 개의 디렉토리가 있습니다. 애플리케이션이 이들 디렉토리에 파일을 작성하기 위해서 세계 쓰기 액세스 권한이 부여됩니다.

- /var/mqm/sockets/QMgrName/@ipcc/ssem/hostname/
- /var/mqm/sockets/QMgrName/@app/ssem/hostname/
- /var/mqm/sockets/QMgrName/zsocketapp/hostname/

여기서 <QMGRNAME>은 큐 관리자의 이름이고 <hostname>은 호스트 이름입니다.

일부 시스템에서는 모든 사용자에게 이들 디렉토리에 대한 쓰기 액세스 권한을 부여하는 것을 허용할 수 없습니다. 예를 들어, 큐 관리자에 액세스할 필요가 없는 사용자에게는 부여하지 않습니다. 제한 모드는 큐 관리자 데이터를 저장하는 디렉토리의 권한을 수정합니다. 해당 디렉토리는 지정된 애플리케이션 그룹의 구성원만 액세스할 수 있습니다. 큐 관리자와 통신하는 데 사용되는 System V IPC 공유 메모리에 대한 권한도 같은 방법으로 수정됩니다.

애플리케이션 그룹은 다음을 수행할 권한을 갖고 있는 그룹의 이름입니다.

- MQI 애플리케이션 실행
- 모든 IPCC 자원 업데이트

- 일부 큐 관리자 디렉토리의 콘텐츠 변경

큐 관리자에 대해 제한 모드를 사용하려면,

- 큐 관리자의 작성자가 mqm 그룹 및 애플리케이션 그룹에 있어야 합니다.
- mqm 사용자 ID가 애플리케이션 그룹에 있어야 합니다.
- 큐 관리자를 관리하려는 모든 사용자는 mqm 그룹 및 애플리케이션 그룹에 있어야 합니다.
- IBM WebSphere MQ 애플리케이션을 실행하려는 모든 사용자는 애플리케이션 그룹에 있어야 합니다.

애플리케이션 그룹에 속하지 않은 사용자가 발행하는 모든 MQCONN 또는 MQCONNX 호출은 이유 코드 MQRC_Q_MGR_NOT_AVAILABLE로 실패합니다.

제한 모드는 IBM WebSphere MQ 권한 서비스와 함께 작동합니다. 따라서 사용자에게 IBM WebSphere MQ에 연결하고 IBM WebSphere MQ 권한 서비스를 사용할 때 필요한 자원에 액세스할 권한도 부여해야 합니다.

 IBM WebSphere MQ 권한 부여 서비스 구성에 대한 추가 정보는 [윈도우, UNIX and Linux 시스템에서 보안 설정](#)에서 찾을 수 있습니다.

권한 서비스가 제공하는 제어가 큐 관리자 자원의 충분한 격리를 제공하지 않을 때만 IBM WebSphere MQ 제한 모드를 사용하십시오.

XA 자원 관리자

IBM WebSphere MQ 탐색기의 XA resource manager 큐 관리자 특성 페이지 또는 qm.ini 파일의 XAResourceManager 스탠자를 사용하여 큐 관리자가 조정하는 글로벌 작업 단위와 관련된 자원 관리자에 대한 다음 정보를 지정하십시오.

글로벌 작업 단위에 참여하는 자원 관리자의 각 인스턴스에 대해 XA 자원 관리자 구성 정보를 수동으로 추가하십시오. 기본값은 제공되지 않습니다.

자원 관리자 속성에 관한 자세한 정보는 [데이터베이스 조정을 참조](#)하십시오.

Name=name(필수)

이 속성은 자원 관리자 인스턴스를 식별합니다.

Name 값은 길이가 최대 31 문자까지 가능합니다. XA-switch 속성에서 정의되는 자원 관리자의 이름을 사용할 수 있습니다. 그러나, 동일한 자원 관리자의 둘 이상의 인스턴스를 사용 중인 경우, 각 인스턴스에 대해 고유한 이름을 구성해야 합니다. 예를 들어 Name 문자열에 데이터베이스의 이름을 포함하여 고유성을 보장할 수 있습니다.

WebSphere MQ 는 dspmqtrn 명령의 메시지 및 출력에서 Name 값을 사용합니다.

연관된 큐 관리자가 시작했고 자원 관리자 이름이 적용된 후에는 자원 관리자 인스턴스의 이름을 변경하거나 구성 정보에서 해당 항목을 삭제하지 마십시오.

SwitchFile=name(필수)

자원 관리자의 XA 스위치 구조를 포함하는 로드 파일의 완전한 이름입니다.

32비트 애플리케이션에서 64비트 큐 관리자를 사용 중인 경우, name 값은 자원 관리자의 XA 스위치 구조를 포함하는 로드 파일의 기본 이름만 포함해야 합니다.

32비트 파일은 ExitsDefaultPath로 지정된 경로에서 애플리케이션으로 로드됩니다.

64비트 파일은 ExitsDefaultPath64로 지정된 경로에서 큐 관리자로 로드됩니다.

XAOpenString=string(선택적)

자원 관리자의 xa_open 시작점으로 전달될 데이터의 문자열입니다. 문자열의 콘텐츠는 자원 관리자 자체에 따라 다릅니다. 예를 들어, 문자열은 자원 관리자의 이 인스턴스가 액세스할 데이터베이스를 식별할 수 있습니다. 이 속성 정의에 대한 자세한 정보는 다음을 참조하십시오.

- [DB2에 대한 자원 관리자 구성 정보 추가](#)
- [Oracle에 대한 자원 관리자 구성 정보 추가](#)
- [Sybase의 자원 관리자 구성 정보 추가](#)

- [Informix® 에 대한 자원 관리자 구성 정보 추가](#)

그리고 적절한 문자열에 대해서는 자원 관리자 문서를 참조하십시오.

XACloseString=string(선택적)

자원 관리자의 xa_close 시작점으로 전달될 데이터의 문자열입니다. 문자열의 콘텐츠는 자원 관리자 자체에 따라 다릅니다. 이 속성 정의에 대한 자세한 정보는 다음을 참조하십시오.

- [DB2 에 대한 자원 관리자 구성 정보 추가](#)
- [Oracle에 대한 자원 관리자 구성 정보 추가](#)
- [Sybase의 자원 관리자 구성 정보 추가](#)
- [Informix 에 대한 자원 관리자 구성 정보 추가](#)

그리고 적절한 문자열에 대해서는 데이터베이스 문서를 참조하십시오.

