

7.5

IBM WebSphere MQ 개요

IBM

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [205 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

IBM WebSphere MQ 정보.....	5
IBM WebSphere MQ에 대한 소개.....	6
IBM WebSphere MQ 라이선스 정보.....	8
IBM WebSphere MQ Telemetry에 대한 소개.....	10
IBM WebSphere MQ Version 7.5 정보 로드맵.....	13
IBM Documentation 오프라인 애플리케이션의 IBM WebSphere MQ Version 7.5.....	16
IBM WebSphere MQ 7.5 PDF 문서.....	16
IBM WebSphere MQ 버전 7.5의 새로운 기능.....	17
HP Integrity NonStop Server용 IBM WebSphere MQ 클라이언트.....	19
IBM WebSphere MQ 버전 7.5의 변경사항.....	22
IBM WebSphere MQ Version 7.5 수정팩의 변경사항.....	23
새로운 기능 및 이전 버전에서 변경된 사항.....	31
이전 IBM WebSphere MQ 발행물과 현재 정보 구조 간 맵핑.....	31
AIX용 빠른 시작.....	31
HP-UX용 빠른 시작.....	31
Linux용 빠른 시작.....	32
Solaris용 빠른 시작.....	32
Windows용 빠른 시작.....	32
애플리케이션 프로그래밍 안내서.....	33
애플리케이션 프로그래밍 참조서.....	33
클라이언트.....	33
Constants.....	34
상호통신.....	34
메시지 및 코드.....	35
마이그레이션.....	35
모니터링.....	35
PCF(Programmable Command Format) 및 Administration Interface.....	35
발행/구독 사용자 안내서.....	35
큐 관리자 클러스터.....	35
스크립트(MQSC) 명령 참조.....	36
보안.....	36
시스템 관리 안내서.....	36
.NET 사용.....	36
C++ 사용.....	37
Java 사용.....	37
웹 서비스.....	37
컴포넌트 오브젝트 모델 인터페이스 사용.....	37
IBM WebSphere MQ 버전 7.5, IBM i 및 z/OS.....	37
주의사항.....	37
기술 개요.....	39
메시지 큐잉 소개.....	39
상호통신의 개념.....	46
IBM WebSphere MQ Telemetry.....	62
오브젝트 관리.....	95
IBM WebSphere MQ 멀티캐스트.....	116
보안.....	117
클라이언트 및 서버.....	118
트랜잭션 관리 및 지원.....	124
큐 관리자 기능 확장.....	126
HP Integrity NonStop Server용 IBM WebSphere MQ 클라이언트 기술 개요.....	126
시나리오.....	128
IBM WebSphere MQ 7.5 시작하기.....	128

스크립트를 사용하는 기본 파일 전송.....	134
세부적인 기본 파일 전송.....	142
스크립트를 사용하는 두 컴퓨터 파일 전송.....	148
세부적인 2 컴퓨터 파일 전송.....	154
관리 파일 전송에 감사 기능 추가.....	162
용어집.....	164
A.....	164
B.....	167
C.....	168
D.....	172
E.....	174
F.....	175
G.....	177
H.....	177
I.....	178
J.....	179
K.....	180
L.....	180
M.....	182
N.....	185
O.....	186
P.....	188
Q.....	190
R.....	191
S.....	194
T.....	198
U.....	200
V.....	201
W.....	201
X.....	202
액세스 용이성.....	203
Windows에서 액세스 용이성.....	204
주의사항.....	205
프로그래밍 인터페이스 정보.....	206
상표.....	206

정보 IBM WebSphere MQ

이 절에서는 IBM WebSphere MQ를 시작하는 데 도움이 되는 소개 정보를 제공합니다.

관련 태스크

[WebSphere MQ 아키텍처 설계](#)

IBM WebSphere MQ

IBM WebSphere MQ Version 7.5 제품 문서를 시작합니다. 여기서 MQ 환경을 작성하고 유지보수하기 위해 수행해야 하는 태스크를 완료하는 방법에 대한 자세한 지시사항을 찾을 수 있습니다. 이 문서는 또한 사용자가 비즈니스 문제를 해결하도록 도움이 되도록 제품을 이해할 수 있도록 돕는 개념적인 정보가 포함되어 있습니다.



IBM WebSphere MQ는 강력하고 안전하며 신뢰할 수 있는 메시징 미들웨어입니다. 이 제품은 메시지 및 큐를 사용하여 애플리케이션, 시스템, 서비스 및 파일 간의 정보 교환을 지원합니다. 다중 플랫폼에서 서로 다른 애플리케이션과 비즈니스 데이터의 통합을 단순화하고 가속합니다.

관련 개념

[IBM WebSphere MQ 정보](#)

관련 태스크

[계획 중](#)

[마이그레이션 및 업그레이드](#)

[설치](#)

[보안](#)

[구성](#)

[관리](#)

[애플리케이션 개발](#)

[모니터링 및 성능](#)

[문제점 해결 및 지원](#)

[WebSphere MQ Managed File Transfer](#)

[WebSphere MQ 탐색기](#)

[MQ GUI 마법사 도움말](#)

[WebSphere MQ 인터넷 Pass-Thru](#)

[Message Service Client for .NET](#)

[WebSphere MQ 하이퍼바이저 에디션](#)

[모바일 메시징과 M2M](#)

관련 참조

[참조](#)

관련 정보

[IBM MQ on Cloud](#)

IBM WebSphere MQ 소개

IBM WebSphere MQ를 사용하여 서로 다른 시간 및 다수의 다양한 컴퓨팅 환경에서 애플리케이션이 통신할 수 있게 할 수 있습니다.

IBM WebSphere MQ의 개념

- IBM WebSphere MQ은 애플리케이션을 위한 메시징입니다. 다양한 컴포넌트로 구성된 네트워크를 통해 메시지를 보냅니다. 애플리케이션은 IBM WebSphere MQ에 연결하여 메시지를 보내거나 받습니다. IBM WebSphere MQ은 메시지를 전송하면서 발견하는 여러 가지 다양한 프로세서, 운영 체제, 서브시스템, 통신 프로토콜을 핸들링합니다. 연결이나 프로세서가 일시적으로 사용 불가능한 경우, IBM WebSphere MQ은 메시지를 큐 처리하고 연결이 다시 온라인 상태로 돌아올 때 이를 전달합니다.
- 애플리케이션은 프로그래밍 인터페이스와 IBM WebSphere MQ에 연결할 프로그래밍 언어를 선택할 수 있습니다.
- IBM WebSphere MQ은 포인트-투-포인트, 발행/구독, 파일 전송 조작 모드를 가진 메시징 및 큐잉 미들웨어입니다. 애플리케이션에서 멀티캐스트를 통해 여러 구독자에게 메시지를 발행할 수 있습니다.

메시징

프로그램은 서로 직접 호출하기 보다는 메시지로 데이터를 서로 송신하여 통신합니다.

큐잉

메시지는 큐에 배치되므로 프로그램이 다른 속도와 시간에, 여러 위치에서, 서로 간에 직접적 연결 없이 독립적으로 실행될 수 있습니다.

포인트-투-포인트

애플리케이션에서 큐나 큐 목록에 메시지를 전송합니다. 송신자는 목적지의 이름을 알아야 하지만, 그 위치는 몰라도 됩니다.

공개/등록

애플리케이션에서 팀이 참가한 게임의 결과 등의 토픽에 대한 메시지를 발행합니다. IBM WebSphere MQ에서 결과 토픽을 구독하는 애플리케이션에 메시지 사본을 보냅니다. 팀이 참가한 게임 결과가 포함된 메시지를 받습니다. 발행자는 구독자의 이름이나 그 위치를 모릅니다.

멀티캐스트

멀티캐스트는 다수의 구독자까지 범위가 미치는 효율적인 발행/구독 메시징 형식입니다. IBM WebSphere MQ에서 네트워크로 각 구독자에게 발행물의 사본을 전송합니다. 발행자와 구독자 간에 발행물의 경로가 설정되면 IBM WebSphere MQ가 발행물 전달에 관여하지 않습니다.

파일 전송

파일은 메시지에 포함되어 전송됩니다. IBM WebSphere MQ File Transfer Edition은 파일 전송과 자동 전송을 설정하고 결과를 로그로 기록하기 위한 관리 작업을 관리합니다. 다른 파일 전송 시스템, IBM WebSphere MQ 메시징 및 웹과 파일 전송을 통합할 수 있습니다.

Telemetry

IBM WebSphere MQ Telemetry는 디바이스를 위한 메시징입니다. IBM WebSphere MQ은 디바이스 및 애플리케이션 메시징을 함께 연결합니다. 그리고 인터넷, 애플리케이션, 서비스 및 의사결정자를 계측 디바이스의 네트워크와 연결합니다. IBM WebSphere MQ 텔레메트리에는 네트워크를 통해 많은 디바이스를 연결하는 효율적인 메시징 프로토콜이 있습니다. 메시징 프로토콜은 디바이스에 통합될 수 있도록 발행됩니다. 또한, 프로토콜에 대해 발행된 프로그래밍 인터페이스 중 하나로 디바이스 프로그램을 개발할 수 있습니다.

사용 목적

- IBM WebSphere MQ는 네트워크에서 해당 애플리케이션 간의 데이터를 송신하고 수신합니다.
- 메시지 전달은 보장되고, 애플리케이션과 구분됩니다. IBM WebSphere MQ이 트랜잭션 방식으로 메시지를 교환하고 분리하고, 애플리케이션이 자신이 보낸 메시지가 안전하게 전달되는지 확인할 필요가 없기 때문에 확실합니다.
- SSL/TLS로 큐 관리자 간에 안전하게 메시지를 배달할 수 있습니다.
- AMS(Advanced Message Security)로 한 애플리케이션에서 메시지를 넣고 다른 애플리케이션에서 이를 검색하는 사이에 메시지를 암호화하고 서명할 수 있습니다.
- 애플리케이션 프로그래머에게 통신 프로그래밍 지식이 요구될 필요가 없습니다.

사용 방법

- IBM WebSphere MQ Explorer GUI를 사용하거나 명령 창 또는 애플리케이션에서 명령을 실행하여 IBM WebSphere MQ를 작성 및 관리하십시오.
- 프로그래밍 인터페이스 중 하나를 호출하여 메시지를 보내고 받도록 애플리케이션을 프로그래밍하십시오. 프로그래밍 인터페이스는 여러 다른 언어로 제공되며 표준 JMS 프로그래밍 인터페이스 및 Windows 통신 기반용 클래스를 포함합니다.
- HTTP 프로토콜을 사용하여 브라우저에서 IBM WebSphere MQ 메시지를 받고 보내십시오.

동작 방법

- 관리자가 명령을 사용하여 큐 관리자를 작성하고 시작합니다. 일반적으로 운영 체제를 부팅할 때 큐 관리자가 계속하여 자동으로 시작됩니다. 그런 다음, 애플리케이션과 기타 큐 관리자가 이 큐 관리자에 연결하여 메시지를 보내고 받을 수 있습니다.
- 애플리케이션 또는 관리자가 큐나 토픽을 작성합니다. 큐와 토픽은 큐 관리자가 소유하고 저장한 오브젝트입니다.
- 애플리케이션이 다른 애플리케이션으로 데이터를 전송하려고 할 때 애플리케이션에서 메시지에 데이터를 넣습니다. 메시지를 큐에 넣거나 메시지를 토픽에 발행합니다. 메시지를 검색할 수 있는 기본 방법은 다음 세 가지입니다.
 - 동일한 큐 관리자에 연결된 포인트-투-포인트 애플리케이션이 동일한 큐에서 메시지를 검색합니다.
예를 들어, 애플리케이션에서 임시 또는 지속 데이터를 저장하는 방법으로 큐에 메시지를 넣습니다. 두 번째 예에서는 애플리케이션이 다른 프로세스에서 실행 중인 다른 애플리케이션과 데이터를 공유합니다.
 - 다른 큐 관리자에 연결된 포인트-투-포인트 애플리케이션이 다른 큐에서 동일한 메시지를 검색합니다.
애플리케이션은 큐의 메시지를 교환하여 서로 통신합니다. IBM WebSphere MQ는 기본적으로 메시지를 보내거나 교환하는 데 사용합니다. 한 애플리케이션에서 한 컴퓨터의 큐에 메시지를 넣고, 다른 애플리케이션에서 다른 컴퓨터의 다른 큐에서 동일한 메시지를 가져옵니다. 두 컴퓨터의 큐 관리자가 함께 작동하여 첫 번째 큐에서 두 번째 큐로 메시지를 전송합니다. 애플리케이션은 서로 통신하지 않지만, 큐 관리자는 서로 통신합니다.
 - 임의의 큐 관리자에 연결된 구독자 애플리케이션이 공용 토픽에서 메시지를 검색합니다.
발행자 애플리케이션에서 메시지를 작성하여 한 컴퓨터의 토픽에 발행합니다. 임의의 수의 구독자 애플리케이션이 다른 컴퓨터에 있는 동일한 토픽을 구독합니다. IBM WebSphere MQ 구독자가 연결되어 있는 큐 관리자에 속하는 큐에 발행물을 전달합니다. 구독자가 큐에서 메시지를 검색합니다.
- MQ 채널은 네트워크 상에서 큐 관리자를 다른 큐 관리자에 연결합니다. 스스로 MQ 채널을 작성할 수 있거나, 큐 관리자 클러스터의 큐 관리자가 필요한 시점에 MQ 채널을 작성합니다.
- 한 큐 관리자 상에 여러 개의 큐와 토픽이 있을 수 있습니다.
- 한 컴퓨터에 하나 이상의 큐 관리자가 있을 수 있습니다.
- 애플리케이션이 큐 관리자와 같은 컴퓨터나 다른 컴퓨터에서 실행될 수 있습니다. 같은 컴퓨터에서 실행되는 경우에는 IBM WebSphere MQ 서버 애플리케이션입니다. 다른 컴퓨터에서 실행되는 경우에는 IBM WebSphere MQ 클라이언트 애플리케이션입니다. IBM WebSphere MQ인지에 상관없이, 클라이언트 또는 서버가 애플리케이션에는 거의 아무런 차이도 만들어내지 않습니다. IBM WebSphere MQ 클라이언트 또는 서버로 클라이언트/서버 애플리케이션을 빌드할 수 있습니다.

IBM WebSphere MQ와 함께 제공되는 도구 및 자원

- 명령행에서 실행되는 제어 명령. 제어 명령으로 큐 관리자를 작성하고 시작하고 중지합니다. 또한, 제어 명령으로 IBM WebSphere MQ 관리 및 문제점 판별 프로그램을 실행합니다.
- 해석기에서 실행하는 IBM WebSphere MQ 스크립트 명령(MQSC). 이 스크립트 명령으로 큐와 토픽을 작성하고 IBM WebSphere MQ를 구성하고 관리하십시오. 파일에서 명령을 편집하여 파일을 **runmqsc** 프로그램으로 전달하여 명령을 해석하십시오. 또한, 다른 큐 관리자를 관리하기 위해 다른 컴퓨터로 명령을 보내는 큐 관리자에서 해석기를 실행할 수 있습니다.
- 프로그램 가능 명령 형식(PCF) 명령은 자체 애플리케이션에서 IBM WebSphere MQ를 관리하기 위해 호출하는 명령입니다. PCF 명령은 스크립트 명령과 같은 기능을 가지고 있지만, 프로그래밍하기는 더 쉽습니다.

- 파생되었습니다.
- Windows 및 Linux® x86/x86-64 플랫폼에서 다음 유틸리티를 실행할 수 있습니다.
 - IBM WebSphere MQ Explorer. 이 탐색기는 스크립트 명령과 같은 관리 태스크를 수행하지만, 대화식으로 사용하기가 훨씬 더 쉽습니다.
 - 메시지를 보여주고 설치를 확인하는 *Postcard* 애플리케이션.
 - 학습서.

관련 개념

17 페이지의 『IBM WebSphere MQ Version 7.5의 새로운 기능』

IBM WebSphere MQ Version 7.5의 주요 새 기능에 대해 학습하십시오.

WebSphere MQ 멀티캐스트

IBM WebSphere MQ 멀티캐스트는 낮은 지연, 높은 팬아웃, 신뢰할 수 있는 멀티캐스트 메시지를 제공합니다.

WebSphere MQ Telemetry

사람과 비즈니스, 정부는 우리가 살며 일하는 환경과 좀 더 스마트한 상호작용을 하기 위해 IBM WebSphere MQ Telemetry를 점점 더 많이 활용하고자 합니다. IBM WebSphere MQ Telemetry는 모든 종류의 디바이스를 인터넷 및 기업과 연결하며 스마트 디바이스를 위한 애플리케이션을 빌드하는 비용을 감소시킵니다.

메시징과 큐잉에 대한 기술적 소개

WebSphere MQ 제품을 사용하면 프로그램이 일관된 애플리케이션 프로그래밍 인터페이스(API)를 사용하여 다른 컴포넌트(프로세서, 운영 체제, 서브시스템 및 통신 프로토콜)의 네트워크에서 서로 통신할 수 있습니다.

WebSphere MQ 클라이언트 및 서버에 대한 기술적 소개

IBM WebSphere MQ가 해당 애플리케이션에 대한 클라이언트 서버 구성을 지원하는 방법에 대한 소개입니다.

큐 관리자 통신에 대한 기술적 소개

WebSphere MQ에서, 상호통신은 하나의 큐 관리자에서 다른 큐 관리자로 메시지를 전송한다는 의미입니다. 수신 큐 관리자는 동일 시스템 또는 다른 쪽 세계 또는 인접한 곳의 다른 시스템에 있을 수 있습니다. 로컬 큐 관리자와 동일한 플랫폼에서 실행 중이거나 WebSphere MQ에서 지원되는 플랫폼에서 실행 중일 수 있습니다. 이를 분산 환경이라고 합니다. WebSphere MQ는 분산 큐 관리(DQM)를 사용하는 것처럼 분산 환경에서 통신을 핸들링합니다.

관련 태스크

[WebSphere MQ Advanced Message Security\(AMS\)](#)

[WebSphere MQ Managed File Transfer](#)

IBM WebSphere MQ 라이선스 정보

IBM WebSphere MQ와 함께 구매할 수 있는 컴포넌트와 개별 구매로 설치할 수 있는 컴포넌트

주의사항: 이 라이선스 안내서는 구입한 권한 내에서 IBM 으로부터 라이선스를 취득한 프로그램을 전개하는 데 도움이 되는 보충 정보를 제공합니다. 귀하의 라이선스 계약 (예: IBM IPLA (International Program License Agreement) 또는 이와 동등한 계약 및 IBM WebSphere MQ 제품 오퍼링에 대한 라이선스 정보를 포함한 거래 문서) 은 본 프로그램의 사용에 관한 귀하와 IBM 간의 유일하고 완전한 계약입니다.

IBM WebSphere MQ와 함께 구매할 수 있는 컴포넌트

분산 플랫폼

분산 플랫폼 기반 IBM WebSphere MQ의 경우, 제품 오퍼링에는 별도로 구매할 수 있는 11가지 유료 컴포넌트가 포함되어 있습니다.

5724-H72 IBM IBM WebSphere MQ

IBM IBM WebSphere MQ(서버)

IBM IBM WebSphere MQ 텔레메트리

IBM IBM WebSphere MQ 고급 메시지 보안

IBM IBM WebSphere MQ 유희 대기

IBM IBM WebSphere MQ Advanced Message Security 유희 대기

IBM IBM WebSphere MQ 고급
IBM IBM WebSphere MQ 고급 유희 대기
개발자용 IBM IBM WebSphere MQ Advanced
IBM IBM WebSphere MQ Managed File Transfer 서비스
IBM IBM WebSphere MQ Managed File Transfer 서비스 유희 대기
IBM IBM WebSphere MQ Managed File Transfer 관리 엔드포인트

내 엔터프라이즈가 설치할 수 있는 컴포넌트

분산 플랫폼 기반 IBM WebSphere MQ의 경우, 아래 컴포넌트는 IBM WebSphere MQ 설치 프로그램이 설치할 수 있는 컴포넌트에 직접 맵핑되므로, 이러한 상황에서는 구매한 컴포넌트와 설치할 수 있는 컴포넌트 간의 맵핑이 용이합니다.

중요사항: IBM WebSphere MQ 설치 매체에는 모든 컴포넌트가 포함되어 있지만, 권한을 구매한 서브세트만 설치해야 합니다.

5724-H72 IBM IBM WebSphere MQ

IBM WebSphere(서버)

포함 사항:

- ClientDevelopment Kit(SDK)
- IBM Global Security Kit(UNIX)
- IBM WebSphere MQExplorer
- Java .NET 메시징 및 웹 서비스
- 샘플 프로그램
- 서버/런타임
- UNIX 맨 페이지

IBM IBM WebSphere MQ 텔레메트리

포함 사항:

- 텔레메트리 서비스

IBM IBM WebSphere MQ 고급 메시지 보안

포함 사항:

- Advanced Message Security

IBM IBM WebSphere MQ Managed File Transfer 서비스

포함 사항:

- IBM WebSphere MQ Managed File Transfer 로거
- IBM WebSphere MQ Managed File Transfer 서비스
- IBM WebSphere MQ Managed File Transfer 도구

IBM WebSphere IBM WebSphere MQ Managed File Transfer 관리 엔드포인트

포함 사항:

- IBM WebSphere MQ Managed File Transfer 에이전트
- IBM WebSphere MQ Managed File Transfer 로거
- IBM WebSphere MQ Managed File Transfer 도구

IBM IBM WebSphere MQ Advanced의 개념

IBM WebSphere MQ Advanced는 권한 구매 프로세스를 단순화하기 위해 도입되었습니다. 엔터프라이즈는 하나의 가격을 지불하고 여러 IBM WebSphere MQ 컴포넌트에 대한 권한을 확보합니다.

IBM WebSphere MQ Advanced 부분은 다음과 같습니다.

5724-H72 IBM IBM WebSphere MQ

IBM IBM WebSphere MQ 고급

개발자용 IBM IBM WebSphere MQ Advanced

분산 플랫폼 기반 IBM IBM WebSphere MQ의 경우, IBM WebSphere MQ Advanced의 100개 PVU(Processor Value Unit)를 구매하면 엔터프라이즈에서 다음을 설치할 수 있습니다.

- IBM IBM WebSphere MQ(서버)의 100개 PVU 및
- IBM IBM WebSphere MQ Advanced Message Security의 100개 PVU 및
- IBM IBM WebSphere MQ Managed File Transfer Service의 100개 PVU 및
- IBM IBM WebSphere MQ Telemetry의 무제한 설치

또한 엔터프라이즈는 필요에 따라 IBM 버전을 혼합하고 일치시킬 수 있습니다. 따라서 IBM IBM WebSphere MQ(서버)의 100개 PVU 권한은 이 컴포넌트의 IBM WebSphere MQ 7.5 버전의 50개 PVU와 IBM WebSphere MQ 7.1 버전의 50개 PVU로 분할될 수 있습니다.

IBM IBM WebSphere MQ Advanced for Developers는 IBM IBM WebSphere MQ Advanced 및 IBM IBM WebSphere MQ Managed File Transfer Managed Endpoint에 포함된 모든 항목에 대한 사용 권한을 개발 용도로만 제공합니다.



주의: IBM 라이선스는 개발 용도로 고려되는 사항을 정의합니다.

Idle Standby 부분의 개념

Idle Standby 부분은 고가용성 환경에 제공하도록 도입되었으며, 이러한 환경에서는 수동 시스템에 IBM WebSphere MQ가 설치되어 사용 가능하지만 해당 시스템은 활성 큐 관리자의 구성 및 활동을 최신 상태로 유지하는 것 외에 어떤 IBM WebSphere MQ 처리 작업이나 활동도 수행하지 않습니다. 이 경우에는 더 낮은 비용을 적용할 수 있습니다.

참고:

1. 또한 IBM WebSphere MQ 다중 인스턴스 큐 관리자 기능 사용 시에도 Idle Standby 권한이 필요합니다.
2. IBM WebSphere MQ Telemetry 컴포넌트에 대한 Idle Standby 부분은 없습니다. 활성 및 수동 시스템에 대해 동일한 IBM IBM WebSphere MQ Telemetry 부분을 구매해야 합니다. 단, IBM IBM WebSphere MQ Advanced Idle Standby가 있는 경우는 제외합니다(이 경우에는 해당 부분이 포함되어 있음).
3. 또한 IBM WebSphere Managed File Transfer Managed Endpoint에 대한 Idle Standby 부분도 없습니다(엔드포인트가 서버 환경의 일부가 아니기 때문).

관련 개념

22 페이지의 『IBM WebSphere MQ Version 7.5 에서 변경된 사항』

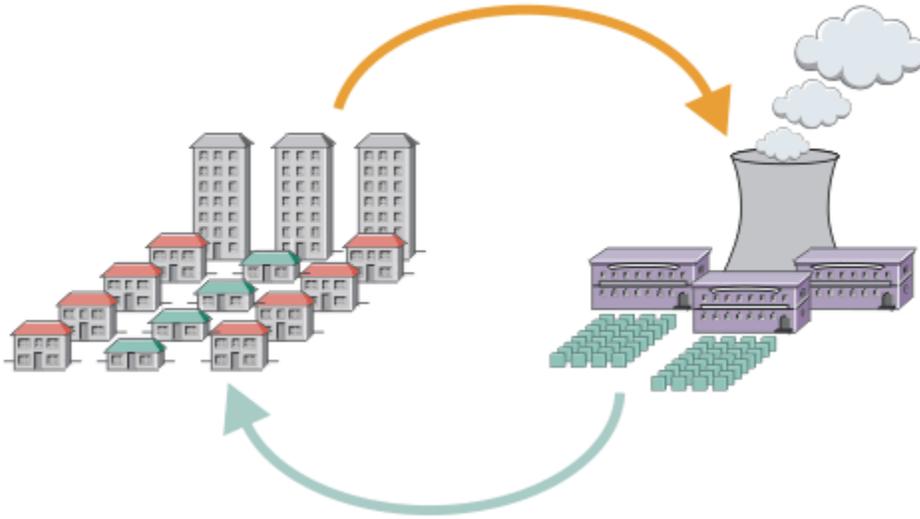
39 페이지의 『IBM WebSphere MQ 기술 개요』

IBM WebSphere MQ를 사용하여 애플리케이션을 연결하고 조직에서 정보 분배를 관리하십시오.

IBM WebSphere MQ Telemetry 소개

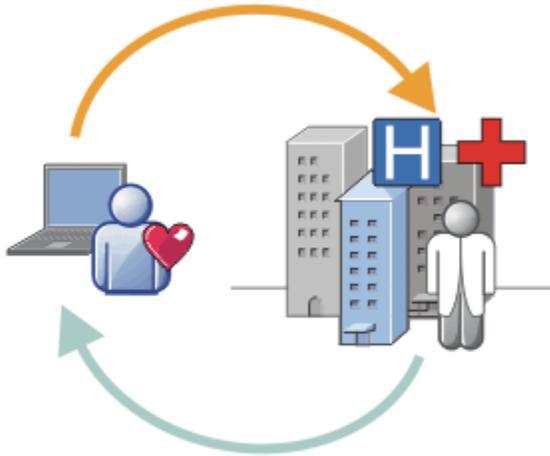
사람과 비즈니스, 정부는 우리가 살며 일하는 환경과 좀 더 스마트한 상호작용을 하기 위해 IBM WebSphere MQ Telemetry를 점점 더 많이 활용하고자 합니다. IBM WebSphere MQ Telemetry는 모든 종류의 디바이스를 인터넷 및 기업과 연결하며 스마트 디바이스를 위한 애플리케이션을 빌드하는 비용을 감소시킵니다.

다음 다이어그램은 IBM WebSphere MQ Telemetry의 일반적인 사용을 보여줍니다.



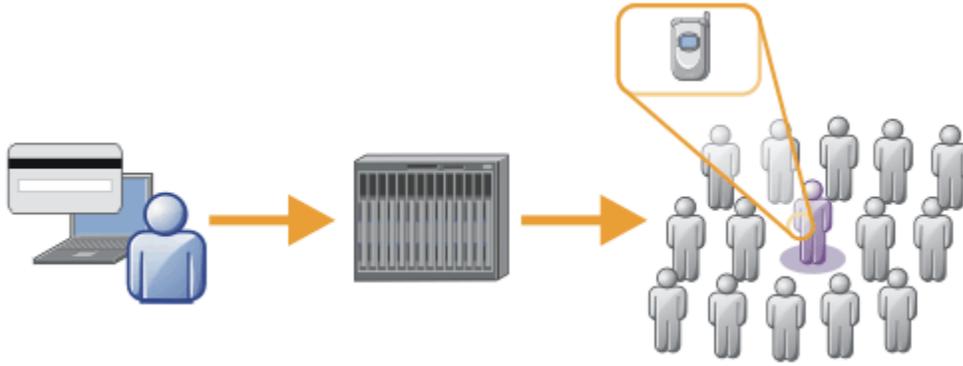
- 서비스 제공자에게 전송되는 에너지 사용 데이터가 들어 있는 MQTT 메시지입니다.
- 텔레메트리 애플리케이션은 에너지 사용량 데이터에 대한 분석을 기반으로 한 제어 명령을 송신합니다.
- 자세한 정보는 67 페이지의 『텔레메트리 시나리오: 가정 에너지 모니터링 및 제어』의 내용을 참조하십시오.

그림 1. 스마트 전기 계량



- 텔레메트리 애플리케이션은 건강 데이터를 병원과 의사에게 송신합니다.
- 의료 데이터의 분석을 기반으로 MQTT 메시지 경고 또는 피드백이 리턴됩니다.
- 자세한 정보는 65 페이지의 『텔레메트리 시나리오: 가정 환자 모니터링』의 내용을 참조하십시오.

그림 2. 스마트 건강 모니터링



- 단순 카드 트랜잭션이 은행의 서버에 전송됩니다.
- IBM WebSphere MQ Telemetry는 수천 명의 사람들 중에서 한 사람을 식별하여 카드를 사용한 고객을 알려줍니다.
- IBM WebSphere MQ Telemetry는 가장 간단한 정보 입력을 사용하여 해당 개인을 찾습니다.

그림 3. Telemetry: 수천 명의 사람들 중 한 사람 식별

WebSphere MQ Telemetry의 개념

- IBM WebSphere MQ 에서 제공하는 범용 메시징 백본을 광범위한 원격 센서, 작동기 및 텔레메트리 디바이스로 확장하는 IBM WebSphere MQ 의 기능입니다. IBM WebSphere MQ Telemetry 는 지능형 엔터프라이즈 애플리케이션, 서비스 및 의사 결정자를 인스트루먼트된 디바이스의 네트워크와 상호 연결할 수 있도록 IBM WebSphere MQ 를 확장합니다.
- WebSphere MQ Telemetry의 두 가지 핵심 파트는 다음과 같습니다.
 1. IBM WebSphere MQ 서버 내부에서 실행되는 IBM WebSphere MQ Telemetry 서비스.
 2. 애플리케이션과 함께 디바이스에 분산된 IBM WebSphere MQ Telemetry 클라이언트

사용 목적

- MQ 텔레메트리는 MQ 텔레메트리 전송 (MQTT) 을 사용하여 애플리케이션과 IBM WebSphere MQ 큐 관리자 간에 데이터를 송신하고 수신합니다.
- MQTT는 다양한 디바이스에 사용할 MQTT 구현을 작성할 수 있게 해주는 개방형 메시징 전송입니다.
- MQTT 클라이언트는 자원이 제한될 수 있는 작은 폼팩터 디바이스입니다.
- MQTT는 대역폭이 낮거나, 송신 데이터 비용이 많이 소요되거나, 불안정한 네트워크에서 효과적으로 작동합니다.
- 메시지 전달은 보장되고, 애플리케이션과 구분됩니다.
- 애플리케이션 프로그래머에게 통신 프로그래밍 지식이 요구될 필요가 없습니다.
- 메시지는 다른 메시징 애플리케이션과 교환될 수 있습니다. 그러한 애플리케이션으로는 다른 텔레메트리 애플리케이션, MQI, JMS 또는 엔터프라이즈 메시징 애플리케이션이 있습니다.

사용 방법

- IBM WebSphere MQ Explorer 및 연관된 도구를 사용하여 MQ의 WebSphere MQ Telemetry 기능을 관리하십시오.
- 사용자 애플리케이션의 MQTT 클라이언트를 사용하여 큐 관리자에 연결하고 메시지를 발행 및 구독합니다.
- MQTT 클라이언트와 함께 사용자 애플리케이션을 이 애플리케이션을 실행할 디바이스에 분배합니다.

동작 방법

- MQ 텔레메트리 (MQXR) 서비스는 IBM WebSphere MQ 큐 관리자를 MQTT 서버로 변환합니다.

- MQTT 서버는 MQTT 메시지 전송을 인식하고 MQTT 클라이언트를 대상으로 메시지를 송수신할 수 있습니다.
- MQ Telemetry는 MQTT 메시지 전송을 구현하는 다양한 텔레메트리 클라이언트와 함께 제공됩니다. 이를 MQTT 클라이언트라고 합니다.
- 기본 텔레메트리 클라이언트는 표준 MQ 클라이언트처럼 작동하지만 더 광범위한 플랫폼 및 네트워크에서 실행될 수 있습니다.
- 고급 텔레메트리 클라이언트는 단일 큐 관리자에 더욱 많은 수의 MQTT 클라이언트가 연결될 수 있도록 네트워크 집선기의 역할을 합니다. 또한 짧은 네트워크 정전 동안 메시지를 보관할 수단이 없는 소형 디바이스를 위해 저장 후 전달 기능도 제공할 수 있습니다.
- IBM WebSphere MQ Telemetry daemon for devices은 IBM WebSphere MQ Telemetry의 부분인 고급 텔레메트리 클라이언트입니다. 자세한 정보는 89 페이지의 『디바이스의 텔레메트리 디먼』의 내용을 참조하십시오.
- MQTT는 발행/구독 프로토콜입니다.
 - MQTT 클라이언트 애플리케이션은 MQTT 서버에 메시지를 발행할 수 있습니다.
 - IBM WebSphere MQ 큐 관리자가 MQTT 서버로 작동하는 경우, 큐 관리자에 연결하는 다른 애플리케이션은 구독할 수 있고 MQTT 클라이언트에서 메시지를 수신할 수 있습니다.
 - MQTT 클라이언트는 MQ 큐 관리자에 연결된 애플리케이션이 송신하는 메시지를 구독할 수 있습니다.
 - 큐 관리자는 발행 애플리케이션에서 구독 애플리케이션으로 메시지를 분배하는 라우터 역할을 합니다.
 - 메시지 서로 다른 유형의 클라이언트 애플리케이션 간에 분배될 수 있습니다. 예를 들어 텔레메트리 클라이언트와 JMS 클라이언트 간에 분배될 수 있습니다.

IBM WebSphere MQ Telemetry는 WebSphere Message Broker의 버전 7에서 제거된 SCADA 노드를 대체하며 Windows, Linux 및 AIX®에서 실행됩니다. [Migration of telemetry applications from using WebSphere Message Broker version 6 to use IBM WebSphere MQ Telemetry and WebSphere Message Broker version 7.0](#)는 WebSphere Message Broker V6에서 SCADA 노드를 사용하여 응용프로그램을 이주하는 데 도움이 되는 정보를 제공합니다. WebSphere Message Broker 버전 7을 사용하는 텔레메트리 애플리케이션은 MQTT 클라이언트와 공용인 토픽을 subscribe합니다. 이는 MQInput 노드를 사용하여 MQTT 클라이언트로부터 발행을 수신하고 발행 노드를 사용하여 MQTT 클라이언트로 발행합니다.

관련 개념

64 페이지의 『모니터링 및 제어를 위한 텔레메트리 개념과 시나리오』

텔레메트리는 자동화된 감지, 데이터 측정 및 원격 디바이스의 제어를 가리킵니다. 주안점은 디바이스에서 중앙 제어 지점으로의 데이터 전송입니다. 또한 텔레메트리는 디바이스로 구성 및 제어 정보를 송신하는 것을 포함합니다.

관련 태스크

[WebSphere MQ Telemetry 설치](#)

[WebSphere MQ Telemetry 관리](#)

[WebSphere Message Broker 버전 6 사용에서 WebSphere MQ Telemetry 및 WebSphere Message Broker 버전 7.0 사용으로 텔레메트리 애플리케이션의 마이그레이션](#)

[WebSphere MQ Telemetry용 애플리케이션 개발](#)

[WebSphere MQ Telemetry의 문제점 해결](#)

관련 참조

[WebSphere MQ Telemetry 참조](#)

관련 정보

62 페이지의 『IBM WebSphere MQ Telemetry』

IBM WebSphere MQ Version 7.5 정보 로드맵

정보 로드맵에는 다양한 IBM WebSphere MQ Version 7.5 자원에 대한 링크가 포함되어 있습니다.

이 로드맵은 IBM WebSphere MQ의 특정 영역에 대해 좀 더 자세히 알아볼 수 있도록 다양한 소스에서 정보를 취합합니다. 로드맵의 각 섹션에 대한 링크를 클릭하여 사용 가능한 자원을 확인하십시오.

- [제품 개요](#)

- [기술 개요](#)
- [시나리오](#)
- [계획](#)
- [마이그레이션 및 업그레이드](#)
- [설치](#)
- [보안](#)
- [구성](#)
- [관리](#)
- [애플리케이션 개발](#)
- [모니터링 및 성능](#)
- [문제점 해결 및 지원](#)
- [참조](#)

표 1. IBM WebSphere MQ 정보 로드맵 테이블	
카테고리	정보 자원
제품 개요	<p>IBM WebSphere MQ의 전체 용도, 기능 및 새 기능의 개요</p> <p>❖ 5 페이지의 『정보 IBM WebSphere MQ』 새로운 기능 및 이번 릴리스에서 변경된 내용에 대한 개요와 제품에 대한 소개를 포함하여 IBM WebSphere MQ Version 7.5을 시작하는 데 도움이 되는 도입 정보입니다.</p> <p>IBM WebSphere MQ Version 7.5</p> <p>이 IBM Redbooks 발행물은 IBM WebSphere MQ Version 7.5의 핵심 기능 확장 및 이해해야 할 개념을 포함합니다.</p> <p>IBM WebSphere MQ 제품 웹 페이지 자원 및 추가 정보의 링크가 있는 제품 웹 페이지입니다.</p> <p>IBM WebSphere MQ 시스템 요구사항 IBM WebSphere MQ의 다양한 릴리스에 대한 시스템 요구사항의 링크가 있는 웹 페이지입니다.</p> <p>16 페이지의 『IBM Documentation 오프라인 애플리케이션의 IBM WebSphere MQ Version 7.5』 IBM WebSphere MQ Version 7.5 메시징 문서를 로컬로 설치하는 IBM Documentation의 오프라인 버전으로 다운로드할 수 있습니다.</p>
기술 개요	<p>❖ 39 페이지의 『IBM WebSphere MQ 기술 개요』</p> <p>IBM WebSphere MQ Version 7.5이 제공하는 메시지 큐잉 및 기타 기능에 대해 알아보는 데 도움이 되는 정보입니다.</p>

표 1. IBM WebSphere MQ 정보 로드맵 테이블 (계속)

카테고리	정보 자원
시나리오	<p>각 시나리오는 중요한 태스크 세트를 설명하고 주요 제품 기능을 구성하는 데 도움을 제공합니다. 시나리오에는 사용자가 관심이 있는 영역을 더 잘 이해할 수 있도록 도움을 주는 다른 콘텐츠에 대한 유용한 링크가 포함되어 있습니다.</p> <p>❖ 128 페이지의 『시나리오』 시작하기 시나리오는 IBM WebSphere MQ를 시작하는 방법에 대해 설명합니다. 이전에 IBM WebSphere MQ를 사용해본 적이 없으며 빨리 시작하고 싶으면 이 시나리오를 사용하십시오. 추가적 시나리오에서는 적절한 태스크 단계를 설명하여 제품 기능을 구성하거나 사용하는 데 도움을 줍니다.</p> <p>❖ IBM WebSphere MQ 에 WebSphere Application Server 연결 다양한 시나리오에서 WebSphere Application Server를 IBM WebSphere MQ에 연결하는 데 필요한 주요 태스크를 설명하는 정보를 포함합니다.</p> <p>❖ IBM WebSphere MQ 에 WebSphere Application Server Liberty 프로파일 연결 다양한 시나리오에서 WebSphere Application Server Liberty 프로파일을 IBM WebSphere MQ에 연결하는 데 필요한 주요 태스크를 설명하는 정보를 포함합니다.</p> <p>❖ IBM MessageSight 를 IBM WebSphere MQ 및 WebSphere Application Server에 연결 다양한 시나리오에서 IBM MessageSight 를 IBM WebSphere MQ 및 WebSphere Application Server 에 연결하는 데 필요한 주요 태스크를 안내하는 정보를 포함합니다.</p>
계획	<p>❖ 계획 IBM WebSphere MQ 환경을 계획할 때, IBM WebSphere MQ가 단일 및 다중 큐 관리자 아키텍처와, 포인트-투-포인트 및 발행/구독 메시징 스타일에 대해 제공하는 지원을 고려하십시오. 또한 자원 요구사항과, 로깅 및 백업 기능 사용을 계획하십시오.</p>
마이그레이션 및 업그레이드	<p>❖ 마이그레이션 및 업그레이드 큐 관리자를 마이그레이션하여 새 레벨의 코드에서 실행하려면, 먼저 새 코드 레벨을 설치할 수 있도록 IBM WebSphere MQ를 업그레이드해야 합니다. 업그레이드가 성공적으로 완료되었음을 확인하면 큐 관리자 및 큐 관리자와 연관된 모든 애플리케이션과 자원을 마이그레이션합니다. 이 프로세스를 시작하기 전에 먼저 해당 마이그레이션 안내서의 정보에 따라 마이그레이션 계획을 작성합니다. 유지보수를 적용하는 경우에는 마이그레이션이 필요하지 않습니다. 그러나 IBM WebSphere MQ 코드의 새 레벨로 응용프로그램을 테스트해야 합니다.</p> <p>❖ WebSphere / IBM MQ 이주 안내서 이 안내서는 이전 버전에서 새 버전으로의 마이그레이션 프로세스를 계획하는 데 도움이 되는 정보를 제공합니다. 이 안내서는 웹 브라우저에서 보거나 PDF 파일로 다운로드할 수 있습니다.</p>
설치	<p>❖ 설치 및 설치 제거 설치를 준비하고 제품을 설치하며 설치를 확인하는 데 도움이 되는 정보입니다. 제품을 설치 제거하는 데 도움이 되는 정보도 있습니다.</p>
보안	<p>❖ 보안 ID 및 인증, 권한 부여, 감사, 기밀성 및 데이터 무결성을 포함하여 IBM WebSphere MQ 설치에서 고려할 보안의 측면입니다.</p>

표 1. IBM WebSphere MQ 정보 로드맵 테이블 (계속)	
카테고리	정보 자원
구성	<p>❖ 구성 하나 이상의 컴퓨터에서 하나 이상의 큐 관리자를 작성하고 개발, 테스트 및 프로덕션 시스템에서 비즈니스 데이터가 포함된 메시지를 처리하도록 해당 큐 관리자 및 관련 자원을 구성하십시오.</p>
관리	<p>❖ 관리IBM WebSphere MQ 큐 관리자 및 연관된 자원을 관리하십시오.</p>
애플리케이션 개발	<p>❖ 애플리케이션 개발 메시지를 송신 및 수신하고 큐 관리자 및 관련 자원을 관리하는 애플리케이션을 개발하십시오. IBM WebSphere MQ는 절차적 언어, 객체 지향 언어 및 프레임워크로 작성된 애플리케이션을 지원합니다.</p>
모니터링 및 성능	<p>❖ 모니터링 및 성능 큐 관리자 네트워크의 성능을 향상시키는 데 도움이 되는 모니터링 정보 및 자세한 내용과, 큐 관리자 네트워크의 성능을 향상시키는 데 도움이 되는 성능 조정 팁입니다.</p>
문제점 해결 및 지원	<p>❖ 문제점 해결 및 지원 큐 관리자 네트워크 또는 IBM WebSphere MQ 애플리케이션의 문제점을 진단하고 해결하는 데 도움이 되는 기술입니다.</p> <p>IBM Support Assistant 웹 페이지 IBM Support Assistant (ISA)는 지원 관련 정보와 문제점 해결 도구에 대한 액세스를 제공하여 IBM 소프트웨어 제품에 대한 질문과 문제점을 해결하도록 돕습니다.</p> <p>IBM 지원 포털 웹 페이지 IBM WebSphere MQ에 대한 포털을 지원하십시오.</p>
참조	<p>❖ 참조 구성, 관리, 개발 애플리케이션, 텔레메트리, 보안, 모니터링, 문제점 해결과 지원 및 진단 메시지에 대한 정보를 참조하십시오.</p>

IBM Documentation 오프라인 애플리케이션의 IBM WebSphere MQ Version 7.5

인터넷에 액세스하지 않고 공중 갭 환경에 있는 경우, 다크 스텝 앱 "IBM Documentation 오프라인" 을 사용하여 IBM WebSphere MQ Version 7.5 제품 문서의 다운로드를 확인하십시오.

IBM Documentation 오프라인에는 두 가지 구성요소가 있습니다.

- **IBM Documentation 오프라인 앱.** 이는 IBM Documentation의 로컬로 설치 가능한 오프라인 버전입니다.
- **IBM Documentation 오프라인 앱에 설치하는 문서 패키지** 이러한 패키지에는 IBM Documentation에서 온라인으로 발행된 것과 동일한 문서가 포함됩니다.

앱을 다운로드하고 IBM WebSphere MQ Version 7.5 문서 패키지를 다운로드하려면 IBM Documentation에 로그인해야 합니다. 추가 정보는 IBM Documentation 오프라인의 내용을 참조하십시오.

IBM WebSphere MQ Version 7.5 PDF 문서

IBM WebSphere MQ Version 7.5 제품 문서를 일련의 PDF 파일로 다운로드할 수 있습니다.

IBM WebSphere MQ Version 7.5 PDF 파일에는 IBM Documentation에 게시된 HTML 버전과 동일한 정보가 포함되어 있습니다. 다음 표의 링크를 클릭하여 다운로드할 수 있습니다.

참고: HTML 형식의 IBM WebSphere MQ Version 7.5 제품 문서를 로컬로 설치하는 IBM Documentation 의 오프라인 버전으로 다운로드할 수도 있습니다. 자세한 정보는 16 페이지의 『IBM Documentation 오프라인 애플리케이션의 IBM WebSphere MQ Version 7.5』의 내용을 참조하십시오.

표 2. 제품 문서의 PDF 파일과 이 파일이 IBM Documentation의 섹션에 대응되는 방법	
PDF 파일 이름 및 다운로드 링크	IBM Documentation의 동등한 섹션
mq75.overview.pdf	정보IBM WebSphere MQ
mq75.plan.pdf	계획 중
mq75.migrate.pdf	마이그레이션 및 업그레이드
mq75.install.pdf	설치
mq75.secure.pdf	보안
mq75.configure.pdf	구성
mq75.administer.pdf	관리
mq75.develop.pdf	애플리케이션 개발
mq75.monitor.pdf	모니터링 및 성능
mq75.troubleshoot.pdf	문제점 해결 및 지원
mq75.reference.pdf	참조
mq75.refcon.pdf	구성 참조
mq75.refadmin.pdf	관리 참조
mq75.refdev.pdf	애플리케이션 참조 개발
mq75.mft.pdf	IBM WebSphere MQ Managed File Transfer
mq75.explorer.pdf	IBM WebSphere MQ Explorer
mq75.guiwizard.pdf	MQ GUI 마법사 도움말
mq75.xms.pdf	Message Service Client for .NET
mq75.hve.pdf	WebSphere MQ Hypervisor Edition
mq75.mmtc.pdf	모바일 메시징과 M2M
MQ_Migration_Guide.pdf	WebSphere / IBM MQ 이주 안내서

참고: PDF 파일은 PDF 파일 간의 링크가 올바르게 작동하기 위해 모두 같은 폴더에 있어야 합니다.

IBM WebSphere MQ Version 7.5의 새로운 기능

IBM WebSphere MQ Version 7.5의 주요 새 기능에 대해 학습하십시오.

- 18 페이지의 『IBM WebSphere MQ 관리 파일 전송』.
- 18 페이지의 『IBM WebSphere MQ 고급 메시지 보안』.
- 18 페이지의 『메시지 채널 에이전트(MCA) 인터셉션』.
- 18 페이지의 『다중 클러스터 전송 큐』.
- 18 페이지의 『코어 클라이언트의 일부가 된 확장 트랜잭션 기능』.
- 18 페이지의 『애플리케이션 이름을 설정하여 큐 관리자에 대한 연결 식별』.
- 18 페이지의 『인증서 유효성 검증 정책』.
- 19 페이지의 『더 뛰어난 트랜잭션 시각성』.

- 19 페이지의 『시나리오』.
- 19 페이지의 『IBM WebSphere MQ Explorer』.

IBM WebSphere MQ 관리 파일 전송

IBM WebSphere MQ Managed File Transfer는 IBM WebSphere MQ를 사용하여 큐 관리자 간에 파일을 전송합니다. 큐 관리자가 없는 워크스테이션 및 서버로 범위를 확장할 수 있습니다. 파일 전송 에이전트와 Apache Ant를 사용하여 확장하고 IBM Sterling Commerce®:Direct, 웹 게이트웨이 및 SFTP, FTP 또는 FTPS 프로토콜 서버와 통합할 수 있습니다.

IBM WebSphere MQ Managed File Transfer를 사용하면 파일 전송을 자동화, 제어, 보안 및 감사할 수 있습니다. [IBM WebSphere MQ Managed File Transfer 소개](#)를 참조하십시오.

IBM WebSphere MQ 고급 메시지 보안

IBM WebSphere MQ AMS (Advanced Message Security)는 별도로 설치된 컴포넌트이며 별도로 비용이 청구됩니다. IBM WebSphere MQ 네트워크를 통해 기밀 데이터 플로우에 대한 상위 레벨 보호를 제공합니다. AMS를 이용하기 위해 기존 애플리케이션을 수정할 필요는 없습니다. [IBM WebSphere MQ 고급 메시지 보안을 참조하십시오.](#)

메시지 채널 에이전트(MCA) 인터셉션

MCA 인터셉션 기능을 사용하면 Advanced Message Security의 라이선스 설치가 있는 IBM WebSphere MQ에서 실행 중인 큐 관리자가 서버 연결 채널에 정책을 선택적으로 적용할 수 있습니다. IBM WebSphere MQ AMS 외부에 남아 있는 클라이언트는 MCA 인터셉션을 통해 큐 관리자에 계속 연결되어 있고 메시지를 암호화하고 복호화할 수 있습니다. [메시지 채널 에이전트\(MCA\) 인터셉션을 참조하십시오.](#)

다중 클러스터 전송 큐

각 클러스터 송신자 채널에 다른 클러스터 전송 큐를 지정하도록 새 큐 관리자 속성 **DEFCLXQ**를 변경할 수 있습니다. 각 클러스터 송신자 채널이 전달하는 메시지는 별도의 클러스터 전송 큐에 배치됩니다. [클러스터 전송 큐 및 클러스터 송신자 채널을 참조하십시오.](#) 또한 새 큐 속성 **CLCHNAME**을 설정하여 수동으로 클러스터 전송 큐를 구성할 수도 있습니다. 어떤 클러스터-송신자 채널이 어떤 전송 큐를 공유하는지, 어떤 채널에 별도의 전송 큐가 있는지, 그리고 어떤 채널이 클러스터 전송 큐를 사용하는지를 결정할 수 있습니다. [클러스터링: 클러스터 전송 큐 구성 방법의 계획](#)을 참조하십시오. 변경사항은 클러스터 큐 관리자 간의 메시지 전송을 관리하는 시스템 관리자를 지원합니다.