ThreadOfControl=THREAD|PROCESS

This attribute is mandatory for WebSphere MQ for 윈도우. 큐 관리자는 자체 멀티스레드 프로세스 중 하나에서 자원 관리자를 호출해야 할 때 직렬화를 위해 이 값을 사용합니다.

THREAD

자원 관리자가 완전히 스레드 인지입니다. 멀티스레드된 WebSphere MQ 프로세스에서, XA 함수 호출이 멀티 스레드로부터 외부 자원 관리자로 동시에 작성될 수 있습니다.

PROCESS

자원 관리자가 스레드 안전하지 않습니다. 멀티스레드된 WebSphere MQ 프로세스에서, 자원 관리자에게 한 번에 하나의 XA 함수 호출만 작성될 수 있습니다.

ThreadOfControl 항목은 멀티스레드 애플리케이션 프로세스에서 큐 관리자에 의해 발행되는 XA 함수 호출에 적용되지 않습니다. 일반적으로 서로 다른 스레드에 동시 작업 단위를 갖는 애플리케이션은 이 조작 모드가 각 자원 관리자에 의해 지정되어야 합니다.

채널 스탠자의 속성

이 속성은 채널의 구성을 판별합니다.

이 정보는 z/OS 플랫폼용 WebSphere MQ에 적용할 수 없습니다.

채널에 대한 정보를 지정하려면 WebSphere MQ 탐색기의 Channels 큐 관리자 특성 페이지 또는 qm.ini 파일의 CHANNELS 스탠자를 사용하십시오.

MaxChannels=100|number

허용되는 현재 채널의 최대 수입입니다.

값은 1 - 65535 범위에 있어야 합니다. 기본값은 100입니다.

MaxActiveChannels=MaxChannels_value

언제든지 활성화될 수 있는 채널의 최대 수입입니다. 기본값은 MaxChannels 속성에 지정된 값입니다.

MaxInitiators=3|number

시작기의 최대 수입입니다. 기본값 및 최대값은 3입니다.

MQIBindType= FASTPATH | STANDARD

애플리케이션에 대한 바인딩:

FASTPATH

채널이 MQCONNX FASTPATH를 사용하여 연결합니다. 에이전트 프로세스가 없습니다.

STANDARD

채널이 STANDARD를 사용하여 연결합니다.

PipeLineLength=1|number

채널이 사용할 동시 스레드의 최대 수입입니다. 기본값은 1입니다. 1보다 큰 모든 값은 2로 처리됩니다.

파이프라이닝을 사용할 때는 채널의 양 끝에 있는 큐 관리자가 1보다 큰 PipeLineLength를 갖도록 구성하십시오.

참고: 파이프라이닝은 TCP/IP 채널에만 유효합니다.

AdoptNewMCA=NO|SVR|SDR|RCVR|CLUSRCVR|ALL|FASTPATH

WebSphere MQ에서 채널을 시작하는 요청을 수신하지만 채널의 인스턴스가 이미 실행 중임을 발견하면, 일부의 경우 새로 시작하기 전에 기존 채널 인스턴스를 중지해야 합니다. AdoptNewMCA 속성을 사용하여 이 방법으로 종료할 수 있는 채널 유형을 제어할 수 있습니다.

특정 채널 유형에 대해 AdoptNewMCA 속성을 지정하지만 일치하는 채널 인스턴스가 이미 실행 중이기 때문에 새 채널이 시작하지 못하는 경우,

1. 새 채널은 이전 채널을 종료하도록 요청하여 이전 채널을 중지하려고 합니다.
2. 이전 채널 서버가 AdoptNewMCATimeout 대기 간격이 만기할 때까지 이 요청에 응답하지 않는 경우 이전 채널 서버에 대한 스레드나 프로세스가 종료됩니다.
3. 이전 채널 서버가 단계 2 후에 종료되지 않고 AdoptNewMCATimeout 대기 간격이 두 번째 만기된 후에도 종료되지 않을 경우 WebSphere MQ는 CHANNEL IN USE 오류로 채널을 종료합니다.

AdoptNewMCA 기능은 서버, 송신자, 수신자 및 클러스터-서버 채널에 적용됩니다. 송신자 또는 서버 채널의 경우에는 특정 이름을 갖는 채널의 한 인스턴스만이 수신 큐 관리자에서 실행 중일 수 있습니다. 장애 복구 또는 클러스터-수신자 채널의 경우, 특정 이름을 갖는 채널의 복수 인스턴스가 수신 큐 관리자에서 실행 중일 수 있지만 임의의 시간에 특정 리모트 큐 관리자의 한 인스턴스만이 실행할 수 있습니다.

참고: AdoptNewMCA는 요청자 또는 서버 연결 채널에서 지원되지 않습니다.

다음 목록에서 하나 이상의 값을 쉼표나 공백으로 구분하여 지정하십시오.

아니오

AdoptNewMCA 기능은 필요하지 않습니다. 기본값입니다.

SVR

서버 채널을 채택합니다.

SDR

송신자 채널을 채택합니다.

RCVR

수신자 채널을 채택합니다.

CLUSRCVR

클러스터 수신자 채널을 채택합니다.

모두

FASTPATH 채널을 제외한 모든 채널 유형을 채택합니다.

FASTPATH

FASTPATH 채널인 경우 이 채널을 채택합니다. 이는 적절한 채널 유형도 지정된 경우에만 발생합니다 (예: AdoptNewMCA=RCVR, SVR, FASTPATH).

주의!: AdoptNewMCA 속성은 FASTPATH 채널에서 예측할 수 없는 형식으로 작동할 수 있습니다. FASTPATH 채널에서 AdoptNewMCA 속성을 사용하는 경우 신중하게 실행하십시오.

AdoptNewMCATimeout=60 |1 - 3600

새 채널 인스턴스가 이전 채널 인스턴스의 종료를 대기하는 시간(초)입니다. 1 - 3600 범위의 값을 지정하십시오. 기본값은 60입니다.

AdoptNewMCACheck=QM|ADDRESS|NAME|ALL

AdoptNewMCA 속성을 사용으로 설정할 때 필요한 검사의 유형입니다. 가능한 경우, 채널이 부주의하게 또는 악의적으로 시스템 종료되는 것을 보호하기 위한 전체 검사를 수행하십시오. 적어도 채널 이름이 일치하는지 확인하십시오.