코어 클라이언트의 일부가 된 확장 트랜잭션 기능

확장 트랜잭션 기능이 IBM WebSphere MQ 코어 클라이언트에 통합되었습니다. 별도로 확장 트랜잭션 클라이언트 라이선스를 구매하거나 별도로 확장 트랜잭션 클라이언트 컴포넌트를 설치할 필요가 없습니다. [122 페이지의 『확장된 트랜잭션 클라이언트 개념』](#)의 내용을 참조하십시오.

애플리케이션 이름을 설정하여 큐 관리자에 대한 연결 식별

애플리케이션은 큐 관리자에 대한 연결을 식별하는 이름을 설정할 수 있습니다. **DISPLAY CONN** 명령을 사용하여 애플리케이션 이름을 표시하십시오. 이름이 **APPLTAG** 필드에서 리턴됩니다. 또는 IBM WebSphere MQ Explorer 애플리케이션 연결 창에 이름을 표시할 수 있습니다. 이 필드를 **App name**라고 합니다. [Java용 WebSphere MQ 클래스에 대한 WebSphere MQ 환경 설정](#)의 내용을 참조하십시오. z/OS®를 제외한 모든 플랫폼에서 애플리케이션 연결의 이름을 설정할 수 있습니다.

인증서 유효성 검증 정책

UNIX, Linux, and Windows에서는 인증서 체인 유효성 검증이 RFC 5280 산업 보안 표준을 얼마나 엄격하게 준수하는지 지정할 수 있습니다. [WebSphere MQ의 인증서 유효성 검증 정책](#)을 참조하십시오.

더 뛰어난 트랜잭션 시각성

dspmqrn 명령에는 비동기 롤백이 발생할 때 자세한 정보를 제공하기 위해 **-a** 및 **-q**의 두개의 새 매개변수가 있습니다. 두 개의 새 메시지 AMQ7486 및 AMQ7487은 롤백되는 트랜잭션에 대한 정보 및 트랜잭션이 연결과 연결되었는지에 대한 정보를 제공합니다.

시나리오

128 페이지의 『시나리오』에서는 새 IBM WebSphere MQ Version 7.5 기능을 쉽고 빠르게 사용하고 결합할 수 있는 방법을 보여줍니다. 시나리오에는 사용자가 관심이 있는 영역을 더 잘 이해할 수 있도록 도움을 주는 제품 문서 콘텐츠에 대한 유용한 링크가 포함되어 있습니다.

IBM WebSphere MQ Explorer

IBM WebSphere MQ Explorer Version 7.5의 새로운 특징에는 IBM WebSphere MQ Managed File Transfer와 IBM WebSphere MQ Advanced Message Security의 통합, 향상된 다중 버전 지원 및 오버헤드 감소가 포함됩니다.

IBM WebSphere MQ Explorer의 새로운 기능에 대한 자세한 정보는 [WebSphere MQ 탐색기의 새로운 기능 및 변경된 기능을 참조하십시오](#).

관련 개념

22 페이지의 『IBM WebSphere MQ Version 7.5 에서 변경된 사항』

23 페이지의 『IBM WebSphere MQ Version 7.5 수정팩의 변경사항』
Version 7.5 수정팩의 기능 및 자원에 대한 변경사항을 이 절에서 설명합니다.

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트

이제 IBM WebSphere MQ는 HP Integrity NonStop Server 플랫폼용 새 클라이언트를 지원합니다.

개요

IBM WebSphere MQ 클라이언트(HP Integrity NonStop Server 플랫폼용 클라이언트 포함) 개요는 [118 페이지의 『IBM WebSphere MQ MQI 클라이언트 개요』](#)의 내용을 참조하십시오.

HP Integrity NonStop Server 플랫폼의 IBM WebSphere MQ 클라이언트에 대한 기술 개요는 [126 페이지의 『HP Integrity NonStop Server 용 IBM WebSphere MQ 클라이언트 기술 개요』](#)의 내용을 참조하십시오.

For details of IBM WebSphere MQ client for HP Integrity NonStop Server supported environments and features, see [127 페이지의 『지원되는 HP Integrity NonStop Server 환경 및 기능에 대한 IBM WebSphere MQ 클라이언트』](#).

계획 중

HP Integrity NonStop Server 환경에 대해 IBM WebSphere MQ 클라이언트를 계획할 때 도움말을 보려면 [HP Integrity NonStop Server에서 IBM WebSphere MQ 클라이언트 환경 계획을 참조하십시오](#).

설치

HP Integrity NonStop Server용 IBM WebSphere MQ 클라이언트 설치에 대한 도움말입니다.

- 설치할 항목을 선택하려면 [HP Integrity NonStop Server용 IBM WebSphere MQ 클라이언트 컴포넌트를 참조하십시오](#).
- 설치 계획에 대해서는 [HP Integrity NonStop Server의 설치 계획을 참조하십시오](#).
 - 파일 시스템
- 하드웨어 및 소프트웨어 요구사항은 [HP Integrity NonStop Server의 하드웨어 및 소프트웨어 요구사항을 참조하십시오](#).
- 올바른 소프트웨어가 있는지 확인하려면 [필수 시스템 소프트웨어 확인을 참조하십시오](#).

- 시스템 준비는 [HP Integrity NonStop Server](#)에서 사용자 및 그룹 설정을 참조하십시오.
- 클라이언트를 설치하려면 [HP Integrity NonStop Server](#) 시스템에 [IBM WebSphere MQ 클라이언트 설치](#)를 참조하십시오.
- 설치를 확인하는 데 대해서는 [클라이언트 설치 확인](#)을 참조하십시오.
- 설치 제거는 [HP Integrity NonStop Server](#)에서 [IBM WebSphere MQ 설치 제거](#)를 참조하십시오.

HP Integrity NonStop Server 클라이언트 명령

다음 명령은 HP Integrity NonStop Server OSS 및 Guardian 환경의 IBM WebSphere MQ 클라이언트에 적용 가능합니다.

- [dspmqver](#)
- [endmqtrc](#)
- [mqrc](#)(MQ 리턴 코드)
- [runmqras](#)
- [runmqtmc](#)
- [strmqtrc](#)

다음 명령은 HP Integrity NonStop Server OSS 환경의 IBM WebSphere MQ 클라이언트에 적용 가능합니다.

- [dspmqtrc](#)

새 제품 ID인 MQNC가 [DISPLAY CHSTATUS](#) 명령 [제품 ID 값](#) 표에 추가되었습니다.

보안

HP Integrity NonStop Server 환경에 대해 IBM WebSphere MQ 클라이언트를 보호하려면 다음을 참조하십시오.

- HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트가 [클라이언트 애플리케이션에 대한 인증 계획](#)에 추가된 큐 관리자에 대한 자체를 식별하는 방법에 대한 정보입니다.
- [HP Integrity NonStop Server](#)에서 [보안 설정](#)
 - [OpenSSL](#)
 - [엔트로피 디먼](#)
- [IBM WebSphere MQ](#)의 [SSL 및 TLS 지원](#)
- [HP Integrity NonStop Server](#)에서 [SSL 또는 TLS에 대해 작업](#)
 - [인증서 관리](#)
 - [개인 인증서 저장소](#)
 - [인증서 신뢰 저장소](#)
 - [암호 문구 보관 파일](#)
 - [인증서 폐기 목록 파일](#)

TMF(Transaction Management Facility)

TMF(Transaction Management Facility)에 대한 정보는 다음 절 및 주제를 참조하십시오.

- [HP Integrity NonStop Server](#)에서 [IBM WebSphere MQ 클라이언트 환경 계획](#)
 - [HP Integrity NonStop Server 환경 준비](#)
 - [IBM WebSphere MQ 및 HP NonStop TMF](#)
 - [HP NonStop TMF 사용](#)
 - [글로벌 작업 단위 사용](#)
 - [장기 실행 트랜잭션 피하기](#)

- 사전 구성된 비활동 간격 후 글로벌 작업 단위가 만기되도록 하는 큐 관리자 구성에 대한 정보가 글로벌 작업 단위 만기에 추가되었습니다.
- HP Integrity NonStop Server 구성
 - 게이트웨이 프로세스 개요
 - Pathway에서 실행되도록 게이트웨이 구성
 - TMF 및 TMF/게이트웨이 스탠자
 - 클라이언트 초기화 파일 구성
 - 채널에 관한 부여
- 관리HP Integrity NonStop Server
 - Pathway에서 수동으로 TMF/게이트웨이 시작
 - Pathway에서 TMF/게이트웨이 중지
- HP Integrity NonStop Server에 대한 IBM WebSphere MQ 클라이언트 문제점 해결

애플리케이션 개발

HP Integrity NonStop Server 플랫폼에서 IBM WebSphere MQ 클라이언트의 애플리케이션 개발에 대한 정보는 다음을 참조하십시오.

- HP Integrity NonStop Server에서 응용프로그램 빌드
 - OSS 및 Guardian 헤더와 공용 라이브러리
 - HP Integrity NonStop Server에서 C 프로그램 준비
 - COBOL 프로그램 준비
 - pTAL 프로그램 준비
- pTAL을 사용한 코드화.
- HP Integrity NonStop Server용 IBM WebSphere MQ 클라이언트에 대한 JMS 프로그램 준비.

새 메시지

다음은 HP Integrity NonStop Server의 IBM WebSphere MQ 클라이언트에 대한 새 메시지입니다.

- AMQ5000-5999: 설치 가능 서비스
 - AMQ5370
 - AMQ5371
 - AMQ5372
 - AMQ5373
 - AMQ5374
 - AMQ5375
 - AMQ5376
 - AMQ5377
 - AMQ5378
 - AMQ5379
 - AMQ5380
 - AMQ5390
 - AMQ5391
 - AMQ5392
 - AMQ5393
 - AMQ5394

- [AMQ5395](#)
- [AMQ5396](#)
- [AMQ5397](#)
- [AMQ5398](#)
- [AMQ5399](#)
- AMQ9000-9999: 원격
 - [AMQ9816](#)
 - [AMQ9817](#)
 - [AMQ9818](#)
 - [AMQ9819](#)
 - [AMQ9820](#)
 - [AMQ9821](#)
 - [AMQ9823](#)
 - [AMQ9824](#)

수정된 API 이유 코드

다음 기존 API 이유 코드는 이제 HP Integrity NonStop Server를 포함합니다.

- [2354\(0932\)\(RC2354\): MQRC_UOW_ENLISTMENT_ERROR](#)
- [2355\(0933\)\(RC2355\): MQRC_UOW_MIX_NOT_SUPPORTED](#)
- [2072 \(0818\) \(RC2072\): MQRC_SYNCPOINT_NOT_AVAILABLE](#)
- [2003\(07D3\)\(RC2003\): MQRC_BACKED_OUT](#)

샘플

샘플 프로그램에서 설명하는 기술에 대한 정보는 [HP Integrity NonStop Server용 IBM WebSphere MQ 클라이언트 샘플](#)을 참조하십시오.

문제점 해결 및 지원

HP Integrity NonStop Server의 IBM WebSphere MQ 클라이언트에 대한 문제점 해결 및 지원 정보는 다음 주제를 참조하십시오.

- [HP Integrity NonStop Server에 대한 IBM WebSphere MQ 클라이언트 문제점 해결](#)
- [HP Integrity NonStop Server의 오류 로그](#)
- [HP Integrity NonStop Server에서 추적 사용](#)
- [FFST: HP Integrity NonStop Server에 대한 IBM WebSphere MQ](#)

IBM WebSphere MQ Version 7.5 에서 변경된 사항

큐 관리자를 IBM WebSphere MQ Version 7.5로 업그레이드하기 위해서는 먼저 변경사항 목록을 주의하여 검토하십시오. Version 7.5로의 시스템 마이그레이션을 시작하기 전에 먼저 기존 애플리케이션, 스크립트 및 프로시저 변경 계획을 세워야 할지를 결정하십시오.

다음 링크는 제품 문서의 마이그레이션 및 업그레이드 절에 있는 정보에 대한 링크입니다. 새 기능, 기존 애플리케이션에 영향을 주지 않는 변경사항, 관리 프로시저 및 관리 스크립트는 여기에 나열되지 않습니다. [17 페이지의 『IBM WebSphere MQ Version 7.5의 새로운 기능』](#)의 내용을 참조하십시오.

[버전, 릴리스 및 유지보수 레벨별 변경사항 목록](#)

- [V7.1에서 V7.5로의 변경](#)

관련 개념

17 페이지의 『IBM WebSphere MQ Version 7.5의 새로운 기능』
IBM WebSphere MQ Version 7.5의 주요 새 기능에 대해 학습하십시오.

23 페이지의 『IBM WebSphere MQ Version 7.5 수정팩의 변경사항』
Version 7.5 수정팩의 기능 및 자원에 대한 변경사항을 이 절에서 설명합니다.

IBM WebSphere MQ Version 7.5 수정팩의 변경사항

Version 7.5 수정팩의 기능 및 자원에 대한 변경사항을 이 절에서 설명합니다.

- **V7.5.0.1** 24 페이지의 『Version 7.5.0, Fix Pack 1: WebSockets 에 대한 MQTT 지원』
- **V7.5.0.4** 25 페이지의 『Version 7.5.0, Fix Pack 4: AMQ_DISABLE_CLIENT_AMS를 사용하여 클라이언트에서 IBM WebSphere MQ Advanced Message Security 사용 안함』
- **V7.5.0.5** 25 페이지의 『Version 7.5.0, Fix Pack 5: mqclient.ini 파일을 사용하여 클라이언트에서 IBM WebSphere MQ Advanced Message Security 사용 안함』
- **Windows V7.5.0.6** 25 페이지의 『Version 7.5.0, Fix Pack 6: 관리 대상 .NET 클라이언트 애플리케이션에 대한 사용자 ID 및 비밀번호』
- **V7.5.0.6** 25 페이지의 『Version 7.5.0, Fix Pack 6: 더 이상 사용되지 않는 cipherspec』
- **V7.5.0.6** 26 페이지의 『Version 7.5.0, Fix Pack 6: IBM WebSphere MQ Managed File Transfer 에 대한 서비스 기능이 향상되었습니다.』
- **V7.5.0.6** 26 페이지의 『Version 7.5.0, Fix Pack 6 새 IBM WebSphere MQ Managed File Transfer 에 이진트 특성 failTransferOnFirstFailure』
- **V7.5.0.7** 26 페이지의 『Version 7.5.0, Fix Pack 7: 더 이상 사용되지 않는 cipherspec』
- **Windows V7.5.0.7** 26 페이지의 『Version 7.5.0, Fix Pack 7: Oracle 12c 데이터베이스에 대한 JDBC 연결을 지원』
- **Windows V7.5.0.7** 26 페이지의 『Version 7.5.0, Fix Pack 7: MQCNO_CLIENT_BINDING 및 MQCNO_LOCAL_BINDING에 대한 .NET 사용 제한 제거』
- **V7.5.0.8** 27 페이지의 『Version 7.5.0, Fix Pack 8: 분배 목록에서 토픽 알리어스 큐 사용 제한』
- **V7.5.0.8** 27 페이지의 『Version 7.5.0, Fix Pack 8: GSKit 버전이 업데이트됨』
- **V7.5.0.8** 27 페이지의 『Version 7.5.0, Fix Pack 8: 더 이상 사용되지 않는 CipherSpecs』
- **V7.5.0.8** 27 페이지의 『Version 7.5.0, Fix Pack 8: JmsConstants 인터페이스에 새 상수인 JMS_IBM_SUBSCRIPTION_USER_DATA가 추가됨』
- **V7.5.0.8** 27 페이지의 『Version 7.5.0, Fix Pack 8: JMS 예외 리스너 업데이트』
- **V7.5.0.8** 28 페이지의 『Version 7.5.0, Fix Pack 8: IBM WebSphere MQ classes for JMS ObjectMessage 에서 클래스 이름 allowlisting에 대한 지원』
- **V7.5.0.8** 28 페이지의 『Version 7.5.0, Fix Pack 8: 새 IBM WebSphere MQ Managed File Transfer 에 이진트 특성 additionalWildcardSandboxChecking』
- **V7.5.0.8** 28 페이지의 『Version 7.5.0, Fix Pack 8: IBM WebSphere MQ Managed File Transfer fteCleanAgent 명령의 작동 변경』
- **V7.5.0.9** 28 페이지의 『Version 7.5.0, Fix Pack 9: 새 환경 변수 AMQ_SSL_LDAP_SERVER_VERSION』
- **V7.5.0.9** 29 페이지의 『Version 7.5.0, Fix Pack 9: IBM WebSphere MQ classes for JMS 의 샘플 구성 파일』

- **V7.5.0.9** 29 페이지의 『Version 7.5.0, Fix Pack 9: dmpmqcfg 명령에 대한 변경』
- **V7.5.0.9** 29 페이지의 『Version 7.5.0, Fix Pack 9: strmqm 명령에 대한 변경』
- **V7.5.0.9** 29 페이지의 『runmqras 유틸리티에 대한 개선사항』
- **V7.5.0.9** 30 페이지의 『Version 7.5.0, Fix Pack 9: 메시지 대 파일 전송을 위해 저장된 groupId 속성의 값과 전송 ID의 비교 검사』
- **V7.5.0.9** 29 페이지의 『Version 7.5.0, Fix Pack 9: 활성화 스펙에서 targetClientMatching 특성에 대해 자원 어댑터 업데이트』
- **V7.5.0.9** 29 페이지의 『Version 7.5.0, Fix Pack 9: runmqras 명령의 새 -caseno 매개변수』
- **V7.5.0.9** 30 페이지의 『Version 7.5.0, Fix Pack 9: 큐 관리자 클러스터링 컴포넌트에서 사용되는 내부 MQ 명령 메시지 내의 손상된 문자 데이터 관련 오류 처리에 대한 변경』
- **V7.5.0.9** **Windows** 30 페이지의 『Version 7.5.0, Fix Pack 9: Windows 에 Microsoft Visual Studio 2013 C/C++ 런타임 추가』
- **V7.5.0.9** 30 페이지의 『Version 7.5.0, Fix Pack 9: 메시지 대 파일 전송을 위해 저장된 groupId 속성의 값과 전송 ID의 비교 검사』
- **V7.5.0.9** 30 페이지의 『Version 7.5.0, Fix Pack 9: 메시지 대 파일 전송을 위해 저장된 groupId 속성의 값과 전송 ID의 비교 검사』
- **V7.5.0.9** 30 페이지의 『Version 7.5.0, Fix Pack 9 . APAR IT26482: 클러스터 큐의 BackoutThreshold 및 BackoutRequeueQName 를 조회하기 위해 IBM WebSphere MQ classes for JMS 에 필요한 권한으로 변경』

Version 7.5.0, Fix Pack 1: WebSockets 에 대한 MQTT 지원

V7.5.0.1

새 통신 프로토콜 매개변수(PROTOCOL)가 MQTT 채널 정의(DEFINE CHANNEL(MQTT))에 추가되었습니다.

- 매개변수가 MQTTV3으로 설정된 경우 채널이 MQTT(MQ Telemetry Transport) 프로토콜 버전 3을 사용하는 클라이언트로부터의 연결만 승인합니다. IBM WebSphere MQ Version 7.5.0, 수정팩 1 이전에서는 이 프로토콜만 지원되었습니다.
- 매개변수가 HTTP로 설정되면 채널은 페이지에 대한 HTTP 요청 또는 IBM WebSphere MQ Telemetry에 대한 WebSockets 연결만 허용합니다.
- 매개변수가 MQTTV3, HTTP로 설정된 경우 채널이 두 프로토콜 중 하나를 사용하는 클라이언트로부터의 연결을 승인합니다. 이는 IBM WebSphere MQ Version 7.5.0, 수정팩 1 이상 버전으로 작성된 새 MQTT 채널의 기본 동작입니다.

자세한 정보는 SSL에서 JavaScript용 MQTT 메시징 클라이언트 연결 및 WebSockets을 참조하십시오.

클라이언트가 SSL을 사용하여 MQTT 채널에 연결할 때, SSLCAUTH 매개변수는 IBM WebSphere MQ 가 클라이언트의 인증서를 필요로 하는지 여부를 판별합니다 (DEFINE CHANNEL (MQTT) 참조). IBM WebSphere MQ Version 7.5.0, 수정팩 1이전에 이 매개변수는 MQTT 채널에 대해 REQUIRED 또는 OPTIONAL 이 될 수 있습니다.

- REQUIRED는 IBM WebSphere MQ가 클라이언트로부터 인증서를 요청하고 클라이언트가 유효한 인증서를 제공해야 함을 의미합니다.
- OPTIONAL은 IBM WebSphere MQ가 클라이언트로부터 인증서를 요청하지만 클라이언트가 해당 인증서를 제공하지 않아도 됨을 의미합니다. 클라이언트가 유효한 인증서를 제공하거나 클라이언트가 인증서를 제공하지 않은 경우 모두 클라이언트 연결이 허용됩니다. 단, 클라이언트가 유효하지 않은 인증서를 제공한 경우에만 클라이언트 연결이 허용되지 않습니다.

IBM WebSphere MQ Version 7.5.0, 수정팩 1 이상에서는 MQTT 채널에 대해 SSLCAUTH 매개변수를 NEVER 로 설정할 수 있습니다. NEVER는 IBM WebSphere MQ가 클라이언트로부터 인증서를 요청하지 않음을 의미합니다. 새 값은 JavaScript에 대한 MQTT 메시징 클라이언트를 사용하는 클라이언트에 대한 지원의 일부로 추가되

었습니다. 이 값은 클라이언트 인증서에 대한 요청을 프로토콜 오류로 처리하는 일부 웹 브라우저의 동작을 수용합니다.

Version 7.5.0, Fix Pack 4: AMQ_DISABLE_CLIENT_AMS를 사용하여 클라이언트에서 IBM WebSphere MQ Advanced Message Security 사용 안함

V 7.5.0.4

Version 7.5.0, Fix Pack 4에서 환경 변수 AMQ_DISABLE_CLIENT_AMS를 사용하여 IBM WebSphere MQ classes for JMS 및 IBM WebSphere MQ classes for Java 클라이언트에서 IBM WebSphere MQ Advanced Message Security (AMS) 를 사용 불가능하게 하여 제품의 이전 버전에서 실행 중인 큐 관리자에 연결할 때 오류를 방지할 수 있습니다. 또는 Version 7.5.0, Fix Pack 4에서 Java 시스템 특성 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS 를 사용하여 클라이언트에서 AMS 를 사용 불가능하게 할 수 있습니다.

자세한 정보는 [클라이언트에서 IBM WebSphere MQ Advanced Message Security 사용 안함을 참조하십시오.](#)

Version 7.5.0, Fix Pack 5: mqclient.ini 파일을 사용하여 클라이언트에서 IBM WebSphere MQ Advanced Message Security 사용 안함

V 7.5.0.5

Version 7.5.0, Fix Pack 4에서 클라이언트에서 IBM WebSphere MQ Advanced Message Security (AMS) 를 사용하지 않도록 설정하면 구성 변수 AMQ_DISABLE_CLIENT_AMS를 설정하여 제품의 이전 버전에서 실행 중인 큐 관리자에 연결할 때 오류를 방지할 수 있습니다. Version 7.5.0, Fix Pack 5에서 mqclient.ini 파일의 **Security** 스탠자 아래에서 DisableClientAMS 특성을 사용하여 AMS 를 사용 안함으로 설정할 수도 있습니다. 자세한 정보는 [클라이언트에서 IBM WebSphere MQ Advanced Message Security 사용 안함을 참조하십시오.](#)

Version 7.5.0, Fix Pack 6: 관리 대상 .NET 클라이언트 애플리케이션에 대한 사용자 ID 및 비밀번호

Windows V 7.5.0.6

IBM WebSphere MQ Version 7.5.0, 수정팩 6부터는 관리 .NET 클라이언트 애플리케이션과 함께 지정되는 사용자 ID 및 비밀번호가 클라이언트 보안 엑시트에 전달되는 IBM WebSphere MQ .NET MQChannelDefinition 클래스에서 설정됩니다. 자세한 정보는 [IBM WebSphere MQ .NET의 채널 엑시트 사용을 참조하십시오.](#)

Version 7.5.0, Fix Pack 6: 더 이상 사용되지 않는 cipherspec

다음 암호 스펙은 IBM WebSphere MQ Version 7.5.0, 수정팩 6에서 더 이상 사용되지 않습니다.

- DES_SHA_EXPORT
-  DES_SHA_EXPORT1024
-  FIPS_WITH_DES_CBC_SHA
- NULL_MD5
- NULL_SHA
- TLS_RSA_WITH_DES_CBC_SHA
- ECDHE_ECDSA_NULL_SHA256
-  ECDHE_RSA_NULL_SHA256
-  TLS_RSA_WITH_NULL_NULL
- TLS_RSA_WITH_NULL_SHA256

자세한 정보는 [더 이상 사용되지 않는 CipherSpecs](#)를 참조하십시오.

Version 7.5.0, Fix Pack 6: IBM WebSphere MQ Managed File Transfer 에 대한 서비스 기능이 향상되었습니다.

V 7.5.0.6

IBM WebSphere MQ Version 7.5.0, 수정팩 6부터 IBM WebSphere MQ Managed File Transfer에 대해 다음과 같이 변경되었습니다.

- `installation.properties` 파일에서 `commandMessage` 우선순위 특성의 기본값은 8로 변경되었습니다. 자세한 정보는 `installation.properties` 파일을 참조하십시오.
- `agent.properties` 파일에 있는 `logTransferRecovery` 특성의 기본값이 `true`로 변경되었습니다. 자세한 정보는 `agent.properties` 파일을 참조하십시오.
- `fteShowAgentDetails` 명령에 `-d` 매개변수를 사용할 때 `installation.properties` 파일에서 설정할 `enableFunctionalFixPack` 특성에 대한 필요성이 제거됩니다. 자세한 정보는 `fteShowAgentDetails`를 참조하십시오.
- 에이전트에 복구할 수 없는 오류가 발생하는 경우 FFDC(First Failure Data Capture)가 생성됩니다.

Version 7.5.0, Fix Pack 6 새 IBM WebSphere MQ Managed File Transfer 에이전트 특성 `failTransferOnFirstFailure`

distributed V 7.5.0.6

IBM WebSphere MQ Version 7.5.0, 수정팩 6부터 `agent.properties` 파일에서 `failTransferOnFirstFailure` 특성을 설정하여 관리 전송 내의 전송 항목이 실패하자마자 관리 전송을 실패하도록 에이전트를 구성할 수 있습니다. 자세한 정보는 `agent.properties` 파일을 참조하십시오.

Version 7.5.0, Fix Pack 7: 더 이상 사용되지 않는 `cipherspec`

다음 암호 스펙은 IBM WebSphere MQ Version 7.5.0, 수정팩 7에서 더 이상 사용되지 않습니다.

- RC2_MD5_EXPORT
- RC4_MD5_EXPORT
- RC4_MD5_US
- RC4_SHA_US
- RC4_56_SHA_EXPORT1024
- ECDHE_ECDSA_RC4_128_SHA256
- ECDHE_RSA_RC4_128_SHA256
- TLS_RSA_WITH_RC4_128_SHA256

자세한 정보는 더 이상 사용되지 않는 CipherSpecs를 참조하십시오.

Version 7.5.0, Fix Pack 7: Oracle 12c 데이터베이스에 대한 JDBC 연결을 지원

Windows V 7.5.0.7

IBM WebSphere MQ Version 7.5.0, 수정팩 7에서 새 파일 `jdbcora12.d11`은 IBM WebSphere MQ Windows 서버 설치 이미지와 함께 제공되어 Oracle 12c 데이터베이스에 대한 JDBC 연결을 지원합니다 ([Windows에서 JTA/JDBC 조정 구성참조](#)).

Version 7.5.0, Fix Pack 7: `MQCNO_CLIENT_BINDING` 및 `MQCNO_LOCAL_BINDING`에 대한 .NET 사용 제한 제거

Windows V 7.5.0.7

IBM WebSphere MQ Version 7.5.0, 수정팩 7에서, 클라이언트 전용 설치에서 실행할 때 올바른 클라이언트 연결 구성이 사용되도록 Microsoft Windows Communication Foundation (WCF)의 IBM WebSphere MQ 사용자

정의 채널이 업데이트되었습니다. 자세한 정보는 [MQCONNX 호출을 사용하여 큐 관리자에 연결을 참조하십시오](#).

Version 7.5.0, Fix Pack 8: 분배 목록에서 토픽 알리어스 큐 사용 제한

V 7.5.0.8

분배 목록에서는 토픽 오브젝트를 가리키는 알리어스 큐 사용을 지원하지 않습니다. Version 7.5.0, Fix Pack 8 부터는 분배 목록에서 알리어스 큐가 토픽 오브젝트를 가리키는 경우 IBM WebSphere MQ가 MQRC_ALIAS_BASE_Q_TYPE_ERROR를 리턴합니다.

Version 7.5.0, Fix Pack 8: GSKit 버전이 업데이트됨

V 7.5.0.8

GSKit 버전이 업데이트되었습니다. 새 버전의 GSKit은 키 데이터베이스 비밀번호를 숨길 .sth 파일을 생성할 때 사용되는 스테쉬 파일 형식을 변경합니다. 이 버전의 GSKit과 함께 생성되는 스테쉬 파일은 이전 버전의 GSKit에서 읽을 수 없습니다. Version 7.5.0, Fix Pack 8 이상과 함께 생성되는 스테쉬 파일이 사용자의 애플리케이션 및 기타 IBM WebSphere MQ 설치와 호환되도록 하려면, 호환되는 버전의 GSKit이 포함된 IBM WebSphere MQ의 버전으로 업데이트해야 합니다. IBM WebSphere MQ Version 7.5의 경우, Version 7.5.0, Fix Pack 8입니다.

사용자의 애플리케이션 또는 기타 IBM WebSphere MQ 설치를 업데이트할 수 없는 경우, 이전 버전과 호환 가능한 스테쉬 파일 형식을 요청할 수 있습니다. **runmqakm** 또는 **runmqckm** 명령을 **-stash** 또는 **-stashpw** 옵션과 함께 사용하는 경우, **-v1stash** 명령행 매개변수를 포함하십시오. iKeyman GUI를 사용하여 이전 버전과 호환 가능한 스테쉬 파일을 생성할 수 없습니다.

Version 7.5.0, Fix Pack 8: 더 이상 사용되지 않는 CipherSpecs

V 7.5.0.8

Version 7.5.0, Fix Pack 8부터는 다음 CipherSpec이 더 이상 사용되지 않습니다.

-  FIPS_WITH_3DES_EDE_CBC_SHA
- TRIPLE_DES_SHA_US
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
-  ECDHE_ECDSA_3DES_EDE_CBC_SHA256
-  ECDHE_RSA_3DES_EDE_CBC_SHA256

자세한 정보는 [더 이상 사용되지 않는 CipherSpecs](#)를 참조하십시오.

Version 7.5.0, Fix Pack 8: JmsConstants 인터페이스에 새 상수인

JMS_IBM_SUBSCRIPTION_USER_DATA가 추가됨

V 7.5.0.8

Version 7.5.0, Fix Pack 8부터는 MQPS 폴더와 RFH2 헤더를 포함하는 큐에서 메시지를 이용할 때 Sud 키(존재하는 경우)와 연관된 값이 IBM WebSphere MQ classes for JMS 애플리케이션에 리턴되는 JMS 메시지 오브젝트에 문자열 특성으로 추가되도록 IBM WebSphere MQ classes for JMS가 업데이트됩니다. 애플리케이션이 메시지에서 이 특성을 검색할 수 있도록 하기 위해 새 상수, JMS_IBM_SUBSCRIPTION_USER_DATA가 JmsConstants 인터페이스에 추가됩니다. 이 새 특성은 구독 사용자 데이터를 검색하기 위해 `javax.jms.Message.getStringProperty(java.lang.String)` 메소드와 함께 사용할 수 있습니다. 자세한 정보는 [사용자 구독 데이터의 검색 및 DEFINE SUB](#)를 참조하십시오.

Version 7.5.0, Fix Pack 8: JMS 예외 리스너 업데이트

V 7.5.0.8

IBM WebSphere MQ Version 7.5.0, 수정팩 8에서 JMS MessageListener 및 JMS ExceptionListener를 구성하는 현재 JMS 응용프로그램의 동작을 유지하기 위해 ASYNC_EXCEPTIONS JMS ConnectionFactory 특성의 기

본값이 Version 7.5의 IBM WebSphere MQ classes for JMS 에 대해 ASYNC_EXCEPTIONS_CONNECTIONBROKEN으로 변경되었습니다. 따라서 기본적으로 중단된 연결 오류 코드에 해당하는 예외만 애플리케이션의 JMS ExceptionListener에 전달됩니다. 연결 중단 예외는 메시지를 동기 또는 비동기로 이용할 때 예외 리스너로 전달됩니다.

애플리케이션이 사용하는 JMS ConnectionFactory 에 ASYNC_EXCEPTIONS_ALL값으로 설정된 ASYNC_EXCEPTIONS 특성이 있는 경우 비동기 메시지 이용자로 메시지 전달 중에 발생하는 연결되지 않은 오류와 관련된 JMS 예외가 등록된 ExceptionListener 로 전달되도록 Version 7.5 에 대한 IBM WebSphere MQ classes for JMS 도 갱신되었습니다.

자세한 정보는 [JMS: 버전 7.5의 예외 리스너 변경사항 및 JMS용 IBM WebSphere MQ 클래스의 예외를 참조하십시오.](#)

Version 7.5.0, Fix Pack 8: IBM WebSphere MQ classes for JMS ObjectMessage 에서 클래스 이름 allowlisting에 대한 지원

V 7.5.0.8

IBM WebSphere MQ Version 7.5.0, 수정팩 8에서 IBM WebSphere MQ classes for JMS 는 JMS ObjectMessage 인터페이스의 구현에서 클래스의 allow리스팅을 지원합니다. 허용 목록은 ObjectMessage.setObject()로 직렬화될 수 있고 ObjectMessage.getObject()로 역직렬화될 수 있는 Java 클래스를 정의합니다.

자세한 정보는 [JMS ObjectMessage 의 클래스 이름 allowlisting 및 Java Security Manager에서 JMS 응용프로그램에 대해 IBM WebSphere MQ 클래스 실행의 내용을 참조하십시오.](#)

Version 7.5.0, Fix Pack 8: 새 IBM WebSphere MQ Managed File Transfer 에이전트 특성 additionalWildcardSandboxChecking

V 7.5.0.8

IBM WebSphere MQ Version 7.5.0, 수정팩 8부터는, 에이전트가 파일을 전송할 수 있는 위치를 제한하기 위해 사용자 또는 에이전트 샌드박스를 사용하여 에이전트를 구성하는 경우, additionalWildcardSandboxChecking 특성을 true로 설정하여 해당 에이전트의 와일드카드 전송에 대해 추가적인 점검이 이루어지게 지정할 수 있습니다. 자세한 정보는 [와일드카드 전송 추가 점검 및 agent.properties 파일을 참조하십시오.](#)

Version 7.5.0, Fix Pack 8: IBM WebSphere MQ Managed File Transfer fteCleanAgent 명령의 작동 변경

V 7.5.0.8

IBM WebSphere MQ Version 7.5.0, 수정팩 8부터는 **fteCleanAgent** 명령이 업데이트되어 에이전트 이름과 적절한 매개변수를 명령에 전달하여 정리하려는 IBM WebSphere MQ Managed File Transfer 상태를 지정해야 합니다. 이와 같이 작동이 변경되어도 **fteCleanAgent**는 기본적으로 모든 진행 중 및 보류 중인 전송, 자원 모니터 정의 및 지정된 에이전트에 스케줄된 전송 정의를 지우지 않습니다.

필요한 경우 command.properties 파일에서 새 failCleanAgentWithNoArguments 특성을 false로 설정하여 **fteCleanAgent**의 이전 작동으로 되돌릴 수 있습니다.

자세한 정보는 [fteCleanAgent\(Managed File Transfer 에이전트 정리\) 및 command.properties 파일을 참조하십시오.](#)

Version 7.5.0, Fix Pack 9: 새 환경 변수 AMQ_SSL_LDAP_SERVER_VERSION

V 7.5.0.9

Version 7.5.0, Fix Pack 9에서 환경 변수 AMQ_SSL_LDAP_SERVER_VERSION을 설정하여, 인증서가 더 높은 버전의 LDAP CRL(Certificate Revocation List) 서버에 대해 유효성 검증되는 경우 IBM WebSphere MQ가 요구되는 LDAP(lightweight directory access protocol) 버전(LDAP v2 또는 LDAP v3)을 GSKit에 전달하도록 할 수 있습니다. 자세한 정보는 [환경 변수를 참조하십시오.](#)

Version 7.5.0, Fix Pack 9: IBM WebSphere MQ classes for JMS 의 샘플 구성 파일

V 7.5.0.9

Version 7.5.0, Fix Pack 9에서 샘플 구성 파일 `mjjava.config`는 IBM WebSphere MQ classes for JMS 설치 디렉토리의 `bin` 서브디렉토리에 제공됩니다. 자세한 정보는 [Java용 IBM MQ 클래스 구성 파일을 참조하십시오.](#)

Version 7.5.0, Fix Pack 9: `dmpmqcfg` 명령에 대한 변경

V 7.5.0.9

Version 7.5.0, Fix Pack 9에서 명령행 스위치 `-w`가 `dmpmqcfg`에 추가됩니다. 이 스위치는 `dmpmqcfg`가 명령 서버에서 응답 메시지에 대해 대기하는 시간(초)을 늘리도록 합니다.

자세한 정보는 `dmpmqcfg`를 참조하십시오.

Version 7.5.0, Fix Pack 9: `strmqm` 명령에 대한 변경

V 7.5.0.9

Version 7.5.0, Fix Pack 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.

runmqras 유틸리티에 대한 개선사항

V 7.5.0.9

Version 7.5.0, Fix Pack 9부터 `runmqras` 유틸리티가 다음과 같이 개선되었습니다.

- ▶ **AIX** ▶ **Solaris** ▶ **Linux** 기본적으로 환경 변수 정보가 검색됩니다.
- ▶ **distributed** 기본적으로 큐 관리자 데이터 디렉토리 목록이 검색됩니다.
- ▶ 다음 두 섹션이 `runmqras` 명령에 추가됩니다.
 - ▶ **Linux** ▶ **UNIX** IBM WebSphere MQ 프로세스 자원 사용량 정보를 수집하는 `leak` 섹션입니다.
 - ▶ **distributed** `fteRas` 명령으로 얻은 데이터를 캡처하기 위한 `mft` 섹션입니다.

자세한 정보는 `runmqras`를 참조하십시오.

Version 7.5.0, Fix Pack 9: 활성화 스펙에서 `targetClientMatching` 특성에 대해 자원 어댑터 업데이트

V 7.5.0.9

IBM WebSphere MQ JCA 자원 어댑터(MQ-RA)가 업데이트되어 `targetClientMatching` 특성이 활성화 스펙에 맞게 구성될 수 있습니다. 요청 메시지에 `MQRFH2` 헤더가 포함되지 않은 경우 응답 메시지에 `MQRFH2` 헤더가 포함되도록 특성을 구성할 수 있습니다. 이는 애플리케이션이 응답 메시지에서 정의하는 메시지 특성이 메시지 전송 시에 포함됨을 의미합니다. 자세한 정보는 [활성화 스펙에 대한 targetClientMatching 특성 구성의 내용](#)을 참조하십시오.

Version 7.5.0, Fix Pack 9: `runmqras` 명령의 새 `-caseno` 매개변수

V 7.5.0.9

`-caseno` 매개변수는 유효한 Salesforce 케이스 번호를 지정할 수 있는 것을 제외하고는 `-pmrno`와 같습니다. `-caseno` 또는 `-pmrno`는 모두 선택적 매개변수이지만 둘 다 함께 제공하는 것은 허용되지 않습니다. 자세한 정보는 `runmqras`(IBM MQ 진단 정보 수집)의 내용을 참조하십시오.

Version 7.5.0, Fix Pack 9: 큐 관리자 클러스터링 컴포넌트에서 사용되는 내부 MQ 명령 메시지 내의 손상된 문자 데이터 관련 오류 처리에 대한 변경

V 7.5.0.9

Version 7.5.0, Fix Pack 9에서 큐 관리자가 큐 관리자의 클러스터링 구성요소에서 사용하는 내부 MQ 명령 메시지 내의 손상된 문자 데이터와 관련된 오류를 처리하는 방법이 변경됩니다. 변환할 수 없는 문자를 포함할 수 있는 필드가 공백이 되고 변환이 재시도됩니다. 변환에 성공하면 업데이트가 업데이트가 공백 설명과 함께 저장되고 관리자에게 알리기 위해 오류 메시지 AMQ6174 및 AMQ9880이 작성됩니다. 메시지를 여전히 변환할 수 없는 경우 데드-레터 큐로 이동되고 오류 메시지가 작성됩니다.

Version 7.5.0, Fix Pack 9: Windows 에 Microsoft Visual Studio 2013 C/C++ 런타임 추가

V 7.5.0.9 Windows

Version 7.5.0, Fix Pack 9에서 IBM WebSphere MQ 의 일부 요소에는 Microsoft Visual Studio 2013 C/C++ (VS2013) 런타임이 필요합니다. 이러한 런타임은 IBM WebSphere MQ Version 7.5.0, 수정팩 9 신규 설치 시 설치됩니다.

VS2013 런타임이 시스템에 없는 경우 새 클라이언트를 설치할 때 설치되며 클라이언트를 7.5.0.9 유지보수 레벨로 업데이트하거나 7.5.0.9 수정팩을 적용합니다. 대부분의 경우 이 프로세스는 사용자 조치를 필요로 하지 않지만, 서버 수정팩을 적용하는 중에 설치 실패가 발생하는 경우에는 런타임을 수동으로 설치할 수 있도록 지정해야 하는 환경 변수가 있습니다. 자세한 정보는 [서버 수정팩의 Microsoft Visual Studio 2013 C/C++ 런타임 설치](#)를 참조하십시오.

Version 7.5.0, Fix Pack 9: 메시지 대 파일 전송을 위해 저장된 groupId 속성의 값과 전송 ID의 비교 검사

V 7.5.0.9

Version 7.5.0, Fix Pack 9에서 IBM WebSphere MQ Managed File Transfer 는 전송 요청 XML 페이로드 내의 groupId 속성 값 및 전송 ID의 Version 7.5.0, Fix Pack 6에 있는 APAR IT18213 에 의해 이전에 제거된 비교 확인을 복원하도록 갱신됩니다. 두 ID가 동등한 경우 소스 에이전트는 메시지 대 파일 전송의 입력 큐에서 작성되는 첫 번째 MQGET 시도에 대해 메시지 ID 일치 옵션(그룹 ID 일치 옵션과 반대임)으로 ID를 사용합니다. [메시지에서 파일로 데이터 전송도](#) 참조하십시오.

Version 7.5.0, Fix Pack 9 . APAR IT26482: 클러스터 큐의 BackoutThreshold 및 BackoutRequeueQName 를 조회하기 위해 IBM WebSphere MQ classes for JMS 에 필요한 권한으로 변경

V 7.5.0.9

Version 7.5.0, Fix Pack 9 에 APAR IT26482의 임시 수정사항이 추가된 경우 클러스터 큐의 **BackoutThreshold** 및 **BackoutRequeueQName** 를 조회하기 위해 조회 액세스만 필요하도록 IBM WebSphere MQ classes for JMS 이 업데이트되었습니다. 다른 모든 버전의 경우에는 찾아보기 및 액세스 권한도 필요합니다. 자세한 정보는 [JMS용 IBM MQ 클래스에서 변조 메시지 처리](#)를 참조하십시오.

관련 개념

[17 페이지의 『IBM WebSphere MQ Version 7.5의 새로운 기능』](#)
IBM WebSphere MQ Version 7.5의 주요 새 기능에 대해 학습하십시오.

[22 페이지의 『IBM WebSphere MQ Version 7.5 에서 변경된 사항』](#)

관련 정보

[IBM WebSphere MQ의 시스템 요구사항](#)

[IBM MQ, WebSphere MQ 및 MQSeries 제품 Readme 웹 페이지](#)

[WebSphere MQ에 대한 권장 수정사항](#)

[WebSphere MQ 계획된 유지보수 릴리스 날짜](#)

새로운 기능 및 이전 버전에서 변경된 사항

IBM WebSphere MQ Version 7.5 이전의 제품 버전에서 발생한 안정화, 폐기 및 제거 등 기능 및 자원에 대한 변경사항 및 새 기능에 대한 내용입니다.

제품의 이전 버전에서 변경된 사항 및 새 기능에 대한 정보는 해당 버전의 제품 문서에서 해당 절을 참조하십시오.

IBM WebSphere MQ Version 7.1

- [IBM WebSphere MQ 버전 7.1의 새 기능](#)
- [IBM WebSphere MQ 버전 7.1의 변경사항](#)
- [IBM WebSphere MQ 버전 7.1 수정팩의 변경사항](#)

IBM WebSphere MQ Version 7.0.1 이전

문서가 IBM Documentation 외부에서 제공되는 이전 버전의 제품의 경우, [WebSphere MQ의 이전 버전에 대한 Documentation](#)를 참조하십시오.

관련 개념

[17 페이지의 『IBM WebSphere MQ Version 7.5의 새로운 기능』](#)
IBM WebSphere MQ Version 7.5의 주요 새 기능에 대해 학습하십시오.

[22 페이지의 『IBM WebSphere MQ Version 7.5에서 변경된 사항』](#)

[23 페이지의 『IBM WebSphere MQ Version 7.5 수정팩의 변경사항』](#)
Version 7.5 수정팩의 기능 및 자원에 대한 변경사항을 이 절에서 설명합니다.

[WebSphere MQ 마이그레이션 소개](#)

이전 IBM WebSphere MQ 발행물과 현재 정보 구조 간 맵핑

IBM Documentation의 정보는 일반 활동 세트에 따라 구조화됩니다. 예를 들어, 설치, 개발, 관리, 구성, 보안. 이전의 발행물("MQ 서적")은 조금 다르게 구조화되었습니다. 이 절에서는 이전 IBM WebSphere MQ 서적과 현재 정보 구조 간의 맵핑을 제공합니다.

AIX용 빠른 시작

이 절에서는 이전 AIX용 빠른 시작 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [AIX: 최신 버전으로의 마이그레이션 계획](#)
- [요구사항 검사](#)
- [시스템 준비](#)
- [AIX에 IBM WebSphere MQ 서버 설치](#)
- [서버 설치 확인](#)
- [AIX 시스템에서 IBM WebSphere MQ 클라이언트 설치](#)
- [클라이언트 설치 확인](#)
- [AIX: IBM WebSphere MQ Version 7.5에서 유지보수 레벨 업그레이드 적용](#)
- [AIX에서 IBM WebSphere MQ 설치 제거](#)

HP-UX용 빠른 시작

이 절에서는 이전 HP-UX용 빠른 시작 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [HP-UX: 최신 버전으로의 마이그레이션 계획](#)
- [요구사항 검사](#)
- [시스템 준비](#)

- [HP-UX에 IBM WebSphere MQ 서버 설치](#)
- [서버 설치 확인](#)
- [HP-UX 시스템에서 IBM WebSphere MQ 클라이언트 설치](#)
- [클라이언트 설치 확인](#)
- [HP-UX: IBM WebSphere MQ Version 7.5 에 유지보수 레벨 업데이트 적용](#)
- [HP-UX에서 IBM WebSphere MQ 설치 제거](#)

Linux 에 대한 빠른 시작

이 절에서는 Linux 서적의 이전 빠른 시작부터 현재 제품 문서 구조로의 맵핑을 제공합니다.

- [Linux: IBM WebSphere MQ Version 7.0.1 - IBM WebSphere MQ Version 7.5에서 이주 계획](#)
- [요구사항 검사](#)
- [시스템 준비](#)
- [Linux에 IBM WebSphere MQ 서버 설치](#)
- [서버 설치 확인](#)
- [Linux에 WebSphere MQ](#)
- [클라이언트 설치 확인](#)
- [Linux: IBM WebSphere MQ Version 7.5 에 유지보수 레벨 업데이트 적용](#)
- [Linux에서 IBM WebSphere MQ 설치 제거](#)

Solaris용 빠른 시작

이 절에서는 이전 Solaris용 빠른 시작 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [Solaris: 최신 버전으로의 마이그레이션 계획](#)
- [요구사항 검사](#)
- [시스템 준비](#)
- [Solaris에 IBM WebSphere MQ 서버 설치](#)
- [서버 설치 확인](#)
- [Solaris에 IBM WebSphere MQ 클라이언트 설치](#)
- [클라이언트 설치 확인](#)
- [Solaris: IBM WebSphere MQ Version 7.5 에 유지보수 레벨 업데이트 적용](#)
- [Solaris에서 IBM WebSphere MQ 설치 제거](#)

Windows용 빠른 시작

이 절에서는 이전 Windows용 빠른 시작 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [Windows: 최신 버전으로의 마이그레이션 계획](#)
- [요구사항 검사](#)
- [시스템 준비](#)
- [Windows에 IBM WebSphere MQ 서버 설치](#)
- [서버 설치 확인](#)
- [Windows 시스템에 IBM WebSphere MQ 클라이언트 설치](#)
- [클라이언트 설치 확인](#)
- [Windows: IBM WebSphere MQ Version 7.5 에서 유지보수 레벨 업그레이드 적용](#)
- [Solaris에서 IBM WebSphere MQ 설치 제거](#)

애플리케이션 프로그래밍 안내서

이 절에서는 이전 애플리케이션 프로그래밍 안내서에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [애플리케이션 개발 개념](#)
- [IBM WebSphere MQ 애플리케이션 설계](#)
- [큐잉 애플리케이션 작성](#)
- [샘플 WebSphere MQ 프로그램](#)
- [C 언어 예](#)
- [COBOL 예](#)
- [System/390® 어셈블러 언어 예](#) [System/390 어셈블러 언어 예](#)
- [WebSphere MQ 데이터 정의 파일](#)
- [64비트 플랫폼에서의 코딩 표준](#)

애플리케이션 프로그래밍 참조서

이 절에서는 이전 애플리케이션 프로그래밍 참조서 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [MQI에서 사용되는 데이터 유형](#)
- [함수 호출](#)
- [오브젝트의 속성](#)
- [리턴 코드](#)
- [MQI 옵션 유효화를 위한 규칙](#)
- [보고 옵션 및 메시지 플래그](#)
- [보고 옵션 및 메시지 플래그](#)
- [데이터 변환](#)
- [MQRFH2 요소로 지정된 특성](#)
- [코드 페이지 변환](#)

클라이언트

이 절에서는 이전 클라이언트 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [118 페이지의 『IBM WebSphere MQ MQI 클라이언트 개요』](#)
- [121 페이지의 『IBM WebSphere MQ 클라이언트에 대한 플랫폼 지원』](#)
- [IBM WebSphere MQ 클라이언트 설치](#)
- [서버와 클라이언트 간의 연결 구성](#)
- [확장 트랜잭션 클라이언트 구성](#)
- [클라이언트 설치 확인](#)
- [WebSphere MQ MQI 클라이언트 보안 설정](#)
- [106 페이지의 『채널』](#)
- [MQI 채널 정의](#)
- [다른 플랫폼에서 서버 연결 및 클라이언트 연결 정의 작성](#)
- [서버에서 서버 연결 및 클라이언트 연결 정의 작성](#)
- [MQI 채널의 채널 엑시트 프로그램](#)
- [큐 공유 그룹에 클라이언트 연결](#)
- [구성 파일을 사용하여 클라이언트 구성](#)
- [WebSphere MQ 환경 변수 사용](#)
- [클라이언트 애플리케이션에서 메시지 큐 인터페이스\(MQI\) 사용](#)

- [WebSphere MQ MQI 클라이언트의 애플리케이션 빌드](#)
- [WebSphere MQ MQI 클라이언트 환경에서 애플리케이션 실행](#)
- [CICS® 및 Tuxedo 애플리케이션 준비 및 실행](#)
- [Microsoft Transaction Server 애플리케이션 준비 및 실행](#)
- [WebSphere MQ JMS 애플리케이션 준비 및 실행](#)
- [IBM WebSphere MQ MQI 클라이언트에 관련된 문제점 해결](#)
- [저장소의 사전 연결 엑시트를 사용하여 연결 정의 참조](#)

Constants

이 절에서는 이전 Constants 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [WebSphere MQ COPY, 헤더, 포함 및 모듈 파일](#)
- [WebSphere MQ COPY, 헤더, 포함 및 모듈 파일](#)
- [Constants](#)

상호통신

이 절에서는 이전 상호통신 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

소개

- [46 페이지의 『상호통신의 개념』](#)
- [분산 큐잉을 사용하여 애플리케이션 연결](#)
- [네트워크 및 네트워크 계획](#)
- [WebSphere MQ 분산 메시징 기술](#)
- [분산 큐 관리 소개](#)
- [채널 속성](#)
- [구성 정보 예](#)

Windows 및 UNIX용 WebSphere MQ 플랫폼의 분산 큐 관리

- [Windows, UNIX and Linux 플랫폼에서 채널 모니터링 및 제어](#)
- [전송 큐 작성](#)
- [채널 트리거](#)
- [채널 프로그램](#)
- [원격 메시징을 위한 보안](#)
- [분산 큐 관리에서 고려해야 할 기타 내용](#)
- [Windows의 통신 설정](#)
- [구성 예 - Windows용 IBM WebSphere MQ](#)
- [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](#)
- [분산 플랫폼에 대한 메시지 채널 계획 예](#)

추가 상호통신 고려사항

- [메시지 채널에 대한 채널 엑시트 프로그램](#)
- [채널 엑시트 호출 및 데이터 구조](#)
- [큐 이름 해석](#)

메시지 및 코드

이 절에서는 이전 메시지 및 코드 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [진단 메시지: AMQ4000-9999](#)
- [API 완료 및 이유 코드](#)
- [PCF 이유 코드](#)
- [SSL\(Secure Sockets Layer\) 및 TLS\(Transport Layer Security\) 리턴 코드](#)
- [WCF 사용자 정의 채널 예외](#)

마이그레이션

Version 7.5의 마이그레이션 섹션 구조는 Version 7.1의 구조와 동일하게 유지됩니다.

Version 7.5에 대한 마이그레이션 토픽이 추가되었으며 Version 7.1에 대한 마이그레이션 토픽은 제거되었습니다. Version 7.1 및 이전 릴리스로의 마이그레이션을 참조하려면 [특정 마이그레이션 경로에 대한 주제를 찾을 수 있는 위치](#)를 참조하십시오.

모니터링

이 절에서는 이전 모니터링 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [이벤트 모니터링](#)
- [메시지 모니터링](#)
- [계정 및 통계 메시지](#)
- [실시간 모니터링](#)
- [구조 데이터 유형](#)
- [이벤트 데이터에 대한 오브젝트 속성](#)

PCF(Programmable Command Format) 및 Administration Interface

이 절에서는 이전 PCF(Programmable Command Format) 및 Administration Interface 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [PCF\(Programmable Command Format\) 소개](#)
- [WebSphere MQ 관리 인터페이스\(MQAI\) 소개](#)

발행/구독 사용자 안내서

이 절에서는 이전 발행/구독 사용자 안내서에서 새 제품 문서 구조로 맵핑을 제공합니다.

- [WebSphere MQ 발행/구독 메시징 소개](#)
- [분산 발행/구독](#)
- [발행/구독 애플리케이션 작성](#)
- [발행/구독 보안](#)
- [Version 6.0에서 발행/구독 마이그레이션](#)
- [WebSphere 이벤트 브로커 및 WebSphere 메시지 브로커에서 발행/구독 브로커 마이그레이션](#)

큐 관리자 클러스터

이 절에서는 이전 큐 관리자 클러스터 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [59 페이지의 『클러스터 작동 방법』](#)
- [큐 관리자 클러스터 구성](#)
- [IBM WebSphere MQ 클러스터 관리](#)
- [클러스터로\(부터\) 메시지 라우팅](#)

- [워크로드 관리에 클러스터 사용](#)
- [클러스터를 안전하게 유지](#)
- [MQI 및 클러스터에 대한 작업](#)
- [WebSphere MQ 클러스터 명령](#)
- [큐 관리자 클러스터 문제점 해결](#)

스크립트(MQSC) 명령 참조

이 절에서는 이전 스크립트(MQSC) 명령 참조서 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [스크립트\(MQSC\) 명령](#)
- [일반 값 및 특수 의미를 갖는 문자](#)
- [명령 스크립트 빌드](#)
- [111 페이지의 『IBM WebSphere MQ 오브젝트 이름 지정 규칙』](#)
- [구문 다이어그램](#)
- [MQSC 명령](#)

보안

이 절에서는 이전 보안 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [보안](#)

시스템 관리 안내서

이 절에서는 이전 시스템 관리 안내서에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [39 페이지의 『IBM WebSphere MQ 기술 개요』](#)
- [관리 IBM WebSphere MQ](#)
- [로컬 WebSphere MQ 오브젝트 관리](#)
- [IBM WebSphere MQ Explorer를 사용한 관리](#)
- [WebSphere MQ 작업 표시줄 애플리케이션 사용\(Windows 전용\)](#)
- [WebSphere MQ 제어 명령](#)

구성 및 관리

- [IBM WebSphere MQ 및 큐 관리자 구성 정보 변경](#)
- [파일 시스템 지원 계획](#)
- [Windows, UNIX and Linux 시스템에서 보안 설정](#)
- [트랜잭션 지원](#)
- [WebSphere MQ 데드-레터 큐 핸들러로 미배달 메시지 핸들링](#)
- [사용가능성, 복구 및 재시작](#)
- [문제점 해결 및 지원](#)
- [WebSphere MQ 및 UNIX System V IPC 자원](#)
- [WebSphere MQ 및 UNIX 프로세스 우선순위](#)
- [사용자 엑시트, API 엑시트 및 WebSphere MQ 설치 가능 서비스](#)

.NET 사용

이 절에서는 이전 .NET 사용 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [.NET 사용](#)
- [WebSphere MQ .NET 프로그램 작성 및 배치](#)

- [WebSphere MQ .NET 클래스 및 인터페이스](#)
- [Microsoft Windows Communication Foundation \(WCF\) 에 대한 IBM WebSphere MQ 사용자 정의 채널](#)

C++ 사용

이 절에서는 이전 C++ 사용 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [C++ 사용](#)
- [WebSphere MQ C++ 클래스](#)

Java 사용

이 절에서는 이전 Java 사용 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [JMS에 대해 WebSphere MQ 클래스 또는 WebSphere MQ 클래스를 사용해야 하나?](#)
- [Java용 WebSphere MQ 클래스 사용](#)
- [JMS용 WebSphere MQ 클래스](#)
- [JMS용 WebSphere MQ 클래스 사용](#)
- [Java용 WebSphere MQ 클래스](#)

웹 서비스

이 절에서는 이전 웹 서비스에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [SOAP용 WebSphere MQ 전송](#)
- [HTTP용 WebSphere MQ 브릿지](#)

컴포넌트 오브젝트 모델 인터페이스 사용

이 절에서는 이전의 컴포넌트 오브젝트 모델 인터페이스 사용 서적에서 새 제품 문서 구조로의 맵핑을 제공합니다.

- [컴포넌트 오브젝트 모델 인터페이스\(WebSphere MQ Automation Classes for ActiveX\) 사용](#)

IBM WebSphere MQ Version 7.5, IBM i 및 z/OS

IBM WebSphere MQ Version 7.5는 IBM i 및 z/OS에 대해 사용 불가능합니다.

이러한 플랫폼은 제품의 최신 버전에서 사용 가능합니다.

IBM i 및 z/OS용 IBM MQ 의 최신 버전에 대한 정보는 [IBM MQ 웹 사이트](#)를 참조하십시오.

주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서 이 책에 포함된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 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.

07326
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.

Intellectual Property Licensing
2-31 Roppongi 3-chome, Minato-Ku
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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
J46A/G4
국제금융로 10, 3IFC
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 추정을 통해 측정되었을 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 애플리케이션 프로그래밍 인터페이스(API)에 부합하는 애플리케이션을 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 진

술하지 않습니다. 본 샘플 프로그램은 일체의 보증 없이 "현상태대로" 제공됩니다. IBM은 귀하의 샘플 프로그램 사용과 관련되는 손해에 대해 책임을 지지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 그 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다. © (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp. 샘플 프로그램에서 파생되었습니다. 샘플 프로그램. © Copyright IBM Corp. 1993, 2024. All rights reserved.

이 정보를 소프트 카피로 보는 경우 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

상표

IBM, IBM 로고, ibm.com®, Passport Advantage®, WebSphere, MQSeries® 및 z/OS 는 International Business Machines Corp.의 상표 또는 등록상표입니다. 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보" (<https://www.ibm.com/legal/copytrade.shtml>) 에 있습니다.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

Java 및 모든 Java 기반 상표와 로고는 Oracle 및/또는 그 계열사의 상표 또는 등록상표입니다.

Adobe, Adobe 로고, PostScript 및 PostScript 로고는 미국 또는 기타 국가에서 사용되는 Adobe Systems Incorporated의 등록상표 또는 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

여기에 포함된 Oracle Outside In Technology는 용도가 제한된 라이선스가 적용되며 본 응용프로그램과 연계해 서만 사용할 수 있습니다.

Intel은 미국 또는 기타 국가에서 사용되는 Intel Corporation 또는 그 계열사의 상표 또는 등록상표입니다.

등록 상표 Linux 는 전세계적으로 표시된 Linus Torvalds의 독점적 라이선스 사용자인 Linux Foundation의 재라이선스에 따라 사용됩니다.

IBM WebSphere MQ 기술 개요

IBM WebSphere MQ를 사용하여 애플리케이션을 연결하고 조직에서 정보 분배를 관리하십시오.

IBM WebSphere MQ에서는 프로그램이 일관된 API(Application Programming Interface)를 사용하여 서로 다른 컴포넌트(프로세서, 운영 체제, 서브시스템 및 통신 프로토콜)가 존재하는 네트워크에서 서로 다른 항목과 통신할 수 있습니다. 이 인터페이스를 사용하여 설계 및 기록된 애플리케이션은 메시지 큐잉 애플리케이션이라고 합니다.

다음 하위 주제를 사용하여 메시지 큐잉 및 IBM WebSphere MQ에서 제공되는 다른 기능에 대해 확인하십시오.

관련 개념

[6 페이지의 『IBM WebSphere MQ 소개』](#)

IBM WebSphere MQ를 사용하여 서로 다른 시간 및 다수의 다양한 컴퓨팅 환경에서 애플리케이션이 통신할 수 있게 할 수 있습니다.

관련 태스크

[WebSphere MQ 아키텍처 설계](#)

[WebSphere MQ Managed File Transfer](#)

관련 참조

[41 페이지의 『메시지 큐잉의 기본 기능 및 이점』](#)

이 정보는 메시지 큐잉의 일부 기능 및 이점을 강조합니다. 이 정보에서는 메시지 큐잉의 보안 및 데이터 무결성과 같은 기능을 설명합니다.

메시지 큐잉 소개

WebSphere MQ 제품을 사용하면 프로그램이 일관된 애플리케이션 프로그래밍 인터페이스(API)를 사용하여 다른 컴포넌트(프로세서, 운영 체제, 서브시스템 및 통신 프로토콜)의 네트워크에서 서로 통신할 수 있습니다.

이 인터페이스를 사용하여 설계되고 작성된 애플리케이션은 메시징 및 큐잉 스타일을 사용하기 때문에 메시지 큐잉 애플리케이션이라고 합니다.

메시징	프로그램은 서로 직접 호출하기 보다는 메시지로 데이터를 서로 송신하여 통신합니다.
큐잉	메시지는 스토리지의 큐에 배치되어 프로그램이 다른 속도와 시간에, 여러 위치에서, 서로 간에 논리적 연결 없이 독립적으로 실행될 수 있도록 합니다.

메시지 큐잉이 다년간 데이터 처리에 사용되어 왔습니다. 메시지 큐잉은 전자 메일에서 가장 많이 사용됩니다. 큐잉 없이 전자 메시지를 장거리 송신하는 경우 라우트의 모든 노드가 메시지 전달에 대해 사용 가능해야 하고 주소가 로그온되어야 하며 사용자가 이 주소에 메시지 송신을 시도한다는 사실에 대해 인식하고 있어야 합니다. 큐잉 시스템에서 시스템이 메시지를 전달할 준비가 될 때까지 메시지는 중간 노드에 저장됩니다. 최종 목적지에서 주소가 메시지를 읽을 준비가 될 때까지 메시지는 전자 편지함에 저장됩니다.

그렇다 하더라도 다수의 복잡한 비즈니스 트랜잭션이 오늘날 큐잉 없이 처리됩니다. 대형 네트워크에서 시스템은 사용 준비 상태인 수천의 연결을 유지보수하고 있을 수 있습니다. 시스템의 한 부분에 문제가 생기는 경우 시스템의 많은 부분이 사용 불가능하게 됩니다.

메시지 큐잉을 프로그램용 전자 메일로 생각할 수 있습니다. 메시지 큐잉 환경에서 애플리케이션 스위트를 구성하는 각 프로그램은 특정 요청에 대한 응답으로 명확하고 독립적인 기능을 수행합니다. 다른 프로그램과 통신하려면 프로그램이 사전정의된 큐에 메시지를 넣어야 합니다. 다른 프로그램은 큐에서 메시지를 검색하고 메시지에 포함된 요청 및 정보를 처리합니다. 따라서 메시지 큐잉은 프로그램 대 프로그램 통신의 스타일입니다.

큐잉은 애플리케이션이 메시지를 처리할 준비가 될 때까지 메시지가 보유되는 메커니즘입니다. 큐잉을 통해 다음을 수행할 수 있습니다.

- 통신 코드를 작성할 필요 없이 프로그램(각각 다른 환경에서 실행될 수 있음) 간에 통신합니다.
- 프로그램이 메시지를 처리하는 순서를 선택하십시오.
- 메시지 수가 임계값을 초과할 때 둘 이상의 프로그램이 큐를 서비스하도록 구성하여 시스템에 로드를 밸런싱합니다.
- 1차 시스템이 사용 가능하지 않은 경우 대체 시스템이 큐를 서비스하도록 구성하여 애플리케이션의 사용가능성을 증가시키십시오.

메시지 큐 개념

간단하게 큐라고 하는 메시지 큐는 메시지를 송신할 수 있는 이름 지정된 목적지입니다. 큐를 서비스하는 프로그램이 메시지를 검색할 때까지 메시지는 큐에 누적됩니다.

큐 관리자에서 큐가 상주하고 관리됩니다(43 페이지의 『메시지 큐잉 용어』 참조). 큐의 물리적 특성은 큐 관리자가 실행되는 운영 체제에 따라 다릅니다. 큐는 컴퓨터 메모리의 휘발성 버퍼이거나 영구 스토리지 디바이스(예: 디스크)에서 데이터 세트일 수 있습니다. 큐의 실제 관리는 큐 관리자가 담당하며 참여하는 애플리케이션 프로그램에 대해 분명하지 않게 작성됩니다.

프로그램은 큐 관리자의 외부 서비스를 통해서만 큐에 액세스할 수 있습니다. 프로그램은 큐를 열고, 큐로(부터) 메시지를 넣고, 메시지를 가져오고 큐를 닫을 수 있습니다. 또한 큐의 속성을 설정 및 조회할 수 있습니다.

메시지 큐잉의 다른 스타일

포인트-투-포인트

큐에는 메시지가 한 개만 배치되며 하나의 애플리케이션이 해당 메시지를 수신합니다.

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

발행/구독

발행 애플리케이션이 발행하는 각 메시지의 사본은 모든 관련 애플리케이션으로 전달됩니다. 관련 애플리케이션이 하나 또는 여러 개일 수 있으며, 전혀 없을 수도 있습니다. 발행/구독에서 관심 애플리케이션은 구독자로 알려져 있으며 메시지는 구독으로 식별되는 큐에 삽입됩니다.

발행/구독 메시지를 사용하면 정보의 제공자를 해당 정보의 이용자와 분리할 수 있습니다. 송신 애플리케이션과 수신 애플리케이션은 송수신할 정보를 위해 서로에 대해 알지 않아도 됩니다. 발행/구독 메시징에 대한 자세한 정보는 [WebSphere MQ 발행/구독 메시징 소개](#)를 참조하십시오.

애플리케이션 설계자 및 개발자에 대한 메시지 큐잉의 이점

WebSphere MQ를 통해 애플리케이션 프로그램은 메시지 큐잉을 사용하여 메시지에 의한 처리에 참여할 수 있습니다. 애플리케이션 프로그램은 적절한 메시지 큐잉 소프트웨어 제품을 사용하여 여러 플랫폼 사이에서 통신할 수 있습니다. For example, HP-UX and z/OS applications can communicate through WebSphere MQ for HP-UX and WebSphere MQ for z/OS. 애플리케이션은 근본적인 통신의 메커니즘으로부터 보호됩니다. 메시지 큐잉의 그 밖의 이점은 다음과 같습니다.

- 여러 애플리케이션에서 공유할 수 있는 소형 프로그램을 사용하여 애플리케이션을 설계할 수 있습니다.
- 이러한 빌딩 블록을 다시 사용하여 새 애플리케이션을 빠르게 빌드할 수 있습니다.
- 메시지 큐잉 기술을 사용하도록 작성된 애플리케이션은 큐 관리자가 작업하는 도중에 발생한 변경사항에 영향을 받지 않습니다.
- 통신 프로토콜을 사용할 필요가 없습니다. 큐 관리자는 사용자를 위해 통신의 모든 측면을 다룹니다.
- 메시지가 프로그램에 송신될 때 메시지를 수신하는 프로그램이 실행 중일 필요는 없습니다. 메시지는 큐에 보유됩니다.

메시지 큐잉을 사용하지 않는 애플리케이션에 비해 프로그래밍 기술에 대한 요구가 낮아지고, 필요한 개발자의 수가 적어지고 개발이 빨라지기 때문에 설계자는 애플리케이션 비용을 줄일 수 있습니다.

WebSphere MQ는 애플리케이션이 실행되는 위치마다 메시지 큐 인터페이스(또는 MQI)라는 공용 애플리케이션 프로그래밍 인터페이스를 구현합니다. 이를 통해 하나의 플랫폼에서 다른 플랫폼으로 애플리케이션 프로그램을 쉽게 포팅시킬 수 있습니다.

MQI에 대한 자세한 내용은 [MQI\(Message Queue Interface\) 개요](#)를 참조하십시오.

메시지 큐잉의 기본 기능 및 이점

이 정보는 메시지 큐잉의 일부 기능 및 이점을 강조합니다. 이 정보에서는 메시지 큐잉의 보안 및 데이터 무결성과 같은 기능을 설명합니다.

메시지 큐잉 기술을 사용하는 애플리케이션의 기본 기능은 다음과 같습니다.

- 프로그램 간의 직접 접속이 없습니다.
- 프로그램 간의 통신은 시간과는 관련이 없을 수 있습니다.
- 작업은 소형 및 자급(self-contained) 프로그램에서 수행할 수 있습니다.
- 통신은 이벤트에 의해 구동될 수 있습니다.
- 애플리케이션은 메시지에 우선순위를 지정할 수 있습니다.
- 보안.
- 데이터 무결성
- 복구 지원

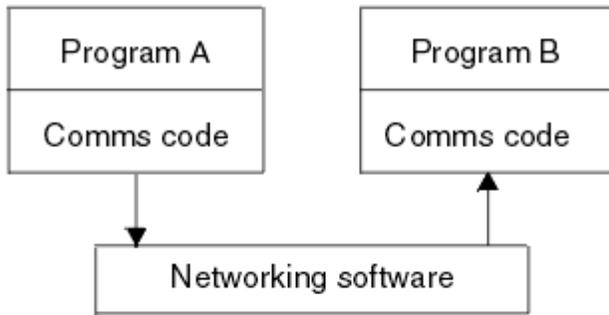
프로그램 간에 직접 접속이 없음

메시지 큐잉은 간접적 프로그램 대 프로그램 통신용 기술입니다. 프로그램이 서로 통신하는 모든 애플리케이션 내에서 이 메시지 큐잉을 사용할 수 있습니다. 메시지를 큐 관리자가 소유한 큐에 넣는 하나의 프로그램과 큐에서 메시지를 가져오는 다른 하나의 프로그램에 의해서 통신이 발생합니다.

프로그램은 다른 프로그램이 큐에 넣는 메시지를 가져올 수 있습니다. 다른 프로그램은 수신 프로그램과 동일한 큐 관리자 또는 다른 큐 관리자에 연결될 수 있습니다. 이러한 다른 큐 관리자는 다른 시스템, 다른 컴퓨터 시스템 상에 있거나 다른 비즈니스 또는 엔터프라이즈 내에 있을 수도 있습니다.

메시지 큐를 사용하여 통신하는 프로그램 간에는 물리적 접속이 없습니다. 프로그램이 큐 관리자가 소유하는 큐에 메시지를 송신하고 다른 프로그램은 해당 큐에서 메시지를 검색합니다(42 페이지의 [그림 4](#) 참조).

Traditional communication between programs



Communication by message queuing

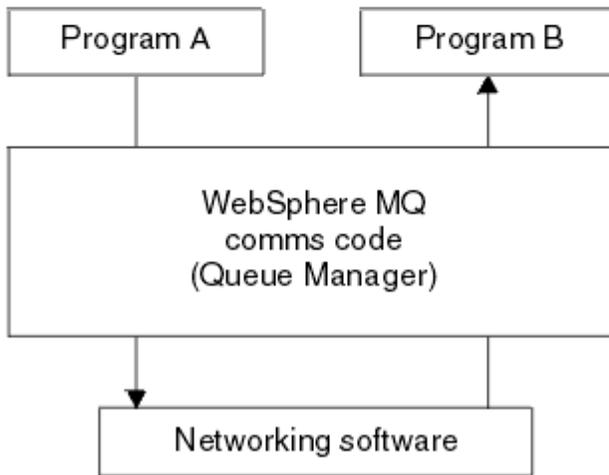


그림 4. 일반 통신과 메시지 큐잉 비교

전자 메일의 경우와 마찬가지로 트랜잭션의 일부인 개별 메시지는 저장 및 전달 네트워크를 통해 이동합니다. 기초 노드 간의 링크가 실패하면 링크가 복원되거나 연산자 또는 프로그램이 메시지를 다시 전송할 때까지 메시지가 보관됩니다.

메시지가 큐에서 큐로 이동하는 메커니즘은 프로그램으로부터 숨겨집니다. 따라서 프로그램이 좀 더 단순해집니다.

시간 독립 통신

다른 프로그램에 작업 수행을 요청하는 프로그램은 요청에 대한 응답을 기다릴 필요가 없습니다. 다른 작업을 수행하고 응답이 도착할 때 또는 나중에 응답을 처리할 수 있습니다. 메시징 애플리케이션을 작성할 때 프로그램이 언제 메시지를 송신하는지 또는 대상이 메시지를 받을 수 있는 상태인지에 대해 알거나 염려할 필요가 없습니다. 메시지는 손실되지 않으며 대상이 메시지를 처리할 준비가 될 때까지 큐 관리자가 보유합니다. 메시지는 프로그램이 제거할 때까지 큐에 남습니다. 이는 송신 및 수신 애플리케이션 프로그램이 연결 해제되었으며 송신자가 수신자의 메시지의 수신 확인을 기다릴 필요 없이 처리를 계속할 수 있음을 의미합니다. 대상 애플리케이션은 메시지가 송신될 때 실행 중인 필요조차 없습니다. 대상 애플리케이션은 시작된 후에 메시지를 검색할 수 있습니다.

소형 프로그램

메시지 큐잉을 사용하면 소형, 자급 프로그램 사용의 이점을 활용할 수 있습니다. 작업의 모든 파트를 순차적으로 수행하는 단일, 대형 프로그램 대신에 작업을 여러 개의 작고 독립적인 프로그램으로 분산시킬 수 있습니다. 요청 프로그램은 개별 프로그램 각각에 해당 기능을 수행하라고 요청하는 메시지를 송신합니다. 각 프로그램이 완료되면 그 결과를 하나 이상의 메시지로 되돌려 보냅니다.

메시지에 의한 처리

메시지가 큐에 도착할 때 이 메시지는 트리거를 사용하여 애플리케이션을 자동으로 시작할 수 있습니다. 필요에 따라 하나의 메시지 또는 다수의 메시지가 처리될 때 애플리케이션을 중지할 수 있습니다.

이벤트 중심 처리

프로그램은 큐의 상태에 따라 제어할 수 있습니다. 예를 들어, 프로그램이 메시지가 큐에 도착하는 즉시 시작 되도록 구성하거나 큐에 특정 우선순위보다 상위에 있는 10개의 메시지 또는 임의의 우선순위를 가지는 10개의 메시지가 있을 때까지 프로그램이 시작되지 않도록 지정할 수 있습니다.

메시지 우선순위

프로그램은 메시지를 큐에 넣을 때 메시지에 우선순위를 지정할 수 있습니다. 이는 큐에서 새 메시지가 추가 되는 위치를 판별합니다.

프로그램은 메시지가 큐에 있는 순서 또는 특정 메시지를 가져오는 순서대로 큐에서 메시지를 가져올 수 있습니다. (프로그램이 이전에 송신한 요청에 대한 응답을 검색하고 있으면 프로그램은 특정 메시지를 가져오려고 할 수 있습니다.)

보안

권한 부여 확인은 WebSphere MQ 관리자가 설정하고 유지보수하는 테이블을 사용하여 각 자원에 대해 수행할 수 있습니다.

- Use Security Server (formerly known as RACF®) or other external security managers on WebSphere MQ for z/OS.
- On WebSphere MQ on UNIX systems, Linux systems, 윈도우 systems, and IBM i, a security manager called the object authority manager (OAM) is provided as an installable service. 기본적으로, OAM은 활성 상태입니다.

데이터 무결성

데이터 무결성은 작업 단위로 제공됩니다. 작업 단위의 시작 및 종료 동기화가 MQGET 또는 MQPUT 각각에 대한 옵션으로 완전하게 지원되어 작업 단위의 결과가 커밋되거나 롤백되도록 허용합니다. 동기점 지원은 애플리케이션에 대해 선택된 동기점 조정의 형식에 따라 WebSphere MQ에 대해서 내부적으로 또는 외부적으로 작동합니다.

복구 지원

가능한 복구의 경우 모든 지속적 WebSphere MQ 업데이트가 로그됩니다. 복구가 필요한 경우에 모든 지속 메시지가 복원되고, 모든 인플라이트 트랜잭션이 롤백되며 모든 동기점 커밋 및 백아웃이 제어 중인 동기점 관리자에 의해 일반적인 방식으로 핸들링됩니다. 지속 메시지에 대한 자세한 정보는 [메시지 지속성](#) 을 참조하십시오.

참고: WebSphere MQ 클라이언트 및 서버를 고려할 때 새 플랫폼에서의 추가 WebSphere MQ MQI 클라이언트 지원을 위해 서버 애플리케이션을 변경할 필요가 없습니다. 이와 유사하게, WebSphere MQ MQI 클라이언트는 변경 없이도 추가 유형의 서버와 함께 작동할 수 있습니다.

메시지 큐잉 용어

이 정보는 메시지 큐잉에 사용되는 일부 용어에 대한 자세한 설명을 제공합니다.

다음에 포함됩니다.

- [메시지](#)
- [메시지 디스크립터](#)
- [큐](#)
- [큐 관리자](#)
- [채널](#)
- [메시지 채널 에이전트](#)
- [CLUSTER](#)
- [WebSphere MQ MQI 클라이언트\(WebSphere MQ 클라이언트\)](#)
- [포인트-투-포인트](#)
- [발행/구독](#)
- [토픽](#)
- [구독](#)

메시지

메시지 큐잉에서 메시지는 하나의 프로그램에서 송신한 데이터의 컬렉션이며 다른 프로그램에도 사용됩니다. [IBM WebSphere MQ 메시지를 참조하십시오](#). 메시지 유형에 대한 정보는 [메시지 유형](#)을 참조하십시오.

메시지 디스크립터

IBM WebSphere MQ 메시지는 제어 정보 및 애플리케이션 데이터로 구성됩니다.

제어 정보는 메시지 디스크립터 구조(MQMD)에 정의되고 다음과 같은 사항을 포함합니다.

- 메시지의 유형
- 메시지의 ID
- 메시지 전달의 우선순위

애플리케이션 데이터의 구조 및 콘텐츠는 IBM WebSphere MQ가 아닌 참여 프로그램에 의해 판별됩니다.

큐

메시지를 전송할 수 있는 이름 지정된 대상입니다. 큐를 서비스하는 프로그램이 메시지를 검색할 때까지 메시지는 큐에 누적됩니다.

큐 관리자

큐 관리자는 애플리케이션에 큐잉 서비스를 제공하는 시스템 프로그램입니다.

큐 관리자는 API(Application Programming Interface)를 제공하여 프로그램이 큐로(부터) 메시지를 넣고 가져올 수 있도록 합니다. 큐 관리자는 관리자가 새 큐를 작성하고, 기존 큐의 특성을 대체하고 큐 관리자의 조작을 제어할 수 있도록 추가 기능을 제공합니다.

IBM WebSphere MQ 메시지 큐잉 서비스를 시스템에서 사용하려면 큐 관리자가 실행 중이어야 합니다. 단일 시스템에는 둘 이상의 실행 중인 큐 관리자가 있을 수 있습니다(예를 들어, 라이브 시스템에서 테스트 시스템을 분리하려는 경우). 애플리케이션에 대해 각 큐 관리자는 연결 핸들(Hconn)로 식별됩니다.

여러 다른 애플리케이션은 큐 관리자의 서비스를 동시에 사용할 수 있으며 이러한 애플리케이션은 전부 비관련될 수 있습니다. 프로그램이 큐 관리자의 서비스를 이용하게 하려면 해당 큐 관리자에 대한 연결을 설정해야 합니다.

애플리케이션이 다른 큐 관리자에 연결된 애플리케이션에 메시지를 송신하게 하려면 큐 관리자는 서로 통신할 수 있어야 합니다. IBM WebSphere MQ에서는 저장 후 전달 프로토콜을 구현하여 이러한 애플리케이션 간의 안전한 메시지 전달을 보장합니다.

채널

채널은 한 큐 관리자에서 다른 큐 관리자로 통신 경로를 제공하는 오브젝트입니다. 채널은 분산 큐잉에서 한 큐 관리자에서 다른 큐 관리자로 메시지를 이동시키는 데 사용되며 기본 통신 프로토콜로부터 애플리케이션을 보호합니다. 큐 관리자들은 동일하거나 서로 다른 플랫폼에 존재할 수 있습니다.

메시지 채널 에이전트

메시지 채널 에이전트는 하나의 큐 관리자에서 다른 큐 관리자로 메시지를 이동시킵니다.

보고 메시지를 처리할 때 이 메시지를 참조하고 애플리케이션을 설계할 때에도 고려할 필요가 있습니다. 자세한 정보는 [고유의 메시지 채널 에이전트 작성](#)을 참조하십시오.

군집

클러스터는 논리적으로 연관된 큐 관리자의 네트워크입니다. IBM WebSphere MQ Version 7.0 및 이후 버전에는 있는 큐 관리자는 클러스터링을 사용할 수 있습니다.

클러스터링 없이 분산 큐잉을 사용하는 IBM WebSphere MQ 네트워크에서 모든 큐 관리자는 독립적입니다. 한 큐 관리자가 다른 큐 관리자에게 메시지를 송신해야 하는 경우 리모트 큐 관리자에 대한 전송 큐 및 채널이 정의되어 있어야 합니다.

클러스터를 사용하는 두 가지 다른 이유는 시스템 관리를 줄이는 것과 사용가능성 및 워크로드 밸런싱을 높이기 위해서입니다.

가장 작은 클러스터라도 설정이 끝나고 나면 단순화된 시스템 관리에 도움이 됩니다. 클러스터의 일부인 큐 관리자는 보다 소수의 정의만을 필요로 하기 때문에 정의에서 오류가 발생하는 위험을 줄일 수 있습니다.

클러스터링에 대한 자세한 정보는 59 페이지의 『클러스터 작동 방법』의 내용을 참조하십시오.

IBM WebSphere MQ MQI client

IBM WebSphere MQ MQI 클라이언트는 IBM WebSphere MQ의 독립적으로 설치 가능한 컴포넌트입니다. MQI 클라이언트를 통해 통신 프로토콜을 사용하는 IBM WebSphere MQ 애플리케이션을 실행하고 다른 플랫폼에 있는 하나 이상의 MQI(Message Queue Interface)와 상호 작용하며 해당 큐 관리자에 접속할 수 있습니다.

IBM WebSphere MQ MQI client 컴포넌트의 설치 및 사용 방법에 대한 자세한 내용은 [IBM WebSphere MQ MQI client 설치 및 서버 및 클라이언트 사이의 연결 구성](#)을 참조하십시오.

포인트-투-포인트 메시징

지점간 메시징에서 각 메시지는 단일 생성 애플리케이션에서 단일 이용 애플리케이션으로 이동합니다. 메시지는 메시지를 큐에 넣는 생성 애플리케이션을 통해 전송되며 이용 애플리케이션은 큐에서 메시지를 받습니다.

발행/구독 메시징

발행/구독 메시징에서 발행 애플리케이션이 발행한 각 메시지의 사본은 모든 관심 애플리케이션으로 전달됩니다. 관심 애플리케이션의 개수는 하나 또는 다수이거나 전혀 없을 수도 있습니다. 발행/구독에서 관심 애플리케이션은 구독자로 알려져 있으며 메시지는 구독으로 식별되는 큐에 삽입됩니다. 발행/구독에 대한 자세한 정보는 [IBM WebSphere MQ 발행/구독 메시징 소개](#)를 참조하십시오.

주제

토픽은 발행/구독 메시징에서 발행된 정보의 제목을 설명하는 문자열입니다.

토픽은 발행/구독 시스템에서 메시지를 성공적으로 전달하기 위한 핵심 사항입니다. 각 메시지에 특정 목적지 주소를 포함하는 대신 발행자가 각 메시지에 대한 토픽을 지정합니다. 큐 관리자가 해당 토픽을 구독하는 구독자 목록과 토픽을 일치시킨 다음 각 해당 구독자에게 메시지를 전달합니다.

구독

발행/구독 애플리케이션은 특정 주제에 대한 메시지에 관심을 등록할 수 있습니다. 이를 수행하는 애플리케이션은 구독자로 알려져 있으며 기간 구독은 일치하는 메시지가 처리를 위해 큐에 대기되는 방식을 정의합니다.

구독에는 구독자의 ID 및 발행이 배치될 목적지 큐의 ID에 관한 정보가 들어있습니다. 또한 발행이 목적지 큐에 배치되는 방법에 대한 정보도 들어있습니다.

메시지 및 큐

메시지 및 큐는 메시지 큐잉 시스템의 기본 컴포넌트입니다.

메시지 개념

메시지는 바이트 문자열이며 문자열을 사용하는 애플리케이션에 의미가 있습니다. 메시지는 하나의 애플리케이션 프로그램에서 다른 애플리케이션 프로그램으로 (또는 동일한 애플리케이션의 다른 파트 사이에서) 정보를 전송하는 데 사용됩니다. 애플리케이션은 동일한 플랫폼 또는 다른 플랫폼에서 실행될 수 있습니다.

IBM WebSphere MQ 메시지에는 다음의 두 부분이 있습니다.

- 애플리케이션 데이터.애플리케이션 데이터의 콘텐츠 및 구조는 이를 사용하는 애플리케이션 프로그램에 의해 정의됩니다.
- 메시지 디스크립터.메시지 디스크립터는 메시지를 식별하고 메시지 유형 및 송신 애플리케이션에 의해 메시지에 지정된 우선순위와 같은 추가 제어 정보를 포함합니다.

메시지 디스크립터의 형식은 IBM WebSphere MQ에 의해 정의됩니다. 메시지 디스크립터에 대한 자세한 설명은 [MQMD - 메시지 디스크립터](#)를 참조하십시오.

메시지 길이

메시지 길이를 100MB의 최대 길이로 증가시킬 수 있기는 하지만 (여기에서 1MB는 1 048 576바이트임) 기본 최대 메시지 길이는 4MB입니다. 실제로 메시지 길이는 다음으로 제한될 수 있습니다.

- 수신 큐에 대해 정의된 최대 메시지 길이
- 큐 관리자에 대해 정의된 최대 메시지 길이
- 큐에 의해 정의된 최대 메시지 길이
- 수신 또는 송신 애플리케이션에 의해 정의된 최대 메시지 길이
- 메시지에 사용 가능한 스토리지 용량

애플리케이션이 요구하는 모든 정보를 송신하는 데 여러 메시지가 필요할 수 있습니다.

애플리케이션이 메시지를 송신하고 수신하는 방법

애플리케이션 프로그램은 **MQI 호출**을 사용하여 메시지를 송신하고 수신합니다.

예를 들어, 큐에 메시지를 배치하기 위해 애플리케이션은 다음을 수행합니다.

1. MQI MQOPEN 호출을 발행하여 필요한 큐를 엽니다.
2. MQI MQPUT 호출을 발행하여 메시지를 큐에 배치합니다.

다른 애플리케이션이 MQI MQGET 호출을 발행하여 동일한 큐에서 메시지를 검색할 수 있습니다.

MQI 호출에 대한 자세한 정보는 [MQI 호출](#)을 참조하십시오.

큐의 개념

큐는 메시지를 저장하는 데 사용되는 데이터 구조입니다.

큐 관리자가 각 큐를 소유합니다. 큐 관리자는 소유한 큐를 유지보수하고 적절한 큐로 수신한 모든 메시지 저장을 담당합니다. 애플리케이션 프로그램 또는 큐 관리자의 일반적인 조작으로 큐에 메시지를 배치할 수 있습니다.

사전정의된 큐 및 동적 큐

큐는 작성되는 방식에 따라 특성화할 수 있습니다.

- **사전정의된 큐**는 적절한 MQSC 또는 PDF 명령을 사용하는 관리자에 의해 작성됩니다. 사전정의된 큐는 영구적이며 이를 사용하는 애플리케이션과는 독립적으로 존재하고 IBM WebSphere MQ를 다시 시작해도 없어지지 않습니다.
- **동적 큐**는 애플리케이션이 모델 큐의 이름을 지정하는 MQOPEN 요청을 발행할 때 작성됩니다. 작성된 큐는 모델 큐라고 하는 템플릿 큐 정의를 기본으로 합니다. MQSC 명령 DEFINE QMODEL을 사용하여 모델 큐를 작성할 수 있습니다. 모델 큐의 속성(예: 모델 큐에 저장될 수 있는 최대 메시지 수)은 모델 큐에서 작성되는 동적 큐로 상속됩니다.

모델 큐에는 동적 큐가 영구적 또는 임시적인지 여부를 지정하는 속성이 있습니다. 영구적 큐는 애플리케이션과 큐 관리자가 재시작해도 남아 있지만 임시적 큐는 재시작 시에 손실됩니다.

큐에서 메시지 검색

적합하게 권한 부여된 애플리케이션은 다음 검색 알고리즘에 따라 큐에서 메시지를 검색할 수 있습니다.

- FIFO(First In, First Out)
- 메시지 디스크립터에 정의된 메시지 우선순위입니다. 동일한 우선순위를 가진 메시지는 FIFO(First In, First Out) 기준으로 검색됩니다.
- 특정 메시지에 대한 프로그램 요청

애플리케이션에서의 MQGET 요청은 사용되는 메소드를 판별합니다.

상호통신의 개념

WebSphere MQ에서, 상호통신은 하나의 큐 관리자에서 다른 큐 관리자로 메시지를 전송한다는 의미입니다. 수신 큐 관리자는 동일 시스템 또는 다른 쪽 세계 또는 인접한 곳의 다른 시스템에 있을 수 있습니다. 로컬 큐 관리자와 동일한 플랫폼에서 실행 중이거나 WebSphere MQ에서 지원되는 플랫폼에서 실행 중일 수 있습니다. 이를 분

산 환경이라고 합니다. WebSphere MQ는 분산 큐 관리(DQM)를 사용하는 것처럼 분산 환경에서 통신을 핸들링 합니다.

로컬 큐 관리자는 소스 큐 관리자라고 하기도 하며 리모트 큐 관리자는 대상 큐 관리자 또는 파트너 큐 관리자라고도 합니다.

분산 큐잉 작동 방법

분산 큐잉을 사용하면 하나의 큐 관리자에서 다른 큐 관리자로 메시지를 전송할 수 있습니다. 수신 큐 관리자는 동일 시스템 또는 원격 시스템에 있을 수 있습니다. 큐 관리자, 큐, 채널 및 연관된 정의는 클러스터링(큐 관리자와 논리적으로 연관된 네트워크)과 함께 파악됩니다.

47 페이지의 그림 5은 분산 큐잉의 컴포넌트 개요를 나타냅니다.

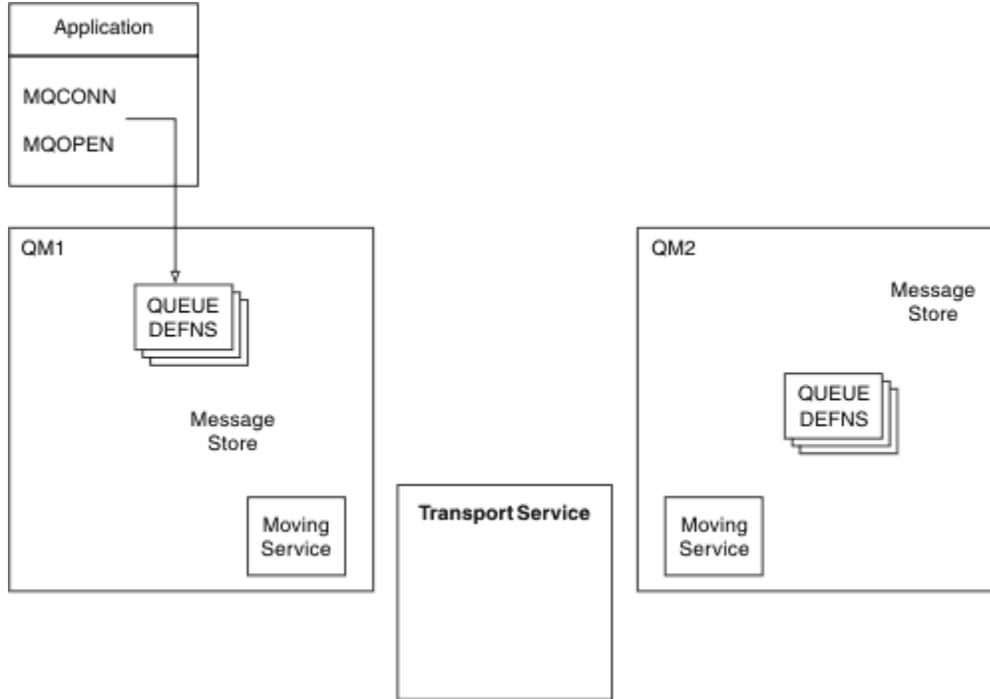


그림 5. 분산 큐잉의 컴포넌트 개요

1. 애플리케이션은 MQCONN 호출을 사용하여 큐 관리자에 연결합니다.
2. 애플리케이션은 MQOPEN 호출을 사용하여 큐를 열어 메시지를 큐에 넣을 수 있습니다.
3. 큐 관리자는 큐에서 허용되는 최대 메시지 수와 같은 정보를 지정하는 각 큐에 대한 정의가 있습니다. 리모트 큐 관리자에 있는 큐에 대한 정의도 있을 수 있습니다.
4. 메시지가 원격 시스템의 큐를 대상으로 하는 경우, 리모트 큐 관리자에 메시지를 전달할 준비가 될 때까지 로컬 큐 관리자는 메시지를 메시지 저장소에 보유합니다. 애플리케이션에 영향을 주지 않습니다.
5. 각 큐 관리자는 이동 서비스 컴포넌트라는 통신 소프트웨어를 포함합니다. 이를 통해 큐 관리자는 다른 큐 관리자와 통신할 수 있습니다.
6. 전송 서비스는 큐 관리자와 무관하며 다음 중 하나일 수 있습니다(플랫폼에 따라).
 - SNA APPC(Systems Network Architecture Advanced Program-to Program Communication)
 - TCP/IP(Transmission Control Protocol/Internet Protocol)
 - NetBIOS(Network Basic Input/Output System)
 - SPX(Sequenced Packet Exchange)

분산 큐잉의 컴포넌트

WebSphere MQ 응용프로그램은 메시지를 로컬 큐(응용프로그램이 연결된 큐 관리자의 큐)에 넣을 수 있습니다.

큐 관리자에는 각 큐에 대한 정의가 있습니다. 다른 큐 관리자에서 소유한 큐에 대한 정의도 있을 수 있습니다. 이를 리모트 큐 정의라고 합니다. WebSphere MQ 애플리케이션은 이 리모트 큐를 대상으로 하는 메시지를 넣을 수도 있습니다.

메시지가 리모트 큐 관리자를 대상으로 하면 리모트 큐 관리자에 메시지를 보낼 준비가 될 때까지 로컬 큐 관리자는 메시지를 전송 큐에 저장합니다. 메시지가 전송되고 리모트 큐 관리자에 저장될 때까지 전송 큐는 메시지를 저장하는 특별한 유형의 로컬 큐입니다.

메시지 송신 및 수신을 핸들링하는 소프트웨어는 메시지 채널 에이전트(MCA)라고 합니다.

메시지는 채널의 큐 관리자 간에 전송됩니다. 채널은 두 큐 관리자 간의 단방향 통신 링크입니다. 리모트 큐 관리자의 임의의 수의 큐를 대상으로 하는 메시지를 전달할 수 있습니다.

메시지를 송신해야 하는 컴포넌트

메시지를 리모트 큐 관리자에 전송하려는 경우 로컬 큐 관리자는 전송 큐 및 채널에 대한 정의가 필요합니다.

각 채널 끝에는 송신 끝 또는 수신 끝과 같이 이를 정의하는 별도의 정의가 있습니다. 단순 채널은 로컬 큐 관리자의 송신자 채널 정의 및 리모트 큐 관리자의 수신자 채널 정의로 구성됩니다. 이 두 정의는 이름이 같아야 하며 함께 한 채널을 구성해야 합니다.

채널의 각 끝에 메시지 채널 에이전트(MCA)도 있습니다.

각 큐 관리자에 데드-레터 큐(미배달 메시지 큐라고도 하는)가 있어야 합니다. 목적지에 메시지를 전달할 수 없는 경우 이 큐에 메시지를 넣습니다.

48 페이지의 그림 6는 큐 관리자, 전송 큐, 채널 및 MCA 간의 관계를 표시합니다.

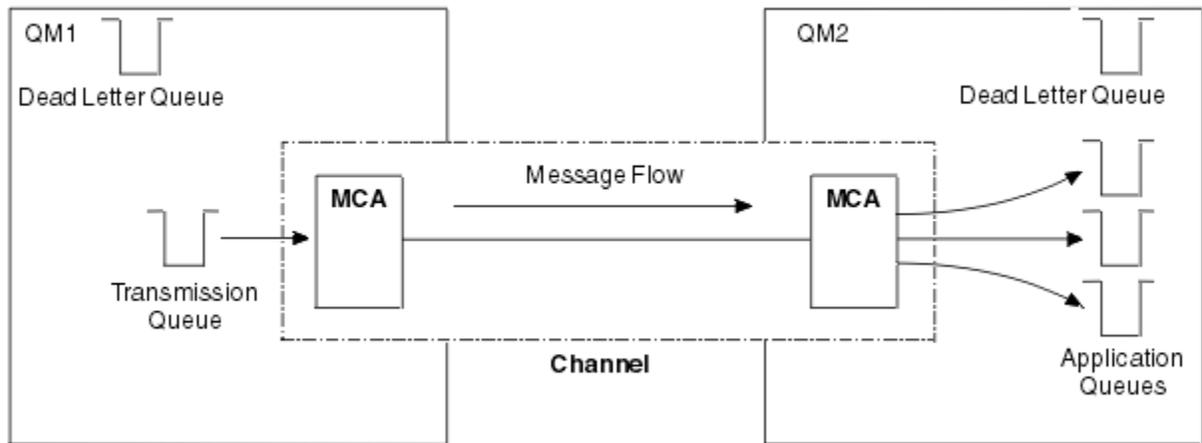


그림 6. 메시지 송신

메시지를 리턴해야 하는 컴포넌트

응용프로그램이 리모트 큐 관리자로부터 메시지가 리턴되어야 한다면, 49 페이지의 그림 7에서와 같이 큐 관리자간에 반대 방향으로 수행되도록 또 다른 채널을 정의해야 합니다.

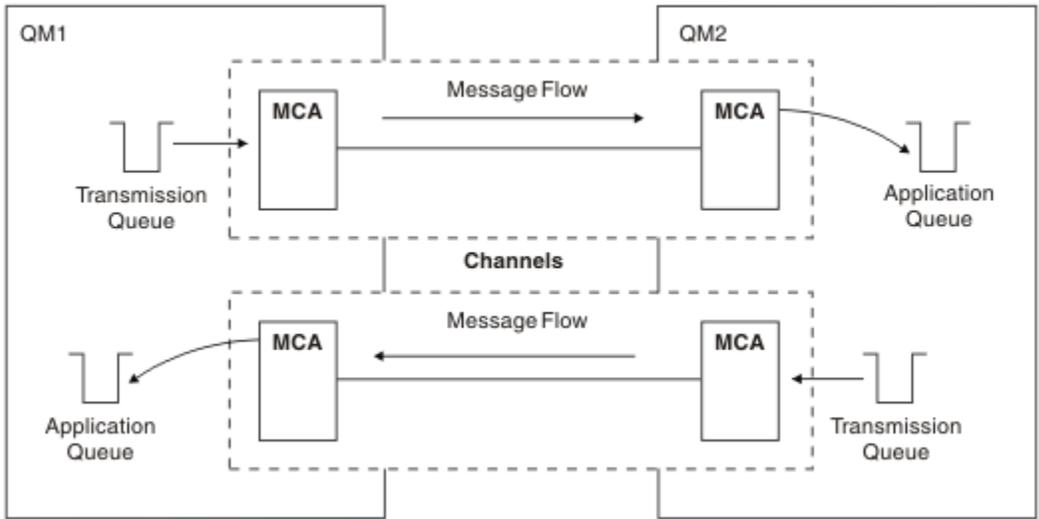


그림 7. 양방향으로 메시지 송신

분산 큐 관리에 대한 자세한 정보는 [분산 큐 관리 소개](#)를 참조하십시오.

클러스터 컴포넌트

수동으로 정의하는 채널을 통해 상호작용하는 기존의 WebSphere MQ 네트워크에 대한 대안은 클러스터를 사용하는 것입니다.

클러스터는 몇 가지 방법으로 논리적으로 연관되는 큐 관리자의 네트워크입니다. 큐 관리자를 클러스터로 그룹화하여 큐 관리자는 호스트하는 큐가 클러스터의 기타 모든 큐 관리자에 대해 사용 가능하도록 작성할 수 있습니다. 필요한 네트워크 인프라가 준비되어 있다고 가정하면서 큐 관리자는 명백한 채널 정의, 리모트 큐 정의 또는 각 목적지에 대한 전송 큐가 필요 없이 메시지를 동일 클러스터의 다른 큐 관리자에 송신할 수 있습니다. 클러스터의 모든 큐 관리자는 클러스터의 다른 큐 관리자에 메시지를 전송하는 단일 전송 큐가 있습니다. 각 큐 관리자는 하나의 클러스터-수신자 채널 및 하나의 클러스터-송신자 채널만을 정의해야 하며, 추가적인 채널은 클러스터가 자동으로 관리합니다.