QM, NAME 또는 ALL의 경우에 하나 이상의 값을 쉼표 또는 공백으로 구분하여 지정하십시오.

QM

큐 관리자 이름이 일치하는지 확인합니다.

큐 관리자 이름 자체는 일치하지만, QMID는 아닙니다.

ADDRESS

통신 소스 IP 주소를 검사하십시오. (예: TCP/IP 주소).

참고: 심표로 구분된 CONNAME 값이 대상 주소에 적용되므로 이 옵션과는 관련이 없습니다.

다중 인스턴스 큐 관리자가 hosta에서 hostb로 페일오버되는 경우, 해당 큐 관리자의 아웃바운드 채널은 hostb의 소스 IP 주소를 사용합니다. 이것이 hosta와 다를 경우 AdoptNewMCACheck=ADDRESS가 일치하지 않습니다.

상호 인증과 함께 SSL 또는 TLS를 사용하여 공격자가 기준에 실행 중인 채널을 방해하지 못하도록 할 수 있습니다. 또는 다중 인스턴스 큐 관리자 대신 IP 인수가 포함된 HACMP 유형 솔루션을 사용하거나 네트워크 로드 밸런서를 사용하여 소스 IP 주소를 마스킹하십시오.

이름

채널 이름이 일치하는지 확인하십시오.

모두

큐 관리자 이름 일치, 통신 주소 및 채널 이름 일치를 확인하십시오.

기본값은 AdoptNewMCACheck=NAME, ADDRESS, QM입니다.

관련 개념

50 페이지의 『채널 상태』

채널은 언제든지 여러 상태 중 하나일 수 있습니다. 일부 상태에는 하위 상태도 있습니다. 채널은 제공된 상태에서 다른 상태로 이동할 수 있습니다.

TCP, LU62, NETBIOS 및 SPX

네트워크 프로토콜 구성 매개변수를 지정하려면 이들 큐 관리자 특성 페이지 또는 qm.ini 파일의 스탠자를 사용하십시오. 채널의 기본 속성을 대체합니다.

TCP

Use the TCP queue manager properties page from the IBM WebSphere MQ Explorer, or the TCP stanza in the qm.ini file, to specify Transmission Control Protocol/Internet Protocol (TCP/IP) configuration parameters.

Port=1414|port_number

TCP/IP 세션에 대한 10진수 표기법의 기본 포트 번호입니다. WebSphere MQ의 잘 알려진 포트 번호는 1414입니다.

Library1=DLLName1 (WebSphere MQ for 윈도우 only)

TCP/IP 소켓 DLL의 이름입니다.

기본값은 WSOCK32입니다.

KeepAlive=NO|YES

KeepAlive 기능을 켜거나 끕니다. KeepAlive=YES로 설정하면 TCP/IP가 연결의 다른 측이 여전히 사용 가능한지 주기적으로 검사합니다. 그렇지 않은 경우 채널이 닫힙니다.

ListenerBacklog=number

TCP/IP 리스너에 대한 미해결 요청의 기본 수를 대체합니다.

TCP/IP에서 수신 시, 최대 미해결 연결 요청 수가 설정됩니다. 이 수는 리스너가 요청을 승인할 때까지 TCP/IP 포트에서 대기 중인 요청의 백로그로 간주될 수 있습니다. 기본 리스너 백로그 값은 [401 페이지](#)의 표 33에 표시됩니다.

표 33. 기본 미해결 연결 요청(TCP)	
플랫폼	기본 ListenerBacklog 값
	100
Windows 워크스테이션	5
Linux	100
Solaris	100
HP-UX	20

표 33. 기본 미해결 연결 요청(TCP) (계속)	
플랫폼	기본 ListenerBacklog 값
AIX V4.2 이상	100
AIX V4.1 또는 이전	10

참고: 일부 운영 체제는 표시된 기본값보다 더 큰 값을 지원합니다. 연결 한계 도달을 피하려면 이것을 사용하십시오.

반대로, 일부 운영 체제는 TCP 백로그의 크기를 제한할 수 있으므로 유효 TCP 백로그가 여기에서 요청된 것보다 더 작을 수 있습니다.

백로그가 401 페이지의 표 33에 표시된 값에 도달하는 경우 TCP/IP 연결은 거부되고 채널은 시작할 수 없습니다. 메시지 채널의 경우에는, 채널이 RETRY 상태로 가서 나중에 연결을 재시도하게 됩니다. 클라이언트 연결의 경우 클라이언트가 MQCONN으로부터 MQRC_Q_MGR_NOT_AVAILABLE 이유 코드를 수신하고 나중에 연결을 재시도합니다.

SvrSndBuffSize=32768|number

클라이언트 연결 서버 연결 채널의 서버 측에서 사용되는 TCP/IP 송신 버퍼의 크기(바이트)입니다.

SvrRcvBuffSize=32768|number

클라이언트 연결 서버 연결 채널의 서버 측에서 사용되는 TCP/IP 수신 버퍼의 크기(바이트)입니다.

Connect_Timeout=0|number

소켓 연결 시도가 제한시간 초과하기 전의 시간(초)입니다. 기본값 0은 연결 제한시간이 없음을 지정합니다.

LU62 (WebSphere MQ for 윈도우 only)

IBM WebSphere MQ 탐색기의 LU6.2 큐 관리자 특성 페이지 또는 qm.ini 파일의 LU62 스탠자를 사용하여 SNA LU 6.2 프로토콜 구성 매개변수를 지정하십시오.

TPName

원격 사이트에서 시작할 TP 이름입니다.

Library1=DLLName 1

APPC DLL의 이름입니다.

기본값은 WCPIC32입니다.

Library2=DLLName2

Library1과 동일하며 코드가 두 개의 별도 라이브러리에 저장된 경우에 사용됩니다.

기본값은 WCPIC32입니다.

NETBIOS (WebSphere MQ for 윈도우 only)

IBM WebSphere MQ 탐색기의 Netbios 큐 관리자 특성 페이지 또는 qm.ini 파일의 NETBIOS 스탠자를 사용하여 NetBIOS 프로토콜 구성 매개변수를 지정하십시오.

LocalName=name

이 머신이 LAN에서 알려지는 이름입니다.

AdapterNum=0|adapter_number

LAN 어댑터의 번호입니다. 기본값은 어댑터 0입니다.

NumSess=1|number_of_sessions

할당할 세션의 수입니다. 기본값은 1입니다.

NumCmds=1|number_of_commands

할당할 명령의 수입니다. 기본값은 1입니다.

NumNames=1|number_of_names

할당할 이름의 수입니다. 기본값은 1입니다.

Library1=DLLName1

NetBIOS DLL의 이름입니다.

기본값은 NETAPI32입니다.

SPX (WebSphere MQ for 윈도우 only)

IBM WebSphere MQ 탐색기의 SPX 큐 관리자 특성 페이지 또는 qm.ini 파일의 SPX 스탠자를 사용하여 SPX 프로토콜 구성 매개변수를 지정하십시오.

Socket=5E86|socket_number

16진 표기법의 SPX 소켓 번호입니다. 기본값은 X'5E86'입니다.

BoardNum=0|adapter_number

LAN 어댑터 번호입니다. 기본값은 어댑터 0입니다.

KeepAlive=NO|YES

KeepAlive 기능을 켜거나 끕니다.

KeepAlive=YES로 설정하면 연결의 다른 편 끝이 여전히 사용 가능한지 SPX가 주기적으로 검사합니다. 그렇지 않은 경우 채널이 닫힙니다.

Library1=DLLName1

SPX DLL의 이름입니다.

기본값은 WSOCK32.DLL입니다.

Library2=DLLName2

LibraryName1과 동일하며 코드가 두 개의 별도 라이브러리에 저장된 경우에 사용됩니다.

기본값은 WSOCK32.DLL입니다.

ListenerBacklog=number

SPX 리스너에 대한 미해결 요청의 기본 수를 대체합니다.

SPX에서 수신 시, 최대 미해결 연결 요청 수가 설정됩니다. 이는 SPX 소켓에서 리스너가 요청을 받아들 이기를 기다리고 있는 요청의 백로그로 간주될 수 있습니다. 기본 리스너 백로그 값은 [403 페이지의 표 34](#)에 표시됩니다.

표 34. 기본 미해결 연결 요청(SPX)	
플랫폼	기본 ListenerBacklog 값
	100
Windows 워크스테이션	5

참고: 일부 운영 체제는 표시된 기본값보다 더 큰 값을 지원합니다. 연결 한계 도달을 피하려면 이것을 사용하십시오.

반대로, 일부 운영 체제는 SPX 백로그의 크기를 제한할 수 있으므로 유효 SPX 백로그가 여기에서 요청된 것보다 더 작을 수 있습니다.

백로그가 [403 페이지의 표 34](#)에 표시된 값에 도달하는 경우 SPX 연결은 거부되고 채널은 시작할 수 없습니다. 메시지 채널의 경우에는, 채널이 RETRY 상태로 가서 나중에 연결을 재시도하게 됩니다. 클라이언트 연결의 경우 클라이언트가 MQCONN으로부터 MQRC_Q_MGR_NOT_AVAILABLE 이유 코드를 수신 하고 나중에 연결을 재시도해야 합니다.

엑시트 경로

IBM WebSphere MQ 탐색기의 Exits 큐 관리자 특성 페이지 또는 qm.ini 파일의 ExitPath 스탠자를 사용하여 큐 관리자 시스템의 사용자 엑시트 프로그램에 대한 경로를 지정하십시오.

ExitsDefaultPath=string

ExitsDefaultPath 속성은 다음의 위치를 지정합니다.

- 클라이언트의 32비트 채널 엑시트
- 서버의 32비트 채널 엑시트 및 데이터 변환 엑시트
- 규정되지 않은 XA 스위치 로드 파일

ExitsDefaultPath64=string

ExitsDefaultPath64 속성은 다음의 위치를 지정합니다.

- 클라이언트의 64비트 채널 엑시트
- 서버의 64비트 채널 엑시트 및 데이터 변환 엑시트
- 규정되지 않은 XA 스위치 로드 파일

API 엑시트

서버의 경우, IBM WebSphere MQ Explorer의 Exits 큐 관리자 특성 페이지 또는 `qm.ini` 파일의 `ApiExitLocal` 스탠자를 사용하여 큐 관리자에 대한 API 엑시트 루틴을 식별하십시오. 클라이언트의 경우 큐 관리자에 대한 API 엑시트 루틴을 식별하려면 `mqclient.ini` 파일의 `ApiExitLocal` 스탠자를 수정하십시오.

Windows 시스템에서는 `amqmdain` 명령을 사용하여 API 엑시트에 대한 항목을 변경할 수 있습니다. (모든 큐 관리자에 대한 API 엑시트 루틴을 식별하려면 391 페이지의 『API 엑시트』에서 설명하는 대로 `ApiExitCommon` 및 `ApiExitTemplate` 스탠자를 사용합니다.)

API 엑시트가 올바르게 작업하기 위해서는 서버의 메시지가 변환되지 않고 클라이언트로 전송되어야 함을 참고하십시오. API 엑시트가 메시지를 처리한 후, 메시지가 클라이언트에서 변환되어야 합니다. 그러므로 이것은 클라이언트에 모든 변환 엑시트를 설치했어야 합니다.

이러한 스탠자의 속성에 대한 전체 설명은 [API 엑시트 구성](#)을 참조하십시오.

UNIX, Linux, and Windows 의 QMErrorLog 스탠자

큐 관리자 오류 로그의 조작 및 콘텐츠를 사용자 정의하려면 WebSphere MQ Explorer에서 Extended 큐 관리자 특성 페이지를 사용하거나 `qm.ini` 파일의 `QMErrorLog` 스탠자를 사용하십시오.



주의: Windows 플랫폼에서 로컬 큐 관리자를 사용하는 경우에만 WebSphere MQ 탐색기를 사용하여 변경할 수 있습니다.

ErrorLogSize=maxsize

백업에 복사되는 큐 관리자 오류 로그의 크기를 지정합니다. `maxsize`는 32768 - 2147483648바이트 범위에 있어야 합니다. `ErrorLogSize`가 지정되지 않으면 기본값 2097152바이트(2MB)가 사용됩니다.