49 페이지의 그림 8에서는 CLUSTER라는 클러스터의 컴포넌트를 보여줍니다.

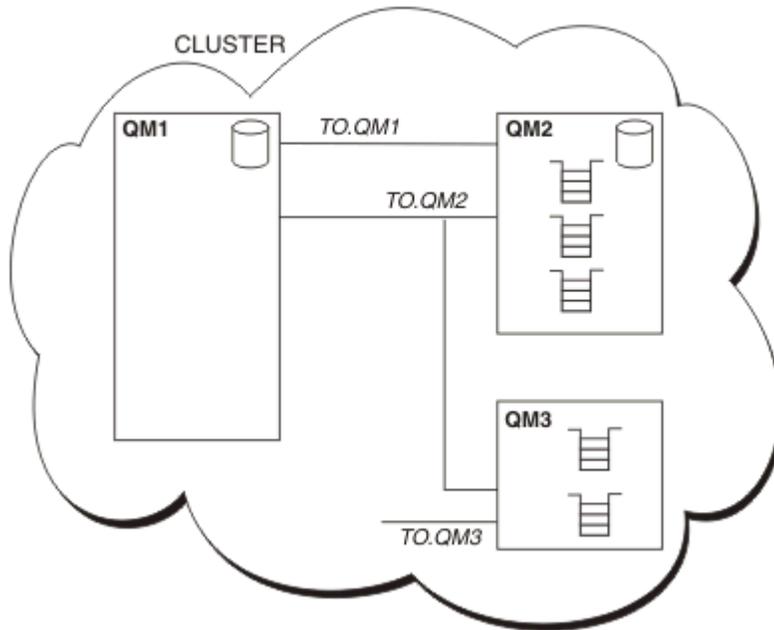


그림 8. 큐 관리자의 클러스터

- CLUSTER에는 세 개의 큐 관리자(QM1, QM2 및 QM3)가 포함되어 있습니다.
- QM1 및 QM2는 클러스터의 큐 및 큐 관리자에 관한 정보의 전체 저장소를 호스트합니다.
- QM2 및 QM3은 일부 클러스터 큐, 즉 클러스터의 다른 큐 관리자에 액세스 가능한 큐를 호스트합니다.
- 각 큐 관리자에는 메시지를 수신할 수 있는 TO.qmgr이라는 클러스터-수신자 채널이 있습니다.
- 각 큐 관리자에는 또한 정보를 저장소 큐 관리자 중의 하나로 송신할 수 있는 클러스터-송신자 채널이 있습니다.
- QM1 및 QM3은 QM2에 있는 저장소에 송신하며 QM2는 QM1에 있는 저장소에 송신합니다.

분산 큐잉과 마찬가지로, MQPUT 호출을 사용하여 메시지를 큐 관리자에 있는 큐에 넣습니다. MQGET 호출을 사용하여 로컬 큐에서 메시지를 검색합니다.

클러스터에 대한 자세한 정보는 35 페이지의 『큐 관리자 클러스터』의 내용을 참조하십시오.

관련 개념

50 페이지의 『분산 큐잉 컴포넌트』

상호통신 사용을 위해 필요한 오브젝트가 있습니다.

53 페이지의 『데드-레터 큐』

데드-레터 큐(또는 미전달 메시지)는 올바른 목적지로 라우트할 수 없는 경우 메시지를 송신하는 큐입니다.

53 페이지의 『리모트 큐 정의』

리모트 큐 정의는 다른 큐 관리자에서 소유하는 큐에 대한 정의입니다.

53 페이지의 『리모트 큐 관리자로 가져오는 방법』

각 소스 및 대상 큐 관리자 간에 하나의 채널이 항상 없을 수 있습니다. 두 관리자 간의 링크 방법으로 멀티호핑, 채널 공유, 다른 채널 사용 및 클러스터링을 포함한 여러 다른 방법이 있습니다.

55 페이지의 『주소 지정 정보』

애플리케이션이 리모트 큐 관리자에 대해 목적지인 메시지를 넣으면, 로컬 큐 관리자는 전송 큐에 넣기 전에 전송 헤더를 추가합니다. 이 헤더에는 목적지 큐와 큐 관리자의 이름, 즉 주소 지정 정보가 포함됩니다.

56 페이지의 『알리어스의 개념』

알리어스는 메시지 서비스 품질을 제공하는 데 사용됩니다. 애플리케이션을 변경하지 않고도 큐 관리자 알리어스를 사용하여 시스템 관리자가 대상 큐 관리자의 이름을 변경할 수 있습니다. 또한 시스템 관리자가 라우트를 목적지 큐 관리자로 변경하거나 다른 여러 큐 관리자를 통한 전달(멀티호핑)과 관련된 라우트를 설정할 수 있습니다. 응답 대상 큐 알리어스는 응답을 위한 서비스 품질을 제공합니다.

56 페이지의 『큐 관리자 알리어스 정의』

큐를 열어 메시지를 넣은 애플리케이션이 큐 이름 및 큐 관리자 이름을 지정하는 경우 큐 관리자 알리어스 정의가 적용됩니다.

57 페이지의 『응답 대상 큐 알리어스 정의』

응답 대상 큐 알리어스 정의는 메시지 디스크립터에서 응답 정보에 대한 대체 이름을 지정합니다. 애플리케이션을 변경하지 않고 큐나 큐 관리자의 이름을 변경할 수 있다는 이점이 있습니다.

분산 큐잉 컴포넌트

상호통신 사용을 위해 필요한 오브젝트가 있습니다.

분산 큐잉의 컴포넌트는 다음과 같습니다.

- 메시지 채널
- 메시지 채널 에이전트
- 전송 큐
- 채널 시작기 및 리스너
- 채널 엑시트 프로그램

메시지 채널은 하나의 큐 관리자에서 다른 큐 관리자로 메시지를 이동시키는 채널입니다. MQI 채널과 메시지 채널을 혼동하지 마십시오. 두 가지 유형의 MQI 채널(서버 연결(SVRCONN) 및 클라이언트 연결(CLNTCONN))이 있습니다. 자세한 정보는 MQI 채널을 참조하십시오.

메시지 채널의 각 끝의 정의는 다음 유형 중 하나일 수 있습니다.

- 송신자(SDR)
- 수신자(RCVR)
- 서버(SVR)
- 요청자(RQSTR)
- 클러스터 송신자(CLUSSDR)
- 클러스터 수신자(CLUSRCVR)

메시지 채널은 한 끝에 정의된 다음 유형 중 하나 및 다른 끝의 호환 가능한 유형을 사용하여 정의됩니다. 가능한 결합은 다음과 같습니다.

- 송신자-수신자
- 요청자-서버
- 요청자-송신자(콜백)
- 서버-수신자
- 클러스터 송신자-클러스터 수신자

송신자-수신자 채널의 작성에 대한 자세한 지시사항은 채널 정의에 포함되어 있습니다(z/OS에는 적용할 수 없음). 송신자-수신자 채널을 설정하기 위해 필요한 매개변수의 예는 사용자 플랫폼에 적용 가능한 구성 정보 예를 참조하십시오. 모든 유형의 채널을 정의하는 데 필요한 매개변수는 DEFINE CHANNEL을 참조하십시오.

송신자-수신자 채널

한 시스템의 송신자는 채널을 시작하여 메시지를 다른 시스템에 송신할 수 있습니다 송신자는 채널의 다른 끝에 있는 수신자가 시작되기를 요청합니다. 송신자는 메시지를 전송 큐에서 수신자에게 송신합니다. 수신자는 메시지를 목적지 큐에 넣습니다. 51 페이지의 그림 9 이것을 보여 주세요.

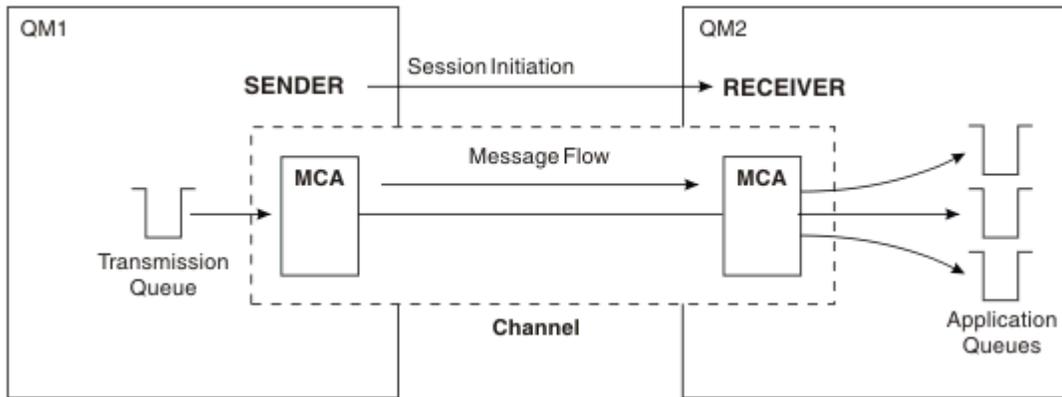


그림 9. 송신자-수신자 채널

요청자-서버 채널

한 시스템의 요청자는 채널을 시작하여 메시지를 다른 시스템에 수신할 수 있습니다 요청자는 채널의 다른 끝에 있는 서버가 시작되기를 요청합니다. 서버는 채널 정의에서 정의된 전송 큐에서 요청자에게 메시지를 송신합니다.

서버 채널은 통신을 시작하여 메시지를 요청자에게도 송신할 수 있습니다. 채널 정의에서 지정된 파트너의 연결 이름이 있는 서버 채널인 완전한 서버에만 적용됩니다. 완전한 서버는 요청자에서 시작되거나 요청자와의 통신을 시작할 수 있습니다.

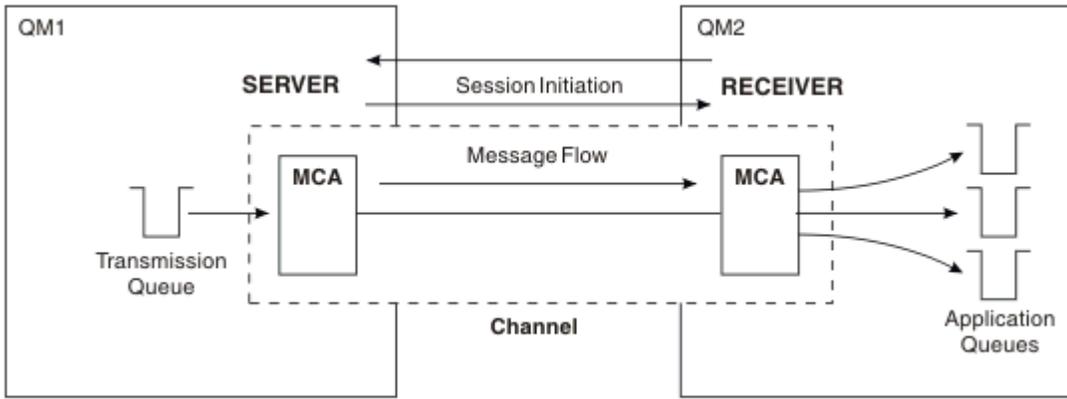


그림 10. 요청자-서버 채널

요청자-송신자 채널

요청자는 채널을 시작하고 송신자는 호출을 종료합니다. 그런 다음 송신자는 채널 정의(콜백이라는)의 정보에 따라 통신을 다시 시작합니다. 전송 큐에서 요청자에게 메시지를 송신합니다.

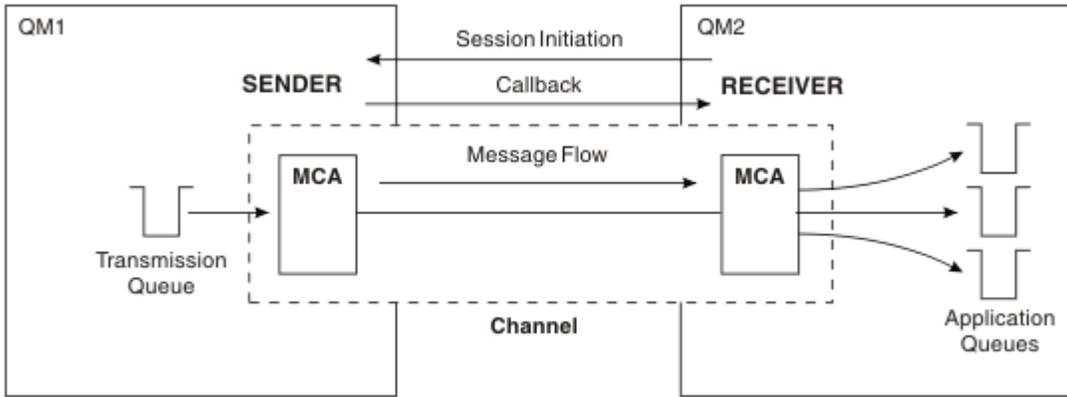


그림 11. 요청자-송신자 채널

서버-수신자 채널

송신자-수신자와 같지만 채널 정의에서 지정된 파트너의 연결 이름이 있는 서버 채널인 완전한 서버에만 적용됩니다. 채널 시동은 링크의 서버 끝에서 시작되어야 합니다. 이 그림은 [51 페이지의 그림 9](#)와 같습니다.

클러스터-송신자 채널

클러스터에서, 각 큐 관리자에는 클러스터 정보를 전체 저장소 큐 관리자 중 하나로 송신할 수 있는 클러스터-송신자 채널이 있습니다. 큐 관리자는 메시지를 클러스터-송신자 채널의 다른 큐 관리자에게도 송신할 수 있습니다.

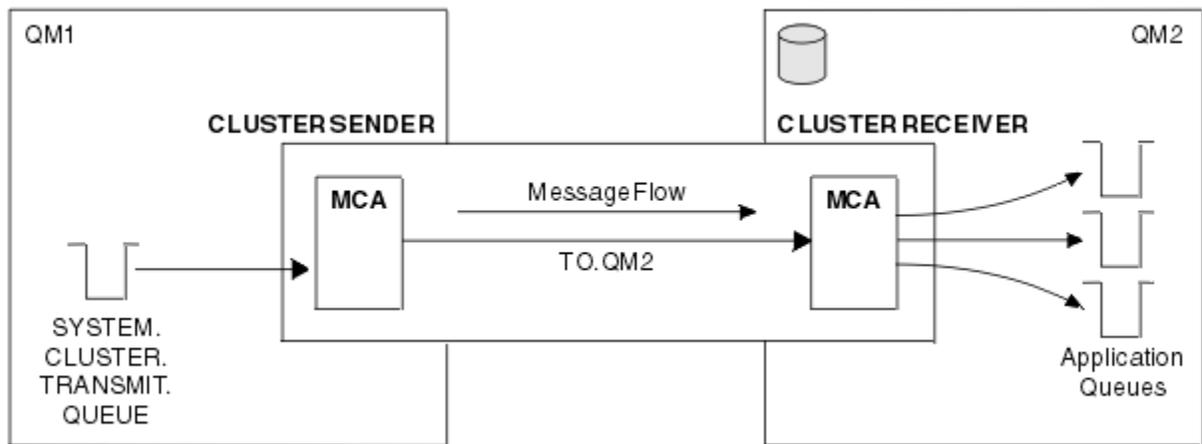


그림 12. 클러스터 송신자 채널

클러스터-수신자 채널

클러스터에는, 각 큐 관리자에 클러스터에 관한 정보 및 메시지를 수신할 수 있는 클러스터-수신자 채널이 있습니다. 이 그림은 53 페이지의 그림 12과 같습니다.

데드-레터 큐

데드-레터 큐(또는 미전달 메시지)는 올바른 목적지로 라우트할 수 없는 경우 메시지를 송신하는 큐입니다.

대상 큐에 넣을 수 없을 때 메시지를 이 큐에 넣습니다. 예를 들어, 큐가 없거나 큐가 가득 찼기 때문입니다. 데드-레터 큐가 데이터 변환 오류에 대한 채널 종료 송신에도 사용됩니다.

각 큐 관리자에 대해 데드 레터 큐 정의를 고려하십시오. 이를 고려하지 않고 MCA가 메시지를 넣을 수 없으면, 전송 큐에 남게 되며 채널이 중지됩니다.

또한 빠른 비지속 메시지(빠른 비지속 메시지 참조)를 전달할 수 없으며 데드-레터 큐가 대상 시스템에 없으면 이 메시지는 제거됩니다.

그러나, 데드-레터 큐 사용은 메시지를 전달한 순서에 영향을 줄 수 있어 이를 사용하지 않도록 선택할 수 있습니다.

USEDLQ 채널 속성을 사용하여 메시지를 전달할 수 없을 때 데드 레터 큐가 사용되는지 여부를 판별할 수 있습니다. 이 속성은 기타 기능이 수행되지 않는 동안 큐 관리자의 일부 기능이 데드-레터 큐를 사용할 수 있도록 구성될 수 있습니다. 다른 MQSC 명령에서 USEDLQ 채널 속성 사용에 대한 자세한 정보는 [DEFINE CHANNEL](#), [DISPLAY CHANNEL](#), [ALTER CHANNEL](#) 및 [DISPLAY CLUSQMGR](#)을 참조하십시오.

리모트 큐 정의

리모트 큐 정의는 다른 큐 관리자에서 소유하는 큐에 대한 정의입니다.

애플리케이션이 로컬 큐에서만 메시지를 검색할 수 있는 반면, 로컬 큐나 리모트 큐에 메시지를 넣을 수 있습니다. 그러므로, 각 로컬 큐에 대한 정의뿐 아니라 큐 관리자에게는 리모트 큐 정의가 있을 수 있습니다. 리모트 큐 정의의 장점은 리모트 큐나 리모트 큐 관리자의 이름, 또는 전송 큐의 이름을 지정하지 않고 애플리케이션을 사용하여 메시지를 리모트 큐에 넣을 수 있다는 것입니다. 리모트 큐 정의는 위치 독립성을 제공합니다.

리모트 큐 정의를 위한 다른 사용이 있으며, 이는 나중에 설명됩니다.

리모트 큐 관리자로 가져오는 방법

각 소스 및 대상 큐 관리자 간에 하나의 채널이 항상 없을 수 있습니다. 두 관리자 간의 링크 방법으로 멀티호핑, 채널 공유, 다른 채널 사용 및 클러스터링을 포함한 여러 다른 방법이 있습니다.

멀티홉

소스 큐 관리자와 대상 큐 관리자 간에 직접적인 통신 링크가 없는 경우, 대상 큐 관리자로 가는 중에 하나 이상의 중간 큐 관리자를 통해 전달할 수 있습니다. 이를 멀티홉이라고 합니다.

모든 큐 관리자와 중간 큐 관리자의 전송 큐 사이에 채널을 정의해야 합니다. 54 페이지의 [그림 13](#)에 설명되어 있습니다.

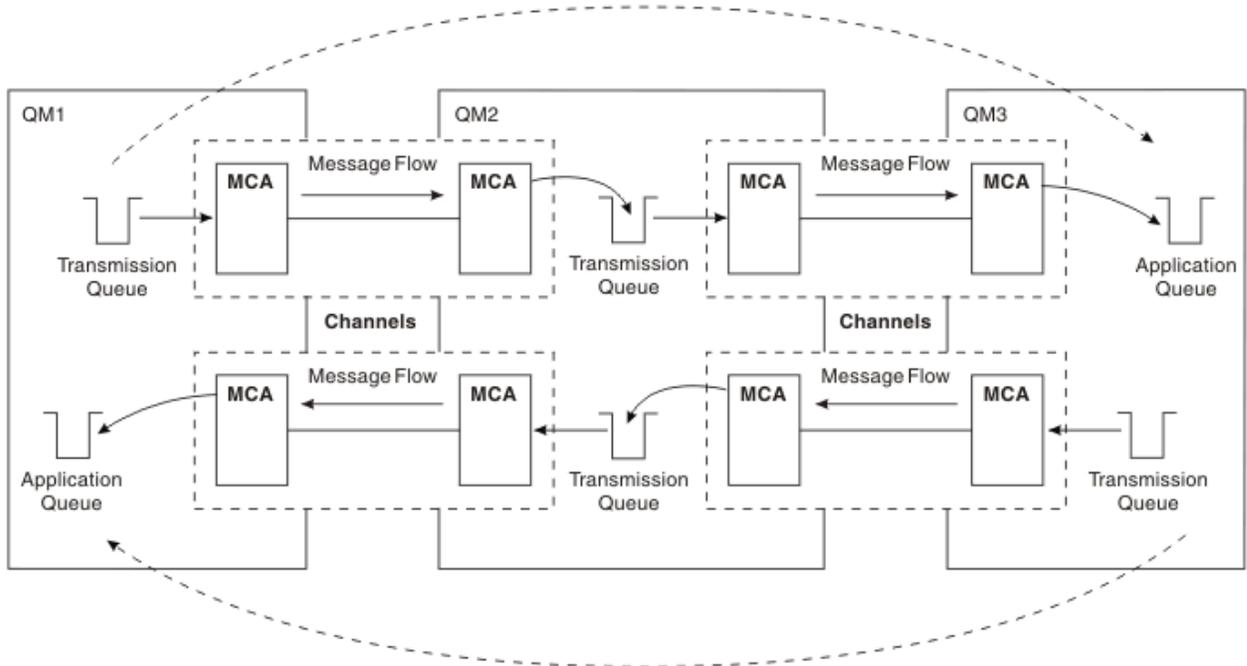


그림 13. 중간 큐 관리자를 통해 전달

채널 공유

애플리케이션 설계자로서 애플리케이션이 큐 이름과 함께 리모트 큐 관리자 이름을 지정하도록 강제 실행하거나 각 리모트 큐에 대해 리모트 큐 정의를 작성하도록 선택할 수 있습니다. 이 정의에는 리모트 큐 관리자 이름, 큐 이름 및 전송 큐의 이름이 있습니다. 이런 방식으로, 동일한 리모트 위치에 큐의 주소를 지정하는 모든 애플리케이션의 모든 메시지에는 동일한 전송 큐를 통해 전송된 메시지가 있습니다. 54 페이지의 [그림 14](#)에 설명되어 있습니다.

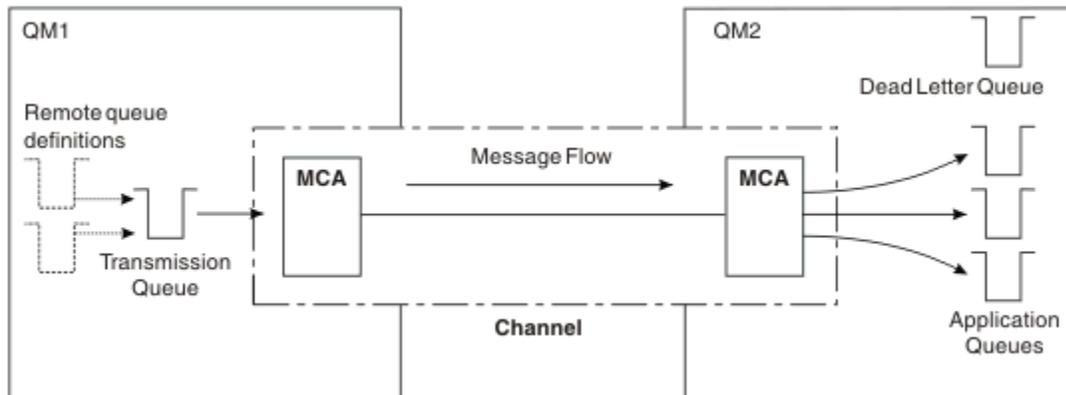


그림 14. 전송 큐 공유

54 페이지의 [그림 14](#)는 메시지가 다중 애플리케이션에서 다중 리모트 큐까지 동일 채널을 사용할 수 있음을 설명합니다.

다른 채널 사용

두 큐 관리자 간에 송신될 다른 유형의 메시지가 있는 경우 두 관리자 간에 둘 이상의 채널을 정의할 수 있습니다. 아마도 보안 용도로, 대량의 메시지 트래픽에 대한 전달 속도의 균형을 유지하기 위해 대체 채널이 필요한 경우가 있습니다.

다른 채널 및 다른 전송 큐를 정의하기 위해 필요한 두 번째 채널을 설정하려면 위치를 지정하는 리모트 큐 정의 및 전송 큐 이름을 작성하십시오. 그런 다음 애플리케이션은 채널을 사용할 수 있지만 메시지는 여전히 동일한 대상 큐에 전달됩니다. 55 페이지의 그림 15에 설명되어 있습니다.

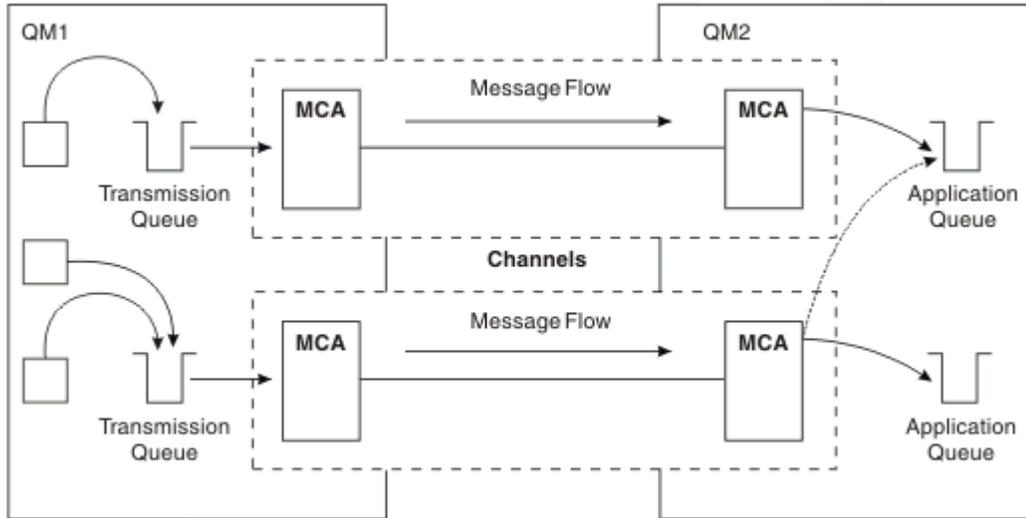


그림 15. 다중 채널 사용

리모트 큐 정의를 사용하여 전송 큐를 지정하는 경우 애플리케이션은 위치(즉, 목적지 큐 관리자) 자체를 지정하지 **않아야** 합니다. 이렇게 하는 경우, 큐 관리자는 리모트 큐 정의를 사용하지 않습니다. 리모트 큐 정의는 위치 독립성을 제공합니다. 애플리케이션은 큐가 있는 위치를 알지 않아도 메시지를 논리 큐에 넣을 수 있으며 애플리케이션을 변경하지 않고도 물리적 큐를 변경할 수 있습니다.

클러스터링 사용

클러스터 내의 모든 큐 관리자는 클러스터 수신자 채널을 정의합니다. 다른 큐 관리자가 메시지를 큐 관리자에 보내려는 경우, 해당 클러스터 송신자 채널을 자동으로 정의합니다. 예를 들어, 클러스터에 둘 이상의 큐 인스턴스가 있는 경우 클러스터 송신자 채널은 큐를 호스팅하는 큐 관리자에 정의될 수 있습니다. WebSphere MQ는 라운드 로빈 루틴을 사용하여 메시지를 라우트할 수 있는 가능한 큐 관리자를 선택하는 워크로드 관리 알고리즘을 사용합니다. 자세한 정보는 115 페이지의 『클러스터』의 내용을 참조하십시오.

주소 지정 정보

애플리케이션이 리모트 큐 관리자에 대해 목적지인 메시지를 넣으면, 로컬 큐 관리자는 전송 큐에 넣기 전에 전송 헤더를 추가합니다. 이 헤더에는 목적지 큐와 큐 관리자의 이름, 즉 주소 지정 정보가 포함됩니다.

단일 큐 관리자 환경에서, 애플리케이션이 메시지를 넣을 큐를 열 때 목적지 큐의 주소가 설정됩니다. 목적지 큐가 동일한 큐 관리자에 있기 때문에, 주소 정보에는 필요하지 않습니다.

분산 환경에서 큐 관리자는 목적지 큐 이름뿐만 아니라 해당 큐의 위치(즉, 큐 관리자 이름) 및 리모트 위치에 대한 라우트(즉, 전송 큐)를 알아야 합니다. 이 주소 지정 정보는 전송 헤더에 포함되어 있습니다. 수신 채널은 전송 헤더를 제거하며 전송 헤더의 정보를 사용하여 목적지 큐를 찾습니다.

리모트 큐 정의를 사용하는 경우 애플리케이션이 목적지 큐 관리자의 이름을 지정하지 않아도 됩니다. 이 정의는 리모트 큐의 이름, 메시지의 대상인 리모트 큐 관리자의 이름 및 메시지를 전송하는 데 사용되는 전송 큐의 이름을 지정합니다.

알리어스의 개념

알리어스는 메시지 서비스 품질을 제공하는 데 사용됩니다. 애플리케이션을 변경하지 않고도 큐 관리자 알리어스를 사용하여 시스템 관리자가 대상 큐 관리자의 이름을 변경할 수 있습니다. 또한 시스템 관리자가 라우트를 목적지 큐 관리자로 변경하거나 다른 여러 큐 관리자를 통한 전달(멀티호핑)과 관련된 라우트를 설정할 수 있습니다. 응답 대상 큐 알리어스는 응답을 위한 서비스 품질을 제공합니다.

큐 관리자 알리어스 및 응답 대상 큐 알리어스는 공백 RNAME이 있는 리모트 큐 정의를 사용하여 작성됩니다. 이 정의는 실제 큐를 정의하지 않습니다. 큐 관리자가 이 정의를 사용하여 물리적인 큐 이름, 큐 관리자 이름 및 전송 큐를 해석합니다.

알리어스 정의는 공백 RNAME이 있다는 특징을 가집니다.

큐 이름 해석

큐가 열릴 때마다 모든 큐 관리자에서 큐 이름 해석이 이루어집니다. 대상 큐, 대상 큐 관리자(로컬일 수 있음) 및 큐 관리자에 대한 라우트(널일 수 있음)를 식별하는 것이 목적입니다. 해석된 이름에는 큐 관리자 이름, 큐 이름 및 큐 관리자가 리모트인 경우 전송 큐의 세 부분이 있습니다.

리모트 큐 정의가 존재하면 알리어스 정의를 참조하지 않습니다. 애플리케이션에서 제공되는 큐 이름은 리모트 큐 정의에서 지정된 전송 큐, 리모트 큐 관리자 및 목적지 큐의 이름으로 해석됩니다. 큐 이름 해석에 대한 자세한 정보는 [큐 이름 해석](#)을 참조하십시오.

리모트 큐 정의가 없고 큐 관리자 이름이 지정되어 있거나 이름 서비스로 해석된 경우, 큐 관리자는 제공된 큐 관리자 이름과 일치하는 큐 관리자 알리어스 정의가 있는지 보고 확인합니다. 있는 경우, 정의 내의 정보를 사용하여 큐 관리자 이름을 목적지 큐 관리자의 이름으로 해석합니다. 큐 관리자 알리어스 정의는 목적지 큐 관리자에 대한 전송 큐를 판별하는 데도 사용될 수 있습니다.

해석된 큐 이름이 로컬 큐가 아닌 경우, 큐 관리자 이름 및 큐 이름 모두 애플리케이션이 전송 큐에 넣은 각 메시지의 전송 헤더에 포함됩니다.

리모트 큐 정의 또는 큐 관리자 알리어스 정의가 변경되지 않는 한 일반적으로 사용된 전송 큐의 이름은 해석된 큐 관리자와 동일합니다. 이러한 전송 큐를 정의하지 않았지만 기본 전송 큐를 정의한 경우 이 이름이 사용됩니다.

z/OS에서 실행 중인 큐 관리자의 이름은 4자로 제한됩니다.

큐 관리자 알리어스 정의

큐를 열어 메시지를 넣은 애플리케이션이 큐 이름 및 큐 관리자 이름을 지정하는 경우 큐 관리자 알리어스 정의가 적용됩니다.

큐 관리자 알리어스 정의는 다음과 같이 3가지로 사용됩니다.

- 메시지 송신 시, 큐 관리자 이름 다시 맵핑
- 메시지 송신 시, 전송 큐 변경 또는 지정
- 메시지 수신 시, 로컬 큐 관리자가 해당 메시지에 대해 의도된 목적지인지 여부 판별

아웃바운드 메시지 - 큐 관리자 이름 다시 맵핑

큐 관리자 알리어스 정의를 사용하여 MQOPEN 호출에서 지정된 큐 관리자 이름을 다시 맵핑할 수 있습니다. 예를 들어, MQOPEN 호출은 THISQ의 큐 이름 및 YOURQM의 큐 관리자 이름을 지정합니다. 로컬 큐 관리자에 다음 예제와 같은 큐 관리자 알리어스 정의가 있습니다.

```
DEFINE QREMOTE (YOURQM) RQMNAME(REALQM)
```

애플리케이션이 메시지를 큐 관리자 YOURQM에 넣을 때 사용할 실제 큐 관리자가 REALQM임을 표시합니다. 로컬 큐 관리자가 REALQM인 경우, 메시지를 로컬 큐인 THISQ 큐에 넣습니다. 로컬 큐 관리자가 REALQM이 아닌 경우, REALQM이라는 전송 큐로 메시지를 라우트합니다. 큐 관리자는 YOURQM 대신 REALQM을 알리도록 전송 헤더를 변경합니다.

아웃바운드 메시지 - 전송 큐 대체 또는 지정

57 페이지의 그림 16에서는 큐 관리자 QM3에 큐 이름을 표시하는 전송 헤더가 있는 큐 관리자 QM1에 메시지가 도달하는 시나리오를 보여줍니다. 이 시나리오에서 QM3는 QM2를 통해 멀티호핑하여 도달할 수 있습니다.

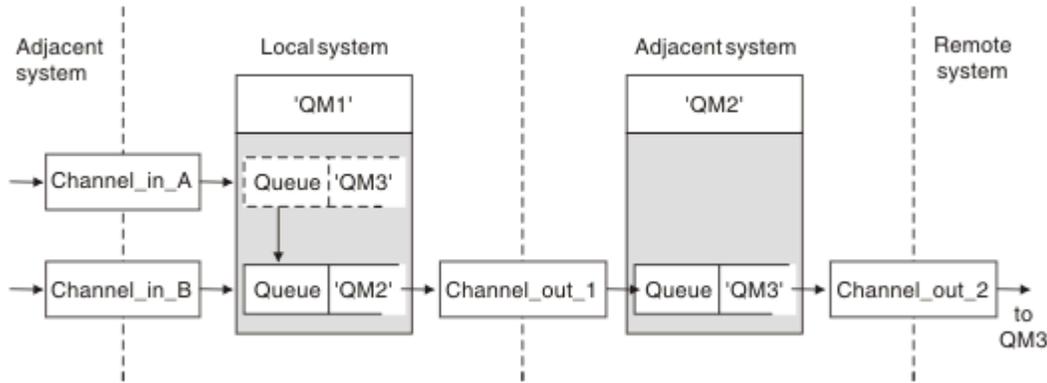


그림 16. 큐 관리자 알리어스

QM3의 모든 메시지는 큐 관리자 알리어스가 있는 QM1에 캡처됩니다. 큐 관리자 알리어스의 이름은 QM3로 지정되며 전송 큐 QM2를 통한 정의 QM3를 포함합니다. 정의는 다음 예제와 같습니다.

```
DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM2)
```

큐 관리자는 메시지를 전송 큐 QM2에 넣지만 목적지 큐 관리자 QM3의 이름이 변경되지 않아 전송 큐 헤더를 변경하지 않습니다.

QM1에 도달하여 QM2의 큐 이름을 포함한 전송 헤더를 표시하는 모든 메시지도 QM2 전송 큐에 넣습니다. 이런 방식으로 목적지가 다른 메시지는 적절한 인접 시스템에 대한 공용 전송 큐로 수집되어 계속 목적지로 전송됩니다.

인바운드 메시지 - 목적지 판별

수신 MCA는 전송 헤더에서 참조된 큐를 엽니다. 큐 관리자 알리어스 정의가 참조된 큐 관리자와 동일한 이름으로 존재하면 전송 헤더에서 수신된 큐 관리자 이름은 해당 목적지의 RQMNAME으로 바뀝니다.

이 프로세스는 다음과 같이 2가지로 사용됩니다.

- 메시지를 다른 큐 관리자에게 전달
- 큐 관리자 이름을 로컬 큐 관리자와 동일하도록 변경

응답 대상 큐 알리어스 정의

응답 대상 큐 알리어스 정의는 메시지 디스크립터에서 응답 정보에 대한 대체 이름을 지정합니다. 애플리케이션을 변경하지 않고 큐나 큐 관리자의 이름을 변경할 수 있다는 이점이 있습니다.

큐 이름 해석

애플리케이션이 메시지에 응답할 때 수신된 메시지의 메시지 디스크립터에 있는 데이터를 사용하여 응답할 큐의 이름을 찾습니다. 송신 애플리케이션은 응답이 송신되어 이 정보를 메시지에 첨부하는 위치를 표시합니다. 이 개념은 애플리케이션 디자인의 일부로 통합될 수 있습니다.

큐 이름 해석은 메시지를 큐에 넣기 전에 애플리케이션의 송신 끝에서 발생합니다. 큐 이름 해석의 일반적이지 않은 사용입니다. 큐가 열리지 않은 때 이름 해석이 이루어지는 상황에서만 해당됩니다. 따라서 큐 이름 해석은 원격 애플리케이션과 상호작용하기 전에 발생합니다.

큐 관리자 알리어스를 사용하여 큐 이름 해석

보통 애플리케이션은 응답 대상 큐를 지정하며 응답 대상 큐 관리자 이름을 공백으로 둡니다. 큐 관리자는 넣기 시 자신의 이름을 완성합니다. 예를 들어, 이 방법은 전송 큐 QM1을 사용하는 기본 리턴 채널 대신 전송 큐

QM1_relief를 사용하는 채널의 경우와 같이 응답에 사용할 대체 채널을 원하는 경우를 제외하고는 잘 작동합니다. 이런 상황에서는 전송 큐 헤더에 지정된 큐 관리자 이름이 "실제" 큐 관리자 이름과 일치하지 않지만 큐 관리자 알리어스 정의를 사용하여 다시 지정됩니다. 대체 라우트와 함께 응답을 리턴하려면 응답 대상 큐 알리어스 정의를 사용하여 응답 대상 큐 데이터에도 맵핑해야 합니다.

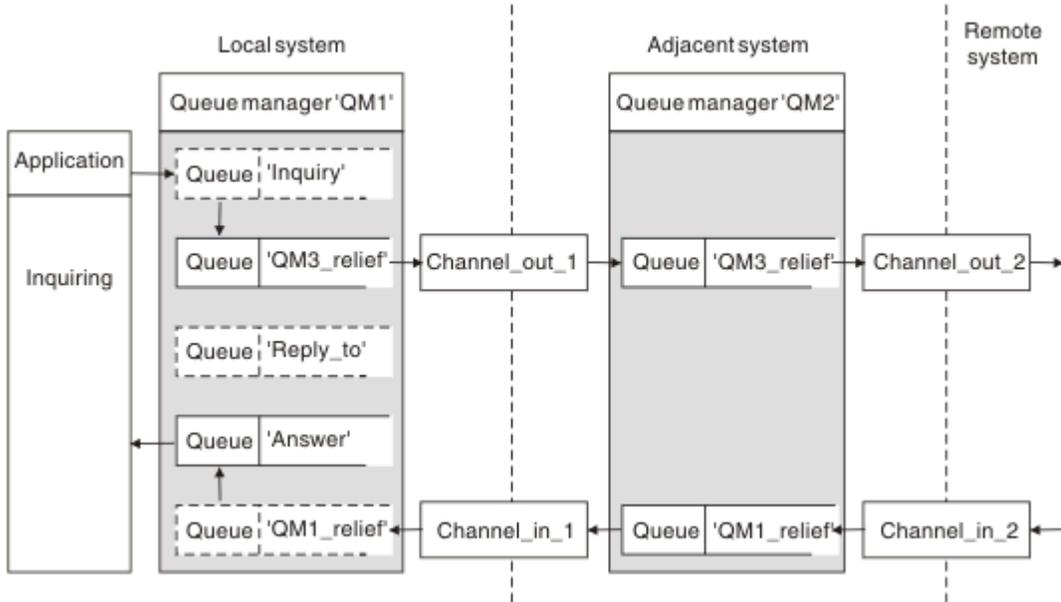


그림 17. 응답 위치 변경에 사용되는 응답 대상 큐 알리어스

58 페이지의 그림 17에서의 예제:

1. 애플리케이션은 MQPUT 호출을 사용하고 메시지 디스크립터에서 다음 정보를 지정하여 메시지를 넣습니다.

```
ReplyToQ='Reply_to'
ReplyToQMgr='QM1_relief'
```

응답 대상 큐 알리어스가 사용되려면 ReplyToQMgr는 공백이어야 합니다.

2. 이름 Answer 및 큐 관리자 이름 QM1_relief가 포함된 Reply_to라는 응답 대상 큐 알리어스 정의를 작성합니다.

```
DEFINE QREMOTE ('Reply_to') RNAME ('Answer')
RQMNAME ('QM1_relief')
```

3. 메시지는 ReplyToQ='Answer' 및 ReplyToQMgr='QM1_relief'를 표시하는 메시지 디스크립터와 함께 전송됩니다.
4. 애플리케이션 스펙에는 응답을 Reply_to가 아닌 큐 Answer에서 찾을 수 있는 정보를 포함해야 합니다.

다음을 정의하는 병렬 리턴 채널을 작성해야 하는 응답을 준비하려면:

- QM2에서, QM1_relief라는 전송 큐

```
DEFINE QLOCAL ('QM1_relief') USAGE(XMITQ)
```

- QM1에서, 큐 관리자 알리어스 QM1_relief

```
DEFINE QREMOTE ('QM1_relief') RNAME() RQMNAME(QM1)
```

이 큐 관리자 알리어스는 병렬 리턴 채널의 체인을 종료하며 QM1의 메시지를 캡처합니다.

나중에 가끔 이를 수행하기를 원한다고 생각되면, 애플리케이션이 시작부터 알리어스 이름을 사용하는지 확인하십시오. 현재까지는 응답 대상 큐에 대한 정상 큐 알리어스이지만 나중에는 큐 관리자 알리어스로 변경될 수 있습니다.

응답 대상 큐 이름

응답 대상 큐 이름 지정에 주의가 필요합니다. 애플리케이션이 응답 대상 큐 이름을 메시지에 넣은 이유는 응답이 전송되는 큐를 지정할 수 있기 때문입니다. 이 이름으로 응답 대상 큐 알리어스 정의를 작성하면 실제 응답 대상 큐(즉, 로컬 큐 정의)를 동일한 이름으로 지정할 수 없습니다. 그러므로 응답 대상 큐 알리어스 정의가 큐 관리자 이름뿐만 아니라 새 큐 이름을 포함해야 하며 애플리케이션 스펙에는 다른 큐에서 응답을 찾을 수 있는 정보를 포함해야 합니다.

이제 애플리케이션은 원본 메시지를 넣을 때의 응답 대상 큐로 이름 지정된 큐와 다른 큐에서 메시지를 검색해야 합니다.

클러스터 작동 방법

클러스터 개념 및 작동 방식에 대한 이해

클러스터는 몇 가지 방법으로 논리적으로 연관되는 큐 관리자의 네트워크입니다. 클러스터의 큐 관리자는 물리적으로 리모트일 수 있습니다. 예를 들어 국제 연쇄점의 지사를 나타내고 물리적으로 서로 다른 국가에 위치할 수 있습니다. 엔터프라이즈 내의 각 클러스터는 고유한 이름을 가져야 합니다.

일반적으로 클러스터에는 몇 가지 방법으로 논리적으로 관련되며 데이터나 애플리케이션을 공유해야 하는 큐 관리자가 들어있습니다. 예를 들어 회사의 각 부서에 대해 해당 부서에 특정한 데이터 및 애플리케이션을 관리하는 하나의 큐 관리자를 가질 수 있습니다. 이들 큐 관리자를 모두 클러스터로 그룹화하여 모두가 급여 애플리케이션에 공급하도록 만들 수 있습니다. 또는 연쇄점의 각 지사에 대해 해당 지사에 대한 재고 수준 및 기타 정보를 관리하는 하나의 큐 관리자를 가질 수 있습니다. 이들 큐 관리자를 클러스터로 그룹화하면 동일한 판매 및 구매 애플리케이션 세트에 액세스할 수 있습니다. 판매 및 구매 애플리케이션은 중앙 집중적으로 본사 큐 관리자에 보유될 수 있습니다.

클러스터를 설정한 후에는 그 안에 있는 큐 관리자가 추가 채널 정의나 리모트 큐 정의를 정의하지 않고도 서로 통신할 수 있습니다.

큐 관리자의 기존 네트워크를 클러스터로 변환할 수 있거나 새 네트워크 설정의 파트로서 클러스터를 설정할 수 있습니다.

IBM WebSphere MQ 클라이언트는 임의의 다른 큐 관리자에게 연결할 수 있는 것처럼 클러스터의 일부인 큐 관리자에 연결할 수 있습니다.

또한 워크로드 관리에 클러스터를 사용할 수도 있습니다. 자세한 정보는 [워크로드 관리를 위한 클러스터 사용](#)을 참조하십시오.

메시지가 클러스터에서 라우트되는 방식

IBM WebSphere MQ 및 분산 큐잉에 친숙한 경우 클러스터를 성실한 시스템 관리자가 유지보수하는 큐 관리자의 네트워크로 생각하십시오. 사용자가 클러스터 큐를 정의할 때마다 시스템 관리자가 자동으로 다른 큐 관리자에서 필요한 대로 대응하는 리모트 큐 정의를 작성합니다.

IBM WebSphere MQ가 클러스터의 각 큐 관리자에 대한 전송 큐를 제공하기 때문에 사용자가 전송 큐 정의를 작성할 필요가 없습니다. 이 단일 전송 큐를 사용하여 클러스터의 다른 모든 큐 관리자로 메시지를 전달할 수 있습니다. 단일 전송 큐를 사용하도록 제한되지는 않습니다. 큐 관리자는 다중 전송 큐를 사용하여 클러스터의 각 큐 관리자로 전달되는 메시지를 구분할 수 있습니다. 일반적으로 큐 관리자는 단일 클러스터 전송 큐를 사용합니다. 큐 관리자에서 클러스터의 각 큐 관리자마다 각기 다른 클러스터 전송 큐를 사용할 수 있도록 큐 관리자 속성 DEFCLXQ를 변경할 수 있습니다. 또한 클러스터 전송 큐를 수동으로 정의할 수도 있습니다.

클러스터에 조인하는 모든 큐 관리자는 이 방식으로 작업하는 데 동의합니다. 자기 자신 및 자신이 호스팅하는 큐에 대한 정보를 송신하고, 클러스터의 다른 멤버에 관한 정보를 수신합니다.

이 정보는 저장소에 저장됩니다. 대부분의 큐 관리자는 필요한 정보만 보유합니다. 즉, 통신해야 하는 큐 및 큐 관리자에 관한 정보만 보유합니다. 각 큐 관리자는 해당 정보를 부분 저장소에 보관합니다. 일부 지정된 큐 관리자가 클러스터의 모든 큐 관리자에 관한 모든 정보의 전체 저장소를 보유합니다.

클러스터의 파트가 되기 위해서는 큐 관리자가 클러스터-송신자 채널과 클러스터-수신자 채널이라는 두 채널을 가져야 합니다.

클러스터-송신자 채널은 송신자 채널 같은 통신 채널입니다. 이미 클러스터의 멤버인 전체 저장소에 연결하기 위해 수동으로 큐 관리자에 하나의 클러스터-송신자 채널을 작성해야 합니다.

클러스터-수신자 채널은 수신자 채널 같은 통신 채널입니다. 하나의 클러스터-수신자 채널을 수동으로 작성해야 합니다. 클러스터 통신을 수신하기 위해 채널이 큐 관리자에 대한 메커니즘으로 작용합니다.

그런 다음 이 큐 관리자와 클러스터의 다른 모든 멤버와의 통신에 필요할 수 있는 다른 모든 채널은 자동으로 작성됩니다.

클러스터를 지원하는 플랫폼의 큐 관리자는 클러스터의 파트일 필요가 없습니다. 클러스터 사용 대신 또는 클러스터 사용뿐 아니라 분산 큐잉 기술을 계속해서 사용할 수 있습니다.

클러스터 예

60 페이지의 그림 18는 CLSTR1이라는 클러스터의 컴포넌트를 보여줍니다.

- 이 클러스터에는 세 개의 큐 관리자(QM1, QM2 및 QM3)가 있습니다.
- QM1 및 QM2에서는 클러스터의 모든 큐 관리자 및 클러스터 관련 오브젝트에 대한 정보의 저장소를 호스팅합니다. 이를 전체 저장소 큐 관리자라 부릅니다. 저장소는 다이어그램에서 음영이 있는 실린더로 표시됩니다.
- QM2 및 QM3에서는 클러스터의 다른 큐 관리자에 액세스할 수 있는 일부 큐를 호스팅합니다. 클러스터의 다른 큐 관리자에 액세스할 수 있는 큐를 클러스터 큐라 합니다. 클러스터 큐는 다이어그램에서 음영이 있는 큐로 표시됩니다. 클러스터 큐는 클러스터의 어디에서나 액세스할 수 있습니다. IBM WebSphere MQ 클러스터링 코드는 클러스터 큐에 대한 리모트 큐 정의가 이 정의를 참조하는 모든 큐 관리자에 작성되도록 합니다.

분산 큐잉과 마찬가지로 애플리케이션은 MQPUT 호출을 사용하여 클러스터의 모든 큐 관리자에 있는 클러스터 큐에 메시지를 넣습니다. 애플리케이션은 MQGET 호출을 사용하여 큐가 상주하는 큐 관리자의 클러스터 큐에서만 메시지를 검색합니다.

- 각 큐 관리자에는 메시지를 수신할 수 있는 *cluster-name.queue-manager*이라는 채널의 수신 측에 대해 수동으로 작성된 정의가 있습니다. 수신 큐 관리자에서 *cluster-name.queue-manager*은 클러스터 수신자 채널입니다. 클러스터 수신자 채널은 분산 큐잉에서 사용되는 수신자 채널과 유사하며 큐 관리자에 대한 메시지를 수신합니다. 클러스터에 관한 정보도 수신합니다.