ExcludeMessage=msgIds

큐 관리자 오류 로그에 기록하지 않을 메시지를 지정합니다. WebSphere MQ 시스템이 많은 채널을 시작하고 중지하면서 과다 사용되는 경우, 많은 수의 정보 메시지가 z/OS 콘솔 및 하드카피 로그로 보내집니다. WebSphere MQ-IMS 브릿지 및 버퍼 관리자 또한 많은 수의 정보 메시지를 생성할 수 있습니다. 따라서 메시지 제외는 필요한 경우 많은 메시지 수가 수신되지 않게 합니다. `msgIds`에는 다음의 심표로 구분된 메시지 ID 목록이 포함됩니다.

- 5211 - 최대 특성 이름 길이가 초과되었습니다.
- 5973 - 분배된 발행/구독 등록이 억제됨
- 5974 - 분배된 발행/구독 등록이 억제됨
- 6254 - 시스템이 공유 라이브러리를 동적으로 로드할 수 없음
- 7234 - 로드된 메시지 수
- 9001 - 채널 프로그램이 정상적으로 종료됨
- 9002 - 채널 프로그램이 시작됨
- 9202 - 원격 호스트가 사용 불가능함
- 9208 - 호스트로부터 수신 시 오류
- 9209 - 연결 종료됨
- 9228 - 채널 응답자를 시작할 수 없음
- 9489 - SVRCONN 최대 인스턴스 한계 초과
- 9490 - 클라이언트당 SVRCONN 최대 인스턴스 한계 초과
- 9508 - 큐 관리자에 연결할 수 없음
- 9524 - 리모트 큐 관리자가 사용 불가능함
- 9528 - 사용자가 채널을 닫도록 요청함
- 9558 - 원격 채널 사용 불가능
- 9637 - 채널에 인증서 결핍
- 9776 - 채널이 사용자 ID에 의해 차단됨

- 9777 - 채널이 NOACCESS 맵에 의해 차단됨
- 9782 - 연결이 주소에 의해 차단됨
- 9999 - 채널 프로그램이 비정상적으로 종료됨

SuppressMessage=msgIds

지정된 시간 간격에 한 번만 큐 관리자 오류 로그에 기록되는 메시지를 지정합니다. WebSphere MQ 시스템이 많은 채널을 시작하고 중지하면서 과다 사용되는 경우, 많은 수의 정보 메시지가 z/OS 콘솔 및 하드카피 로그로 보내집니다. WebSphere MQ-IMS 브릿지 및 버퍼 관리자 또한 많은 수의 정보 메시지를 생성할 수 있습니다. 따라서 메시지를 억제하면 필요한 경우 반복되는 여러 메시지가 수신되지 않게 합니다. 시간 간격은 SuppressInterval에 의해 지정됩니다. msgIds에는 다음의 심표로 구분된 메시지 ID 목록이 포함됩니다.

- 5211 - 최대 특성 이름 길이가 초과되었습니다.
- 5973 - 분배된 발행/구독 등록이 억제됨
- 5974 - 분배된 발행/구독 등록이 억제됨
- 6254 - 시스템이 공유 라이브러리를 동적으로 로드할 수 없음
- 7234 - 로드된 메시지 수
- 9001 - 채널 프로그램이 정상적으로 종료됨
- 9002 - 채널 프로그램이 시작됨
- 9202 - 원격 호스트가 사용 불가능함
- 9208 - 호스트로부터 수신 시 오류
- 9209 - 연결 종료됨
- 9228 - 채널 응답자를 시작할 수 없음
- 9489 - SVRCONN 최대 인스턴스 한계 초과
- 9490 - 클라이언트당 SVRCONN 최대 인스턴스 한계 초과
- 9508 - 큐 관리자에 연결할 수 없음
- 9524 - 리모트 큐 관리자가 사용 불가능함
- 9528 - 사용자가 채널을 닫도록 요청함
- 9558 - 원격 채널 사용 불가능
- 9637 - 채널에 인증서 결핍
- 9776 - 채널이 사용자 ID에 의해 차단됨
- 9777 - 채널이 NOACCESS 맵에 의해 차단됨
- 9782 - 연결이 주소에 의해 차단됨
- 9999 - 채널 프로그램이 비정상적으로 종료됨

SuppressMessage와 ExcludeMessage에 같은 메시지 ID가 지정될 경우 해당 메시지가 제외됩니다.

SuppressInterval=length

SuppressMessage에 지정된 메시지가 큐 관리자 오류 로그에 한 번만 기록되는 시간 간격(초)을 지정합니다. length는 1 - 86400초 범위에 있어야 합니다. SuppressInterval이 지정되지 않으면 기본값 30초가 사용됩니다.

큐 관리자 기본 바인드 유형

IBM WebSphere MQ 탐색기의 Extended 큐 관리자 특성 페이지 또는 qm.ini 파일의 Connection 스탠자를 사용하여 기본 바인드 유형을 지정하십시오.

DefaultBindType=SHARED|ISOLATED

DefaultBindType이 ISOLATED로 설정되는 경우, 애플리케이션과 큐 관리자가 별도의 프로세스에서 실행하며 그들 사이의 자원이 공유되지 않습니다.

DefaultBindType이 SHARED로 설정되는 경우, 애플리케이션과 큐 관리자가 별도의 프로세스에서 실행하지만 그들 사이에 일부 자원이 공유됩니다.

기본값은 SHARED입니다.

큐 관리자 구성 파일의 SSL 및 TLS 스탠자

큐 관리자에서 SSL 또는 TLS 채널을 구성하려면 큐 관리자 구성 파일의 SSL 스탠자를 사용하십시오.

OCSP(Online Certificate Status Protocol)

인증서는 AuthorityInfoAccess 확장을 포함할 수 있습니다. 이 확장은 OCSP(Online Certificate Status Protocol)를 통해 접속될 서버를 지정합니다. 큐 관리자의 SSL 또는 TLS 채널이 AuthorityInfoAccess 확장을 사용하도록 허용하려면, 그 안에서 이름 지정된 OCSP 서버가 사용 가능하고 올바르게 구성되고 네트워크를 통해 액세스 가능한지 확인하십시오. 자세한 정보는 [폐기된 인증서에 대한 작업을 참조하십시오](#).

CrldistributionPoint(CDP)

인증서는 CrldistributionPoint 확장을 포함할 수 있습니다. 이 확장은 인증서 폐기 목록(CRL)을 다운로드하는 데 사용되는 프로토콜 및 접속될 서버를 둘 다 지정하는 URL을 포함합니다.