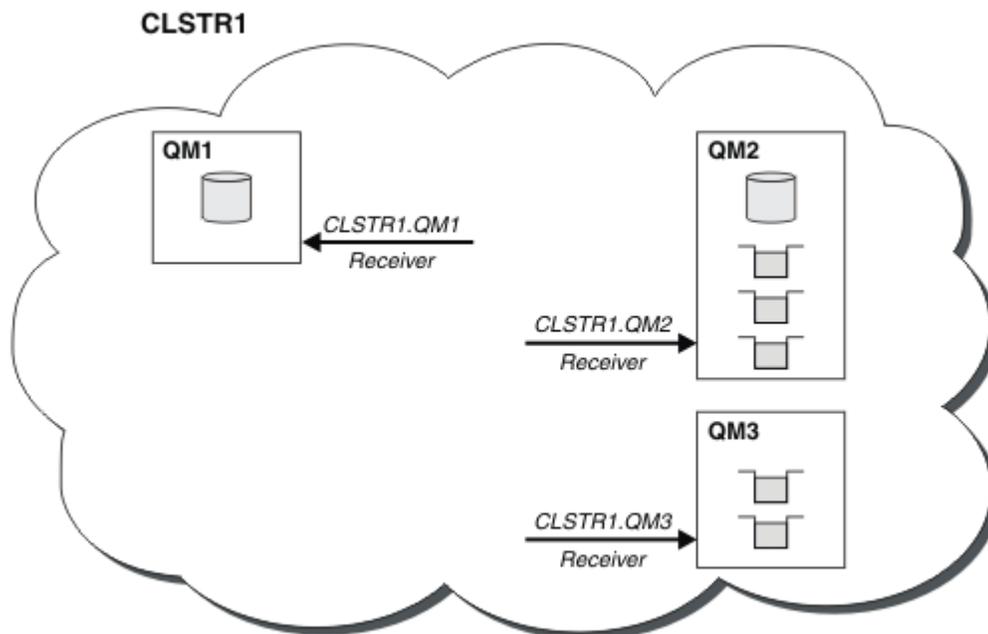


그림 18. 큐 관리자의 클러스터

- 61 페이지의 그림 19에는 각 큐 관리자마다 채널의 송신 측에 대한 정의도 있습니다. 각 큐 관리자는 전체 저장소 큐 관리자 중 하나의 클러스터 수신자 채널에 연결합니다. 송신 큐 관리자에서 `cluster-name.queue-manager`은(는) 클러스터 전송자 채널입니다. QM1 및 QM3에 CLSTR1.QM2에 연결하는 클러스터 송신자 채널이 있습니다(점선 "2" 참조).

QM2에는 CLSTR1.QM1에 연결하는 클러스터 송신자 채널이 있습니다(점선 "3" 참조). 클러스터-송신자 채널은 분산 큐잉에서 사용되는 송신자 채널과 비슷하며, 수신 큐 관리자로 메시지를 송신합니다. 클러스터에 관한 정보도 보냅니다.

클러스터 수신자 채널의 끝과 클러스터 송신자 채널의 끝이 모두 정의되고 나면 채널이 자동으로 시작됩니다.

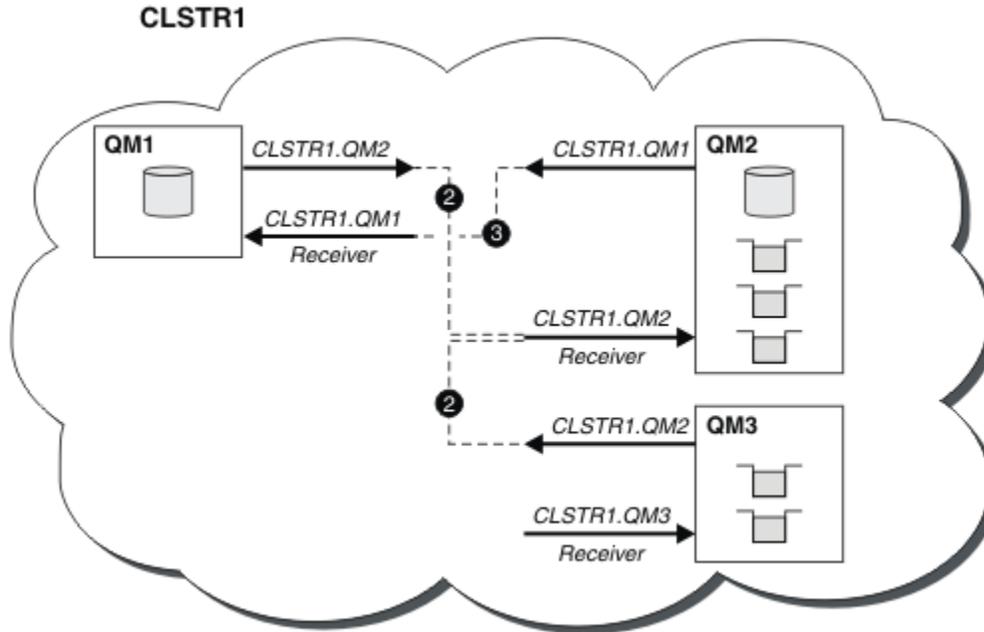


그림 19. 송신자 채널이 있는 큐 관리자의 클러스터

클러스터링 작동 원리

로컬 큐 관리자에 클러스터 송신자 채널을 정의하면 해당 큐 관리자가 전체 저장소 큐 관리자 중 하나에 도입됩니다. 전체 저장소 큐 관리자는 그에 따라 전체 저장소에서 정보를 업데이트합니다. 그런 다음 자동으로 원래 큐 관리자에 다시 클러스터 송신자 채널을 작성하고 클러스터에 대한 이 큐 관리자 정보를 보냅니다. 따라서 큐 관리자는 클러스터에 대해 학습하고 클러스터는 큐 관리자에 대해 학습합니다.

60 페이지의 그림 18를 다시 보십시오. 큐 관리자 QM3에 연결된 애플리케이션이 QM2의 큐에 일부 메시지를 보내려 한다고 가정합니다. QM3이(가) 해당 큐에 처음으로 액세스해야 하는 경우 전체 저장소를 참조하여 해당 큐를 발견합니다. 이 경우 전체 저장소는 송신자 채널 CLSTR1.QM2을(를) 사용하여 액세스하는 QM2입니다. 저장소의 정보를 사용하여 해당 큐에 대한 리모트 정의를 자동으로 작성할 수 있습니다. 큐가 QM1에 있으면 QM2은(는) 전체 저장소이므로 이 메커니즘은 계속 작동합니다. 전체 저장소에 클러스터의 모든 오브젝트에 대한 전체 레코드가 있습니다. 후자의 경우, QM3에서 QM1의 클러스터 수신자 채널에 해당하는 클러스터 송신자 채널을 자동으로 작성하므로 둘 간에 직접 통신할 수도 있습니다.

62 페이지의 그림 20는 자동으로 작성된 두 개의 클러스터 송신자 채널이 있는 동일한 클러스터를 보여줍니다. 클러스터 송신자 채널은 클러스터 수신자 채널 CLSTR1.QM3과(와) 결합하는 두 개의 점선으로 표시됩니다. QM1이(가) 메시지를 전송하는 데 사용하는 클러스터 전송 큐, SYSTEM.CLUSTER.TRANSMIT.QUEUE도 표시합니다. 클러스터의 모든 큐 관리자에는 클러스터 전송 큐가 있어서 여기에서 동일한 클러스터의 다른 모든 큐 관리자로 메시지를 보낼 수 있습니다.

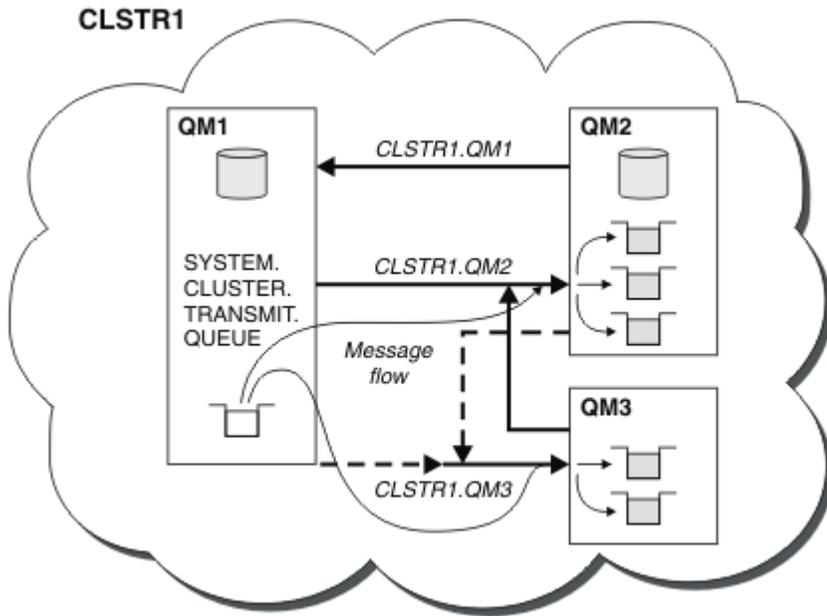


그림 20. 자동 정의된 채널이 표시된 큐 관리자의 클러스터

참고: 기타 다이어그램은 수동 정의하는 채널의 수신 측만을 표시합니다. 송신 끝은 대부분 필요할 때 자동으로 정의되므로 생략되었습니다. 대부분의 클러스터 송신자 채널의 자동 정의는 클러스터의 기능 및 효율에 결정적 요소입니다.

관련 개념

클러스터

클러스터에서 큐 관리자를 그룹화할 수 있습니다. 클러스터에 있는 큐 관리자는 호스트하는 큐를 클러스터에 있는 다른 큐 관리자에 대해서도 사용 가능하도록 작성할 수 있습니다. 모든 큐 관리자는 표준 분산 큐잉에 필요한 여러 오브젝트 정의 없이도 동일한 클러스터 내에 있는 다른 큐 관리자에게 메시지를 송신할 수 있습니다.

클러스터링과 분산 큐잉의 비교

클러스터의 컴포넌트

관련 태스크

큐 관리자 클러스터 구성

새 클러스터 설정

WebSphere MQ 클러스터 관리

IBM WebSphere MQ Telemetry

IBM WebSphere MQ Telemetry는 큐 관리자의 일부인 텔레메트리(MQXR) 서비스, 직접 작성하거나 제공되는 클라이언트 중 하나를 사용할 수 있는 텔레메트리 클라이언트, 명령행 및 탐색기 관리 인터페이스로 구성됩니다. 텔레메트리는 다양한 원격 장치에서 데이터를 수집하고 이런 장치를 관리하는 것을 말합니다. IBM WebSphere MQ Telemetry를 사용하면 데이터 수집과 장치 제어를 웹 애플리케이션으로 통합할 수 있습니다.

MQTT 지원은 WebSphere MQ Telemetry가 별도의 기능이던 WebSphere Message Broker 또는 WebSphere MQ 버전 7.0.1에서 이전에 사용 가능했습니다. WebSphere MQ Telemetry는 WebSphere MQ 버전 7.1 이상의 컴포넌트이므로 업그레이드 시 기본적으로 WebSphere MQ Telemetry 버전 7.0.1이 설치 제거되고

WebSphere MQ 버전 7.1이 설치됩니다. WebSphere MQ Telemetry는 기본 제품과 함께 설치되거나, 버전 7.1 이상이 이미 설치된 이후 설치될 수도 있습니다. 마이그레이션 정보는 [Version 7.0.1 - Version 7.5 에서 IBM WebSphere MQ Telemetry 이주](#) 또는 [Migration of telemetry applications from using WebSphere Message Broker version 6 to use IBM WebSphere MQ Telemetry and WebSphere Message Broker version 7.0](#)를 참조하십시오.

IBM WebSphere MQ Telemetry에는 다음 컴포넌트가 포함되어 있습니다.

텔레메트리 채널

MQTT 클라이언트와 IBM WebSphere MQ 사이의 연결을 관리하려면 텔레메트리 채널을 사용하십시오. 텔레메트리 채널은 SYSTEM.MQTT.TRANSMIT.QUEUE와 같은 새 IBM WebSphere MQ 오브젝트를 사용하여 IBM WebSphere MQ와 상호작용합니다.

텔레메트리(MQXR) 서비스

MQTT 클라이언트는 텔레메트리 채널과 연결하는 데 SYSTEM.MQXR.SERVICE 텔레메트리 서비스를 사용합니다.

IBM WebSphere MQ Explorer 지원 IBM WebSphere MQ Telemetry

IBM WebSphere MQ Telemetry 는 IBM WebSphere MQ Explorer를 사용하여 관리할 수 있습니다.

클라이언트 소프트웨어 개발 키트(SDK, software development kit)

클라이언트 SDK에는 다음과 같은 네 가지 부분이 있습니다.

1. Java SE 및 Java ME를 위한 MQTT v3 클라이언트 라이브러리. Java SE 또는 Java ME를 지원하는 디바이스를 위한 Java 클라이언트를 쓰려면 Java 라이브러리를 사용하십시오.
2. C의 경우 v3 라이브러리 C 라이브러리를 사용하여 다수의 플랫폼에 C 클라이언트를 작성한다.
3. 다수의 플랫폼에서 실행되는, C로 쓰여진 고급 클라이언트인 IBM WebSphere MQ Telemetry daemon for devices.
4. MQTT v3 프로토콜. MQTT v3 프로토콜이 발행되었으며 재사용 라이선스가 있습니다. 다양한 플랫폼과 언어를 위한 MQTT 클라이언트를 쓰려면 프로토콜과 참조 MQTT 클라이언트 구현을 사용하십시오.

Documentation

IBM WebSphere MQ Telemetry 문서는 Version 7.1의 표준 IBM WebSphere MQ 제품 문서에 포함되어 있습니다. Java와 C 클라이언트를 위한 SDK 문서는 Javadoc 및 HTML 형식으로 제품 문서에서 제공됩니다.

Telemetry 개념

우리는 주변 환경에서 정보를 수집하여 무엇을 할 지 결정합니다. 소비자로서 음식을 구입하기 전에 가게에 어떤 물건이 있는지 확인합니다. 환승 예약을 하기 전에 지금 떠나면 얼마나 오래 걸릴 지 알고 싶어합니다. 의사를 방문할까 결정하기 전에 자신의 증상을 확인합니다. 좀 더 기다릴까 정하기 전에 버스가 언제 도착할지 확인합니다. 이러한 결정을 내리게 되는 근거가 되는 정보는 측정기와 디바이스, 문서와 화면, 그리고 우리에게서 나옵니다. 어디에 있든, 필요할 때면 언제나 우리는 정보를 수집하고 종합하여 이를 토대로 행동합니다.

정보의 소스가 넓게 분산되어 있거나 액세스할 수 없으면 가장 정확한 정보를 모으기 어려우며 비용이 많이 들게 됩니다. 큰 변화를 가져오려 하거나 변화를 만들기 어려울 경우 변화시키지 못한 것이 생기거나 덜 효과적일 때 변화가 일어나게 됩니다.

넓게 분산되어 있는 디바이스를 디지털 기술로 인터넷에 연결시켜 이를 제어하고 여기서 정보를 수집하는 비용이 획기적으로 줄어들 수 있다면 어떠십니까? 이 정보는 인터넷과 기업의 자원을 사용하여 분석할 수 있습니다. 정보에 기반하여 결정을 내리고 행동을 취할 수 있는 기회가 더 많아집니다.

기술 트렌드, 환경 및 경제적 요구로 인해 다음과 같은 변화가 일어나고 있습니다.

1. 표준화와 저비용 디지털 프로세서와의 연결을 통해 센서와 작동기를 연결하고 제어하는 비용이 줄어들고 있습니다.
2. 디바이스를 연결하는 데 인터넷과 인터넷 기술이 점점 더 많이 사용되고 있습니다. 몇몇 나라에서는 인터넷 애플리케이션에 대한 연결 수에서 휴대전화가 개인용 컴퓨터를 앞서고 있습니다. 다른 디바이스들도 곧 이를 따르게 될 것입니다.
3. 인터넷과 인터넷 기술이 애플리케이션이 훨씬 쉽게 데이터를 얻게 해 줍니다. 쉽게 데이터에 액세스할 수 있게 되면서 센서에서 얻은 데이터를 많은 솔루션에 유용한 정보로 바꿔 주는 데이터 분석의 사용이 크게 늘고 있습니다.
4. 자원을 지능적으로 사용하는 것이 탄소 배출량과 비용을 절감하는 더 빠르고 저렴한 방법입니다. 새 자원을 찾거나 기존 자원을 사용하는 새 기술을 개발하는 것과 같은 대안 솔루션은 오랜 기간이 걸릴 수 있습니다. 단 기간에 새 기술을 개발하거나 새 자원을 찾는 것은 기존 솔루션을 개선하는 것보다 종종 더 위험하고, 느리며 비용이 많이 듭니다.

예

다음 예는 어떻게 이러한 트렌드가 환경과 지능적으로 상호작용할 수 있는 새 기회를 제공하는지 보여줍니다.

국제해상안전협약(International Convention for the Safety of Life at Sea)은 많은 배에 자동 식별 시스템(AIS, automatic identification system)을 배치하도록 요구합니다. 300톤이 넘는 상선과 여객선에는 필수 사항입니다. AIS는 기본적으로 연안 해운의 충돌 회피 시스템입니다. 해군 당국에서 연안을 모니터링하고 제어하는 데에도 사용됩니다.

전 세계의 열정적인 사람들이 저비용 AIS 추적 기지를 배치하고 연안 해운 정보를 인터넷에 올립니다. AIS의 정보를 인터넷의 다른 정보와 결합시키는 애플리케이션을 작성하는 사람도 있습니다. 결과는 웹 사이트에 게재되며 Twitter와 SMS를 통해 발행됩니다.

한 애플리케이션에서 사우스햄턴 가까이 있는 AIS 기지의 정보가 배 소유주 및 지리적 정보와 결합됩니다. 이 애플리케이션은 선박 도착 및 출발에 대한 실시간 정보를 Twitter에 제공합니다. 사우스햄턴과 아일랜드 화이트 사이에서 연락선을 사용하는 정기 통근자들은 Twitter 또는 SMS를 사용하여 새 피드를 구독합니다. 피드에 연락선이 느리게 운행 중이라고 표시되면 통근자들은 출발을 늦추고 연락선이 예정된 도착 시간보다 늦게 정박하면 연락선을 탈 수 있습니다.

다른 예는 64 페이지의 『모니터링 및 제어를 위한 텔레메트리 개념과 시나리오』의 내용을 참조하십시오.

관련 태스크

[WebSphere MQ Telemetry 설치](#)

[WebSphere MQ Telemetry 관리](#)

[WebSphere Message Broker 버전 6 사용에서 WebSphere MQ Telemetry 및 WebSphere Message Broker 버전 7.0 사용으로 텔레메트리 애플리케이션의 마이그레이션](#)

[버전 7.0.1에서 버전 7.5로 WebSphere MQ Telemetry 마이그레이션](#)

[WebSphere MQ Telemetry용 애플리케이션 개발](#)

[WebSphere MQ Telemetry의 문제점 해결](#)

관련 참조

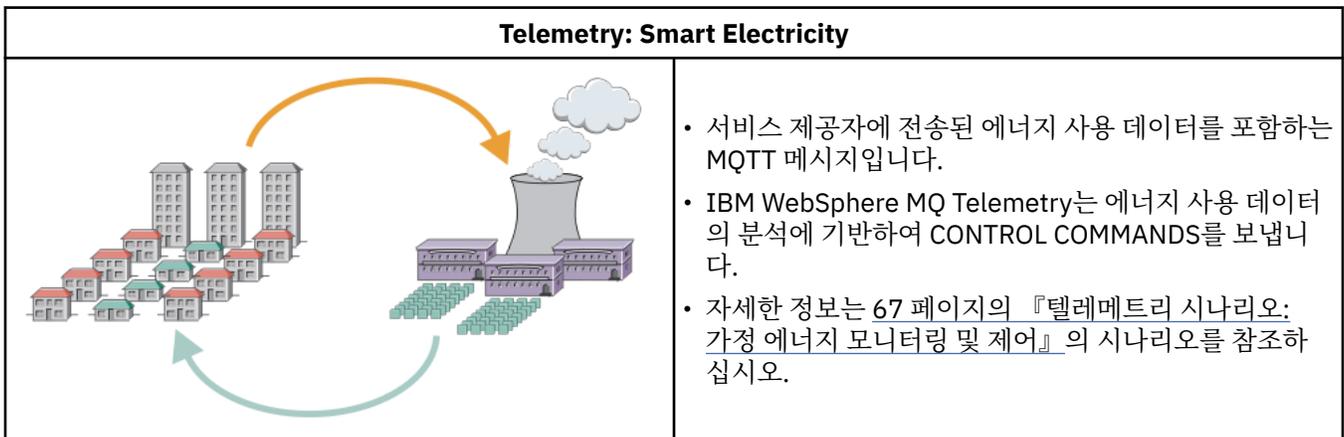
[WebSphere MQ Telemetry 참조](#)

모니터링 및 제어를 위한 텔레메트리 개념과 시나리오

텔레메트리는 자동화된 감지, 데이터 측정 및 원격 디바이스의 제어를 가리킵니다. 주안점은 디바이스에서 중앙 제어 지점으로의 데이터 전송입니다. 또한 텔레메트리는 디바이스로 구성 및 제어 정보를 송신하는 것을 포함합니다.

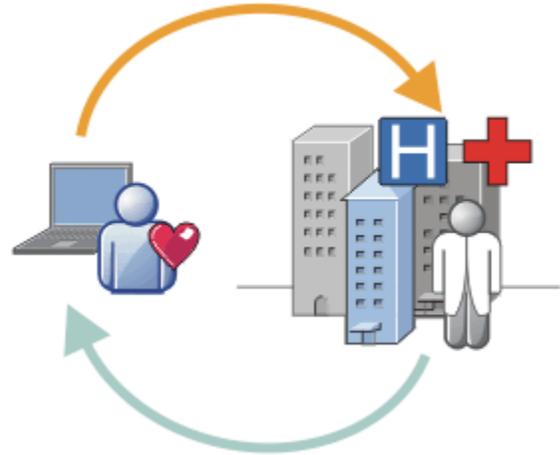
IBM WebSphere MQ Telemetry는 MQTT 프로토콜을 사용하는 소형 디바이스를 연결하고 디바이스를 IBM WebSphere MQ를 사용하는 다른 애플리케이션으로 연결합니다. IBM WebSphere MQ Telemetry 장치와 인터넷 사이의 간극을 매우며 "스마트 솔루션"을 쉽게 구축할 수 있다. 스마트 솔루션은 디바이스를 모니터링하고 제어하는 애플리케이션을 위해 인터넷과 엔터프라이즈 애플리케이션으로부터 사용 가능한 풍부한 정보를 제공한다.

다음 다이어그램은 IBM WebSphere MQ Telemetry의 일반적인 사용을 보여줍니다.

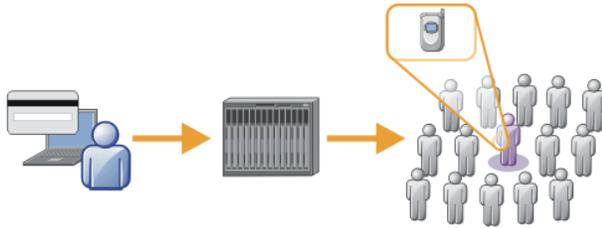


Telemetry: Smart Health Services

- IBM WebSphere MQ Telemetry 는 건강 데이터를 병원 및 의사 닥터에 보냅니다.
- MQTT 메시지 경보 또는 피드백은 의료 데이터의 분석을 기초로 송신될 수 있습니다.
- 자세한 정보는 65 페이지의 『텔레메트리 시나리오: 가정 환자 모니터링』의 시나리오를 참조하십시오.



Telemetry: 수천 명의 사람들 중 한 사람 식별



- 단순 카드 트랜잭션이 은행의 서버에 전송됩니다.
- IBM WebSphere MQ Telemetry는 수천 명의 사람들 중에서 한 사람을 식별하여 카드를 사용한 고객을 알려줍니다.
- IBM WebSphere MQ Telemetry는 가장 간단한 정보 입력을 사용하여 해당 개인을 찾습니다.

실제 사례를 참고해 만들어진 후속 시나리오는 텔레메트리를 사용하는 몇 가지 방법과 텔레메트리 기술이 해결해야 할 공통된 문제점을 설명합니다.

관련 개념

65 페이지의 『텔레메트리 시나리오: 가정 환자 모니터링』

심장병 환자 간호 시스템에서의 IBM과 보건 의료 서비스 제공자와의 협업에서는 이식된 제세동기가 병원과 통신합니다. 환자와 이식된 디바이스에 대한 데이터는 RF 텔레메트리를 이용해 환자의 가정에 있는 MQTT 디바이스로 전송됩니다.

67 페이지의 『텔레메트리 시나리오: 가정 에너지 모니터링 및 제어』

68 페이지의 『텔레메트리 시나리오: 라디오 주파수 식별(RFID, radio frequency identification)』

69 페이지의 『텔레메트리 시나리오: 환경 감지』

환경 감지는 강물의 깊이와 수질, 대기 오염원 및 기타 환경 데이터를 수집하는 데 텔레메트리를 사용합니다.

69 페이지의 『텔레메트리 시나리오: 모바일 애플리케이션』

모바일 애플리케이션은 무선 디바이스에서 실행되는 애플리케이션입니다. 이 디바이스는 일반 애플리케이션 플랫폼이거나 특수 디바이스입니다.

텔레메트리 시나리오: 가정 환자 모니터링

심장병 환자 간호 시스템에서의 IBM과 보건 의료 서비스 제공자와의 협업에서는 이식된 제세동기가 병원과 통신합니다. 환자와 이식된 디바이스에 대한 데이터는 RF 텔레메트리를 이용해 환자의 가정에 있는 MQTT 디바이스로 전송됩니다.

일반적으로 전송은 병상에 설치된 송신기에서 밤에 진행됩니다. 송신기는 전화 시스템을 이용해 데이터를 안전하게 병원으로 전송하며, 병원에서 이 데이터를 분석합니다.

이 시스템은 환자가 의사를 만나야 하는 횟수를 줄여 줍니다. 환자나 디바이스에 주의가 필요할 때를 감지하며 응급시에는 통화 가능한 의사에게 알립니다.

IBM과 보건 의료 서비스 제공자 사이의 협업은 다수의 텔레메트리 시나리오에 공통되는 다음과 같은 몇 가지 특징이 있습니다.

투명성

디바이스는 전원, 전화 회선을 공급하는 것과 디바이스의 범위 안에 있을 것을 제외하고 사용자 개입을 필요로 하지 않습니다. 조작은 안정적이며 사용하기 간단합니다.

환자가 디바이스를 설정해야 할 필요를 없애기 위해 디바이스 공급자는 디바이스를 사전 구성합니다. 환자는 전원을 연결하기만 하면 됩니다. 환자가 직접 구성하지 않도록 함으로써 조작이 간편해지며 디바이스가 잘못 구성될 확률이 줄어듭니다.

MQTT 클라이언트가 디바이스의 한 부분으로서 임베드되어 있습니다. 디바이스 개발자는 MQTT 클라이언트 구현을 디바이스에 임베드하며 개발자 또는 공급자가 사전 구성의 일부로서 MQTT 클라이언트를 구성합니다.

MQTT 클라이언트는 Java SE and Java ME jar 파일로 제공되며 개발자가 Java 애플리케이션에 포함합니다. 이 디바이스와 같은 비Java 환경에서 디바이스 개발자는 발행된 MQTT 포맷 및 프로토콜을 이용하여 다른 언어로 클라이언트를 구현할 수 있습니다. 또는 개발자는 Linux 및 ARM 플랫폼용 공유 라이브러리로 제공되는 C 클라이언트 중 하나를 사용할 수 있습니다.

불규칙한 연결

제세동기와 병원 사이의 통신에는 불규칙한 네트워크 특성이 있습니다. 환자에게서 데이터를 수집하고, 데이터를 병원으로 송신하는 서로 다른 문제를 해결하기 위해 두 개의 서로 다른 네트워크가 사용됩니다. 환자와 MQTT 디바이스 사이에는 단거리 저전력 RF 네트워크가 사용됩니다. 송신기는 VPN TCP/IP 연결을 사용하여 낮은 대역폭 전화 회선으로 병원과 연결합니다.

모든 디바이스를 인터넷 프로토콜 네트워크와 직접 연결하는 방법을 찾는 것은 보통 실용적이지 않습니다. 허브로 연결된 두 네트워크를 사용하는 것이 보통 솔루션입니다. MQTT 디바이스는 간단한 허브로, 환자에게서 정보를 저장하고 이를 병원으로 전달합니다.

보안

의사는 환자 데이터의 신뢰성을 믿을 수 있어야 하며 환자는 개인정보인 자신의 데이터가 보호되기를 바랍니다.

몇몇 시나리오에서는 VPN 또는 SSL을 사용하여 연결을 암호화하는 것으로 충분합니다. 다른 시나리오에서는 데이터가 저장된 후에도 이를 보호하는 것이 더 좋은 경우도 있습니다.

텔레메트리 디바이스가 안전하지 않은 경우도 있습니다. 예를 들면, 공동 주택에 있거나 하는 경우입니다. 데이터가 올바른 환자의 것임을 보장할 수 있도록 디바이스 사용자는 반드시 인증을 받아야 합니다. 디바이스 자체도 SSL을 사용하여 서버의 인증을 받을 수 있고, 서버 또한 디바이스의 인증을 받을 수 있습니다.

디바이스와 큐 관리자 사이의 텔레메트리 채널은 사용자 인증을 위해 JAAS를 지원하며 통신 암호화와 디바이스 인증을 위해 SSL을 지원합니다. 발행에 대한 액세스는 WebSphere MQ의 오브젝트 권한 관리자에 의해 제어됩니다.

사용자 인증에 사용된 ID는 공용 환자 ID와 같은 다른 ID에 맵핑될 수 있습니다. 공통 ID는 WebSphere MQ의 발행 토픽 구성을 단순화시켜줍니다.

연결성

MQTT 디바이스와 병원 사이의 연결은 전화 접속을 사용하며 1초에 300보오 정도의 낮은 대역폭으로 작동합니다.

300 보오에서 효율적으로 작업하기 위해서 MQTT 프로토콜은 TCP/IP 헤더 외에 추가 바이트를 메시지에 조금 추가합니다.

MQTT 프로토콜은 단일 전송 "전송 후 삭제" 메시징을 제공하며 이는 대기 시간을 낮게 유지시켜 줍니다. 또한 보장된 전달이 응답 시간보다 더 중요한 경우에는 "적어도 한 번" 및 "정확히 한 번" 전달을 보장하기 위해 다중 전송을 사용할 수 있습니다. 전달을 보장하기 위해 메시지는 전달이 완료될 때까지 디바이스에 저장됩니다. 디바이스가 무선으로 연결되어 있을 경우 보장된 전달은 특히 유용합니다.

확장성

텔레메트리 디바이스는 보통 수만에서 수백만에 이르는 규모로 많은 수가 배치됩니다.

시스템에 많은 디바이스를 연결할 경우 많은 솔루션이 필요하게 됩니다. 디바이스와 그 소프트웨어에 대한 비용과 같은 비즈니스적 요구, 라이선스, 디바이스와 사용자를 관리하는 관리 요구가 이에 해당합니다. 기술적 요구는 네트워크나 서버의 부하를 포함합니다.

연결을 여는 것은 열린 연결을 유지하는 것보다 많은 서버 자원을 사용합니다. 하지만 전화 회선을 사용하는 이와 같은 시나리오에서 연결 비용은 필요할 때를 제외하고 연결이 열려 있지 않다는 것을 의미합니다. 데이터 전송은 대부분 일괄 처리로 이뤄집니다. 연결은 취침 시간에 갑자기 증가하는 연결을 피하기 위해 심야에 스케줄될 수 있습니다.

클라이언트 측면에서 클라이언트의 확장성은 필요한 클라이언트 구성이 적을수록 향상됩니다. MQTT 클라이언트는 디바이스 안에 임베드되어 있습니다. 디바이스를 환자에 배치하는 데 있어서 구성이나 MQTT 클라이언트 라이선스 수락 단계와 같은 요구사항은 없습니다.

서버 측면에서 WebSphere MQ Telemetry에는 큐 관리자 당 50,000개의 열린 연결이라는 초기 목표가 있습니다.

연결은 WebSphere MQ Explorer를 사용하여 관리됩니다. 탐색기는 관리 가능한 숫자만을 표시하도록 연결을 필터합니다. 적절하게 선택된, 클라이언트에 대한 ID 할당 설계를 사용하면 지정학적으로, 또는 환자 이름의 영문자 순서대로 연결을 필터할 수도 있습니다.

텔레메트리 시나리오: 가정 에너지 모니터링 및 제어

스마트 측정기는 기존 측정기보다 에너지 소비에 대한 세부 사항을 더 많이 수집합니다.

스마트 측정기는 가정의 개별 어플라이언스를 모니터 및 제어하기 위해 보통 로컬 텔레메트리 네트워크와 결합됩니다. 몇몇은 원거리에서 모니터링과 제어를 할 수 있도록 원격으로도 연결됩니다.

원격 연결은 개별 기기, 전원 유틸리티 및 중앙 제어 지점에 의해 설정될 수 있습니다. 원격 제어 지점은 전력을 사용하고 사용 데이터를 제공할 수 있습니다. 이는 연속적인 가격 설정이나 날씨 정보에 영향을 줄 수 있는 데이터를 제공할 수 있습니다. 전체적인 전력 생산 효율을 높이기 위해 부하를 제한할 수도 있습니다.

스마트 측정기는 점점 더 널리 보급되고 있습니다. 예를 들면, 영국 정부는 2020년까지 모든 국내 가정에 스마트 측정기를 보급하는 것에 대해 협의 중에 있습니다.

가정 측정 시나리오에는 다음과 같은 몇 가지 공통되는 특성이 있습니다.

투명성

사용자가 측정기를 사용하여 에너지를 절약하고자 하지 않는 이상 측정기는 사용자 개입을 필요로 하지 않습니다. 이는 개별 어플라이언스에 대한 에너지 공급 안정성을 저하시키지 않아야 합니다.

MQTT 클라이언트는 측정기와 함께 설치된 소프트웨어에 임베드될 수 있으며 별도의 설치나 구성을 필요로 하지 않습니다.

불규칙한 연결

어플라이언스와 스마트 측정기는 측정기와 원격 연결 지점 사이의 연결과는 다른 연결성 기준을 요구합니다.

스마트 측정기에서 어플라이언스로의 연결은 가용성이 높아야 하며 홈 네트워크의 네트워크 기준에 부합해야 합니다.

원격 네트워크는 다양한 실제 접속을 사용할 가능성이 높습니다. 이 중 무선과 같은 몇몇은 전송 비용이 높으며 단속적일 수 있습니다. MQTT v3 스펙은 원격 연결 및 로컬 어댑터와 스마트 측정기 사이의 연결을 주요 대상으로 하고 있습니다.

전원 콘센트와 어플라이언스, 그리고 측정기 사이의 연결은 Zigbee와 같은 홈 네트워크를 사용합니다. 센서 네트워크용 MQTT(MQTT-S)는 Zigbee 및 기타 낮은 대역폭 네트워크 프로토콜과 작업하도록 설계되었습니다. WebSphere MQ Telemetry는 직접 MQTT-S를 지원하지는 않습니다. MQTT-S를 MQTT v3에 연결하는 데는 게이트웨이가 필요합니다.

가정 환자 모니터링과 마찬가지로 가정 에너지 모니터링 및 제어에 대한 솔루션은 다중 네트워크를 필요로 하며 이는 스마트 측정기를 허브로 하여 연결되어 있어야 합니다.

보안

스마트 측정기와 연관된 보안 문제에는 몇 가지 있습니다. 이 문제에는 트랜잭션 부인 방지, 시작된 모든 제어 조치의 권한 부여와 전력 소비 데이터의 개인정보 보호가 포함되어 있습니다.

확실한 개인정보 보호를 위해 측정기와 원격 제어 지점 사이에 MQTT로 전송된 데이터는 SSL을 사용해 암호화될 수 있습니다. 제어 조치에 대한 확실한 권한 부여를 위해 측정기와 원격 제어 지점 사이의 MQTT 연결은 SSL을 사용해 상호 인증하도록 할 수 있습니다.

연결성

원격 네트워크의 실제 환경은 매우 다양할 수 있습니다. 기존 광대역 연결을 사용하거나 높은 비용을 지불하며 모바일 네트워크를 사용할 수도 있고, 가용성은 단속적일 수 있습니다. 높은 비용, 간헐적 연결의 경우, 연결은 효율적이고 신뢰할 수 있는 프로토콜입니다. 65 페이지의 『텔레메트리 시나리오: 가정 환자 모니터링』을 참조하십시오.

확장성

전력 회사나 중앙 제어 지점은 언젠가 수천만 개의 스마트 측정기를 설치하려고 계획하고 있습니다. 처음 배치 당 측정기의 수는 수만에서 수십만 단위입니다. 이 숫자는 초기 MQTT 목표였던 큐 관리자 당 50,000개의 열린 클라이언트 연결과 비슷합니다.

가정 에너지 모니터링 및 제어의 아키텍처가 가진 중요한 면은 스마트 측정기를 네트워크 집선기로서 사용한다는 점입니다. 각 어플라이언트 어댑터는 별도의 센서입니다. 이들을 MQTT를 사용해 로컬 허브에 연결함으로써 허브는 중앙 제어 지점과의 단일 TCP/IP 세션으로 데이터 흐름을 집중시킬 수 있으며 세션 정전 시를 대비해 짧은 시간 동안 메시지를 저장할 수도 있습니다.

가정 에너지 시나리오에서 원격 연결은 두 가지 이유로 인해 열려있어야 합니다. 첫째, 연결을 여는 것은 요청을 송신하는 것과 비교해 시간이 오래 걸리기 때문입니다. 짧은 간격 안에 "부하 제한" 요청을 보내기 위해 많은 연결을 여는 데는 시간이 너무 오래 걸립니다. 둘째, 전력 회사로부터 부하 제한 요청을 수신하려면 클라이언트로부터 먼저 연결이 열려야 합니다. MQTT를 사용하면 연결은 항상 클라이언트에 의해 시작되며 전력 회사로부터 부하 제한 요청을 수신하기 위해서 연결은 열린 채로 남겨져 있어야 합니다.

연결 열기 등급이 중요이거나 서버가 시간에 쫓기는 요청을 시작할 경우 보통 솔루션은 많은 열린 연결을 유지하는 것입니다.

텔레메트리 시나리오: 라디오 주파수 식별(RFID, radio frequency identification)

RFID는 오브젝트를 무선으로 식별하고 추적하는 임베드된 RFID 태그 사용법입니다. RFID 태그는 수 미터 범위 안에서, 태그와 RFID 리더 사이에 물체가 있더라도 읽을 수 있습니다. 수동 태그는 RFID 리더에 의해 활성화됩니다. 활성 태그는 외부에서 작동시키지 않더라도 전송됩니다. 활성 태그에는 전원이 있어야 합니다. 수동 태그는 그 범위를 늘리기 위해 전원이 있을 수 있습니다.

RFID는 많은 애플리케이션에서 사용되며 수많은 유형의 시나리오가 있습니다. RFID 시나리오에는 가정 환자 모니터링, 가정 에너지 모니터링 및 제어 시나리오와 몇몇 유사한 점 및 다른 점이 있습니다.

투명성

많은 시나리오에서 RFID 리더는 다수 배치되며 사용자 개입 없이 작동해야 합니다. 리더는 중앙 제어 지점과의 통신을 위한 임베드된 MQTT 클라이언트를 포함하고 있습니다.

예를 들면, 유통 창고에서 리더는 팔릿을 감지하는 데 동작 센서를 사용합니다. 이는 팔릿 위에 놓인 물건의 RFID 태그를 활성화시키며 데이터와 요청을 중앙 애플리케이션으로 송신합니다. 이 데이터는 재고의 위치를 업데이트하는 데 사용됩니다. 요청은 팔릿에 수행될 다음 작업(예: 특정 공간으로 이를 이동시키는 등)을 제어합니다. 항공사나 공항 수화물 시스템은 이런 방식으로 RFID를 사용합니다.

일부 RFID 시나리오에서, 독자는 자바 ME와 같은 표준 컴퓨팅 환경을 갖는다. 이런 경우 MQTT 클라이언트는 제조 후 특정한 구성 단계를 거쳐 배치될 수 있습니다.

불규칙한 연결

RFID 리더는 MQTT 클라이언트를 포함하고 있는 로컬 제어 디바이스와 분리될 수 있으며, 리더가 MQTT 클라이언트를 임베드할 수도 있습니다. 일반적으로 지형적 또는 통신상의 요인이 토폴로지를 결정하게 됩니다.

보안

RFID 태그를 사용에 있어서 개인정보 보호 및 신뢰성은 주요 보안 대상입니다. RFID 태그는 눈에 잘 띄지 않으며 비밀리에 모니터, 위조나 도용의 대상이 될 수 있습니다.

RFID 보안 문제에 대한 솔루션은 새 RFID 솔루션이 배치될 기회를 증가시킵니다. 그러나 보안 취약점은 주로 RFID 태그에 있으며, 중앙 정보 처리를 사용하는 로컬 리더는 다양한 위협을 제거할 방법을 제시해 줍니다. 예를 들면, 태그 도용은 입고와 출고에 대해 동적으로 상호 연관된 재고량을 통해 발견될 수 있습니다.

연결성

RFID 애플리케이션은 보통 RFID 리더에서 수집한 일괄 처리되는 저장 후 전달 정보, 그리고 즉각적 조회를 모두 포함하고 있습니다. 유통 창고 시나리오에서 RFID 리더는 항상 연결되어 있습니다. 태그가 읽히면 이는 리더에 관한 정보와 함께 발행됩니다. 창고 애플리케이션은 리더에게 응답을 발행합니다.

창고 애플리케이션에서 네트워크는 일반적으로 안정적이며 즉각적 요청은 낮은 지연시간 성능을 위해 "전송 후 삭제" 메시지를 사용할 수 있습니다. 일괄 처리되는 저장 후 전달 데이터는 데이터 삭제와 연관된 관리 비용을 최소화하기 위해 "정확히 한 번" 메시지를 사용할 수 있습니다.

확장성

RFID 애플리케이션이 1 - 2초 정도 소요되는 즉각적 응답을 필요로 할 경우 RFID 리더는 계속 연결되어 있어야 합니다.

텔레메트리 시나리오: 환경 감지

환경 감지는 강물의 깊이와 수질, 대기 오염원 및 기타 환경 데이터를 수집하는 데 텔레메트리를 사용합니다.

센서는 보통 외진 곳에, 유선 통신에 대한 액세스 없이 설치됩니다. 무선 대역폭은 비경제적이며 안정성은 낮을 수 있습니다. 작은 공간에 있는 다수의 환경 센서가 안전한 장소의 로컬 모니터링 디바이스에 연결되어 있는 것이 전형입니다. 로컬 연결은 유선 또는 무선입니다.

투명성

센서 디바이스는 보통 중앙 모니터링 디바이스보다 액세스하기 어렵고, 전력을 덜 사용하며 다수 배치됩니다. 센서는 때때로 "조용하며" 로컬 모니터링 디바이스는 센서 데이터를 변환하고 저장하는 어댑터를 포함하고 있습니다. 모니터링 디바이스는 Java SE 또는 ME를 지원하는 범용 컴퓨터를 포함할 가능성이 있다. 투명성은 MQTT 클라이언트를 구성하는 데 주된 요구사항은 아닙니다.

불규칙한 연결

센서의 기능, 원격 연결의 비용과 대역폭 때문에 보통 로컬 모니터링 허브는 중앙 서버에 연결됩니다.

보안

솔루션이 군 또는 방어 시나리오에서 사용되는 것이 아닐 경우 보안은 주요 요구사항이 아닙니다.

연결성

대부분의 경우 연속적 모니터링이나 데이터의 즉시 가용성이 필요하지 않습니다. 홍수 레벨 경보와 같은 예외 데이터는 즉시 전달되어야 할 필요가 있습니다. 센서 데이터는 연결 및 통신 비용 절감을 위해 로컬 모니터에서 집계되어 스케줄된 연결을 사용하여 전송됩니다. 예외 데이터는 모니터에서 감지되는 즉시 전달됩니다.

확장성

센서는 로컬 허브에 집중되어 있으며 센서 데이터는 스케줄에 따라 전송되는 패킷에 집계됩니다. 이 두 요소가 직접 연결된 센서를 사용할 때 발생할 수 있는 중앙 서버의 부담을 덜어줍니다.

텔레메트리 시나리오: 모바일 애플리케이션

모바일 애플리케이션은 무선 디바이스에서 실행되는 애플리케이션입니다. 이 디바이스는 일반 애플리케이션 플랫폼이거나 특수 디바이스입니다.

일반 플랫폼에는 휴대전화나 PDA와 같은 소형 디바이스나 노트북 컴퓨터와 같은 휴대용 디바이스가 포함됩니다. 특수 디바이스는 특정 애플리케이션에 맞춰진 특수 목적 하드웨어를 사용합니다. 특수 모바일 디바이스의 한 예로는 "전달 확인" 소포 전달을 기록하는 디바이스가 있습니다. 특수 모바일 디바이스의 애플리케이션은 보통 일반 소프트웨어 플랫폼에서 빌드됩니다.

투명성

특수 모바일 애플리케이션의 배치가 관리되며 MQTT 클라이언트 애플리케이션 구성을 포함할 수 있습니다. 투명성은 MQTT 클라이언트를 구성하는 데 주된 요구사항은 아닙니다.

불규칙한 연결

선형 시나리오들의 로컬 허브 토폴로지와는 달리 모바일 클라이언트는 원격으로 연결합니다. 클라이언트 애플리케이션 계층은 중앙 허브의 애플리케이션으로 직접 연결합니다.

보안

물리적 보안성이 높지 않기 때문에 모바일 디바이스와 모바일 사용자는 인증을 받아야 합니다. 디바이스 ID를 확인하는 데는 SSL이, 사용자를 인증하는 데는 JAAS가 사용됩니다.

연결성

모바일 애플리케이션이 무선 범위에 종속적일 경우 이는 오프라인으로 조작이 가능하며 인터럽트된 연결을 효율적으로 다룰 수 있어야 합니다. 이와 같은 환경에서 목표는 연결을 유지하는 동시에 애플리케이션에서 메시지를 저장하고 전달할 수 있게 하는 것입니다. 메시지는 종종 명령이나 전달 확인인 경우가 있으며 중요한 비즈니스 가치가 있습니다. 이들은 안정적으로 저장 및 전달되어야 합니다.

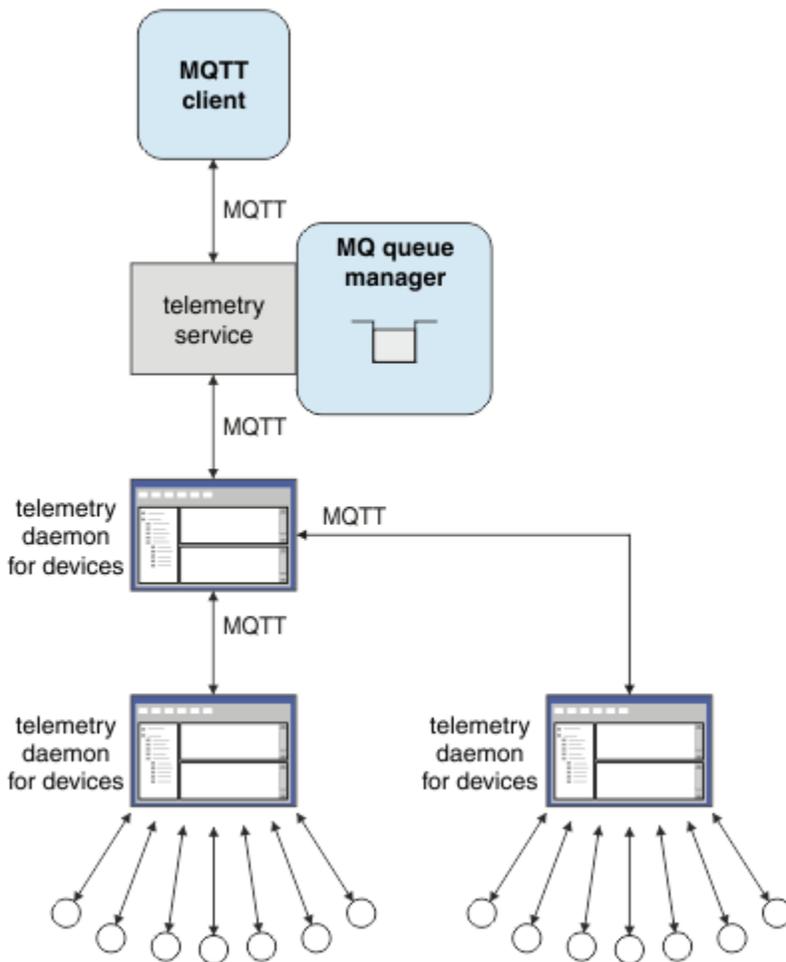
확장성

확장성은 주요 문제는 아닙니다. 특수 모바일 애플리케이션 시나리오에서 애플리케이션 클라이언트의 수는 천이나 만 단위를 넘어가지 않을 가능성이 큼니다.

텔레메트리 디바이스를 큐 관리자에 연결

텔레메트리 디바이스는 MQTT v3 클라이언트를 사용하여 큐 관리자에 연결합니다. MQTT v3 클라이언트는 TCP/IP를 사용하여 텔레메트리(MQXR) 서비스라고 하는 TCP/IP 리스너에 연결합니다.

텔레메트리 디바이스를 직접 텔레메트리 서비스에 연결하는 것에 대한 대안으로, 디바이스를 디바이스용 WebSphere MQ Telemetry 디먼에 연결할 수 있습니다. 디먼은 그 자체가 MQTT v3 클라이언트입니다. 이는 디바이스 연결을 풀링하며 텔레메트리(MQXR) 서비스에 대한 단일 연결을 작성합니다. 계층로 디먼을 연결하여 많은 크기 순서를 통해 IBM WebSphere MQ에 간접적으로 연결될 수 있는 디바이스의 수를 증가시킬 수 있습니다.



MQTT 클라이언트는 `MqttClient.connect` 메소드를 사용하여 TCP/IP 연결을 시작합니다. IBM WebSphere MQ 클라이언트와 마찬가지로 MQTT 클라이언트도 메시지를 송수신하려면 큐 관리자에 연결되어야 합니다.

IBM WebSphere MQ Telemetry와 함께 설치되며 텔레메트리(MQXR) 서비스라고 하는 TCP/IP 리스너를 사용하여 서버에서 연결이 이루어집니다. 각각의 큐 관리자는 최대 하나의 텔레메트리(MQXR) 서비스를 실행합니다.

텔레메트리(MQXR) 서비스는 텔레메트리 채널에 대한 연결을 할당하기 위해 `MqttClient.connect` 메소드에서 각 클라이언트가 설정한 원격 소켓 주소를 사용합니다. 소켓 주소는 TCP/IP 호스트 이름과 포트 번호의 조합입니다. 동일한 원격 소켓 주소를 사용하는 다수의 클라이언트는 텔레메트리(MQXR) 서비스에 의해 동일한 텔레메트리 채널에 연결됩니다.

서버에 다수의 큐 관리자가 있을 경우 큐 관리자 사이에 텔레메트리 채널을 분할하십시오. 큐 관리자 사이에 원격 소켓 주소를 할당하십시오. 각 텔레메트리 채널을 고유한 원격 소켓 주소로 정의하십시오. 두 텔레메트리 채널은 같은 소켓 주소를 사용해서는 안 됩니다.

다수의 큐 관리자에 텔레메트리 채널에 대해 같은 소켓 주소가 구성되어 있을 경우 처음 연결하는 텔레메트리 채널이 남게 됩니다. 같은 주소에 연결하는 후속 채널은 실패하며 첫 번째 오류 데이터 캡처(FFDC) 파일을 작성합니다.

서버에 다수의 네트워크 어댑터가 있을 경우 텔레메트리 채널 사이에 원격 소켓 주소를 분할하십시오. 하나의 특정 소켓 주소라도 유일한 텔레메트리 채널에 구성되어 있는 한 소켓 주소 할당은 완전히 임의로 이뤄집니다.

IBM WebSphere MQ Explorer의 WebSphere MQ Telemetry 공급에서 제공되는 마법사를 사용하여 MQTT 클라이언트를 연결하도록 IBM WebSphere MQ를 구성하십시오. 또는 수동으로 텔레메트리를 구성하려면 Linux 및 AIX에서 텔레메트리에 대해 큐 관리자 구성 및 Windows에서 텔레메트리에 대해 큐 관리자 구성의 지시사항을 따르십시오.

관련 참조

[MQXR 특성](#)

텔레메트리 연결 프로토콜

WebSphere MQ Telemetry는 TCP/IP IPv4, IPv6과 SSL을 지원합니다.

텔레메트리(MQXR) 서비스

텔레메트리(MQXR) 서비스는 TCP/IP 리스너이며, IBM WebSphere MQ 서비스로 관리됩니다. IBM WebSphere MQ Explorer 마법사나 `runmqsc` 명령을 사용해 서비스를 작성하십시오.

IBM WebSphere MQ 텔레메트리 (MQXR) 서비스는 `SYSTEM.MQXR.SERVICE`입니다.

IBM WebSphere MQ Explorer에 대한 IBM WebSphere MQ 텔레메트리 보충물에 제공된 **Telemetry sample configuration** 마법사는 텔레메트리 서비스 및 샘플 텔레메트리 채널을 작성합니다. [IBM WebSphere MQ Explorer 를 사용하여 IBM WebSphere MQ Telemetry 설치 확인](#) 을 참조하십시오. 명령행에서 샘플 구성을 작성하십시오. [명령행을 사용하여 IBM WebSphere MQ Telemetry 설치 확인](#) 을 참조하십시오.

텔레메트리(MQXR) 서비스는 큐 관리자에서 자동으로 시작되고 중지됩니다. IBM WebSphere MQ Explorer의 서비스 폴더를 사용하여 서비스를 제어하십시오. 서비스를 보려면 표시에서 SYSTEM 오브젝트를 필터하는 탐색기를 중지하기 위해 아이콘을 클릭해야 합니다.

`installMQXRService_unix.mqsc`는 AIX 및 Linux에서 수동으로 서비스를 작성하는 방법의 예를 표시합니다.

`installMQXRService_win.mqsc`는 Windows에서 수동으로 서비스를 작성하는 방법을 표시합니다.

텔레메트리 채널

Java 인증 및 권한 부여 서비스(JAAS, Java authentication and authorization service)나 SSL 인증과 같은 다양한 특성과 연결을 작성하기 위해서나 클라이언트 그룹을 관리하려면 텔레메트리 채널을 작성하십시오.

IBM WebSphere MQ Explorer에 대한 IBM WebSphere MQ Telemetry 보충물에 제공된 **New Telemetry Channel** 마법사를 사용하여 텔레메트리 채널을 작성하십시오. 특정 TCP/IP의 MQTT 클라이언트로부터 연결을 허용하려면 마법사를 사용하여 채널을 구성하십시오. Version 7.1이후에는 명령행 프로그램 `runmqsc`를 사용하여 IBM WebSphere MQ Telemetry 를 구성할 수 있습니다.

클라이언트를 그룹으로 나눠 대규모 클라이언트 연결을 관리하기 쉽도록 서로 다른 포트에 다중 텔레메트리 채널을 작성하십시오. 각 텔레메트리 채널은 서로 다른 이름을 가지고 있습니다.

다양한 유형의 연결을 작성하기 위해 다양한 보안 속성으로 텔레메트리 채널을 구성할 수 있습니다. 다양한 TCP/IP 주소로부터 클라이언트 연결을 수락하려면 다중 채널을 작성하십시오. SSL을 사용하면 메시지를 암호화하고 텔레메트리 채널 및 클라이언트를 인증할 수 있습니다. [MQTT 클라이언트 및 텔레메트리 채널의 SSL 구성](#)을 참조하십시오. WebSphere MQ 오브젝트에 대한 액세스 권한을 부여하는 작업을 단순화하려면 사용자 ID를 지정하십시오. JAAS 를 사용하여 MQTT 사용자를 인증하려면 JAAS 구성을 지정하십시오. [MQTT 클라이언트 식별, 권한 부여 및 인증을 참조하십시오.](#)

MQTT 프로토콜

MQ Telemetry Transport(MQTT) v3 프로토콜은 낮은 대역폭 또는 비경제적인 연결 상의 소형 디바이스 사이에서 메시지를 교환하고 안정적으로 메시지를 전송할 수 있도록 설계되었습니다. TCP/IP를 사용합니다.

MQTT 프로토콜이 발행되었습니다. [MQ Telemetry Transport 형식 및 프로토콜을 참조하십시오.](#) 프로토콜 버전 3은 발행/구독을 사용하며 다음과 같은 세 가지 서비스 품질을 제공합니다. "전송 후 삭제", "적어도 한 번", "정확히 한 번".

프로토콜 헤더의 작은 크기와 바이트 배열 메시지 페이로드로 인해 메시지 용량은 작습니다. 헤더는 2바이트의 고정된 헤더와 12바이트까지의 추가 변수 헤더를 포함합니다. 프로토콜은 구독 및 연결을 하는 데 12바이트의 변수 헤더를 사용하며 대부분의 발행에 대한 변수 헤더에는 2바이트만을 사용합니다.

세 가지 서비스 품질(QoS)을 사용하여 낮은 지연 시간과 안정성 사이에서 적절하게 균형을 유지할 수 있습니다. MQTT 클라이언트가 제공하는 서비스 품질(QoS)을 참조하십시오. "전송 후 삭제"는 지속적 디바이스 스토리지를 사용하지 않으며 발행물 송신 및 수신에 하나의 전송만을 사용합니다. "적어도 한 번"과 "정확히 한 번"에는 인식될 때까지 프로토콜 상태를 유지하고 메시지를 저장할 지속적 스토리지가 디바이스에 필요합니다.

프로토콜은 다른 제품에서 사용되는 MQTT 프로토콜 제품군 중 하나입니다.

MQTT 클라이언트

MQTT 클라이언트 애플리케이션은 Telemetry 디바이스에서 정보를 수집하고 서버에 연결하며 정보를 서버에 발행하는 역할을 합니다. 또한 토픽을 구독하고 서적을 수신하며 텔레메트리 디바이스를 제어할 수도 있습니다.

IBM WebSphere MQ 클라이언트 애플리케이션과 달리 MQTT 클라이언트 애플리케이션은 IBM WebSphere MQ 애플리케이션이 아닙니다. 이 클라이언트는 연결할 큐 관리자를 지정하지 않습니다. 특정 IBM WebSphere MQ 프로그래밍 인터페이스를 사용해야 하는 제약도 받지 않습니다. 대신에, MQTT 클라이언트는 MQTT version 3 프로토콜을 구현합니다. 프로그래밍 언어를 사용하여 선택한 플랫폼에서 MQTT 프로토콜에 대한 클라이언트 라이브러리의 인터페이스를 작성할 수 있습니다. [MQ Telemetry Transport 형식 및 프로토콜을 참조하십시오.](#)

MQTT 클라이언트 애플리케이션 작성을 단순화하려면 여러 플랫폼에 대해 MQTT 프로토콜을 캡슐화하는 C, Java, JavaScript 클라이언트 라이브러리를 사용하십시오. MQTT 클라이언트 라이브러리에 대한 클라이언트 API 문서에 대한 링크는 [MQTT 클라이언트 프로그래밍 참조](#)를 참조하십시오. 이러한 라이브러리를 MQTT 앱에 통합하면 완전한 기능을 갖춘 MQTT 클라이언트가 15행의 코드로 짧을 수 있습니다. [PubSync.java](#)를 참조하십시오.

com.ibm.micro.client.mqttv3.jar JAR 파일의 두 개의 사본이 설치됩니다. 하나의 사본은 파일 이름에 버전 번호가 포함됩니다. 예: com.ibm.micro.client.mqttv3_3.0.2.0-20100723.jar. OSGi 애플리케이션에서 버전 지정된 사본을 사용하십시오. JAR 파일의 콘텐츠는 동일합니다.

MQTT 클라이언트 애플리케이션은 항상 텔레메트리 채널로 연결을 시작합니다. 연결된 후 MQTT 클라이언트 애플리케이션이나 IBM WebSphere MQ 애플리케이션 모두에서 메시지 교환을 시작할 수 있습니다.

MQTT 클라이언트 애플리케이션과 IBM WebSphere MQ 애플리케이션은 동일한 토픽 세트에 발행 및 구독합니다. 또한 IBM WebSphere MQ 애플리케이션은 클라이언트 앱이 먼저 등록을 작성하지 않고 MQTT 클라이언트 앱에 직접 메시지를 보낼 수 있습니다. [MQTT 클라이언트에 메시지를 보내도록 분산 큐잉 구성을 참조하십시오.](#)

MQTT 클라이언트 애플리케이션은 Telemetry 채널을 사용하여 IBM WebSphere MQ에 연결됩니다. 텔레메트리 채널은 MQTT 및 IBM WebSphere MQ가 사용하는 서로 다른 유형의 메시지 사이에서 브릿지 역할을 수행합니다. 이는 MQTT 클라이언트 애플리케이션을 대신하여 큐 관리자에서 발행 및 구독을 작성합니다. 텔레메트리 채널은 MQTT 클라이언트 애플리케이션의 구독과 일치하는 발행을 큐 관리자에서 MQTT 클라이언트 애플리케이션으로 송신합니다.

MQTT 클라이언트에 메시지 송신

WebSphere MQ 애플리케이션은 클라이언트가 작성한 구독을 발행하거나 직접 메시지를 보내서 MQTT v3 클라이언트 메시지를 보낼 수 있습니다. MQTT 클라이언트는 다른 클라이언트가 subscribe한 토픽을 발행하는 방식으로 메시지를 다른 클라이언트에게 보낼 수 있습니다.

MQTT 클라이언트가 WebSphere MQ에서 수신한 발행을 구독함

WebSphere MQ에서 MQTT 클라이언트로 발행을 보내려면 75 페이지의 『[IBM WebSphere MQ Explorer 에서 MQTT 클라이언트 유틸리티에 메시지 게시](#)』 태스크를 수행하십시오.

MQTT v3 클라이언트가 메시지를 수신하는 일반적인 방법은 토픽 또는 토픽 세트에 구독을 작성하는 것입니다. 이 예제 코드 스니펫 74 페이지의 [그림 21](#)에서는 MQTT 클라이언트가 토픽 문자열 "MQTT Examples"을 사용하여 구독합니다. WebSphere MQ C 애플리케이션인 74 페이지의 [그림 22](#)는 토픽 문자열 "MQTT Examples"을 사용하여 토픽을 publish합니다. 이 코드 스니펫 74 페이지의 [그림 23](#)에서는 MQTT 클라이언트가 콜백 메소드 `messageArrived`로 publication을 수신합니다.

MQTT 클라이언트의 구독에 대한 응답으로 발행물을 송신하도록 WebSphere MQ 를 구성하는 방법에 대한 자세한 정보는 [MQTT 클라이언트 구독에 대한 응답으로 메시지 발행](#) 을 참조하십시오.

WebSphere MQ 애플리케이션이 MQTT 클라이언트로 직접 메시지를 보냄

WebSphere MQ에서 MQTT 클라이언트로 직접 메시지를 보내려면 79 페이지의 『[IBM WebSphere MQ Explorer 를 사용하여 메시지를 MQTT 클라이언트로 송신](#)』 태스크를 수행하십시오.

이러한 방식으로 MQTT 클라이언트로 보낸 메시지를 요청하지 않은 메시지라고 합니다. MQTT v3 클라이언트는 토픽 이름 세트가 포함된 발행으로 요청하지 않은 메시지를 수신합니다. 텔레메트리(MQXR) 서비스는 토픽 이름을 리모트 큐 이름으로 설정합니다.

디바이스용 WebSphere MQ 디먼으로 요청하지 않은 메시지를 보낼 수 없습니다. 디먼이 요청하지 않은 메시지를 수신하면 종료될 수 있습니다. MQTT v3 클라이언트는 요청하지 않은 메시지를 다른 MQTT v3 클라이언트 또는 WebSphere MQ 큐로 보낼 수 없습니다.

MQTT 클라이언트에 직접 메시지를 보내도록 WebSphere MQ를 구성하는 방법에 대한 자세한 정보는 [클라이언트에 직접 메시지 전송](#) 을 참조하십시오.

MQTT 클라이언트가 메시지를 발행함

MQTT v3 클라이언트는 다른 MQTT v3 클라이언트가 수신한 메시지를 발행할 수 있지만 요청하지 않은 메시지는 보낼 수 없습니다. 코드 스니펫, 75 페이지의 [그림 24](#)에서는 Java로 작성된 MQTT v3 클라이언트가 메시지를 발행하는 방법을 보여줍니다.

특정 MQTT v3 클라이언트로 메시지를 보내는 일반적인 방법은 각 클라이언트가 고유의 `ClientIdentifier`에 구독을 작성하는 것입니다. `ClientIdentifier`를 토픽 문자열로 사용하여 특정 MQTT 클라이언트에서 다른 MQTT 클라이언트로 메시지를 발행하려면 81 페이지의 『[특정 MQTT v3 클라이언트에 메시지 발행](#)』 태스크를 수행하십시오.

예제 코드 스니펫

74 페이지의 [그림 21](#)의 코드 스니펫은 Java로 작성된 MQTT 클라이언트가 구독을 작성하는 방법을 보여줍니다. 구독에 대한 발행을 수신하려면 `messageArrived`라는 콜백 메소드도 필요합니다. 코드 스니펫은 [Java를 사용하여 MQ 텔레메트리에 대한 구독자 작성태스크](#)에서 추출됩니다.

```
String    clientId = String.format("%-23.23s",
                                   System.getProperty("user.name") + "_" +
                                   (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient("localhost", clientId);
String topicString = "MQTT Examples";
int      QoS = 1;
client.subscribe(topicString, QoS);
```

그림 21. MQTT v3 클라이언트 구독자

74 페이지의 그림 22의 코드 스니펫에서는 C로 작성된 WebSphere MQ 애플리케이션이 발행을 보내는 방법을 보여줍니다. 코드 스니펫은 변수 토픽에 발행자 작성 태스크를 통해 추출됩니다.

```
/* Define and set variables to defaults */
/* Omitted lines declaring variables */
char * topicName = ""
char * topicString = "MQTT Examples"
char * publication = "Hello world!";
do {
    MQCONN(qMgrName, &Hconn, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    td.ObjectType = MQOT_TOPIC; /* Object is a topic */
    td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
    strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
    td.ObjectString.VSPtr = topicString;
    td.ObjectString.VSLength = (MQLONG)strlen(topicString);
    MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF_QUIESCING, &Hobj, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    pmo.Options = MQPMO_FAIL_IF_QUIESCING | MQPMO_RETAIN;
    MQPUT(Hconn, Hobj, &md, &pmo, (MQLONG)strlen(publication)+1, publication, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQCLOSE(Hconn, &Hobj, MQCO_NONE, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQDISC(&Hconn, &CompCode, &Reason);
} while (0);
```

그림 22. WebSphere MQ 발행자

발행이 도착하면 MQTT 클라이언트가 MQTT 애플리케이션 클라이언트 MqttCallback 클래스의 messageArrived 메소드를 호출합니다. 코드 스니펫은 Java를 사용하여 MQ 텔레메트리에 대한 구독자 작성 태스크에서 추출됩니다.

```
public class CallBack implements MqttCallback {
    public void messageArrived(MqttTopic topic, MqttMessage message) {
        try {
            System.out.println("Message arrived: \"" + message.toString()
                               + "\" on topic \"" + topic.toString() + "\"");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
// ... Other callback methods
}
```

그림 23. messageArrived 메소드

75 페이지의 그림 24에서는 MQTT v3가 74 페이지의 그림 21에서 작성된 구독에 메시지를 발행하는 방법을 보여줍니다. 코드 스니펫은 Java를 사용하여 첫 번째 MQ 텔레메트리 전송 공개자 애플리케이션 작성 태스크에서 추출됩니다.

```

String      address = "localhost";
String      clientId = String.format("%-23.23s",
                                     System.getProperty("user.name") + "-" +
                                     (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient  client = new MqttClient(address, clientId);
String      topicString = "MQTT Examples";
MqttTopic   topic = client.getTopic(Example.topicString);
String      publication = "Hello world";
MqttMessage message = new MqttMessage(publication.getBytes());
MqttDeliveryToken token = topic.publish(message);

```

그림 24. MQTT v3 클라이언트 발행자

IBM WebSphere MQ Explorer 에서 MQTT 클라이언트 유틸리티에 메시지 게시

IBM WebSphere MQ Explorer를 사용하여 메시지를 발행하고 MQTT 클라이언트 유틸리티를 사용하여 해당 메시지를 구독하려면 이 태스크의 단계를 수행하십시오. 다음 태스크에서는 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하지 않고 큐 관리자 알리언스를 구성하는 방법을 보여줍니다.

시작하기 전에

이 태스크에서는 사용자가 IBM WebSphere MQ 및 IBM WebSphere MQ Explorer에 익숙하며 IBM WebSphere MQ 및 IBM WebSphere MQ Telemetry 기능이 설치되어 있는 것으로 가정합니다.

이 태스크에서 큐 관리자 자원을 작성하는 사용자는 이에 대한 충분한 권한을 가지고 있어야 합니다. 데모를 위해 IBM WebSphere MQ Explorer 사용자 ID가 mqm 그룹의 구성원이라고 가정합니다.

이 태스크 정보

이 태스크에서는 IBM WebSphere MQ에 토픽을 작성하고 MQTT 클라이언트 유틸리티를 사용하여 해당 토픽을 구독합니다. IBM WebSphere MQ Explorer를 사용하여 토픽을 발행하면 MQTT 클라이언트가 해당 발행을 수신합니다.

프로시저

다음 태스크 중 하나를 수행하십시오.

- WebSphere MQ Telemetry를 설치했지만 아직 시작하지 않았습니다. [76 페이지의 『텔레메트리\(MQXR\) 서비스가 아직 정의되지 않은 태스크 시작』](#) 태스크를 수행하십시오.
- 이전에 WebSphere MQ Telemetry를 실행했지만 데모를 위해 새 큐 관리자를 사용하려고 합니다. [76 페이지의 『텔레메트리\(MQXR\) 서비스가 아직 정의되지 않은 태스크 시작』](#) 태스크를 수행하십시오.
- 텔레메트리 자원이 정의되지 않은 기존의 큐 관리자를 사용하는 태스크를 수행하려고 합니다. **샘플 구성 정의** 마법사를 실행하지 않으려고 합니다.
 - a. 텔레메트리를 설정하려면 다음 태스크 중 하나를 수행하십시오.
 - [Linux 및 AIX에서 텔레메트리를 큐 관리자 구성](#)
 - [Windows에서 텔레메트리를 큐 관리자 구성](#)
 - b. [77 페이지의 『실행 중인 텔레메트리\(MQXR\) 서비스가 있는 태스크 시작』](#) 태스크를 수행하십시오.
- 텔레메트리 자원이 이미 정의되어 있는 기존의 큐 관리자를 사용하는 태스크를 수행하려는 경우 [77 페이지의 『실행 중인 텔레메트리\(MQXR\) 서비스가 있는 태스크 시작』](#) 태스크를 수행하십시오.

다음에 수행할 작업

클라이언트 유틸리티로 직접 메시지를 보내려면 [79 페이지의 『IBM WebSphere MQ Explorer 를 사용하여 메시지를 MQTT 클라이언트로 송신』](#) 태스크를 수행하십시오.

관련 태스크

[IBM WebSphere MQ Explorer 를 사용하여 메시지를 MQTT 클라이언트로 송신](#)

IBM WebSphere MQ Explorer를 사용하여 IBM WebSphere MQ큐에 메시지를 넣음으로써 메시지를 MQTT 클라이언트 유틸리티로 송신합니다. 다음 태스크에서는 MQTT 클라이언트로 직접 메시지를 보내도록 리모트 큐 정의를 구성하는 방법을 보여줍니다.

특정 MQTT v3 클라이언트에 메시지 발행

ClientIdentifier를 토픽 이름으로 사용하고 IBM WebSphere MQ를 발행/구독 브로커로 사용하여 특정 MQTT v3 클라이언트에서 다른 MQTT v3 클라이언트로 메시지를 발행합니다. 디바이스용 WebSphere MQ Telemetry 디먼을 발행/구독 브로커로 사용하여 태스크를 반복하십시오.

텔레메트리(MQXR) 서비스가 아직 정의되지 않은 태스크 시작

큐 관리자를 작성하고 샘플 구성 정의를 실행하여 큐 관리자에 대한 샘플 텔레메트리 자원을 정의합니다. IBM WebSphere MQ Explorer를 사용하여 메시지를 발행하고 MQTT 클라이언트 유틸리티로 해당 메시지를 구독합니다.

이 태스크 정보

샘플 구성 정의 마법사를 사용하여 샘플 텔레메트리 자원을 설정할 때 해당 마법사가 게스트 사용자 ID 권한을 설정합니다. 이러한 방식으로 게스트 사용자 ID에 권한을 부여할지 신중하게 고려하십시오. 윈도우의 guest 및 Linux의 nobody 에 토픽 트리의 루트를 공개 및 등록하고 SYSTEM.MQTT.TRANSMIT.QUEUE에 메시지를 넣을 수 있는 권한이 부여됩니다.

또한 마법사는 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하는데, 이는 기존의 큐 관리자에서 실행 중인 애플리케이션에 지장을 줄 수 있습니다. 기본 전송 큐를 사용하지 않고 텔레메트리를 구성하는 것은 가능하지만 번거로운 작업입니다. 78 페이지의 『큐 관리자 알리어스 사용』 태스크에 따라 작업을 수행하십시오. 이 태스크에서는 큐 관리자를 작성하여 기존의 기본 전송 큐에 지장을 주지 않도록 합니다.

프로시저

1. IBM WebSphere MQ Explorer를 사용하여, 새 큐 관리자를 작성하고 시작하십시오.
 - a) Queue Managers 폴더 > 새로 작성 > 큐 관리자 ...를 마우스 오른쪽 단추로 클릭하십시오. 큐 관리자 이름을 입력하고 마침을 클릭하십시오.
큐 관리자 이름을 작성하십시오(예: MQTQMGR).
2. 텔레메트리(MQXR) 서비스를 작성 및 시작하고 샘플 텔레메트리 채널을 작성하십시오.
 - a) Queue Managers\QmgrName\Telemetry 폴더를 여십시오.
 - b) 샘플 구성 정의... > 마침을 클릭하십시오.
MQTT 클라이언트 유틸리티 시작 선택란을 선택하십시오.
3. MQTT 클라이언트 유틸리티를 사용하여 MQTT Example에 대한 구독을 작성하십시오.
 - a) 연결을 클릭하십시오.
클라이언트 실행 기록은 Connected 이벤트를 기록합니다.
 - b) 등록/주제 필드 > 등록에 MQTT Example 을 입력하십시오.
클라이언트 실행 기록은 Subscribed 이벤트를 기록합니다.
4. WebSphere MQ에 MQTTExampleTopic을 작성하십시오.
 - a) WebSphere MQ 탐색기 > 새로운 > 주제에서 Queue Managers\QmgrName\Topics 폴더를 마우스 오른쪽 단추로 누르십시오.
 - b) MQTTExampleTopic을 이름으로 입력하고 다음을 클릭하십시오.
 - c) 주제 문자열 > 완료로 MQTT Example 를 입력하십시오.
 - d) 수신확인 창을 닫으려면 확인을 클릭하십시오.
5. Publish Hello World! to the topic MQTT Example using IBM WebSphere MQ Explorer.
 - a) IBM WebSphere MQ Explorer에서 Queue Managers\QmgrName\Topics 폴더를 클릭하십시오.
 - b) MQTTExampleTopic을 마우스의 오른쪽 단추로 클릭하고 발행 테스트...를 클릭하십시오.

- c) 메시지 데이터 필드에 Hello World! 을 입력하고 > 메시지 게시 > MQTT 클라이언트 유틸리티 창으로 전환하십시오.

클라이언트 실행 기록은 Received 이벤트를 기록합니다.

실행 중인 텔레메트리(MQXR) 서비스가 있는 태스크 시작 텔레메트리 채널 및 토픽을 작성합니다. 사용자에게 토픽 및 텔레메트리 전송 큐를 사용할 수 있는 권한을 부여합니다. IBM WebSphere MQ Explorer를 사용하여 메시지를 발행하고 MQTT 클라이언트 유틸리티로 해당 메시지를 구독합니다.

시작하기 전에

이 태스크 버전에서는 큐 관리자 *QmgrName*이 정의되어 실행되고 있습니다. 텔레메트리(MQXR) 서비스가 정의되어 실행 중입니다. 텔레메트리(MQXR) 서비스는 수동으로 또는 샘플 구성 정의 마법사를 실행하여 작성되었을 수 있습니다.

이 태스크 정보

이 태스크에서는 기존 큐 관리자를 구성하여 발행을 MQTT 클라이언트 유틸리티로 보냅니다.

태스크의 77 페이지의 『1』 단계에서는 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하는데, 이는 기존의 큐 관리자에서 실행 중인 애플리케이션에 지장을 줄 수 있습니다. 기본 전송 큐를 사용하지 않고 텔레메트리를 구성하는 것은 가능하지만 번거로운 작업입니다. 78 페이지의 『큐 관리자 알리어스 사용』 태스크에 따라 작업을 수행하십시오.

프로시저

1. SYSTEM.MQTT.TRANSMIT.QUEUE를 기본 전송 큐로 설정하십시오.
 - a) Queue Managers*QmgrName* folder를 마우스의 오른쪽 단추로 클릭하고 특성...을 클릭하십시오.
 - b) 네비게이터에서 통신을 클릭하십시오.
 - c) 선택...을 클릭하고 SYSTEM.MQTT.TRANSMIT.QUEUE > 확인 > 확인을 선택하십시오.
2. Telemetry 채널 MQTTExampleChannel을 작성하여 MQTT 클라이언트 유틸리티를 WebSphere MQ에 연결하고 MQTT 클라이언트 유틸리티를 시작하십시오.
 - a) WebSphere MQ 탐색기 > 새로운 > 텔레메트리 채널 ...에서 Queue Managers*QmgrName*\Telemetry\Channels 폴더를 마우스 오른쪽 단추로 누르십시오.
 - b) MQTTExampleChannel을 채널 이름 필드에 입력하고 > 다음 > 다음을 클릭하십시오.
 - c) 클라이언트 권한 패널의 고정된 사용자 ID를 발행하려는 사용자 ID로 변경하고 MQTTExample을 구독한 후 다음을 클릭하십시오.
 - d) 클라이언트 유틸리티 실행 을 선택한 채로 완료를 선택하십시오.
3. MQTT 클라이언트 유틸리티를 사용하여 MQTT Example에 대한 구독을 작성하십시오.
 - a) 연결을 클릭하십시오.
클라이언트 실행 기록은 Connected 이벤트를 기록합니다.
 - b) 등록\주제 필드 > 등록에 MQTT Example 을 입력하십시오.
클라이언트 실행 기록은 Subscribed 이벤트를 기록합니다.
4. WebSphere MQ에 MQTTExampleTopic을 작성하십시오.
 - a) WebSphere MQ 탐색기 > 새로운 > 주제에서 Queue Managers*QmgrName*\Topics 폴더를 마우스 오른쪽 단추로 누르십시오.
 - b) MQTTExampleTopic을 이름으로 입력하고 다음을 클릭하십시오.
 - c) 주제 문자열 > 완료로 MQTT Example 를 입력하십시오.
 - d) 수신확인 창을 닫으려면 확인을 클릭하십시오.

5. mqm 그룹에 포함되지 않은 사용자가 MQTTExample 토픽을 발행 및 구독하게 하려면 다음을 수행하십시오.

a) 사용자에게 MQTTExampleTopic 토픽을 발행 및 구독할 수 있는 권한을 부여하십시오.

```
setmqaut -m qMgrName -t topic -n MQTTExampleTopic -p User ID -all +pub +sub
```

b) 사용자에게 SYSTEM.MQTT.TRANSMIT.QUEUE에 메시지를 넣을 수 있는 권한을 부여하십시오.

```
setmqaut -m qMgrName -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p User ID -all +put
```

6. Publish Hello World! to the topic MQTT Example using IBM WebSphere MQ Explorer.

a) IBM WebSphere MQ Explorer에서 Queue Managers*QmgrName*\Topics 폴더를 클릭하십시오.

b) MQTTExampleTopic을 마우스의 오른쪽 단추로 클릭하고 **발행 테스트...**를 클릭하십시오.

c) 메시지 데이터 필드에 Hello World! 을 입력하고 > 메시지 게시 > **MQTT 클라이언트 유틸리티** 창으로 전환하십시오.

클라이언트 실행 기록은 Received 이벤트를 기록합니다.

큐 관리자 알리어스 사용

기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하지 않고 IBM WebSphere MQ Explorer 를 사용하여 MQTT 클라이언트 유틸리티에 메시지를 발행합니다.

이 태스크는 이전 태스크와 연결되는 것으로, 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하지 않도록 하기 위해 큐 관리자 알리어스를 사용합니다.

시작하기 전에

76 페이지의 『텔레메트리(MQXR) 서비스가 아직 정의되지 않은 태스크 시작』 태스크 또는 77 페이지의 『실행 중인 텔레메트리(MQXR) 서비스가 있는 태스크 시작』 태스크를 완료하십시오.

이 태스크 정보

MQTT 클라이언트가 구독을 작성하면 IBM WebSphere MQ가 원격 큐 관리자 이름으로 ClientIdentifier 를 사용하여 해당 응답을 송신합니다. 이 태스크에서는 ClientIdentifier, MyClient를 사용합니다.

MyClient라는 큐 관리자 알리어스나 전송 큐가 없으면 응답이 기본 전송 큐에 배치됩니다. 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하면 MQTT 클라이언트가 응답을 받습니다.

큐 관리자 알리어스를 사용하면 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하지 않아도 됩니다. 각 ClientIdentifier에 대해 큐 관리자 알리어스를 설정해야 합니다. 일반적으로 클라이언트가 지나치게 많아 실제로 큐 관리자 알리어스를 사용할 수 없습니다. 대개 ClientIdentifier는 예측이 불가능하므로 이러한 방식으로 텔레메트리를 구성하는 것은 불가능합니다.

그럼에도 불구하고 특정 환경에서는 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE이외의 다른 설정으로 구성해야 합니다. [프로시저](#)의 단계에서는 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하는 대신 큐 관리자 알리어스를 구성합니다.

프로시저

1. SYSTEM.MQTT.TRANSMIT.QUEUE를 기본 전송 큐로 제거하십시오.

a) Queue Managers*QmgrName* folder를 마우스의 오른쪽 단추로 클릭하고 **특성...**을 클릭하십시오.

b) 네비게이터에서 **통신**을 클릭하십시오.

c) 기본 전송 큐 필드에서 SYSTEM.MQTT.TRANSMIT.QUEUE를 제거하고 **확인**을 클릭하십시오.

2. MQTT 클라이언트 유틸리티에서 더 이상 구독을 작성할 수 없는지 확인하십시오.

a) **연결**을 클릭하십시오.

클라이언트 실행 기록은 Connected 이벤트를 기록합니다.

b) **등록\주제** 필드 > **등록**에 MQTT Example 을 입력하십시오.

클라이언트 히스토리는 Subscribe failed 및 Connection lost 이벤트를 기록합니다.

3. ClientIdentifier, MyClient에 대한 큐 관리자 알리언스를 작성하십시오.

- Queue Managers\QmgrName\Queues 폴더 > 새로 작성 > 리모트 큐 정의를 마우스 오른쪽 단추로 클릭하십시오.
- 정의 이름을 MyClient로 지정하고 다음을 클릭하십시오.
- 리모트 큐 관리자 필드에 MyClient를 입력하십시오.
- 전송 큐 필드에 SYSTEM.MQTT.TRANSMIT.QUEUE를 입력하고 마침을 클릭하십시오.

4. MQTT 클라이언트 유틸리티를 다시 연결하십시오.

- 클라이언트 ID가 MyClient로 설정되어 있는지 확인하십시오.
- 연결

클라이언트 실행 기록은 Connected 이벤트를 기록합니다.

5. MQTT 클라이언트 유틸리티를 사용하여 MQTT Example에 대한 구독을 작성하십시오.

- 연결을 클릭하십시오.

클라이언트 실행 기록은 Connected 이벤트를 기록합니다.

- 등록\주제 필드 > 등록에 MQTT Example 을 입력하십시오.

클라이언트 실행 기록은 Subscribed 이벤트를 기록합니다.

6. Publish Hello World! to the topic MQTT Example using IBM WebSphere MQ Explorer.

- IBM WebSphere MQ Explorer에서 Queue Managers\QmgrName\Topics 폴더를 클릭하십시오.
- MQTTExampleTopic을 마우스의 오른쪽 단추로 클릭하고 발행 테스트...를 클릭하십시오.
- 메시지 데이터 필드에 Hello World! 을 입력하고 > 메시지 게시 > MQTT 클라이언트 유틸리티 창으로 전환하십시오.

클라이언트 실행 기록은 Received 이벤트를 기록합니다.

IBM WebSphere MQ Explorer 를 사용하여 메시지를 MQTT 클라이언트로 송신

IBM WebSphere MQ Explorer를 사용하여 IBM WebSphere MQ큐에 메시지를 넣음으로써 메시지를 MQTT 클라이언트 유틸리티로 송신합니다. 다음 태스크에서는 MQTT 클라이언트로 직접 메시지를 보내도록 리모트 큐 정의를 구성하는 방법을 보여줍니다.

시작하기 전에

75 페이지의 『IBM WebSphere MQ Explorer 에서 MQTT 클라이언트 유틸리티에 메시지 게시』 태스크를 수행하십시오. MQTT 클라이언트 유틸리티를 연결된 상태로 두십시오.

이 태스크 정보

다음 태스크에서는 토픽을 발행하지 않고 큐를 사용하여 MQTT 클라이언트로 메시지를 보내는 방법을 보여줍니다. 클라이언트에서 구독을 작성하지 않습니다. 이 태스크의 80 페이지의 『2』 단계에서는 삭제된 이전 구독에 대해 설명합니다.

프로시저

1. MQTT 클라이언트 유틸리티의 연결을 끊었다가 다시 연결하여 기존의 구독을 모두 제거하십시오.

기본값을 변경하지 않으면 MQTT 클라이언트 유틸리티가 정리 세션과 연결되므로 구독이 제거됩니다. 정리 세션을 참조하십시오.

태스크를 더욱 간편하게 수행하려면 MQTT 클라이언트 유틸리티가 작성하여 생성된 ClientIdentifier 를 사용하는 대신 고유의 ClientIdentifier를 입력하십시오.

- 연결 끊기를 클릭하여 Telemetry 채널에서 MQTT 클라이언트 유틸리티의 연결을 끊으십시오.

클라이언트 실행 기록은 Disconnected 이벤트를 기록합니다.

- b) **클라이언트 ID**를 MyClient로 변경하십시오.
 - c) **연결**을 클릭하십시오.
- 클라이언트 실행 기록**은 Connected 이벤트를 기록합니다.
2. MQTT 클라이언트 유틸리티가 MQTTExampleTopic에 대한 발행을 더 이상 수신하지 않는지 확인하십시오.
 - a) IBM WebSphere MQ Explorer에서 Queue Managers*QmgrName*\Topics 폴더를 클릭하십시오.
 - b) MQTTExampleTopic을 마우스의 오른쪽 단추로 클릭하고 **발행 테스트...**를 클릭하십시오.
 - c) **메시지 데이터** 필드에 Hello World! 을 입력하고 > **메시지 게시** > **MQTT 클라이언트 유틸리티** 창으로 전환하십시오.
 이 경우 **클라이언트 실행 기록**에 이벤트가 기록되지 않습니다.
 3. 클라이언트에 대한 리모트 큐 정의를 작성하십시오.

ClientIdentifier, MyClient를 리모트 큐 정의의 리모트 큐 관리자 이름으로 설정하십시오. 원하는 이름을 리모트 큐 이름으로 사용하십시오. 리모트 큐 이름이 MQTT 클라이언트에 토픽 이름으로 전달됩니다.

 - a) Queue Managers*QmgrName*\Queues 폴더 > **새로 작성** > **리모트 큐 정의**를 마우스 오른쪽 단추로 클릭하십시오.
 - b) 정의 이름을 MyClientRemoteQueue로 지정하고 **다음**을 클릭하십시오.
 - c) **리모트 큐** 필드에 MQTTExampleQueue를 입력하십시오.
 - d) **리모트 큐 관리자** 필드에 MyClient를 입력하십시오.
 - e) **전송 큐** 필드에 SYSTEM.MQTT.TRANSMIT.QUEUE를 입력하고 **마침**을 클릭하십시오.
 4. 테스트 메시지를 MyClientRemoteQueue에 넣으십시오.
 - a) **MyClientRemoteQueue**를 마우스의 오른쪽 단추로 클릭하고 **테스트 메시지 넣기...**를 클릭하십시오.
 - b) 메시지 데이터 필드에 Hello queue!를 입력하고 **메시지 넣기** > **닫기**를 클릭하십시오.

클라이언트 실행 기록은 Received 이벤트를 기록합니다.
 5. SYSTEM.MQTT.TRANSMIT.QUEUE를 기본 전송 큐로 제거하십시오.
 - a) Queue Managers*QmgrName* folder를 마우스의 오른쪽 단추로 클릭하고 **특성...**을 클릭하십시오.
 - b) 네비게이터에서 **통신**을 클릭하십시오.
 - c) **기본 전송 큐** 필드에서 SYSTEM.MQTT.TRANSMIT.QUEUE를 제거하고 **확인**을 클릭하십시오.
 6. 80 페이지의 『4』 단계를 다시 실행하십시오.

MyClientRemoteQueue는 전송 큐에 명시적으로 이름 지정된 리모트 큐 정의입니다. 기본 전송 큐를 정의하여 메시지를 MyClient로 보내지 않아도 됩니다.

다음에 수행할 작업

기본 전송 큐가 더 이상 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정되지 않은 경우 ClientIdentifier, MyClient에 대한 큐 관리자 알리어스가 정의되어 있어야 MQTT 클라이언트 유틸리티가 새 구독을 작성할 수 있습니다. 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 복원하십시오.

관련 태스크

IBM WebSphere MQ Explorer 에서 MQTT 클라이언트 유틸리티에 메시지 게시
 IBM WebSphere MQ Explorer를 사용하여 메시지를 발행하고 MQTT 클라이언트 유틸리티를 사용하여 해당 메시지를 구독하려면 이 태스크의 단계를 수행하십시오. 다음 태스크에서는 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하지 않고 큐 관리자 알리어스를 구성하는 방법을 보여줍니다.

특정 MQTT v3 클라이언트에 메시지 발행

ClientIdentifier를 토픽 이름으로 사용하고 IBM WebSphere MQ를 발행/구독 브로커로 사용하여 특정 MQTT v3 클라이언트에서 다른 MQTT v3 클라이언트로 메시지를 발행합니다. 디바이스용 WebSphere MQ Telemetry 디먼을 발행/구독 브로커로 사용하여 태스크를 반복하십시오.

특정 MQTT v3 클라이언트에 메시지 발행

ClientIdentifier를 토픽 이름으로 사용하고 IBM WebSphere MQ를 발행/구독 브로커로 사용하여 특정 MQTT v3 클라이언트에서 다른 MQTT v3 클라이언트로 메시지를 발행합니다. 디바이스용 WebSphere MQ Telemetry 디먼을 발행/구독 브로커로 사용하여 태스크를 반복하십시오.

시작하기 전에

75 페이지의 『IBM WebSphere MQ Explorer 에서 MQTT 클라이언트 유틸리티에 메시지 게시』 태스크를 수행하십시오. MQTT 클라이언트 유틸리티를 연결된 상태로 두십시오.

이 태스크 정보

이 태스크에서는 다음 두 가지 사항에 대해 설명합니다.

1. 한 MQTT 클라이언트의 토픽을 구독하고 다른 MQTT 클라이언트에서 발행을 수신합니다.
2. ClientIdentifier를 토픽 문자열로 사용하여 "포인트-투-포인트" 구독을 설정합니다.

추가 태스크인 82 페이지의 『디바이스용 WebSphere MQ Telemetry 디먼을 발행/구독 브로커로 사용』에서는 WebSphere MQ 대신 디바이스용 WebSphere MQ Telemetry 디먼을 발행/구독 브로커로 사용합니다.

프로시저

1. MQTT 클라이언트 유틸리티의 연결을 끊었다가 다시 연결하여 기존의 구독을 모두 제거하십시오.

기본값을 변경하지 않으면 MQTT 클라이언트 유틸리티가 정리 세션과 연결되므로 구독이 제거됩니다. 정리 세션을 참조하십시오.

태스크를 더욱 간편하게 수행하려면 MQTT 클라이언트 유틸리티가 작성하여 생성된 ClientIdentifier를 사용하는 대신 고유의 ClientIdentifier를 입력하십시오.

- a) 연결 끊기를 클릭하여 Telemetry 채널에서 MQTT 클라이언트 유틸리티의 연결을 끊으십시오.

클라이언트 실행 기록은 Disconnected 이벤트를 기록합니다.

- b) 클라이언트 ID를 MyClient로 변경하십시오.

- c) 연결을 클릭하십시오.

클라이언트 실행 기록은 Connected 이벤트를 기록합니다.

2. 토픽, MyClient에 대한 구독을 작성하십시오.

MyClient는 이 클라이언트의 ClientIdentifier입니다.

- a) Subscription\Topic 필드 > Subscribe에 MyClient 를 입력하십시오.

클라이언트 실행 기록은 Subscribed 이벤트를 기록합니다.

3. 다른 MQTT 클라이언트 유틸리티를 시작하십시오.

- a) Queue Managers\QmgrName\Telemetry\channels 폴더를 여십시오.

- b) PlainText 채널 > MQTT 클라이언트 유틸리티 실행 ... 을 마우스 오른쪽 단추로 클릭하십시오.

- c) 연결을 클릭하십시오.

클라이언트 실행 기록은 Connected 이벤트를 기록합니다.

4. Hello MyClient!를 MyClient 토픽에 발행하십시오.

- a) ClientIdentifier, MyClient를 사용하여 실행 중인 MQTT 클라이언트 유틸리티에서 구독 토픽 MyClient를 복사하십시오.

- b) MyClient를 각 MQTT 클라이언트 유틸리티 인스턴스의 Publication\Topic 필드에 붙여 넣으십시오.

- c) Publication\message 필드에 Hello MyClient! 를 입력하십시오.

- d) 두 인스턴스 모두에서 발행을 클릭하십시오.

결과

MQTT 클라이언트 유틸리티에서 ClientIdentifier, MyClient를 사용하는 클라이언트 실행 기록은 두 개의 수신된 이벤트와 한 개의 발행된 이벤트를 기록합니다. 다른 MQTT 클라이언트 유틸리티 인스턴스는 한 개의 발행된 이벤트를 기록합니다.

하나의 수신된 이벤트만 표시되는 경우 다음 가능한 원인을 확인하십시오.

1. 큐 관리자에 대한 기본 전송 큐가 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정되었습니까?
2. 다른 실습에서 MyClient를 참조하는 리모트 큐 정의나 큐 관리자 알리언스를 작성했습니까? 구성 문제점이 있는 경우 큐 관리자 알리언스 또는 전송 큐와 같은 MyClient를 참조하는 자원을 모두 삭제하십시오. 클라이언트 유틸리티의 연결을 끊고 텔레메트리(MQXR) 서비스를 중지한 후에 다시 시작하십시오.

다음에 수행할 작업

다음 82 페이지의 『[디바이스용 WebSphere MQ Telemetry 디먼을 발행/구독 브로커로 사용](#)』 태스크를 수행하십시오. MQTT 클라이언트 유틸리티가 Telemetry 채널 대신 디바이스용 WebSphere MQ Telemetry 디먼에 연결합니다.

관련 태스크

[IBM WebSphere MQ Explorer 에서 MQTT 클라이언트 유틸리티에 메시지 게시](#)

IBM WebSphere MQ Explorer를 사용하여 메시지를 발행하고 MQTT 클라이언트 유틸리티를 사용하여 해당 메시지를 구독하려면 이 태스크의 단계를 수행하십시오. 다음 태스크에서는 기본 전송 큐를 SYSTEM.MQTT.TRANSMIT.QUEUE로 설정하지 않고 큐 관리자 알리언스를 구성하는 방법을 보여줍니다.

[IBM WebSphere MQ Explorer 를 사용하여 메시지를 MQTT 클라이언트로 송신](#)

IBM WebSphere MQ Explorer를 사용하여 IBM WebSphere MQ큐에 메시지를 넣음으로써 메시지를 MQTT 클라이언트 유틸리티로 송신합니다. 다음 태스크에서는 MQTT 클라이언트로 직접 메시지를 보내도록 리모트 큐 정의를 구성하는 방법을 보여줍니다.

디바이스용 *WebSphere MQ Telemetry* 디먼을 발행/구독 브로커로 사용

WebSphere MQ 대신 디바이스용 WebSphere MQ Telemetry 디먼을 발행/구독 브로커로 사용할 수 있습니다. ClientIdentifier를 토픽 문자열로 구독하면 MQTT 클라이언트 유틸리티의 한 인스턴스가 포함된 메시지를 다른 인스턴스로 발행할 수 있습니다.

시작하기 전에

아직 설치하지 않은 경우 디먼을 설치하십시오.

PlainText Telemetry 채널에서 이미 사용 중인 포트 1883을 디먼이 사용하는지에 대해서는 확인하지 마십시오.

이 태스크 정보

이 태스크에서는 기본 TCP/IP 포트 이외의 포트를 사용하여 디바이스용 WebSphere MQ Telemetry 디먼에 MQTT 클라이언트 유틸리티를 연결합니다. 한 클라이언트는 해당 ClientIdentifier를 토픽 문자열로 사용하여 구독하고 다른 클라이언트는 이전 태스크에서 수행한 것과 동일한 방식으로 ClientIdentifier를 발행합니다. 자세한 정보는 [프로시저](#)의 내용을 참조하십시오.

참고: 이 태스크에서는 Windows에서 디먼을 실행하는 방법에 대해 설명합니다. Linux에서 디먼을 실행하려면 amqtdd의 경로 및 권한을 수정하십시오.

프로시저

1. 디바이스용 WebSphere MQ Telemetry 디먼을 포함하는 디렉토리에서 명령 창을 여십시오.

Windows의 디렉토리 경로는 *WebSphere MQ installation directory\mqxr\SDK\advanced\DeviceDaemon\windows_ia32*입니다.

2. 다른 TCP/IP 포트에서 디먼을 실행하십시오.

a) 디먼과 같은 디렉토리에서 amqtdd.cfg라는 파일을 작성하십시오.

b) 디먼에 대한 다른 기본 포트를 구성하려면 다음 행을 파일에 추가하십시오.

```
port 1884
```

c) 파일을 저장하십시오.

3. 디먼을 시작하십시오.

```
amqtd
```

디먼은 명령 창에 해당 콘솔 로그를 작성합니다.

```
20100712 123133.857 CWNAN9999I IBM WebSphere MQ Telemetry daemon for devices
20100712 123133.857 CWNAN9997I Licensed Materials - Property of IBM
20100712 123133.857 CWNAN9996I Copyright IBM Corp. 2007, 2024. All Rights Reserved
20100712 123133.857 CWNAN9995I US Government Users Restricted Rights ...
20100712 123133.857 CWNAN0049I Configuration file name is .\amqtd.cfg
20100712 123133.873 CWNAN0054I Features included: bridge
20100712 123134.060 CWNAN0014I MQTT protocol starting, listening on port 1884
```

4. MQTT 클라이언트 유틸리티의 인스턴스를 시작하십시오.

Telemetry 채널을 통해서만 MQTT 클라이언트 유틸리티를 시작하십시오. 그러면 해당 디먼에 연결할 수 있습니다. 또는 IBM WebSphere MQ SupportPac, IA92를 설치할 수 있습니다. SupportPac 은 [IA92: WBI Brokers- WebSphere MQ Telemetry 전송의 Java 구현](#)에서 사용 가능합니다.

- Queue Managers*QmgrName*\Telemetry\channels 폴더를 여십시오.
- PlainText** 채널 > **MQTT 클라이언트 유틸리티 실행 ...** 을 마우스 오른쪽 단추로 클릭하십시오.
- 포트를 1884로 변경하십시오.
- 클라이언트 ID를 MyClient로 변경하십시오.
- 연결을 클릭하십시오.

클라이언트 실행 기록은 Connected 이벤트를 기록합니다.

5. 토픽, MyClient에 대한 구독을 작성하십시오.

MyClient는 이 클라이언트의 ClientIdentifier입니다.

- Subscription\Topic** 필드 > **Subscribe**에 MyClient 를 입력하십시오.

클라이언트 실행 기록은 Subscribed 이벤트를 기록합니다.

6. 다른 MQTT 클라이언트 유틸리티를 시작하십시오.

- Queue Managers*QmgrName*\Telemetry\channels 폴더를 여십시오.
- PlainText** 채널 > **MQTT 클라이언트 유틸리티 실행 ...** 을 마우스 오른쪽 단추로 클릭하십시오.
- 포트를 1884로 변경하십시오.
- 연결을 클릭하십시오.

클라이언트 실행 기록은 Connected 이벤트를 기록합니다.

7. Hello MyClient!를 MyClient 토픽에 발행하십시오.

- ClientIdentifier, MyClient를 사용하여 실행 중인 MQTT 클라이언트 유틸리티에서 구독 토픽 MyClient를 복사하십시오.
- MyClient를 각 MQTT 클라이언트 유틸리티 인스턴스의 **Publication\Topic** 필드에 붙여 넣으십시오.
- Publication\message** 필드에 Hello MyClient! 를 입력하십시오.
- 두 인스턴스 모두에서 **발행**을 클릭하십시오.

결과

MQTT 클라이언트 유틸리티에서 ClientIdentifier, MyClient를 사용하는 **클라이언트 실행 기록** 은 두 개의 수신된 이벤트와 한 개의 **발행된** 이벤트를 기록합니다. 다른 MQTT 클라이언트 유틸리티 인스턴스는 한 개의 **발행된** 이벤트를 기록합니다.

명령 창에서 디바이스용 WebSphere MQ Telemetry 디먼에 대한 연결 및 연결 끊기 이벤트를 모니터링할 수도 있습니다.

MQTT 클라이언트에서 IBM WebSphere MQ 애플리케이션에 메시지 송신

IBM WebSphere MQ 애플리케이션은 토픽에 구독하여 MQTT v3 클라이언트에서 메시지를 수신할 수 있습니다. MQTT 클라이언트는 텔레메트리 채널을 사용하여 IBM WebSphere MQ 에 연결하고 동일한 토픽을 발행하여 IBM WebSphere MQ 애플리케이션에 메시지를 보냅니다.

발행물을 IBM WebSphere MQ에 정의된 구독으로 발행물을 송신하는 방법을 학습하려면 [84 페이지의 『MQTT 클라이언트에서 IBM WebSphere MQ Explorer로 메시지 발행』](#) 태스크를 수행하십시오.

토픽이 클러스터되어 있거나 발행/구독 계층 구조를 사용해 분배되어 있을 경우 구독은 다른 큐 관리자에서 MQTT 클라이언트가 연결되어 있는 큐 관리자로 될 수 있습니다.

MQTT 클라이언트에서 IBM WebSphere MQ Explorer로 메시지 발행

WebSphere MQ Explorer를 사용해 토픽의 구독을 작성하고 WebSphere MQTT 클라이언트 유틸리티를 사용해 토픽에 발행하십시오.

시작하기 전에

75 페이지의 『IBM WebSphere MQ Explorer 에서 MQTT 클라이언트 유틸리티에 메시지 게시』 태스크를 수행하십시오. MQTT 클라이언트 유틸리티를 연결된 상태로 두십시오.