큐 관리자의 SSL 또는 TLS 채널이 CrldistributionPoint 확장을 사용하도록 허용하려면, 그 안에서 이름 지정된 CDP 서버가 사용 가능하고 올바르게 구성되고 네트워크를 통해 액세스 가능한지 확인하십시오.

SSL 스탠자

큐 관리자의 SSL 또는 TLS 채널이 다음 기능을 사용하려고 시도하는 방법 및 사용 시 문제점이 발생하는 경우 반응하는 방법을 구성하려면 `qm.ini` 파일에서 SSL 스탠자를 사용하십시오.

다음의 각 경우에서 제공되는 값이 나열된 올바른 값의 하나가 아닌 경우 기본값이 사용됩니다. 올바르지 않은 값이 지정됨을 언급하는 오류 메시지가 기록되지 않습니다.

CDPCheckExtensions=YES|NO

CDPCheckExtensions는 이 큐 관리자의 SSL 또는 TLS 채널이 CrldistributionPoint 인증서 확장에서 이름 지정되는 CDP 서버를 검사하려고 시도할지 여부를 지정합니다.

- YES: SSL 또는 TLS 채널이 디지털 인증서가 취소되었는지 여부를 판별하기 위해 CDP 서버를 검사하려고 시도합니다.
- NO: SSL 또는 TLS 채널이 CDP 서버를 검사하려고 시도하지 않습니다. 이 값은 기본값입니다.

OCSPAAuthentication=REQUIRED|WARN|OPTIONAL

OCSPAAuthentication은 OCSP 서버로부터 폐기 상태를 판별할 수 없을 때 취할 조치를 지정합니다.

OCSP 검사가 가능한 경우 SSL 또는 TLS 채널 프로그램이 OCSP 서버에 접속하려고 시도합니다.

채널 프로그램이 어떤 OCSP 서버에도 접속할 수 없는 경우 또는 인증서의 폐기 상태를 제공할 수 있는 서버가 없는 경우 OCSPAAuthentication 매개변수의 값이 사용됩니다.

- REQUIRED: 폐기 상태를 판별하지 못하면 연결이 오류와 함께 닫힙니다. 이 값은 기본값입니다.
- WARN: 폐기 상태를 판별하지 못하면 경고 메시지가 큐 관리자 오류 로그에 기록되지만, 연결은 계속될 수 있습니다.
- OPTIONAL: 폐기 상태를 판별하지 못하면 연결이 자동으로 진행할 수 있습니다. 경고나 오류가 제공되지 않습니다.

OCSPCheckExtensions=YES|NO

OCSPCheckExtensions는 이 큐 관리자의 SSL 및 TLS 채널이 AuthorityInfoAccess 인증서 확장자에서 이름 지정된 OCSP 서버를 검사하려고 시도할지 여부를 지정합니다.

- YES: SSL 및 TLS 채널이 디지털 인증서가 해지되는지 여부를 판별하기 위해 OCSP 서버를 검사하려고 시도합니다. 이 값은 기본값입니다.
- NO: SSL 및 TLS 채널이 OCSP 서버를 검사하려고 시도하지 않습니다.

SSLHTTPProxyName=string

문자열은 OCSP 검사를 위해 GSKit에서 사용될 HTTP 프록시 서버의 호스트 이름 또는 네트워크 주소입니다. 이 주소 뒤에 선택적 포트 번호를 괄호로 묶어 표시할 수 있습니다. 포트 번호가 지정되지 않으면 기본 HTTP

포트 80이 사용됩니다. HP-UX PA-RISC, Sun Solaris SPARC 플랫폼 및 AIX의 32-bit비트 클라이언트의 경우 네트워크 주소는 IPv4 주소만 사용할 수 있습니다. 기타 플랫폼에서는 IPv4 또는 IPv6 주소를 사용할 수 있습니다.

예를 들어 방화벽이 OCSP 응답자의 URL에 대한 액세스를 막는 경우 이 속성이 필요할 수 있습니다.

엑시트 특성

큐 관리자의 엑시트 특성에 관한 정보를 지정하려면 IBM WebSphere MQ Explorer의 클러스터 큐 관리자 특성 페이지 또는 qm.ini 파일의 ExitPropertiesLocal 스탠자를 사용하십시오. 그렇지 않으면 **amqmdain** 명령을 사용하여 설정할 수 있습니다.

기본적으로 이러한 설정값은 시스템 전반의 구성에 있는 ExitProperties 스탠자의 CLWLMode 속성에서 상속됩니다(388 페이지의 『엑시트 특성』에 설명됨). 이 큐 관리자를 다른 방법으로 구성하려는 경우에만 이 설정을 변경하십시오. 이 값은 클러스터 큐 관리자 특성 페이지의 클러스터 워크로드 모드 속성을 사용하여 개별 큐 관리자에 대해 대체될 수 있습니다.

CLWLMode=SAFE|FAST

클러스터 워크로드 (CLWL) 종료를 사용하면 MQI 호출에 대한 응답으로 열리는 클러스터의 클러스터 큐 (예: MQOPEN, MQPUT) 를 지정할 수 있습니다. CLWL 종료는 CLWLMode 속성에 지정한 값에 따라 FAST 모드 또는 SAFE 모드에서 실행된다. CLWLMode 속성을 생략하면 클러스터 워크로드 엑시트가 SAFE 모드로 실행됩니다.

SAFE

큐 관리자와는 별도의 프로세스로 CLWL 엑시트를 실행하십시오. 기본값입니다.

SAFE 모드로 실행할 때 사용자 작성 CLWL 엑시트에서 문제점이 발생하면 다음이 발생합니다.

- CLWL 서버 프로세스(amqzlw0)가 실패합니다.
- 큐 관리자가 CLWL 서버 프로세스를 재시작합니다.
- 오류 로그에 오류가 보고됩니다. MQI 호출이 진행 중인 경우 리턴 코드 양식으로 된 알림이 수신됩니다.

큐 관리자의 무결성은 보존됩니다.

참고: 별도의 프로세스로 CLWL 엑시트를 실행하면 성능에 영향을 줄 수 있습니다.

FAST

큐 관리자 프로세스에서 클러스터 엑시트 인라인을 실행합니다.

이 옵션을 지정하면 SAFE 모드 실행과 연관된 프로세스 전환 비용이 발생하는 것을 방지하여 성능을 향상시키지만 큐 관리자 무결성은 손상됩니다. CLWL 엑시트에 문제점이 **없음**을 확신하고 성능에 대해 특히 걱정되는 경우에만 CLWL 엑시트를 FAST 모드로 실행해야 합니다.

CLWL 엑시트를 FAST 모드로 실행하는 동안 문제점이 발생하면 큐 관리자는 실패하고 큐 관리자의 무결성이 손상될 위험이 있습니다.

서브풀

이 스탠자는 WebSphere MQ에 의해 작성되어 있습니다. 변경하지 마십시오.

스탠자 Subpool 및 이 스탠자 내의 속성 ShortSubpoolName은 큐 관리자를 작성할 때 WebSphere MQ에 의해 자동으로 작성됩니다. WebSphere MQ는 ShortSubpoolName의 값을 선택합니다. 이 값을 변경하지 마십시오.

이 이름은 WebSphere MQ가 실행 프로세스 사이에서 내부 통신에 사용하는 /var/mqm/sockets 디렉토리에 작성된 디렉토리 및 기호 링크에 해당합니다.

HP Integrity NonStop Server 구성

이 정보를 사용하여 HP Integrity NonStop Server 설치를 위한 IBM WebSphere MQ 클라이언트를 구성할 수 있습니다.

구성 파일을 사용하여 클라이언트를 구성하는 데 대한 자세한 내용은 [114 페이지의 『구성 파일을 사용하여 클라이언트 구성』](#)의 내용을 참조하십시오.

환경 변수를 사용하여 클라이언트를 구성하는 데 대한 자세한 내용은 [130 페이지의 『WebSphere MQ 환경 변수 사용』](#)의 내용을 참조하십시오.

TMF/Gateway에서 HP Integrity NonStop Server 조작에 대해 IBM WebSphere MQ 클라이언트를 수행하는 경우 TMF/Gateway를 구성하는 방법에 대한 정보는 하위 주제를 참조하십시오. 게이트웨이 프로세스 개요, 경로로 실행할 게이트웨이 구성 및 HP Integrity NonStop Server에 대한 IBM WebSphere MQ 클라이언트가 TMF 게이트웨이에 도달할 수 있도록 클라이언트 초기화 파일 구성이 포함됩니다.

이 절에는 채널에 대한 권한 부여에 대한 HP Integrity NonStop Server 특정 정보에 대한 IBM WebSphere MQ 클라이언트도 포함되어 있습니다.

게이트웨이 프로세스 개요

HP NonStop TMF(Transaction Management Facility)는 게이트웨이 프로세스를 자원 관리자로 등록할 수 있게 해 주는 서비스를 제공합니다. IBM WebSphere MQ 제공 TMF/게이트웨이 프로세스는 Pathway에서 실행됩니다.

IBM WebSphere MQ는 TMF에서 통합된 각 큐 관리자에 대해 하나의 게이트웨이 프로세스를 등록하므로 TMF 통합 작업 단위에 참여하는 각 큐 관리자에 대해 별도의 TMF/게이트웨이를 구성해야 합니다. 이 등록은 각 큐 관리자가 독립적인 자원 관리자가 되도록, 그리고 관리 목적을 위해, 이해하기 쉬운 맵핑을 통해 각 큐 관리자를 우선 HP NonStop TMF 결과에 등록합니다.

IBM WebSphere MQ의 다중 설치를 위해서는 TMF에서 통합할 각 큐 관리자에 대해 이러한 설치 중 하나의 단일 게이트웨이 프로세스를 지정해야 합니다.

게이트웨이 프로세스에 대한 인터페이스는 같은 버전 또는 이전 버전의 클라이언트를 지원합니다.

[HP Integrity NonStop Server 관리](#)를 참조하십시오.

Pathway에서 실행되도록 게이트웨이 구성

TMF/게이트웨이는 HP NonStop TMF(Transaction Management Facility)와 TMF가 IBM WebSphere MQ 트랜잭션을 위한 트랜잭션 통합기가 될 수 있게 해 주는 IBM WebSphere MQ 간의 인터페이스입니다.

IBM WebSphere MQ 제공 TMF/게이트웨이는 리모트 큐 관리자와 통신하기 위해 TMF 통합의 트랜잭션을 XA(eXtended Architecture) 트랜잭션 통합으로 변환합니다.

조정을 필요로 하는 큐 관리자당 하나의 TMF/게이트웨이를 갖고 있어야 하며 클라이언트가 올바른 게이트웨이에 연결할 수 있게 하기 위해 클라이언트 구성이 필요합니다.

TMF/게이트웨이는 큐 관리자와 통신하기 위해 클라이언트에서 사용 가능한 모든 메커니즘을 사용할 수 있습니다. 다른 애플리케이션과 같은 방식으로 TMF/게이트웨이를 구성하십시오.

TMF/게이트웨이는 HP Integrity NonStop Server 프로세스 쌍이 아니며 Pathway 환경에서 실행되도록 설계되었습니다. TMF/게이트웨이는 후속 실행에 재사용하는 영구 자원을 TMF 내에 작성하므로 TMF/게이트웨이는 항상 같은 사용자 권한으로 실행되어야 합니다.

serverclass 정의

TMF/게이트웨이는 Pathway 환경 내의 serverclass로 호스팅됩니다. serverclass를 정의하려면 다음 서버 속성을 설정해야 합니다.

PROCESSTYPE=OSS

serverclass의 서버 유형을 지정합니다. 게이트웨이 프로세스는 멀티스레드 OSS 프로그램입니다. 이 속성은 필수이며 OSS로 설정되어야 합니다.

MAXSERVERS=1

이 serverclass에서 동시에 실행될 수 있는 최대 서버 프로세스 개수를 지정합니다. 큐 관리자에 대해서는 하나의 게이트웨이 프로세스만 있을 수 있습니다. 이 속성은 필수이며 1로 설정되어야 합니다.

NUMSTATIC=1

이 serverclass 내의 최대 정적 서버 개수를 지정합니다. 게이트웨이 프로세스는 정적 서버로 실행되어야 합니다. 이 속성은 필수이며 1로 설정되어야 합니다.

TMF=ON

이 serverclass의 서버가 TMF 서브시스템이 감사한 데이터 파일을 잠그고 업데이트할 수 있는지 지정합니다. 게이트웨이 프로세스는 IBM WebSphere MQ 클라이언트 애플리케이션의 TMF 트랜잭션에 참여하므로 이 속성은 ON으로 설정되어야 합니다.