이 태스크 정보

이 태스크는 MQTT 클라이언트로 메시지를 발행하는 것과 WebSphere MQ Explorer로 작성된 관리되지 않은 지속적 구독을 사용해 발행을 수신하는 것을 보여줍니다.

프로시저

1. 토픽 문자열 MQTT Example에 대한 지속 가능한 구독을 작성하십시오. 다음 프로시저 중 하나를 수행하십시오.

- [결과](#)에 설명된 명령 스크립트를 실행하십시오.
- WebSphere MQ Explorer를 사용해 큐와 구독을 작성하려면 다음 단계를 따르십시오.
 - a) WebSphere MQ 탐색기 > **새로운** > **로컬 큐 ...**에서 Queue Managers*QmgrName*\Queues 폴더를 마우스 오른쪽 단추로 누르십시오.
 - b) MQTTExampleQueue를 큐 이름으로 입력하고 **마침**을 클릭하십시오.
 - c) WebSphere MQ 탐색기 > **새로운** > **등록 ...**에서 Queue Managers*QmgrName*\Subscriptions 폴더를 마우스 오른쪽 단추로 누르십시오.
 - d) MQTTExampleSubscription을 큐 이름으로 입력하고 **다음**을 클릭하십시오.
 - e) **선택...** > MQTTExampleTopic > **확인**을 클릭하십시오.

[75 페이지의 『IBM WebSphere MQ Explorer 에서 MQTT 클라이언트 유틸리티에 메시지 게시』](#)의 단계 [76 페이지의 『4』](#)에서 이미 토픽 MQTTExampleTopic을 작성했습니다.

- f) MQTTExampleQueue를 대상 이름으로 입력하고 **마침**을 클릭하십시오.

2. 선택적 단계로, mqm 권한이 없는 다른 사용자가 사용하도록 큐를 설정하십시오.

mqm보다 낮은 권한을 가진 사용자에 대한 구성을 설정할 경우 MQTTExampleQueue에 put 및 get 권한을 부여해야 합니다. 토픽과 전송 큐에 대한 액세스는 [75 페이지의 『IBM WebSphere MQ Explorer 에서 MQTT 클라이언트 유틸리티에 메시지 게시』](#)에서 구성되었습니다.

- a) 사용자에게 MQTTExampleQueue 큐에 대한 넣기 및 가져오기 권한을 부여하십시오.

```
setmqaut -m qMgrName -t queue -n MQTTExampleQueue -p User ID -all +put +get
```

3. Hello WebSphere MQ! 를 MQTT 클라이언트 유틸리티를 사용하여 MQTT Example 토픽에 공개하십시오.

MQTT 클라이언트 유틸리티를 연결하지 않은 경우, **PlainText** 채널 > **Run MQTT Client Utility ... > Connect**를 마우스 오른쪽 단추로 클릭하십시오.

a) **Publication\Topic** 필드에 MQTT Example을 입력하십시오.

b) **공개\메시지 필드 > 공개**에 Hello WebSphere MQ! 을 입력하십시오.

4. Queue Managers\QmgrName\Queues 폴더를 열고 MQTTExampleQueue를 찾으십시오.

현재 큐 용량 필드는 1입니다.

5. MQTTExampleQueue > **메시지 찾아보기 ...** 를 마우스 오른쪽 단추로 클릭하십시오. 그리고 발행을 조사한다.

디바이스용 IBM WebSphere MQ 텔레메트리 디먼과 IBM WebSphere MQ 사이에 메시지 전송

디바이스용 IBM WebSphere MQ Telemetry 디먼에 명령을 송신하는 방법을 알아보려면 이 태스크를 수행하십시오. 사용자가 작성하는 명령은 IBM WebSphere MQ에서 디먼으로, 디먼에서 IBM WebSphere MQ로 메시지를 전송하는 브릿지를 작성합니다.

시작하기 전에

MQTT 클라이언트 유틸리티 사용에 익숙해지기 위해서 81 페이지의 『[특정 MQTT v3 클라이언트에 메시지 발행](#)』 및 82 페이지의 『[디바이스용 WebSphere MQ Telemetry 디먼을 발행/구독 브로커로 사용](#)』 태스크를 수행하십시오. 태스크를 완료하고 나면 MQTT 클라이언트 유틸리티의 인스턴스 하나를 디바이스용 텔레메트리 디먼에 연결한 채로 두십시오. 다른 인스턴스는 텔레메트리 채널에 연결한 채로 두십시오.

이 태스크에서는 127.0.0.1주소에서 1883 포트를 청취하는 텔레메트리 서비스에 채널을 정의한 것으로 가정합니다. 마찬가지로, 기본 디먼 리스너는 127.0.0.1주소에서 1884 포트를 청취하도록 구성됩니다. 디먼 amqtdc와 동일한 디렉토리에 저장된 amqtdc.cfg파일의 한 행은 기본 디먼 리스너 포트를 구성합니다.

```
port 1884
```

이 태스크 정보

이 태스크에서는 실행 중인 디먼을 업데이트하여 WebSphere MQ 텔레메트리(MQXR) 서비스에 대한 연결 브릿지를 작성한 후에 디먼과 메시지를 교환합니다.

팁: 업데이트 파일 amqtdc.upd는 사용된 후 디먼에 의해 삭제됩니다. 나중에 사용하기 위해 작성한 명령을 유지하기 위해서 명령을 다른 파일에 작성하여 이를 amqtdc.upd에 전송할 수도 있습니다.

프로시저

1. 실행 중인 MQTT 클라이언트 유틸리티 인스턴스가 두 개 있는지 확인하십시오. 하나는 포트 1884의 디먼에 연결되어 있고 다른 하나는 포트 1883에서 실행 중인 텔레메트리 채널에 연결되어 있습니다..
2. 다음 명령을 파일에 넣어 디먼 amqtdc와 같은 디렉토리의 파일 amqtdc.upd를 작성합니다.

```
connection daemon1
address 127.0.0.1:1883
topic # in import/ export/
topic # out export/ import/
try_private false
```

- 브릿지를 daemon1이라고 하며, 이는 소켓 주소 127.0.0.1:1883에서 실행 중인 텔레메트리(MQXR) 서비스를 위해 구성된 채널에 연결됩니다. try_private 명령은 선택적입니다. 기본값은 true입니다. 이 행이 없으면 브릿지는 우선 디바이스용 WebSphere MQ Telemetry 디먼에서 이해하는 개인용 프로토콜을 사용해 연결하려 합니다. 명령에 try_private false를 추가하면 이 단계를 피할 수 있으며 연결을 완료하는 시간을 줄여줍니다.
- topic # in import/ export/행은 daemon1에게 큐 관리자에서 작성된 토픽 문자열 export/#와 일치하는 모든 토픽을 구독하도록 지시합니다. 이는 토픽 문자열의 시작을 export/에서 import/로 변경

하며 일치하는 발행을 큐 관리자에서 디먼으로 전송합니다. `topic # out export/ import/` 행은 로컬 디먼에서 구독을 작성합니다. 브릿지는 디먼에서 작성된 토픽 문자열 `export/#`과 일치하는 모든 토픽을 구독합니다. 이는 토픽 문자열의 시작을 `export/`에서 `import/`로 변경하며 발행을 디먼에서 큐 관리자로 전송합니다.

86 페이지의 그림 25은 결과 콘솔 로그를 보여줍니다.

```
CWNAN0124I Starting bridge connection daemon1
CWNAN0133I Bridge connection daemon1 to 127.0.0.1:1883 now established
```

그림 25. 시작 연결 브릿지의 콘솔 로그

3. MQTT 클라이언트 유틸리티의 각 인스턴스에서 **구독/토픽:** 입력 필드 > **구독**에 `import/#` 를 입력하십시오.
 4. MQTT 클라이언트 유틸리티의 각 인스턴스에서 **Publication/Topic:** 입력 필드에 `export/#` 를 입력하십시오.
 - a) 텔레메트리 채널 포트 1883에 연결된 MQTT 클라이언트 유틸리티에서 **Publication/Message:** 입력 필드 > **공개**에 `From the queue manager` 를 입력하십시오.
 - b) 1884포트에 연결된 MQTT 클라이언트 유틸리티에서 텔레메트리 디먼, **Publication/Message:** 입력 필드 > **공개**에 `From the daemon` 를 입력하십시오.
- 각 MQTT 클라이언트 유틸리티의 클라이언트 실행 기록은 발행이 한 브로커에서 다른 브로커로 전송되었음을 표시합니다.

MQTT 발행/구독 애플리케이션

MQTT 애플리케이션 쓰기에 토픽 기반 발행/구독을 사용하십시오.

MQTT 클라이언트가 연결되면 발행은 클라이언트와 서버 사이에서 오고 가게 됩니다. 발행은 클라이언트에서 정보가 발행되면 클라이언트에서 전송됩니다. 메시지가 클라이언트에서 작성한 구독과 일치하는 토픽에 발행되었을 경우에는 클라이언트에서 발행을 수신합니다.

WebSphere MQ 발행/구독 브로커는 MQTT 클라이언트에서 작성한 토픽과 구독을 관리합니다. MQTT에서 작성된 토픽은 WebSphere MQ 애플리케이션에서 작성된 토픽과 동일한 토픽 공간을 공유합니다.

MQTT 클라이언트 구독에서 토픽 문자열과 일치하는 발행은 클라이언트의 `ClientIdentifier`에 설정된 리모트 큐 관리자 이름과 함께 `SYSTEM.MQTT.TRANSMIT.QUEUE`에 저장됩니다. 텔레메트리(MQXR) 서비스는 구독을 작성한 클라이언트에 발행물을 전달합니다. 클라이언트를 식별하기 위해 리모트 큐 관리자 이름으로 설정된 `ClientIdentifier`를 사용합니다.

일반적으로 `SYSTEM.MQTT.TRANSMIT.QUEUE`는 기본 전송 큐로 정의되어야 합니다. 기본 전송 큐를 사용하지 않도록 MQ를 구성하는 것이 가능합니다. 기본 전송 큐를 사용하지 않도록 하십시오. [MQTT 클라이언트로 메시지를 송신하도록 분산 큐잉 구성](#)을 참조하십시오.

MQTT 클라이언트는 지속 세션을 작성할 수 있습니다. 90 페이지의 『MQTT 상태 비저장 및 상태 저장 세션』을 참조하십시오. 지속 세션에 작성된 구독은 지속 가능합니다. 지속 세션과 함께 클라이언트에 도착하는 발행은 `SYSTEM.MQTT.TRANSMIT.QUEUE`에 저장되며, 다시 연결될 때 클라이언트로 전달됩니다.

MQTT 클라이언트는 보유한 발행물에 발행하고 구독할 수도 있습니다. 보유한 발행물 및 MQTT 클라이언트를 참조하십시오. 보유한 발행물 토픽에 대한 구독자는 토픽의 최근 발행을 수신합니다. 구독자는 보유한 발행물이 구독을 작성하거나 이전 세션에 다시 연결할 때 보유한 발행물을 수신합니다.

관련 태스크

[Java를 사용하여 첫 번째 MQ 텔레메트리 전송 공개자 애플리케이션 작성](#)

[Java를 사용하여 MQ 텔레메트리 전송에 대한 등록자 작성](#)

텔레메트리 애플리케이션

WebSphere MQ 또는 WebSphere Message Broker 메시지 플로우를 사용하여 텔레메트리 애플리케이션을 쓰십시오.

WebSphere MQ에 텔레메트리 애플리케이션을 프로그램하려면 JMS, MQI 또는 다른 WebSphere MQ 프로그래밍 인터페이스를 사용하십시오.

텔레메트리(MQXR) 서비스는 MQTT v3 메시지 및 WebSphere MQ 메시지 간에 변환합니다. MQTT 클라이언트를 대신해 구독과 발행을 작성하며 발행을 MQTT 클라이언트에 전달합니다. 발행은 MQTT v3 메시지의 페이로드입니다. 페이로드는 메시지 헤더와 `json-bytes` 형식의 바이트 배열을 포함합니다. 텔레메트리 서버는 MQTT v3 메시지와 WebSphere MQ 메시지를 상호 매핑합니다. 87 페이지의 『[큐 관리자와 WebSphere MQ Telemetry의 통합](#)』을 참조하십시오.

WebSphere Message Broker와 MQTT 클라이언트 사이에서 발행을 송신 및 수신하려면 `Publication`, `MQInput`과 `JMSInput` 노드를 사용하십시오.

메시지 플로우를 사용하여 텔레메트리와 HTTP를 사용하는 웹 사이트를, 텔레메트리와 WebSphere MQ, WebSphere Adapters를 사용하는 다른 애플리케이션을 통합할 수 있습니다.

WebSphere MQ 텔레메트리는 WebSphere Message Broker 버전 7의 SCADA 노드를 대체합니다. SCADAInput 및 SCADAOutput 노드를 사용하여 버전 6 WebSphere Message Broker 메시지 플로우를 버전 7로 이주하는 방법에 대한 정보는 [Migration of telemetry applications from using WebSphere Message Broker version 6 to use IBM WebSphere MQ Telemetry and WebSphere Message Broker version 7.0](#) 를 참조하십시오.

큐 관리자와 WebSphere MQ Telemetry의 통합

MQTT 클라이언트는 발행/구독 애플리케이션으로서 WebSphere MQ와 통합되어 있습니다. WebSphere MQ의 토픽을 발행 또는 구독하고 새 토픽을 작성하며 기존 토픽을 사용할 수 있습니다. 자신을 포함한 MQTT 클라이언트들의 결과로서 WebSphere MQ에서 발행을 수신하거나 다른 WebSphere MQ 애플리케이션이 그 구독의 토픽에 발행하는 발행을 수신합니다. 발행의 속성을 결정하는 규칙이 적용됩니다.

WebSphere MQ에서 제공되는 토픽, 발행, 구독 및 메시지와 연관된 많은 속성은 지원되지 않습니다. 87 페이지의 『[MQTT 클라이언트에서 WebSphere MQ 발행/구독 브로커로](#)』와 88 페이지의 『[WebSphere MQ에서 MQTT 클라이언트로](#)』에서 `publication`의 속성을 설정하는 방법에 대해 설명합니다. 설정은 발행이 WebSphere MQ 발행/구독 브로커로 가는가, 혹은 브로커에서 오는가에 따라 달라집니다.

WebSphere MQ 발행/구독 토픽은 관리 토픽 오브젝트와 연관되어 있습니다. MQTT 클라이언트에서 작성되는 토픽도 이와 마찬가지로입니다. MQTT 클라이언트가 발행의 토픽 문자열을 작성하면 WebSphere MQ 발행/구독 브로커는 이를 관리 토픽 오브젝트와 연관시킵니다. 브로커는 발행의 토픽 문자열을 제일 가까운 관리 토픽 오브젝트 상위에 매핑합니다. 매핑은 WebSphere MQ 애플리케이션에 대한 것과 동일합니다. 사용자가 작성한 토픽이 없을 경우 발행 토픽은 `SYSTEM.BASE.TOPIC`에 매핑됩니다. 발행에 적용되는 토픽은 토픽 오브젝트에서 도출된 것입니다.

WebSphere MQ 애플리케이션이나 관리자가 구독을 작성하면 구독에 이름이 지정됩니다. WebSphere MQ 탐색기를 사용하거나 `runmqsc` 또는 PCF 명령을 사용하여 등록을 나열하십시오. 모든 MQTT 클라이언트 구독이 이름 지정됩니다. 이름에는 다음 양식의 이름이 지정됩니다. `ClientIdentifier:Topic name`

MQTT 클라이언트에서 WebSphere MQ 발행/구독 브로커로

MQTT 클라이언트가 발행을 WebSphere MQ로 전송했습니다. 텔레메트리(MQXR) 서비스는 WebSphere MQ 메시지로 발행물을 변환합니다. WebSphere MQ 메시지는 다음 세 파트를 포함합니다.

1. MQMD
2. RFH2
3. 메시지

MQMD 특성은 87 페이지의 표 3에 기록되어 있지 않은 이상 각각 기본값으로 설정됩니다.

표 3. MQMD		
MQMD 필드	유형	가치
Format	MQCHAR8	MQFMT_RF_HEADER_2

표 3. MQMD (계속)		
MQMD 필드	유형	가치
UserIdentifier	MQCHAR12	다음 중 하나로 설정하십시오. MqttClient.ClientIdentifier MqttConnectOptions.UserName 텔레메트리 채널을 위해 WebSphere MQ 관리자에 의해 설정된 사용자 ID
Priority	MQLONG	MQPRI_PRIORITY_AS_Q_DEF (WebSphere MQ의 기본값이며 이는 4가 기본값인 JMS와는 다릅니다.)
Persistence	MQLONG	QoS=0→MQPER_NOT_PERSISTENT QoS=1→MQPER_PERSISTENT QoS=2→MQPER_PERSISTENT

RFH2 헤더에는 JMS 메시지의 유형을 정의하기 위한 <msd> 폴더가 없습니다. 텔레메트리(MQXR) 서비스는 WebSphere MQ 메시지를 기본 JMS 메시지로 작성합니다. 기본 JMS 메시지-유형은 jms-bytes 메시지입니다. 애플리케이션은 메시지 특성으로서의 추가 헤더 정보에 액세스할 수 있습니다. 메시지 특성을 참조하십시오.

RFH2 값은 88 페이지의 표 4에 표시된 것과 같이 설정됩니다. 형식 특성은 RFH2 고정된 헤더에 설정되며 다른 값은 RFH2 폴더 안에 설정됩니다.

표 4. RFH2		
RFH2 특성	유형/폴더	헤더
Format	MQCHAR8	MQFMT_NONE
ClientIdentifier	mqtt/clientId	MqttClient.ClientIdentifier를 1 - 23바이트 길이로 복사하십시오.
QoS	mqtt/qos	수신되는 MQTT 메시지에서 QoS를 복사하십시오.
메시지 ID	mqtt/msgid	QoS가 1 또는 2인 경우 수신되는 MQTT 메시지에서 Message ID를 복사하십시오.
MQIsRetained	mqs/Ret	기존 MQTT 발행이 RETAIN 특성이 설정되어 전송되었으며 메시지가 보유된 발행물로서 수신된 경우 설정하십시오.
MQTopicString	mqs/Top	MQTT 메시지 발행 대상이 된 토픽입니다.

MQTT 발행의 페이로드를 WebSphere MQ 메시지의 콘텐츠에 맵핑합니다.

표 5. 메시지 콘텐츠		
메시지 콘텐츠	유형	콘텐츠
Buffer	MQBYTEn	수신되는 MQTT 메시지의 바이트 사본입니다. 길이는 0이 될 수도 있습니다.

WebSphere MQ에서 MQTT 클라이언트로

클라이언트가 발행 토픽을 구독했습니다. WebSphere MQ 애플리케이션이 토픽에 발행되었으며 이로 인해 발행이 WebSphere MQ 발행/구독 브로커에 의해 MQTT 구독자에게 전송되었습니다. 또는 WebSphere MQ 애플리

케이션이 요청되지 않은 메시지를 MQTT 클라이언트로 직접 전송했습니다. 89 페이지의 표 6에서는 고정된 메시지 헤더를 MQTT 클라이언트로 전송된 메시지에 설정하는 방법에 대해 설명합니다. WebSphere MQ 메시지 헤더의 다른 모든 데이터 또는 다른 모든 헤더는 버려집니다. WebSphere MQ 메시지의 메시지 데이터는 변경 없이 MQTT 메시지 안의 메시지 페이로드로서 전송됩니다. MQTT 메시지는 텔레메트리(MQXR) 서비스에 의해 MQTT 클라이언트로 전송됩니다.

표 6. MQTT 고정된 헤더 특성		
MQTT 필드	유형	가치
DUP	부울	QoS = 1 또는 2이며 이전 전송에서 메시지가 이 클라이언트로 전송되었고 이 메시지가 시간이 지난 뒤에도 수신확인되지 않았을 경우 설정하십시오.
QoS	int	<p>WebSphere MQ의 발행/구독 브로커에서 발신되는 발행의 QoS 값이 설정되는 방법은 수신되는 발행에 따라 다릅니다. 수신되는 발행이 MQTT에서 송신한 것인가 또는 WebSphere MQ 애플리케이션에서 송신한 것인가에 따라 달라집니다.</p> <p>MQTT 수신되는 발행과 구독자에 의해 요청된 QoS의 QoS 값을 낮추십시오.</p> <p>WebSphere MQ 수신되는 발행에서 도출되는 QoS의 값을 낮추십시오.</p> <p>MQPER_NOT_PERSISTENT→QoS=0 MQPER_PERSISTENT→QoS=2</p> <p>구독자에 의해 요청된 QoS의 QoS 값도 낮추십시오. 구독 없이 메시지가 클라이언트에 전송된 경우 QoS는 기본값인 2로 설정됩니다. 클라이언트는 다른 QoS로 DEFAULT. QoS를 구독하여 이 값을 변경할 수 있습니다.</p>
RETAIN	부울	수신되는 발행에 보유된 특성이 설정되어 있을 경우 설정하십시오.

89 페이지의 표 7에서는 MQTT 클라이언트로 전송된 MQTT 메시지에 변수 메시지 헤더가 설정되는 방법에 대해 설명합니다.

표 7. MQTT 변수 헤더 특성		
MQTT 필드	유형	가치
Topic name	문자열	메시지와 같이 발행된 토픽 문자열입니다.
Message ID	문자열	발행을 SYSTEM.MQTT.TRANSMIT.QUEUE에 넣었을 때 그 발행에 있는 MQMD.MsgId 특성의 마지막 2바이트입니다.
Payload	byte[]	수신되는 발행의 바이트 사본을 발행/구독 브로커에 전달하십시오. 길이는 0이 될 수도 있습니다.

디바이스의 텔레메트리 디먼

디바이스용 WebSphere MQ Telemetry 디먼은 고급 MQTT V3 클라이언트 애플리케이션입니다. 다른 MQTT 클라이언트에서 온 메시지를 저장 후 전달하려면 이를 사용하십시오. 이는 MQTT 클라이언트처럼 WebSphere MQ에 연결하지만 다른 MQTT 클라이언트도 여기 연결할 수 있습니다. 다른 텔레메트리 디먼도 여기 연결할 수 있습니다.

이는 다음과 같은 네 가지 기본 목적을 충족시킵니다.

1. 로컬 MQTT 클라이언트들을 발행/구독 네트워크 안에 모아 연결합니다.

디바이스의 센서와 작동 장치를 서로 다른 MQTT 클라이언트로서 디먼에 연결할 수 있습니다. 센서는 자신의 게이지 값을 발행하고 작동 장치는 값을 구독하여 여기에 기초해 자기 작동을 수정합니다. 값은 부분적으로 영향을 줍니다.

2. 어느 구독, 어느 메시지가 큐 관리자와 디바이스로 발행되었는지 필터합니다.

이전 예에서 WebSphere Message Broker 메시지 플로우가 디먼이 값을 발행한 토픽을 구독할 수 있습니다. 플로우는 웹 페이지를 업데이트하고 디바이스의 상태를 표시합니다. 디먼은 또한 작동 장치가 큐 관리자에게 작성한 구독을 전달합니다. WebSphere Message Broker 플로우는 작동 장치를 서비스하고 있는 MQTT 클라이언트가 구독하고 있는 토픽에 메시지를 발행합니다. MQTT 클라이언트는 디바이스 세팅을 수정합니다. 메시지 플로우는 WebSphere Message Broker HTTPInput 노드를 사용하여 웹 페이지를 시작할 수도 있습니다.

3. 다중 MQTT 클라이언트를 텔레메트리 서버에 대한 하나의 연결로 집중합니다.

각 디바이스가 따로 텔레메트리 서버에 연결하는 대신 디먼은 발행과 구독을 단일 TCP/IP 연결로 전달합니다. 디먼은 텔레메트리(MQXR) 서비스가 관리하는 TCP/IP 연결의 수를 줄입니다. 개별 MQTT 클라이언트가 디먼에 연결합니다. 개별 클라이언트는 큐 관리자에게 보이지 않습니다. 디먼은 자신에게 연결되어 있는 모든 클라이언트를 대신하여 큐 관리자와 하나의 연결을 작성합니다.

4. 디바이스와 큐 관리자 사이에서 메시지를 저장 후 전달합니다.

디먼은 큐 관리자와의 연결 중 단명 연결 실패로부터 텔레메트리 디바이스를 보호합니다. 디바이스는 "전송 후 삭제" 메시징만을 지원할 수 있습니다. 큐 관리자와의 연결이 단속적으로만 사용 가능할 경우나 안정적이지 못할 경우 디바이스는 정보를 예상 가능하며 안정적으로 전송할 방법이 없습니다. 솔루션은 디바이스를 항상 사용 가능한 로컬 연결을 사용하는 디먼에 연결하는 것입니다. 디먼은 자신의 메모리에 큐 관리자에 오고가는 메시지를 중개할 수 있습니다. 안정적이지 못한 연결의 큐 관리자와 안정적인 서비스 품질을 사용하여 메시지를 주고 받을 수 있습니다.

참고: 디먼에는 "인플라이트" 메시지에 대한 지속적 스토리지가 없습니다. 메시지는 메모리에 저장됩니다.

MQTT 상태 비저장 및 상태 저장 세션

MQTT 클라이언트는 큐 관리자를 사용하여 상태 저장 세션을 작성할 수 있습니다. 상태 저장 MQTT 클라이언트의 연결이 끊어지면 큐 관리자가 클라이언트에 의해 작성된 구독 및 작업 시 메시지를 유지합니다. 클라이언트가 다시 연결되면 인플라이트 메시지를 해석합니다. 전달을 위해 큐에 대기된 모든 메시지를 보내고 연결이 끊어진 동안 구독에 대해 발행한 모든 메시지를 수신합니다.

MQTT 클라이언트가 Telemetry 채널에 연결되면 새 세션을 시작하거나 기존 세션을 재개합니다. 새 세션에는 수신확인되지 않은 미해결 메시지, 구독 및 전달되기를 기다리는 발행이 포함되어 있지 않습니다. 클라이언트가 연결되면 정리 세션으로 시작하는지 또는 기존 세션을 재개하는지 여부를 지정합니다. [정리 세션](#)을 참조하십시오.

클라이언트가 기존 세션을 재개하면 연결이 끊어지지 않았던 것처럼 계속 진행됩니다. 전달을 기다리는 발행은 클라이언트로 전송되고 커밋되지 않은 메시지 전송이 완료됩니다. 지속 세션의 클라이언트와 텔레메트리(MQXR) 서비스와의 연결이 끊어질 때 클라이언트가 작성한 모든 구독은 그대로 남아 있습니다. 클라이언트가 다시 연결되면 해당 클라이언트에게 구독에 대한 발행이 전송됩니다. 이전 세션을 재개하지 않고 다시 연결하면 텔레메트리(MQXR) 서비스가 발행물을 제거합니다.

큐 관리자가 `SYSTEM.MQTT.PERSISTENT.STATE` 큐에 세션 상태 정보를 저장합니다.

WebSphere MQ 관리자는 세션의 연결을 끊고 세션을 제거할 수 있습니다.

MQTT 클라이언트가 연결되지 않았을 때

클라이언트가 연결되지 않았을 때 큐 관리자는 이를 대신해 발행을 계속 수신할 수 있습니다. 발행은 클라이언트가 다시 연결할 때 여기로 전달됩니다. 클라이언트는 예상치 못하게 연결이 끊어질 경우 큐 관리자가 자신을 대신해 발행하는 "이상 종료 시 메시지"를 작성할 수 있습니다.

클라이언트가 예상치 않게 연결을 끊었을 때 알림을 받으려면 마지막 유언장 및 유언장을 등록할 수 있습니다. [마지막 유언장 및 유언장 공개](#)를 참조하십시오. 클라이언트에 대한 연결이 클라이언트의 요청 없이 중단되었음을 감지한 경우, 이는 텔레메트리(MQXR) 서비스에 의해 전송됩니다.

클라이언트는 언제라도 보유된 발행물을 공개할 수 있습니다. [보유된 공개 및 MQTT 클라이언트](#)를 참조하십시오. 토픽에 대한 새 구독은 토픽과 연관된 어떤 보유된 발행물도 송신해 달라고 요청할 수 있습니다. 이상 종료 시 메시지를 보유된 발행물로서 작성할 경우 이를 클라이언트 상태를 모니터링하는 데 사용할 수 있습니다.

예를 들면, 한 클라이언트는 연결할 때 자신의 가용성을 알리며 보유된 발행물을 발행합니다. 동시에, 비가용성을 알리는 보유된 이상 종료 시 메시지 발행을 작성합니다. 또한 계획된 연결 종료를 하기 전에 보유된 발행물로서 자신의 비가용성을 발행합니다. 클라이언트가 사용 가능한지 알아보려면 이 보유된 발행의 토픽을 구독합니다. 사용자는 항상 세 가지 발행 중 하나를 수신합니다.

클라이언트가 연결이 끊어져 있을 때 publish된 메시지를 수신하고자 하는 경우 클라이언트를 이전 세션에 다시 연결하십시오. [90 페이지의 『MQTT 상태 비저장 및 상태 저장 세션』](#)을 참조하십시오. 그 구독은 삭제될 때까지 또는 클라이언트가 정리 세션을 작성할 때까지 활성 상태입니다.

MQTT 클라이언트와 WebSphere MQ 애플리케이션 사이의 느슨한 결합

MQTT 클라이언트와 WebSphere MQ 애플리케이션 사이의 발행 흐름은 느슨히 결합되어 있습니다. 발행은 MQTT 클라이언트 또는 WebSphere MQ 애플리케이션에서 설정된 순서 없이 생성될 수 있습니다. 발행자와 구독자는 느슨히 결합되어 있습니다. 이들은 발행과 구독을 통해 서로 상호작용합니다. 또한 WebSphere MQ 애플리케이션에서 MQTT 클라이언트로 메시지를 직접 송신할 수 있습니다.

MQTT 클라이언트와 WebSphere MQ 애플리케이션은 다음과 같은 두 가지 측면에서 느슨히 결합되어 있습니다.

1. 발행자와 구독자는 발행, 구독과 토픽의 연관을 통해 느슨히 결합되어 있습니다. 발행자와 구독자는 보통 다른 발행 또는 구독 소스의 주소나 ID를 알고 있습니다.
2. MQTT 클라이언트는 발행을 발행, 구독, 수신하며 분리된 스레드에서 전달 수신확인을 처리합니다.

MQTT 클라이언트 애플리케이션은 발행이 전달될 때까지 대기하지 않습니다. 애플리케이션은 MQTT 클라이언트로 메시지를 전달하고 그 후 애플리케이션은 자신의 스레드에서 계속됩니다. 발행물의 전달과 애플리케이션을 동기화하기 위해 전달 토큰이 사용됩니다. [전달 토큰](#)을 참조하십시오.

MQTT 클라이언트에 메시지를 전달한 후 애플리케이션은 전달 토큰을 기다릴지 선택할 수 있습니다. 기다리는 대신 클라이언트는 WebSphere MQ에 발행이 전달되었을 때 호출되는 콜백 메소드를 제공할 수 있습니다. 이는 또한 전달 토큰을 무시할 수 있습니다.

메시지의 서비스 품질에 따라 전달 토큰은 콜백 메소드로 즉시 리턴되거나 상당한 시간이 지난 뒤 리턴될 수 있습니다. 전달 토큰은 클라이언트 연결이 끊어진 후 다시 연결되었을 때 리턴될 수도 있습니다. 서비스 품질이 "전송 후 삭제"인 경우 전달 토큰은 즉시 리턴됩니다. 다른 두 경우 전달 토큰은 클라이언트가 발행이 구독자로 송신되었다는 수신확인을 수신했을 때만 리턴됩니다.

클라이언트 구독의 결과로 MQTT 클라이언트로 송신된 발행은 `messageArrived` 콜백 메소드로 전달됩니다. `messageArrived`는 기본 애플리케이션과 다른 스레드에서 실행됩니다.

MQTT 클라이언트로 직접 메시지 송신

둘 중 한 가지 방법으로 특정 MQTT 클라이언트에 메시지를 송신할 수 있습니다.

1. WebSphere MQ 응용프로그램은 등록 없이 메시지를 직접적으로 MQTT 클라이언트로 전송할 수 있습니다. [클라이언트에 직접 메시지 보내기](#)를 참조하십시오.
2. 또 다른 방법은 `ClientIdentifier` 이름 지정 규칙을 사용하는 것입니다. 모든 MQTT 구독자가 자신의 고유한 `ClientIdentifier`를 토픽으로 사용해 구독을 작성하도록 하십시오. `ClientIdentifier`에 발행하십시오. 발행은 토픽 `ClientIdentifier`를 구독한 클라이언트에게 송신됩니다. 이 기술을 사용하면 발행을 특정 MQTT 구독자에게 송신할 수 있습니다.

WebSphere MQ Telemetry 보안

텔레메트리 디바이스는 대부분 휴대용이며 잘 제어되지 않은 환경에서 사용하게 될 경우가 많으므로 디바이스를 보호하는 것은 중요합니다. VPN을 사용하여 MQTT 디바이스에서 텔레메트리(MQXR) 서비스로 이어지는 연결을

보호할 수 있습니다. WebSphere MQ Telemetry는 SSL과 JAAS라는 다른 두 가지 보안 메커니즘을 더 제공합니다.

SSL은 기본적으로 디바이스 및 텔레메트리 채널 간의 통신을 암호화하며 디바이스가 올바른 서버에 연결 중인지 인증하기 위해 사용됩니다. SSL을 사용한 텔레메트리 채널 인증을 참조하십시오. SSL을 사용하여 클라이언트 디바이스가 서버에 연결할 수 있는지를 확인할 수도 있습니다. SSL을 사용한 MQTT 클라이언트 인증을 참조하십시오.

JAAS는 기본적으로 디바이스의 사용자가 서버 애플리케이션을 사용할 수 있는지 확인하는 데 사용됩니다. 비밀번호를 사용한 MQTT 클라이언트 인증을 참조하십시오. JAAS는 싱글 사인온 디렉토리를 사용하는 비밀번호를 확인하기 위해 LDAP와 함께 사용할 수 있습니다.

SSL 및 JAAS는 두 가지 인증을 제공하기 위해 결합해 사용할 수 있습니다. SSL에서 사용하는 암호를 FIPS 표준을 만족시키는 암호로 제한할 수 있습니다.

최소 수만 명의 사용자가 있을 경우 개별 보안 프로파일을 제공하는 것은 실용적이지 않을 수 있습니다. WebSphere MQ 오브젝트에 액세스하려는 개별 사용자를 인증하는 데 프로파일을 사용하는 것 또한 마찬가지입니다. 대신 토픽에 대한 발행 및 구독 인증과, 클라이언트로의 발행물 송신을 위해 사용자를 클래스로 그룹 지으십시오.

클라이언트를 공용 클라이언트 사용자 ID로 맵핑하도록 각 텔레메트리 채널을 구성하십시오. 특정 채널에서 연결하는 모든 클라이언트에 대해 공용 사용자 ID를 사용하십시오. MQTT 클라이언트 ID 및 권한을 참조하십시오.

사용자 그룹에 권한을 부여하는 것이 각 개인 인증에 문제를 일으키지는 않습니다. 각 개별 사용자는 클라이언트와 서버에서 자신의 Username과 Password로 인증을 받은 후 공용 사용자 ID를 사용하는 서버에서 권한 부여를 받을 수 있습니다.

WebSphere MQ Telemetry 다국어 지원

MQTT v3 프로토콜의 메시지 페이로드는 바이트 배열로 인코딩됩니다. 일반적으로 텍스트를 처리하는 애플리케이션은 UTF-8로 메시지 페이로드를 작성합니다. 텔레메트리 채널은 메시지 페이로드를 UTF-8로 설명하지만 어떤 코드 페이지 변환도 수행하지 않습니다. 발행 토픽 문자열은 UTF-8이어야 합니다.

영문자 데이터를 올바른 코드 페이지로, 숫자 데이터를 올바른 숫자 인코딩으로 변환하는 것은 애플리케이션의 역할입니다.

MQTT Java 클라이언트에는 편리한 `MqttMessage.toString` 메소드가 있습니다. 이 메소드는 메시지 페이로드를 로컬 플랫폼의 기본 문자 세트(일반적으로 UTF-8)로 인코딩된 것으로 취급합니다. 이는 페이로드를 Java 문자열로 변환한다. Java에는 문자열 메소드인 `getBytes`가 있으며, 이는 문자열을 로컬 플랫폼 기본 문자 세트를 사용하여 인코딩된 바이트 배열로 변환합니다. 메시지 페이로드에서 텍스트를 교환하는 두 개의 MQTT Java 프로그램은 동일한 기본 문자 세트가 있는 플랫폼 간에 UTF-8에서 쉽고 효율적으로 수행됩니다.

한 플랫폼의 기본 문자 세트가 UTF-8이 아닐 경우에 애플리케이션은 메시지를 교환하는 데 변환을 설정해야 합니다. 예를 들면, 발행자가 `getBytes("UTF8")` 메소드를 사용하여 문자열에서 UTF-8로의 변환을 지정하는 경우가 있습니다. 메시지의 텍스트를 수신하기 위해 구독자는 메시지가 UTF-8 문자 세트로 인코딩되어 있다고 추정합니다.

텔레메트리(MQXR) 서비스는 MQTT 클라이언트 메시지에서 수신되는 모든 발행물의 인코딩이 UTF-8임을 설명합니다. 이는 `MQMD.CodedCharSetId`를 UTF-8로, `RFH2.CodedCharSetId`를 `MQCCSI_INHERIT`로 설정합니다. 87 페이지의 『큐 관리자와 WebSphere MQ Telemetry의 통합』의 내용을 참조하십시오. 발행의 형식은 `MQFMT_NONE`으로 설정되어 채널이나 `MQGET`에 의해 변환이 수행되지 않도록 합니다.

WebSphere MQ Telemetry의 성능 및 확장성

다수의 클라이언트를 관리하고 WebSphere MQ Telemetry의 확장성을 향상시킬 때 다음과 같은 요인을 고려하십시오.

용량 계획

WebSphere MQ 텔레메트리에 대한 성능 보고서에 대한 정보는 WebSphere MQ 제품군-성능 보고서에서 WebSphere MQ 텔레메트리 성능 평가 보고서를 선택하십시오.

연결

연결과 관련된 비용은 다음을 포함하고 있습니다.

- 프로세서 사용 및 시간의 관점에서 본 연결 자체를 설정하는 비용.
- 네트워크 비용.
- 연결을 열어놓고 사용하지 않는 동안 사용되는 메모리.

클라이언트가 연결된 채일 때 발생하는 추가적인 부하가 있습니다. 연결이 열린 채인 경우 TCP/IP에는 플로우가 계속 발생하며 MQTT 메시지는 연결이 계속 있는지 확인하기 위해 네트워크를 사용합니다. 또한 서버에서 열려 있는 채인 각 클라이언트 연결에 대해 메모리가 사용됩니다.

1분에 하나 이상 메시지를 송신할 경우 새 연결을 시작하는 비용을 피하기 위해 연결을 계속 열어두십시오. 10 - 15분 동안 하나 이하의 메시지를 송신할 경우 연결을 열어두는 비용을 피하기 위해 연결을 끄는 것을 고려하십시오. SSL 연결은 설정하는 데 비용이 많이 소요되기 때문에 열어둔 채로 두는 편이 좋을 수 있습니다.

추가적으로 클라이언트의 기능을 고려하십시오. 클라이언트에 저장 후 전달 기능이 있는 경우 메시지를 배치해 두고 배치를 송신하는 사이에 연결을 끊을 수 있습니다. 그러나 클라이언트의 연결이 끊긴 경우에는 클라이언트가 서버에서 메시지를 수신할 수 없습니다. 따라서 애플리케이션의 목적이 의사결정에 영향을 미칩니다.

파일 전송과 같이 시스템에 많은 메시지를 송신하는 클라이언트가 있을 경우 메시지마다 서버 응답을 기다리지 마십시오. 대신 모든 메시지를 송신하고 마지막에 메시지가 모두 수신되었는지 확인하십시오. 또는 [QoS\(Quality of Service\)](#)를 사용하십시오.

중요하지 않은 메시지는 QoS 0을 사용하여 전달하고 중요한 메시지는 QoS 2를 사용하여 전달하는 것처럼 메시지마다 QoS를 달리 할 수 있습니다. 메시지 처리량은 QoS가 2일 때보다 QoS가 0일 때 약 두 배 더 높을 수 있습니다.

이름 지정 규칙

다수의 클라이언트를 대상으로 애플리케이션을 설계할 경우 효율적인 이름 지정 규칙을 구현하십시오. 각 클라이언트를 올바르게 `ClientIdentifier`에 맵핑하려면 `ClientIdentifier`를 의미있게 작성하십시오. 좋은 이름 지정 규칙은 어느 클라이언트가 실행 중인지 관리자가 쉽게 알 수 있게 해 줍니다. 이름 지정 규칙은 관리자가 WebSphere MQ Explorer에서 긴 클라이언트 목록을 필터링하고 문제점을 판별하는 데 도움이 됩니다. [클라이언트 ID](#)를 참조하십시오.

처리량

토픽 이름의 길이는 네트워크를 흘러가는 바이트 수에 영향을 줍니다. 발행하거나 구독할 때 메시지의 바이트 수는 중요할 수 있습니다. 그러므로 토픽 이름의 문자 개수를 제한하십시오. MQTT 클라이언트가 토픽을 구독할 때 WebSphere MQ는 여기에 다음과 같은 형식의 이름을 부여합니다.

```
ClientIdentifier:TopicName
```

MQTT 클라이언트의 모든 구독을 보려면 WebSphere MQ MQSC **DISPLAY** 명령을 사용할 수 있습니다.

```
DISPLAY SUB('ClientID1:*')
```

WebSphere MQ에 MQTT 클라이언트에서 사용하기 위한 자원 정의

MQTT 클라이언트는 IBM WebSphere MQ 리모트 큐 관리자에 연결됩니다. WebSphereMQ 애플리케이션이 MQTT 클라이언트에 메시지를 송신하는 기본 방법이 두 가지 있습니다. 기본 전송 큐를 `SYSTEM.MQTT.TRANSMIT.QUEUE`로 설정하거나 큐 관리자 알리아스를 사용하는 것입니다. 다수의 MQTT 클라이언트가 있을 경우 큐 관리자에 기본 전송 큐를 정의하십시오. 기본 전송 큐 설정을 사용하면 관리 작업이 단 순화됩니다. [MQTT 클라이언트로 메시지를 송신하도록 분산 큐잉 구성](#)을 참조하십시오.

구독하지 않음으로서 오는 확장성 향상

MQTT V3 클라이언트가 토픽을 구독하는 경우, 구독은 WebSphere MQ의 텔레메트리(MQXR) 서비스에 의해 작성됩니다. 구독은 클라이언트가 구독한 발행을 `SYSTEM.MQTT.TRANSMIT.QUEUE`로 라우팅합니다. 각 발행의

전송 헤더에 있는 리모트 큐 관리자 이름은 구독을 작성한 MQTT 클라이언트의 ClientIdentifier로 설정됩니다. 각각 자신의 구독을 작성하고 있는 다수의 클라이언트가 있을 경우 WebSphere MQ 발행/구독 클러스터 또는 계층 전반에 많은 프록시 구독이 유지되는 결과가 발생합니다. 발행/구독을 사용하는 대신 포인트-투-포인트 기반 솔루션 사용에 대한 정보는 [클라이언트에 직접 메시지 보내기를 참조하십시오](#).

다수의 클라이언트 관리

다수의 동시에 연결된 클라이언트를 지원하려면 JVM 매개변수 **-Xms** 및 **-Xmx**를 설정하여 텔레메트리(MQXR) 서비스에 사용할 수 있는 메모리를 늘리십시오. 다음 단계를 수행하십시오.

1. 텔레메트리 서비스 구성 디렉토리에서 `java.properties` 파일을 찾으십시오. [Windows의 텔레메트리\(MQXR\) 서비스 구성 디렉토리](#) 또는 [Linux의 텔레메트리 서비스 구성 디렉토리](#)를 참조하십시오.
2. 파일의 지시사항을 따르십시오. 50,000개의 동시 연결된 클라이언트를 유지하는 데는 1GB 힙 정도면 충분합니다.

```
# Heap sizing options - uncomment the following lines to set the heap to 1G
#-Xmx1024m
#-Xms1024m
```

3. `java.properties` 파일에 텔레메트리(MQXR) 서비스를 실행 중인 JVM에 전달할 다른 명령행 인수를 추가하십시오. [텔레메트리\(MQXR\) 서비스에 JVM 매개변수 전달을 참조하십시오](#).

Linux에서 열린 파일 디스크립터의 수를 늘리려면 `/etc/security/limits.conf/`에 다음 행을 추가하고 다시 로그인하십시오.

```
@mqm soft nofile 65000
@mqm hard nofile 65000
```

각 소켓은 하나의 파일 디스크립터를 필요로 합니다. 텔레메트리 서비스는 몇몇 추가 파일 디스크립터를 필요로 하며 따라서 이 숫자는 요구되는 열린 소켓의 개수보다 커야 합니다.

큐 관리자는 각 지속 불가능 구독에 오브젝트 핸들을 사용합니다. 많은 활성 지속 불가능 구독을 지원하려면 큐 관리자에서 활성 핸들의 최대 개수를 증가시키십시오. 예:

```
echo ALTER QMGR MAXHANDS(99999999) | runmqsc qMgrName
```

그림 26. Windows에서 핸들 최대 개수 변경

```
echo "ALTER QMGR MAXHANDS(99999999)" | runmqsc qMgrName
```

그림 27. Linux에서 최대 핸들 수 대체

기타 고려사항

시스템 요구사항을 계획할 때 다시 시작하는 데 드는 시간의 길이를 고려하십시오. 계획된 중단 시간은 처리될 때까지 큐에 들어가 대기하는 메시지의 개수에 영향을 줄 수 있습니다. 메시지가 허용 가능한 범위의 시간 안에 처리 완료될 수 있도록 시스템을 구성하십시오. 디스크 스토리지, 메모리 및 처리 능력을 검토하십시오. 몇몇 클라이언트 애플리케이션의 경우 클라이언트가 다시 연결할 때 메시지를 버릴 수도 있습니다. 메시지를 제거하려면 클라이언트 연결 매개변수의 CleanSession을 설정하십시오. 정리 세션을 참조하십시오. 또는 MQTT 클라이언트에서 가장 효율적인 서비스 품질인 0을 사용하여 발행과 구독하십시오. [서비스 품질을 참조하십시오](#).

WebSphere MQ에서 메시지를 송신할 때 non-persistent 메시지를 사용하십시오. 이 서비스 품질의 메시지는 시스템이나 연결이 다시 시작할 때 복구되지 않습니다.

IBM WebSphere MQ Telemetry에서 지원되는 디바이스

MQTT 클라이언트는 센서 및 작동기부터 휴대용 디바이스 및 차량 시스템까지 다양한 범위의 디바이스에서 시행할 수 있습니다.

MQTT 클라이언트는 크기가 작으며, 적은 메모리를 갖고 처리 능력이 낮은 디바이스에서 실행됩니다. MQTT 프로토콜은 안정적이며 헤더가 작고, 이는 대역폭이 낮고 사용 비용이 높으며 단속적인 네트워크에 적합합니다.

IBM WebSphere MQ Telemetry에서는 다음 세 가지 클라이언트를 제공하고 이들 모두는 MQTT v3 프로토콜을 구현합니다.

- 가장 작은 CLDC (Connected Limited Device Configuration) /MIDP (Mobile Information Device Profile) 에서 CDC (Connected Device Configuration) /Foundation, J2SE (Java Platform, Standard Edition) 및 J2EE (Java Platform, Enterprise Edition) 에서 Java의 모든 변형에서 실행할 수 있는 Java 클라이언트입니다. IBM jclIRM 사용자 정의된 클래스 라이브러리 또한 지원됩니다.
- C 참조 구현은 Linux 시스템용 사전 빌드된 기본 클라이언트와 함께 구현됩니다. C 참조 구현은 MQTT가 다양한 디바이스와 플랫폼으로 포트될 수 있도록 해 줍니다.
- 고급 클라이언트인 IBM WebSphere MQ 디바이스용 텔레메트리 디먼은 C로 작성되며 적합한 Linux 또는 플랫폼 실행할 수 있습니다.

Intel의 일부 윈도우 시스템 (예: 윈도우 XP, RedHat, Ubuntu) 및 ARM 플랫폼의 일부 Linux 시스템 (예: Eurotech Viper는 C 클라이언트를 실행하는 Linux 버전을 구현하지만 IBM 은 플랫폼에 대한 서비스 지원을 제공하지 않음) 사용자의 IBM 지원 센터를 호출하려면 지원되는 플랫폼에서 클라이언트로 다시 이 문제점을 다시 발생시켜야 합니다.

자바 ME 플랫폼은 일반적으로 작동기, 센서, 이동 전화 및 기타 임베디드 장치와 같은 소형 장치에서 사용된다. 일반적으로 Java SE 플랫폼은 데스크탑 컴퓨터 및 서버와 같은 고급 임베디드 디바이스에 설치된다.

참고: Eclipse paho 프로젝트 및 mqtt.org에서는 다양한 프로그래밍 언어에 대한 샘플 및 최신 텔레메트리 클라이언트를 무료로 다운로드할 수 있습니다.

관련 개념

[89 페이지의 『디바이스의 텔레메트리 디먼』](#)

디바이스용 WebSphere MQ Telemetry 디먼은 고급 MQTT V3 클라이언트 애플리케이션입니다. 다른 MQTT 클라이언트에서 온 메시지를 저장 후 전달하려면 이를 사용하십시오. 이는 MQTT 클라이언트처럼 WebSphere MQ 에 연결하지만 다른 MQTT 클라이언트도 여기 연결할 수 있습니다. 다른 텔레메트리 디먼도 여기 연결할 수 있습니다.

오브젝트 관리

큐 관리자는 IBM WebSphere MQ 오브젝트의 특성 또는 속성을 정의합니다. 이들 속성의 값은 WebSphere MQ 가 이러한 오브젝트를 처리하는 방법에 영향을 줍니다. 애플리케이션에서 MQI(Message Queue Interface)를 사용하여 이들 오브젝트를 제어합니다. 오브젝트는 프로그램에서 취급할 때 MQOD(MQ object descriptor)에서 식별합니다.

WebSphere MQ 명령을 실행하여 오브젝트 관리 조작(예: 오브젝트 정의, 수정 또는 삭제)를 수행하는 경우 큐 관리자에 조작을 수행하는 데 필요한 필수 권한 레벨이 있는지 확인하십시오. 마찬가지로, 애플리케이션이 MQOPEN 호출을 사용하여 오브젝트를 열 때 큐 관리자는 애플리케이션이 필요한 권한 레벨을 가지고 있는지 확인한 후 해당 오브젝트에 대한 액세스를 허용합니다. 열려 있는 오브젝트의 이름에 대해 확인합니다.

오브젝트 관리에는 다음 태스크가 포함됩니다.

- 큐 관리자 시작 및 중지
- 애플리케이션용 오브젝트(특히 큐)를 작성
- 다른(원격) 시스템에 있는 큐 관리자에 대한 통신 경로를 작성하기 위해 채널에 대한 작업
- 큐 관리자의 클러스터를 작성하여 전체 관리 프로세스를 단순화하고 워크로드를 밸런스화함

WebSphere MQ 오브젝트 작성 및 관리에 대한 메소드 개요에 대해서는 [114 페이지의 『오브젝트 관리』](#)의 내용을 참조하십시오.

오브젝트 관리 태스크에 대한 추가 세부사항은 다음 하위 주제를 참조하십시오.

오브젝트

관리 태스크 중 다수가 다양한 IBM WebSphere MQ 오브젝트 유형의 조작을 포함합니다.

IBM WebSphere MQ 오브젝트 이름 지정에 대한 정보는 [110 페이지의 『IBM WebSphere MQ 오브젝트 이름 지정』](#)의 내용을 참조하십시오.

큐 관리자에 작성된 기본 오브젝트에 대한 정보는 [115 페이지의 『시스템 기본 오브젝트』](#)의 내용을 참조하십시오.

다른 유형의 IBM WebSphere MQ 오브젝트에 대한 정보는 다음 하위 주제를 참조하십시오.

관련 개념

[39 페이지의 『메시지 큐잉 소개』](#)

WebSphere MQ 제품을 사용하면 프로그램이 일관된 애플리케이션 프로그래밍 인터페이스(API)를 사용하여 다른 컴포넌트(프로세서, 운영 체제, 서브시스템 및 통신 프로토콜)의 네트워크에서 서로 통신할 수 있습니다.

[115 페이지의 『오브젝트 속성』](#)

오브젝트의 특성은 그 속성으로 정의됩니다. 일부는 지정할 수 있고 그 외에는 보기만 가능합니다.

관련 참조

[MQSC 명령](#)

큐

WebSphere MQ 큐 및 큐 속성에 대한 소개입니다.

WebSphere MQ 큐는 애플리케이션이 메시지를 넣을 수 있고 애플리케이션이 메시지를 가져올 수 있는 이름 지정된 오브젝트입니다.

메시지가 큐에 저장되어 넣기 애플리케이션이 해당 메시지에 대한 응답을 기다리는 경우 이 응답을 기다리는 동안 다른 작업을 할 수 있습니다. 애플리케이션은 [메시지 큐 인터페이스 개요](#)에 설명된 MQI(Message Queue Interface)를 사용하여 큐에 액세스합니다.

메시지를 큐에 넣기 전에 먼저 큐가 작성되어 있어야만 합니다. 큐 관리자가 큐를 소유하며 해당 큐 관리자는 다수의 큐를 소유할 수 있습니다. 하지만 각 큐는 큐 관리자 내에서 고유한 이름을 가져야 합니다.

큐는 큐 관리자를 통해 유지보수됩니다. 대부분의 경우 각 큐는 해당 큐 관리자가 실제로 관리하지만 애플리케이션 프로그램에 대해서는 분명하지 않습니다. WebSphere MQ for z/OS shared queues can be managed by any queue manager in the queue-sharing group.

To create a queue you can use WebSphere MQ commands (MQSC), PCF commands, or platform-specific interfaces such as the WebSphere MQ for z/OS operations and control panels.

애플리케이션에서 임시 작업용 로컬 큐를 동적으로 작성할 수 있습니다. 예를 들어, 응답 대상 큐를 작성할 수 있습니다(애플리케이션이 종료된 후에는 필요하지 않음). 자세한 정보는 [101 페이지의 『동적 및 모델 큐』](#)의 내용을 참조하십시오.

큐를 사용하기 전에 큐를 열어 큐에 대해 수행하려는 작업을 지정해야 합니다. 예를 들어, 다음의 작업을 위해 큐를 열 수 있습니다.

- 메시지 찾아보기 전용(메시지 검색은 아님)
- 메시지 검색(및 다른 프로그램과 액세스 공유 또는 독점 액세스 사용)
- 큐에 메시지 넣기
- 큐의 속성 조회
- 큐의 속성 설정

큐를 열 때 지정할 수 있는 전체 옵션 목록은 [MQOPEN - 오브젝트 열기](#)를 참조하십시오.

큐의 속성

큐의 일부 속성은 큐가 정의될 때 지정되며 이후에 변경할 수 없습니다(예: 큐의 유형). 다른 큐 속성은 변경할 수 있는 속성으로 그룹화할 수 있습니다.

- 큐 처리 중에 큐 관리자에 의해 지정(예: 큐의 현재 용량)
- 명령에 의해서만 지정(예: 큐의 텍스트 설명)
- MQSET 호출을 사용하는 애플리케이션에 의해 지정(예: Put 조작이 큐에서 허용되는지 여부)

MQING 호출을 사용하여 모든 속성의 값을 찾을 수 있습니다.

둘 이상의 큐 유형에 공용인 속성은 다음과 같습니다.

QName

큐의 이름

QType

큐의 유형

QDesc

큐에 대한 텍스트 설명

InhibitGet

프로그램이 큐에서 메시지를 가져오도록 허용하는지 여부(리모트 큐에서 메시지를 가져올 수 없는 경우에도 해당됨)

InhibitPut

프로그램이 큐에 메시지를 넣도록 허용하는지 여부

DefPriority

큐에 넣은 메시지의 기본 우선순위

DefPersistence

큐에 넣은 메시지의 기본 지속성

Scope(z/OS에서는 지원되지 않음)

이 큐에 대한 항목이 이름 서비스에 존재하는지 여부를 제어

이 속성에 대한 자세한 설명은 [큐의 속성을 참조하십시오](#).

관련 개념**99 페이지의 『리모트 큐』**

프로그램에 있어서 큐가 다른 큐 관리자에 의해 소유되는 경우 큐는 프로그램이 연결되는 큐 관리자에 대해 리모트입니다.

99 페이지의 『알리어스 큐』

알리어스 큐는 다른 큐 또는 토픽에 액세스하는 데 사용할 수 있는 WebSphere MQ 오브젝트입니다. 이는 둘 이상의 프로그램이 다른 이름을 사용하여 액세스하여 동일한 큐에 대해 작업할 수 있음을 의미합니다.

103 페이지의 『큐 정의』

MQSC 명령인 DEFINE 또는 PCF의 큐 작성 명령을 사용하여 IBM WebSphere MQ에 큐를 정의합니다.

103 페이지의 『IBM WebSphere MQ에서 사용되는 큐』

IBM WebSphere MQ에서는 그 조작과 관련된 특정 목적의 로컬 큐를 사용합니다.

관련 태스크

[애플리케이션 참조 개발](#)

관련 참조**MQSC 명령****97 페이지의 『로컬 큐』**

전송, 시작, 데드 레터, 명령, 기본값, 채널 및 이벤트 큐는 로컬 큐의 유형입니다.

100 페이지의 『공유 및 클러스터 큐』

이 정보는 공유 큐와 클러스터 큐라는 용어를 정의하고 설명하며 둘 사이의 비교를 제공합니다.

101 페이지의 『동적 및 모델 큐』

이 정보는 동적 큐, 임시 및 영구적 동적 큐의 특성, 동적 큐의 사용, 동적 큐를 사용할 때의 고려사항 및 모델 큐에 대한 통찰을 제공합니다.

로컬 큐

전송, 시작, 데드 레터, 명령, 기본값, 채널 및 이벤트 큐는 로컬 큐의 유형입니다.

프로그램이 연결된 큐 관리자가 큐를 소유하고 있는 경우 큐는 프로그램에 로컬로 표시됩니다. 메시지를 로컬 큐에서 가져오거나 로컬 큐에 넣을 수 있습니다.

큐 정의 오브젝트는 큐에 넣은 실제 메시지뿐만 아니라 큐 정의 정보도 포함하고 있습니다.

각 큐 관리자는 특수 용도로 사용하는 몇몇 로컬 큐를 가질 수 있습니다.

전송 큐

애플리케이션이 리모트 큐로 메시지를 송신할 때 로컬 큐 관리자는 전송 큐라는 특수 로컬 큐에 메시지를 저장합니다.

메시지 채널 에이전트는 전송 큐와 연관된 채널 프로그램으로서 메시지를 다음 목적지로 전달합니다. 다음 목적지란 메시지 채널이 연결된 큐 관리자를 말합니다. 메시지의 마지막 목적지와 큐 관리자는 다를 수 있습니다. 메시지가 다음 목적지로 전달되면, 전송 큐에서 삭제됩니다. 메시지는 최종 목적지에 도달하는 과정에서 많은 큐 관리자를 통과해야 할 수 있습니다. 라우트에 따라 각각 다음 목적지로 전송하기 위해 대기 중인 메시지를 보유하는 각 큐 관리자에 전송 큐를 정의해야 합니다. 일반적인 전송 큐는 메시지의 최종 목적지가 서로 달라도 다음 목적지에 대한 메시지를 보유합니다. 클러스터 전송 큐는 여러 목적지에 대한 메시지를 보유합니다. 각 메시지의 `correlID`는 다음 목적지로 전송할 메시지가 있는 채널을 식별합니다.

큐 관리자에 여러 전송 큐를 정의할 수 있습니다. 다른 클래스의 서비스에 사용 중인 각 전송 큐가 있는 동일한 목적지에 대해 여러 전송 큐를 정의할 수도 있습니다. 예를 들어, 동일한 목적지로 가는 소형 메시지와 대형 메시지에 각각 다른 전송 큐를 작성할 수 있습니다. 서로 다른 메시지 채널을 사용하여 메시지를 전송하면 대형 메시지로 인한 소형 메시지 지연이 발생하지 않습니다. z/OS이외의 플랫폼에서는 메시지가 기본적으로 단일 클러스터 전송 큐 (SYSTEM.CLUSTER.TRANSMIT.QUEUE) 로 이동합니다. 기타 플랫폼에 적용된 옵션에 따라 기본값을 변경하여 다른 클러스터 큐 관리자로 가는 메시지 트래픽을 다른 클러스터 전송 큐로 분리할 수 있습니다. 큐 관리자 속성 DEFCLXQ 를 CHANNEL로 설정하면 각 클러스터 송신자 채널이 별도의 클러스터 전송 큐를 작성합니다. 기타 플랫폼에서 다른 옵션은 클러스터 송신자 채널에서 사용할 클러스터 전송 큐를 수동으로 정의하는 것입니다.

전송 큐는 메시지 채널 에이전트를 트리거하여 메시지를 전송할 수 있습니다 ([트리거를 사용하여 WebSphere MQ 애플리케이션 시작 참조](#)).

이니시에이션 큐

이니시에이션 큐는 애플리케이션 큐에서 트리거 이벤트가 발생할 때 큐 관리자가 트리거 메시지를 넣는 로컬 큐입니다.

트리거 이벤트는 프로그램이 큐 처리를 시작하도록 하는 이벤트입니다. 예를 들어 이벤트는 10개가 넘는 수신 메시지가 될 수 있습니다. 트리거링 작동 방법에 대한 자세한 정보는 [트리거를 사용하여 WebSphere MQ 애플리케이션 시작](#)의 내용을 참조하십시오.

데드-레터(미전달 메시지) 큐

데드-레터(미전달 메시지) 큐는 큐 관리자가 전달할 수 없는 메시지를 넣는 로컬 큐입니다.

큐 관리자가 메시지를 데드-레터 큐에 넣을 때 메시지에 헤더를 추가합니다. 헤더 정보에는 큐 관리자가 데드-레터 큐에 메시지를 넣는 이유가 포함됩니다. 또한 원래 메시지의 목적지, 큐 관리자가 데드-레터 큐에 메시지를 넣는 날짜 및 시간도 포함됩니다.

애플리케이션은 전달할 수 없는 메시지에 대한 큐를 사용할 수도 있습니다. 자세한 정보는 [데드-레터\(미배달 메시지\) 큐 사용](#)을 참조하십시오.

시스템 명령 큐

시스템 명령 큐는 적절하게 권한 부여된 애플리케이션이 WebSphere MQ 명령을 송신할 수 있는 큐입니다. 이러한 큐는 플랫폼의 지원에 따라 PCF, MQSC 및 CL 명령을 수신하며 이러한 명령에 대해 조치하는 큐 관리자에 대해 준비된 상태입니다.

시스템 기본 큐

시스템 기본 큐에는 시스템에 대한 큐의 초기 정의가 포함됩니다. 큐 정의를 작성할 때, 큐 관리자는 적절한 시스템 기본 큐에서 정의를 복사합니다. 큐 정의 작성은 동적 큐 작성과 다릅니다. 동적 큐 정의는 동적 큐 템플릿으로 선택한 모델 큐를 기준으로 합니다.

이벤트 큐

이벤트 큐는 이벤트 메시지를 보유합니다. 이러한 메시지는 큐 관리자 또는 채널이 보고합니다.

리모트 큐

프로그램에 있어서 큐가 다른 큐 관리자에 의해 소유되는 경우 큐는 프로그램이 연결되는 큐 관리자에 대해 리모트입니다.

통신 링크가 설정된 위치에서 프로그램은 리모트 큐로 메시지를 송신할 수 있습니다. 프로그램은 리모트 큐에서 메시지를 가져올 수 없습니다.

리모트 큐를 정의할 때 작성된 큐 정의 오브젝트는 로컬 큐 관리자가 메시지를 송신하려는 큐를 찾는 데 필요한 정보만을 보유합니다. 이 오브젝트를 리모트 큐의 로컬 정의라고 합니다. 리모트 큐의 모든 속성은 리모트 큐를 소유하는 큐 관리자가 보유하며 이는 리모트 큐가 큐 관리자에게는 로컬 큐이기 때문입니다.

리모트 큐를 열 때 이 큐를 식별하려면 다음 중 하나를 지정해야 합니다.

- 리모트 큐를 정의하는 로컬 정의의 이름

리모트 큐의 로컬 정의를 작성하려면 DEFINE QREMOTE 명령을 사용하십시오. WebSphere MQ for IBM i에서 CRTMQMQ 명령을 사용하십시오.

애플리케이션의 관점에서 이는 로컬 큐를 여는 것과 같습니다. 애플리케이션은 큐가 로컬 또는 리모트인지를 알 필요가 없습니다.

- 해당 리모트 큐 관리자에게 알려진 대로의 리모트 큐 관리자의 이름 및 큐의 이름

리모트 큐의 로컬 정의에는 96 페이지의 『큐의 속성』에 설명된 공용 속성에 추가하여 세 가지 속성이 더 있습니다. 이는 *RemoteQName* (큐의 소유 큐 관리자가 이를 인식하는 이름), *RemoteQMGrName* (소유하는 큐 관리자의 이름) 및 *XmitQName* (다른 큐 관리자로 메시지를 전달할 때 사용되는 로컬 전송 큐의 이름)입니다. 이 속성에 대한 자세한 설명은 [큐의 속성](#)을 참조하십시오.

리모트 큐의 로컬 정의에 대하여 MQINQ 호출을 사용하는 경우 큐 관리자는 원격 시스템에서 일치하는 로컬 큐의 속성이 아닌 로컬 정의의 속성(리모트 큐 이름, 리모트 큐 관리자 이름 및 전송 큐 이름)만을 리턴합니다.

[전송 큐의 내용](#)도 참조하십시오.

알리어스 큐

알리어스 큐는 다른 큐 또는 토픽에 액세스하는 데 사용할 수 있는 WebSphere MQ 오브젝트입니다. 이는 둘 이상의 프로그램이 다른 이름을 사용하여 액세스하여 동일한 큐에 대해 작업할 수 있음을 의미합니다.

알리어스 이름의 해석 결과인 큐(기본 큐라고 함) can be a local queue, the local definition of a remote queue, or a shared queue (a type of local queue only available on WebSphere MQ for z/OS). 이 큐는 플랫폼의 지원에 따라 사전정의된 큐 또는 동적 큐가 될 수 있습니다.

알리어스 이름은 토픽으로 해석할 수도 있습니다. 현재 애플리케이션이 메시지를 큐에 넣고 있으면 큐 이름을 토픽의 알리어스로 작성하여 토픽으로 발행하도록 애플리케이션을 작성할 수 있습니다. 애플리케이션 코드의 변경은 필요하지 않습니다.

참고: 별명은 로컬로 정의된 다른 알리어스 큐로 해석할 수 없습니다.

알리어스 큐 사용에 대한 예는 시스템 관리자가 기본 큐 이름(알리어스가 해석의 대상 큐) 및 다른 큐 이름에 다른 액세스 권한을 부여하는 것입니다. 이는 프로그램 또는 사용자에게 알리어스 큐 사용에 대한 권한은 부여되지만 기본 큐에 대한 권한은 부여될 수 없음을 의미합니다.

그렇지 않은 경우 알리어스 이름에 대해 Put 조작을 금지하지만 기본 큐에 대해서는 Put 조작을 허용하도록 권한 부여를 설정할 수 있습니다.

일부 애플리케이션에서 알리어스 큐 사용은 시스템 관리자가 애플리케이션을 변경할 필요 없이 알리어스 큐 오브젝트의 정의를 쉽게 변경할 수 있음을 의미합니다.

WebSphere MQ는 프로그램이 알리어스 이름을 사용하려고 할 때 이 이름에 대해 권한 검사를 수행합니다. 권한 검사에서는 알리어스가 해석하는 이름에 액세스하기 위한 권한이 프로그램에 부여되었는지 확인하지 않습니다. 따라서 프로그램은 해석된 큐 이름이 아닌 알리어스 큐 이름에 액세스할 수 있는 권한을 부여받을 수 있습니다.

96 페이지의 『큐』에 설명된 일반 큐 속성 이외에도 알리어스 큐에는 *BaseQName* 속성이 있습니다. 알리어스 이름이 해석되는 기본 큐의 이름입니다. 이 속성에 대한 자세한 설명은 [BaseQName \(MQCHAR48\)](#)을 참조하십시오.

별명 큐의 *InhibitGet* 및 *InhibitPut* 속성 (96 페이지의 『큐』 참조) 은 별명 이름에 속합니다. 예를 들어, 알리어스 큐 이름 ALIAS1이 기본 큐 이름 BASE로 해석하는 경우 ALIAS1에 대한 금지가 ALIAS1에만 영향을 주며 BASE는 금지되지 않습니다. 하지만 BASE에 대한 금지는 ALIAS1에 영향을 줍니다.

DefPriority 및 *DefPersistence* 속성도 별명 이름에 속합니다. 예를 들어 동일한 기본 큐의 다른 알리어스에 다른 기본 우선순위를 지정할 수 있습니다. 또한 알리어스를 사용하는 애플리케이션에 대한 변경없이 이러한 우선순위를 변경할 수 있습니다.

공유 및 클러스터 큐

이 정보는 공유 큐와 클러스터 큐라는 용어를 정의하고 설명하며 둘 사이의 비교를 제공합니다.

공유 큐

공유 큐는 큐 공유 그룹에 있는 하나 이상의 큐 관리자가 액세스할 수 있는 메시지가 있는 로컬 큐의 유형입니다. **Shared queues are available only on WebSphere MQ for z/OS.** (이 큐는 동일한 큐 관리자를 사용하여 둘 이상의 애플리케이션에서 공유하는 큐와 같지 않습니다.) 공유 큐는 커플링 기능(CF)에 의해 보유되며 큐 공유 그룹의 모든 큐 관리자가 액세스할 수 있습니다. 큐 공유 그룹의 각 공유 큐에는 해당 그룹 내에서 고유한 이름이 있어야 합니다.

클러스터 큐

클러스터 큐는 클러스터 큐 관리자가 호스팅하며 클러스터 내의 다른 큐 관리자가 사용할 수 있는 큐입니다.

클러스터 큐 관리자는 큐에 대한 로컬 큐 정의가 큐를 사용할 수 있는 클러스터의 이름을 지정하도록 합니다. 이러한 정의는 큐를 클러스터에 있는 다른 큐 관리자에게 광고합니다. 다른 큐 관리자는 해당하는 리모트 큐 정의 없이도 클러스터 큐에 메시지를 넣을 수 있습니다. 클러스터 큐는 둘 이상의 클러스터에서 광고할 수 있습니다. 추가적인 정보는 [클러스터 및 큐 관리자 클러스터 구성](#)을 참조하십시오.

공유 큐와 클러스터 큐의 비교

이 정보는 공유 큐와 클러스터 큐를 비교하고 사용자의 시스템에 더 적합한 것을 결정하도록 돕기 위해 설계되었습니다.

이동기 비용

클러스터 큐에서 메시지는 이동기가 전송하므로 애플리케이션 비용에 이동기 비용이 추가됩니다. 채널이 메시지를 보내고 받으므로 네트워크에 비용이 발생합니다. 공유 큐에서는 이러한 비용이 발생하지 않으므로 큐 공유 그룹의 큐 관리자 사이에 메시지를 이동하는 경우 클러스터 큐보다 더 적은 처리 능력을 사용합니다.

메시지의 사용가능성

큐에 넣는 경우 클러스터 큐는 사용자의 큐 관리자에 연결된 활성 채널이 있는 큐 관리자 중 하나로 메시지를 송신합니다. 리모트 큐 관리자에서 메시지를 처리하는 데 사용되는 애플리케이션이 작동하지 않는 경우 메시지는 처리되지 않으며 애플리케이션이 시작될 때까지 대기합니다. 마찬가지로, 큐 관리자가 종료된 경우 큐 관리자의 모든 메시지는 큐 관리자가 다시 시작될 때까지 사용할 수 없습니다. 이러한 인스턴스는 공유 큐를 사용하는 경우보다 더 낮은 메시지 가용성을 나타냅니다.

공유 큐를 사용하는 경우, 큐 공유 그룹의 모든 애플리케이션은 전송된 메시지를 받을 수 있습니다. 큐 공유 그룹에서 단일 큐 관리자를 종료하는 경우에는 다른 큐 관리자에 대해 메시지를 사용할 수 있으므로 클러스터 큐를 사용할 때보다 높은 메시지 가용성을 제공합니다.

용량

커플링 기능은 디스크보다 비쌉니다. 그러므로 로컬 큐에 1,000,000개의 메시지를 저장하는 것이 동일한 수의 메시지를 저장할 수 있는 충분한 용량이 있는 커플링 기능보다 비용이 덜 듭니다.

다른 큐 관리자로 송신

공유 큐 메시지는 큐 공유 그룹 내에서만 사용 가능합니다. 큐 공유 그룹 외부에서 큐 관리자를 사용하려면 이동기를 사용해야 합니다. 클러스터링을 사용하여 여러 원격 분산 큐 관리자 간에 작업을 분산할 수 있습니다.

워크로드 밸런싱

클러스터링을 사용하여 메시지 전송의 일부를 받을 채널 및 큐 관리자에 가중치를 부여할 수 있습니다. 예를 들어, 메시지의 60%를 하나의 큐 관리자로 송신하고 메시지의 40%를 다른 큐 관리자로 송신할 수 있습니다. 이 인스턴스는 작업을 처리할 리모트 큐 관리자의 능력에 따라 달라지지 않습니다. 첫 번째 큐 관리자가 있는 시스템이 과부하되고 두 번째 큐 관리자가 있는 시스템이 유휴 상태여도 대부분의 메시지는 여전히 첫 번째 큐 관리자로 이동합니다.

공유 큐를 사용하는 경우에는 두 CICS 시스템이 메시지를 받을 수 있습니다. 한 시스템이 과부하되면 다른 시스템이 대부분의 과부하를 넘겨 받습니다.

동적 및 모델 큐

이 정보는 동적 큐, 임시 및 영구적 동적 큐의 특성, 동적 큐의 사용, 동적 큐를 사용할 때의 고려사항 및 모델 큐에 대한 통찰을 제공합니다.

애플리케이션 프로그램이 MQOPEN 호출을 발행하여 모델 큐를 열 때 큐 관리자는 모델 큐와 같은 속성을 가진 로컬 큐의 인스턴스를 동적으로 작성합니다. 모델 큐의 *DefinitionType* 필드 값에 따라 큐 관리자가 임시 또는 영구 동적 큐를 작성합니다 ([동적 큐 작성 참조](#)).

임시 동적 큐의 특성

임시 동적 큐에는 다음 특성이 있습니다.

- 이는 공유 큐일 수 없으며 큐 공유 그룹의 큐 관리자에서 액세스할 수 없습니다.
Note that queue-sharing groups are only available on WebSphere MQ for z/OS.
- 비지속 메시지만 보유합니다.
- 복구 가능하지 않습니다.
- 큐 관리자가 시작될 때 삭제됩니다.
- 큐를 작성한 MQOPEN 호출을 발행한 애플리케이션이 큐를 닫거나 종료할 때 삭제됩니다.
 - 큐에 커밋된 메시지가 있는 경우 삭제됩니다.
 - 이 때 큐에 대해 미해결된 커밋되지 않은 MQGET, MQPUT 또는 MQPUT1 호출이 있을 경우 이러한 호출이 커밋된 후에 큐는 논리적으로 삭제된 것으로 표시되며 닫기 처리의 일부로 또는 애플리케이션이 종료될 때에만 실제로 삭제됩니다.
 - 이때 큐가 작성 또는 다른 애플리케이션에 의해 사용 중이면 큐가 논리적으로 삭제된 것으로 표시되며 큐를 사용하여 마지막 애플리케이션을 종료할 때에만 큐가 실제로 삭제됩니다.
 - 논리적으로 삭제된 큐(큐를 닫은 경우는 제외)에 대한 액세스 시도가 이유 코드 MQRC_Q_DELETED로 실패했습니다.
 - 큐를 작성한 해당 MQOPEN 호출에 대해 MQCO_NONE, MQCO_DELETE 및 MQCO_DELETE_PURGE가 MQCLOSE 호출에 지정되었을 때 이 호출 모두 MQCO_NONE으로 처리됩니다.

영구적 동적 큐의 특성

영구적 동적 큐에는 다음의 특성이 있습니다.

- 지속적 또는 비지속 메시지를 보유합니다.
- 시스템 실패 이벤트에서 복구 가능합니다.
- 애플리케이션(큐를 작성한 MQOPEN 호출 발행 애플리케이션일 필요는 없음)이 MQCO_DELETE 또는 MQCO_DELETE_PURGE 옵션을 사용하여 큐를 성공적으로 닫을 때 삭제됩니다.
 - 큐에 여전히 메시지(커밋된 또는 커밋되지 않은 메시지)가 있는 경우 MQCO_DELETE 옵션을 사용한 닫기 요청이 실패합니다. 큐에 커밋된 메시지(닫기의 일부로 삭제되는 메시지)가 있는 경우에도 MQCO_DELETE_PURGE 옵션을 사용한 닫기 요청은 성공하지만 큐에 대해 미해결된 커밋되지 않은 MQGET, MQPUT 또는 MQPUT1 호출이 있을 경우에는 실패합니다.

- 삭제 요청이 성공했지만 큐가 사용 중(작성 또는 다른 애플리케이션에 의해)이면 이 큐는 논리적으로 삭제된 것으로 표시되며 큐를 사용하여 마지막 애플리케이션이 큐를 닫았을 때에만 실제로 삭제됩니다.
- 닫기 애플리케이션이 큐를 작성한 MQOPEN 호출을 발행하지 않는 한 큐 삭제에 대한 권한이 부여되지 않은 애플리케이션이 큐를 닫는 경우 삭제되지 않습니다. 해당하는 MQOPEN 호출을 유효성 검증하는 데 사용되는 사용자 ID(또는 MQOO_ALTERNATE_USER_AUTHORITY가 지정된 경우 대체 사용자 ID)에 대해 권한 검사를 수행합니다.
- 정상 큐와 같은 방식으로 삭제될 수 있습니다.

동적 큐의 사용

다음에 대해서 동적 큐를 사용할 수 있습니다.

- 애플리케이션이 종료된 후 보유될 큐를 요구하지 않는 애플리케이션입니다.
- 다른 애플리케이션이 처리하는 메시지에 대한 응답을 요구하는 애플리케이션입니다. 이런 애플리케이션은 모델 큐를 열어서 응답 대상 큐를 동적으로 작성할 수 있습니다. 예를 들어, 클라이언트 애플리케이션은 다음을 수행할 수 있습니다.
 1. 동적 큐를 작성하십시오.
 2. 요청 메시지의 메시지 디스크립터 구조의 *ReplyToQ* 필드에 동적 큐의 이름을 제공하십시오.
 3. 서버에 의해 처리되고 있는 큐에 요청을 배치하십시오.

서버는 응답 대상 큐에 응답 메시지를 배치할 수 있습니다. 마지막으로 클라이언트는 응답을 처리하고 삭제 옵션을 사용하여 응답 대상 큐를 닫습니다.

동적 큐 사용 시 고려사항

동적 큐 사용 시 다음 사항을 고려하십시오.

- 클라이언트 서버 모델에서 각 클라이언트는 고유 동적 응답 대상 큐를 작성하고 사용해야 합니다. 동적 응답 대상 큐가 둘 이상의 클라이언트 사이에서 공유되는 경우 응답 대상 큐 삭제가 지연될 수 있습니다. 이는 큐에 대해 미해결된 커밋되지 않은 활동이 있거나 큐가 다른 클라이언트에 의해 사용 중이기 때문입니다. 추가적으로 큐는 논리적으로 삭제된 것으로 표시될 수 있으며 후속 API 요청(MQCLOSE 제외)에 대해 액세스 가능하지 않을 수 있습니다.
- 애플리케이션 환경이 동적 큐가 애플리케이션 사이에서 공유되어야 함을 요구하는 경우 큐에 대한 모든 활동이 커밋될 때에만 큐가 닫히는지(삭제 옵션 사용) 확인하십시오. 이는 마지막 사용자에게 의해 수행되어야 합니다. 이 큐의 삭제가 지연되지 않는지 확인하고 논리적으로 삭제된 것으로 표시되어 큐에 액세스가 불가능한 시간을 최소화합니다.

모델 큐

모델 큐는 사용자가 동적 큐를 작성할 때 사용하는 큐 정의의 템플릿입니다.

큐 속성의 템플릿으로 사용하려는 모델 큐에 이름을 지정하여 WebSphere MQ 프로그램에서 동적으로 로컬 큐를 작성할 수 있습니다. 이때 새 큐의 일부 속성을 변경할 수 있습니다. 그러나 *DefinitionType*은 변경할 수 없습니다. 예를 들어 영구적 큐가 필요한 경우 정의 유형이 영구적으로 설정된 모델 큐를 선택하십시오. 일부 대화식 애플리케이션은 응답을 처리한 후 이러한 큐를 유지보수할 필요가 없기 때문에 조회에 대한 응답을 보유하는 데 동적 큐를 사용할 수 있습니다.

모델 큐의 이름을 MQOPEN 호출의 오브젝트 디스크립터(MQOD)에 지정합니다. 모델 큐의 속성을 사용하여 큐 관리자는 로컬 큐를 동적으로 작성합니다.

동적 큐에 대한 전체 이름 또는 이름의 어간(예: ABC)을 지정하고 큐 관리자가 여기에 고유한 파트를 추가하도록 하거나 큐 관리자가 고유한 전체 이름을 지정하도록 할 수 있습니다. 큐 관리자가 이름을 지정하는 경우 이 이름을 MQOD 구조에 삽입합니다.

MQPUT1 호출을 모델 큐에 직접 발행할 수 없지만 MQPUT1을 모델 큐를 열어 작성된 동적 큐에 MQPUT1을 발행할 수 있습니다.

MQSET 및 MQINQ는 모델 큐에서 실행될 수 없습니다. MQOO_INQUIRE 또는 MQOO_SET를 포함하는 모델 큐를 열면 동적으로 작성된 큐에서 후속 MQINQ 및 MQSET 호출이 작성됩니다.

모델 큐의 속성은 로컬 큐의 속성 서브세트입니다. 자세한 설명은 큐의 속성을 참조하십시오.

큐 정의

MQSC 명령인 DEFINE 또는 PCF의 큐 작성 명령을 사용하여 IBM WebSphere MQ에 큐를 정의합니다.

이 명령은 큐의 유형 및 해당 속성을 지정합니다. 예를 들어, 로컬 큐 오브젝트에는 애플리케이션이 MQI 호출에서 해당 큐를 참조할 때 발생하는 일을 지정하는 속성이 있습니다. 속성의 예는 다음과 같습니다.

- 애플리케이션이 큐에서 메시지를 검색할 수 있는지의 여부(GET 사용)
- 애플리케이션이 메시지를 큐에 넣을 수 있는지의 여부(PUT 사용)
- 큐에 대한 액세스가 하나의 애플리케이션에만 독점적인지 또는 애플리케이션 사이에서 공유되는지 여부
- 큐에 동시에 저장될 수 있는 최대 메시지 수 (최대 큐 용량)
- 큐에 넣을 수 있는 메시지의 최대 길이

큐 오브젝트 정의에 대한 자세한 내용은 [스크립트 \(MQSC\) 명령](#) 을 참조하십시오.

IBM WebSphere MQ 에서 사용되는 큐

IBM WebSphere MQ에서는 그 조작과 관련된 특정 목적의 로컬 큐를 사용합니다.

IBM WebSphere MQ에서 이러한 큐를 사용할 수 있도록 하려면 먼저 큐를 정의해야 합니다.

이니시에이션 큐

이니시에이션 큐는 트리거에서 사용되는 큐입니다. 큐 관리자는 트리거 이벤트가 발생할 때 트리거 메시지를 이니시에이션 큐에 넣습니다. 트리거 이벤트는 큐 관리자가 감지하는 조건의 논리적 조합입니다. 예를 들어, 큐의 메시지 수가 사전정의된 용량에 도달하면 트리거 이벤트가 생성될 수 있습니다. 이러한 이벤트는 큐 관리자가 지정된 이니시에이션 큐에 트리거 메시지를 넣는 원인이 됩니다. 이 트리거 메시지는 이니시에이션 큐를 모니터링하는 특수 애플리케이션인 트리거 모니터로 검색할 수 있습니다. 트리거 모니터는 트리거 메시지에 지정된 애플리케이션 프로그램을 시작합니다.

큐 관리자가 트리거를 사용하는 경우 최소 하나 이상의 이니시에이션 큐가 해당 큐 관리자에 대해 정의되어야 합니다. [트리거할 오브젝트 관리](#), [runmqtrm](#) 및 [트리거를 사용한 WebSphere MQ 애플리케이션 시작](#)을 참조하십시오.

전송 큐

전송 큐는 리모트 큐 관리자를 목적지로 하는 메시지를 임시로 저장하는 큐입니다. 로컬 큐 관리자가 메시지를 직접 송신하는 각 리모트 큐 관리자에 대해 최소 하나 이상의 전송 큐를 정의해야 합니다. 이 큐는 원격 관리에서도 사용됩니다. [로컬 큐 관리자에서 원격 관리를 참조하십시오](#). 분산 큐잉에서 전송 큐의 사용에 대한 정보는 [WebSphere MQ 분산 메시징 기술](#) 을 참조하십시오.

각 큐 관리자에게는 기본 전송 큐가 있을 수 있습니다. 클러스터에 포함되지 않는 큐 관리자가 메시지를 리모트 큐에 넣는 경우, 기본 조치는 기본 전송 큐를 사용하는 것입니다. 목적지 큐 관리자와 이름이 동일한 전송 큐가 있는 경우, 메시지는 해당 전송 큐에 배치됩니다. **RQMNAME** 매개변수가 목적지 큐 관리자와 일치하며 **XMITQ** 매개변수가 정의되어 있는 큐 관리자 알리어스 정의가 있는 경우, 메시지는 **XMITQ**로 이름 지정된 전송 큐에 배치됩니다. **XMITQ** 매개변수가 없으면 메시지가 해당 메시지에 이름 지정된 로컬 큐에 배치됩니다.

클러스터 전송 큐

클러스터 내의 각 큐 관리자에게는 클러스터 전송 큐 **SYSTEM.CLUSTER.TRANSMIT.QUEUE**와 모델 클러스터 전송 큐 **SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE**가 있습니다. 이러한 큐의 정의는 큐 관리자를 정의할 때 기본적으로 작성됩니다. 큐 관리자 속성 **DEFCLXQ**를 CHANNEL로 설정한 경우 생성된 각 클러스터-송신자 채널에 대해 영구적인 동적 클러스터 전송 큐가 자동으로 생성됩니다. 큐를 **SYSTEM.CLUSTER.TRANSMIT.ChannelName**이라고 합니다. 또한 클러스터 전송 큐를 수동으로 정의할 수도 있습니다.

클러스터에 포함되는 큐 관리자는 이러한 큐 중 하나에서 동일한 클러스터에 있는 다른 큐 관리자로 메시지를 송신합니다.

이름 해석 중에, 클러스터 전송 큐는 기본 전송 큐보다 우선하고 특정 클러스터 전송 큐는 **SYSTEM.CLUSTER.TRANSMIT.QUEUE**보다 우선합니다.

데드-레터 큐

데드-레터(미전달 메시지) 큐는 올바른 목적지로 라우트되지 못한 메시지를 저장하는 큐입니다. 예를 들어, 목적지 큐가 가득 찬 경우에는 메시지를 라우트할 수 없습니다. 제공된 데드-레터 큐를 **SYSTEM.DEAD.LETTER.QUEUE**라고 합니다.

분산 큐잉의 경우 포함된 각 큐 관리자에서 데드-레터 큐를 정의하십시오.

명령 큐

명령 큐인 SYSTEM.ADMIN.COMMAND.QUEUE는 적합하게 권한 부여된 애플리케이션이 처리를 위한 MQSC 명령을 보낼 수 있는 로컬 큐입니다. 이러한 명령은 명령 서버라고 하는 IBM WebSphere MQ 컴포넌트로 검색됩니다. 명령 서버는 명령을 유효성 검증하고 큐 관리자에 의한 처리를 위해 올바른 명령을 전달하며 적절한 응답 대상 큐로 응답을 리턴합니다.

명령 큐는 해당 큐 관리자가 작성될 때 각 큐 관리자에 대해 자동으로 작성됩니다.

응답 대상 큐

애플리케이션이 요청 메시지를 송신하면 메시지를 수신하는 애플리케이션은 응답 메시지를 송신 애플리케이션으로 돌려 보낼 수 있습니다. 이 메시지를 큐(응답 대상 큐라고 함)에 넣고 이 큐는 일반적으로 송신 애플리케이션에 대한 로컬 큐입니다. 응답 대상 큐의 이름은 메시지 디스크립터의 일부로 송신 애플리케이션에 의해 지정됩니다.

이벤트 큐

도구 이벤트는 MQI 애플리케이션과는 독립적으로 큐 관리자를 모니터링하는 데 사용할 수 있습니다.

도구 이벤트가 발생하면 큐 관리자는 이벤트 메시지를 이벤트 큐에 넣습니다. 이 메시지는 모니터링 애플리케이션이 읽을 수 있으며 이벤트가 문제점을 표시할 때 관리자에게 알려거나 일부 보수 조치를 시작할 수 있습니다.

참고: 트리거 이벤트는 도구 이벤트와 다릅니다. 트리거 이벤트는 동일한 조건에 의해 발생하지 않으며 이벤트 메시지를 생성하지 않습니다.

도구 이벤트에 대한 자세한 정보는 [도구 이벤트를 참조하십시오](#).

IBM WebSphere MQ 큐 관리자

애플리케이션에 제공하는 큐 관리자 및 큐잉 서비스 소개입니다.

프로그램이 해당 큐 관리자의 서비스를 사용할 수 있으려면 큐 관리자에 대한 연결이 있어야 합니다. 프로그램은 이 연결을 MQCONN 또는 MQCONNX 호출을 사용하여 명시적으로 수행할 수 있거나 암묵적으로 수행할 수도 있습니다 (이는 프로그램이 실행되는 플랫폼 및 환경에 따라 다름).

큐 관리자는 애플리케이션에 큐잉 서비스를 제공하며 그에 포함된 큐를 관리합니다. 큐 관리자는 다음 조치를 확인합니다.

- 오브젝트 속성이 수신된 명령에 따라 변경됩니다.
- 적절한 조건에 부합되는 경우 트리거 이벤트 또는 도구 이벤트와 같은 특수 이벤트가 생성됩니다.
- MQPUT 호출을 작성하는 애플리케이션이 요청한 대로 올바른 큐에 메시지를 넣습니다. 이러한 조치가 수행될 수 없는 경우에는 애플리케이션에 알려지며 적절한 이유 코드가 제공됩니다.

각 큐는 단일 큐 관리자에 속하며 해당 큐 관리자에 대한 로컬 큐라고 합니다. 애플리케이션이 연결되는 큐 관리자는 해당 애플리케이션에 대한 로컬 큐 관리자라고 합니다. 애플리케이션의 경우 해당 로컬 큐 관리자에 속하는 큐는 로컬 큐입니다.

리모트 큐는 다른 큐 관리자에 속하는 큐입니다. 리모트 큐 관리자는 로컬 큐 관리자를 제외한 다른 모든 큐 관리자입니다. 리모트 큐 관리자는 네트워크 전체의 원격 시스템에 존재하거나 로컬 큐 관리자와 동일한 시스템에 존재할 수 있습니다. WebSphere MQ는 동일한 시스템에서 다중 큐 관리자를 지원합니다.

큐 관리자 오브젝트는 일부 MQI 호출에서 사용할 수 있습니다. 예를 들어, MQI 호출 MQINQ를 사용하여 큐 관리자 오브젝트의 속성에 대해 조회할 수 있습니다.

큐 관리자의 속성

각 큐 관리자에 연관된 속성(또는 특성) 세트는 큐 관리자의 특성을 정의합니다. 큐 관리자의 일부 속성은 작성될 때 이미 고정되며 그 외의 속성은 WebSphere MQ 명령을 사용하여 변경할 수 있습니다. SSL(Secure Sockets Layer) 암호화에 사용된 속성을 제외한 모든 속성 값에 대해서는 MQINQ 호출을 사용하여 조회할 수 있습니다.

고정된 속성에는 다음이 포함됩니다.

- 큐 매니저의 이름
- 큐 관리자가 실행되는 플랫폼(예를 들어, Windows)

- 큐 관리자가 지원하는 시스템 제어 명령의 레벨
- 큐 관리자에 의해 처리되는 메시지에 지정할 수 있는 최고 우선순위
- 프로그램이 WebSphere MQ 명령을 송신할 수 있는 큐의 이름
- 큐 관리자가 처리할 수 있는 메시지의 최대 길이
- 프로그램이 메시지를 넣고 가져올 때 큐 관리자가 동기점을 지원하는지 여부

변경 가능한 속성에는 다음이 포함됩니다.

- 큐 관리자에 대한 텍스트 설명
- 큐 관리자가 MQI 호출을 처리할 때 문자열에 사용하는 문자 세트의 ID
- 큐 관리자가 트리거 메시지의 수를 제한하는 데 사용하는 시간 간격
- 큐 관리자의 데드-레터(미전달 메시지) 큐 이름
- 큐 관리자의 기본 전송 큐 이름
- 하나의 연결에 대한 최대 열린 핸들 수
- 다양한 범주의 이벤트 보고 사용 및 사용 안함
- 작업 단위 내에서 커밋되지 않은 메시지의 최대 수

큐 관리자 및 워크로드 관리

동일한 큐에 대해 둘 이상의 정의를 가지고 있는 큐 관리자의 클러스터를 설정할 수 있습니다(예를 들어, 클러스터에 있는 큐 관리자는 서로의 복제본일 수 있음). 특정 큐에 대한 메시지는 큐의 인스턴스를 호스팅하는 큐 관리자로 핸들링할 수 있습니다. 워크로드 관리 알고리즘은 메시지를 핸들링할 큐 관리자를 결정하여 큐 관리자 간에 워크로드를 분배합니다. 자세한 정보는 [클러스터 워크로드 관리 알고리즘](#)을 참조하십시오.

프로세스 정의

프로세스 정의 오브젝트를 통해 큐 관리자 사용에 대한 애플리케이션의 속성을 정의하여 운영자가 개입할 필요 없이 애플리케이션을 시작할 수 있습니다.

프로세스 정의 오브젝트는 IBM WebSphere MQ 큐 관리자에서 트리거 이벤트에 대한 응답으로 시작하는 애플리케이션을 정의합니다. 프로세스 정의 속성에는 애플리케이션 ID, 애플리케이션 유형 및 애플리케이션에 특정한 데이터가 포함됩니다. 자세한 정보는 [103 페이지의 『IBM WebSphere MQ에서 사용되는 큐』](#) 아래의 "이니셔이션 큐" 항목을 참조하십시오.

운영자 개입이 없이도 애플리케이션이 시작될 수 있도록 허용하려면(트리거를 사용한 WebSphere MQ 애플리케이션 시작 참조), 애플리케이션의 속성을 큐 관리자에 알려야 합니다. 이러한 속성은 프로세스 정의 오브젝트에 정의되어 있습니다.

ProcessName 속성은 오브젝트가 작성될 때 고정됩니다. IBM WebSphere MQ 명령을 사용하여 다른 속성을 변경할 수 있습니다.

[MQINQ - 오브젝트 속성 조회](#)를 사용하여 모든 속성의 값에 대해 조회할 수 있습니다.

프로세스 정의의 속성에 대한 자세한 설명은 [프로세스 정의의 속성](#)을 참조하십시오.

이름 목록

이름 목록은 클러스터 이름, 큐 이름 또는 인증 정보 오브젝트 이름 목록이 포함된 WebSphere MQ 오브젝트입니다. 클러스터에서 이름 목록은 큐 관리자가 저장소에 보유하는 클러스터의 목록을 식별하는 데 사용될 수 있습니다.

이름 목록은 다른 WebSphere MQ 오브젝트 목록이 포함된 WebSphere MQ 오브젝트입니다. 일반적으로 이름 목록은 큐 그룹을 식별하는 데 사용되는 트리거 모니터와 같은 애플리케이션에 의해 사용됩니다. 이름 목록 사용의 장점은 애플리케이션과 독립적으로 유지보수된다는 점이며 이름 목록을 사용하는 애플리케이션을 중지하지 않고도 업데이트할 수 있다는 것입니다. 또한 하나의 애플리케이션이 실패하는 경우 이름 목록은 영향을 받지 않으며 다른 애플리케이션은 계속해서 이름 목록을 사용할 수 있습니다.

이름 목록은 또한 큐 관리자 클러스터와 함께 사용되어 둘 이상의 WebSphere MQ 오브젝트를 통해서 참조되는 클러스터의 목록을 유지보수합니다.

MQSC 명령을 사용하여 이름 목록을 정의하고 수정할 수 있습니다.

프로그램은 MQI를 사용하여 이러한 이름 목록에 포함되는 큐를 알아낼 수 있습니다. 이름 목록의 조직은 애플리케이션 설계자 및 시스템 관리자가 담당합니다.

이름 목록 속성에 대한 자세한 설명은 [이름 목록 속성](#)을 참조하십시오.

인증 정보 오브젝트

큐 관리자 인증 정보 오브젝트에 대한 소개 및 추가 정보에 대한 링크입니다.

WebSphere MQ에 속하는 큐 관리자 인증 정보 오브젝트 형식은 SSL(Secure Sockets Layer) 및 TLS(Transport Layer Security)를 지원합니다. 폐기된 인증서를 확인하는 데 필요한 정의를 제공합니다. 인증 기관은 더 이상 신뢰할 수 없는 인증서를 폐기합니다.

이 절에서는 setmqaut, dspmqaut, dmpmqaut, rcrmqobj, rcdmqimg 및 dspmqfls 명령을 인증 정보 오브젝트와 함께 사용하는 방법을 설명합니다. SSL 및 TLS의 개요와 인증 정보 오브젝트 사용에 대한 정보는 [SSL 및 TLS에 대한 WebSphere MQ 지원](#)의 내용을 참조하십시오.

SSL 및 TLS에 대한 자세한 정보는 [SSL \(Secure Sockets Layer\) 및 TLS \(Transport Layer Security\) 개념](#)을 참조하십시오.

인증 정보 오브젝트는 인증서 폐기 확인을 수행하는 데 필요한 정의를 제공합니다.

인증 정보 오브젝트의 속성에 대한 자세한 정보는 [인증 정보 오브젝트](#)를 참조하십시오.

통신 정보 오브젝트

IBM WebSphere MQ 멀티캐스트에서는 낮은 지연, 높은 팬아웃, 신뢰할 수 있는 멀티캐스트 메시징을 제공합니다. 통신 정보(COMMINFO) 오브젝트는 멀티캐스트 전송을 사용하는 데 필요합니다.

COMMINFO 오브젝트는 멀티캐스트 전송과 연관된 속성을 포함하는 IBM WebSphere MQ 오브젝트입니다. 이러한 속성에 대한 자세한 정보는 [DEFINE COMMINFO](#)를 참조하십시오. COMMINFO 오브젝트의 작성에 대한 자세한 정보는 [멀티캐스트 시작하기](#)를 참조하십시오.

관련 개념

[116 페이지의 『IBM WebSphere MQ 멀티캐스트』](#)

IBM WebSphere MQ 멀티캐스트는 낮은 지연, 높은 팬아웃, 신뢰할 수 있는 멀티캐스트 메시징을 제공합니다.

채널

채널은 분산 큐 관리자에서 사용되는 통신 링크입니다.

채널은 한 큐 관리자에서 다른 큐 관리자로 통신 경로를 제공하는 오브젝트입니다. 채널은 분산 큐잉에서 한 큐 관리자에서 다른 큐 관리자로 메시지를 이동시키는 데 사용되며 기본 통신 프로토콜로부터 애플리케이션을 보호합니다. 큐 관리자들은 동일하거나 서로 다른 플랫폼에 존재할 수 있습니다.

큐 관리자가 서로 통신하기 위해서는 다른 큐 관리자로 메시지를 보낼 큐 관리자에서 하나의 채널 오브젝트를 정의하고 메시지를 받을 큐 관리자에서 보충 채널 오브젝트를 정의해야 합니다.

WebSphere MQ에는 두 가지 범주의 채널이 있습니다.

- 단방향이며, 한 큐 관리자에서 다른 큐 관리자로 메시지를 전송하는 메시지 채널입니다. 자세한 정보는 [채널 엑시트 호출 및 데이터 구조](#)를 참조하십시오.
- MQI 채널은 양방향이며 WebSphere MQ MQI 클라이언트에서 큐 관리자로 MQI 호출을 전송하며, 큐 관리자에서 WebSphere MQ 클라이언트로 응답합니다. 자세한 정보는 [107 페이지의 『채널의 개념』](#)를 참조하십시오.

관련 개념

[46 페이지의 『상호통신의 개념』](#)

WebSphere MQ에서, 상호통신은 하나의 큐 관리자에서 다른 큐 관리자로 메시지를 전송한다는 의미입니다. 수신 큐 관리자는 동일 시스템 또는 다른 쪽 세계 또는 인접한 곳의 다른 시스템에 있을 수 있습니다. 로컬 큐 관리자와 동일한 플랫폼에서 실행 중이거나 WebSphere MQ에서 지원되는 플랫폼에서 실행 중일 수 있습니다. 이를 분산 환경이라고 합니다. WebSphere MQ는 분산 큐 관리(DQM)를 사용하는 것처럼 분산 환경에서 통신을 핸들링합니다.

관련 태스크

[원격 WebSphere MQ 오브젝트 관리](#)

관련 참조

[채널 엑시트 호출 및 데이터 구조](#)

[109 페이지의 『통신』](#)

WebSphere MQ MQI 클라이언트는 MQI 채널을 사용하여 서버와 통신합니다.

채널의 개념

채널은 WebSphere MQ MQI 클라이언트와 WebSphere MQ 서버 간 또는 두 개의 WebSphere MQ 서버 간 논리 통신 링크입니다.

채널에는 두 가지 정의가 있으며 연결의 양 끝에 하나씩 있습니다. 동일한 채널 이름이 연결의 양 끝에서 사용되어야 하며 사용된 채널 유형은 호환 가능해야 합니다.

WebSphere MQ에는 두 개의 채널 범주가 있으며 이 채널 범주에는 다른 채널 유형이 포함됩니다.

관련 개념

[107 페이지의 『메시지 채널』](#)

메시지 채널은 단방향 링크입니다. 이 링크는 메시지 및 에이전트(MCAs)를 사용하여 두 큐 관리자에 연결됩니다.

[107 페이지의 『MQI 채널』](#)

MQI 채널은 WebSphere MQ MQI 클라이언트를 서버 시스템의 큐 관리자에 연결하며, WebSphere MQ MQI 클라이언트 애플리케이션에서 MQCONN 또는 MQCONNX 호출을 실행할 때 설정됩니다.

[108 페이지의 『채널 중지』](#)

WebSphere MQ에서 서버 연결 채널에 대해 STOP CHANNEL 명령을 실행할 때 