PROGRAM=<OSS installation path>/opt/mqm/bin/runmqtmf

IBM WebSphere MQ용 IBM WebSphere MQ 클라이언트의 경우 이 속성은 runmqtmf가 되어야 합니다. 이 속성은 절대 OSS 경로 이름이어야 합니다. 대소문자를 구분해야 합니다.

ARGLIST=-m<QMgr name>[-c<channel name>][,-p<port>][,-h<host name>][,-n<max threads>]

이 속성은 매개변수를 게이트웨이 프로세스에게 제공하며 여기서 각 부분의 의미는 다음과 같습니다.

- QMgrName은 이 게이트웨이 프로세스에 대한 큐 관리자의 이름입니다. 큐 공유 그룹(또는 다른 포트 분배 기술)을 사용 중인 경우 이 매개변수를 특정 큐 관리자로 대상 지정해야 합니다. 이 매개변수는 필수입니다.
- channel name은 게이트웨이 프로세스에 의해 사용될 큐 관리자에 서버 채널의 이름입니다. 이 매개변수는 선택사항입니다.
- port는 큐 관리자의 TCP/IP 포트입니다. 이 매개변수는 선택사항입니다.
- host name은 큐 관리자의 호스트 이름입니다. 이 매개변수는 선택사항입니다.
- max threads는 게이트웨이 프로세스에 의해 작성되는 최대 작업 프로그램 스레드 개수입니다. 이 매개변수의 값은 10 이상일 수 있습니다. 10보다 낮은 값이 지정되더라도 사용되는 가장 낮은 값은 10입니다. 어떤 값도 제공되지 않으면 게이트웨이 프로세스는 최대 50개의 스레드를 작성합니다.

409 페이지의 『환경 변수를 사용하여 TMF/게이트웨이 구성』에 설명된 방법 외에, 게이트웨이에 연결 정보를 제공하는 대체 방법으로서 -c, -p 및 -h 속성을 사용하십시오. -c, -p 및 -h 속성을 전부 지정하지는 않으면서 하나 이상 지정하는 경우에는 지정하지 않은 속성이 다음 값으로 기본 설정됩니다.

- channel name의 기본값은 SYSTEM.DEF.SVRCONN입니다.
- host name의 기본값은 localhost입니다.
- port의 기본값은 1414입니다.

제공한 매개변수가 올바르지 않은 경우 TMF/게이트웨이는 오류 로그에 진단 메시지 AMQ5379를 발행하고 종료됩니다.

OWNER=ID

게이트웨이를 실행하며 큐 관리자에 대한 연결 권한이 부여되어야 하는 사용자 ID입니다.

SECURITY="value"

Owner 속성과 관련하여, IBM WebSphere MQ 클라이언트 애플리케이션에서 게이트웨이에 액세스할 수 있는 사용자를 지정합니다.

LINKDEPTH 및 MAXLINKS는 게이트웨이와 동시에 통신하기 원할 수 있는 IBM WebSphere MQ 클라이언트 애플리케이션의 예상되는 수에 적절한 값으로 구성되어야 합니다. 이 값이 너무 낮게 설정되면 클라이언트 애플리케이션에서 오류 메시지 AMQ5399가 발행될 수 있습니다.

이러한 서버 속성에 대한 자세한 정보는 HP NonStop TS/mp 2.5 시스템 관리 매뉴얼을 참조하십시오.

환경 변수를 사용하여 TMF/게이트웨이 구성

TMF/게이트웨이를 정의하는 데 가장 일반적으로 사용되는 방법 중 하나는 MQSERVER 환경 변수를 설정하는 것입니다. 예:

```
ENV MQSERVER=<channel name>/<transport>/<host name>(<listener port>)
```

명령의 맨 처음에 있는 ENV는 Pathway 표기법입니다.

클라이언트 초기화 파일 구성

TMF (NonStop Transaction Management Facility) 를 사용하는 경우 HP Integrity NonStop Server 클라이언트가 TMF 게이트웨이에 도달할 수 있도록 IBM WebSphere MQ 클라이언트 초기화 파일이 있어야 합니다.

HP Integrity NonStop Server 에 대한 IBM WebSphere MQ 클라이언트 초기화 파일을 여러 위치에 보유할 수 있습니다. 자세한 정보는 [116 페이지의 『클라이언트 구성 파일의 위치』](#)의 내용을 참조하십시오.

구성 파일의 콘텐츠에 대한 자세한 내용과 예는 [114 페이지의 『구성 파일을 사용하여 클라이언트 구성』](#)의 내용을 참조하십시오. TMF 큐 관리자 및 서버 세부사항을 지정하려면 TMF 스탠자를 사용하십시오. 자세한 정보는 [130 페이지의 『TMF 및 TMF/게이트웨이 스탠자』](#)의 내용을 참조하십시오.

HP Integrity NonStop Server 에 대한 IBM WebSphere MQ 클라이언트의 항목 예제는 다음과 같습니다.

```
TMF:
  PathMon=$PSD1P

TmfGateway:
  QManager=MQ5B
  Server=MQ-MQ5B

TmfGateway:
  QManager=MQ5C
  Server=MQ-MQ5C
```

환경 변수를 사용하여 클라이언트를 구성하는 데 대한 자세한 정보는 [130 페이지의 『WebSphere MQ 환경 변수 사용』](#)의 내용을 참조하십시오.

채널에 권한 부여

HP Integrity NonStop Server 에 대한 IBM WebSphere MQ 클라이언트의 채널에 대한 권한을 부여하는 것은 다른 운영 체제와 동일하지만, 게이트웨이가 실행 중인 소유자의 식별을 알아야 합니다.

이를 알고 있으면 게이트웨이 소유자의 ID를 사용하여 적절한 권한을 부여할 수 있습니다. 중요한 차이는 큐 관리자 채널에 권한을 부여하는 것이 어느 애플리케이션의 권한도 아니라는 점입니다.

setmqaut 명령을 사용하여 IBM WebSphere MQ 프린시פל 또는 사용자 그룹에 권한을 부여하여 조작을 수행할 수 있고, 권한을 취소하고, 조작을 수행할 수 있는 권한을 제거합니다.

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 디버그를 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: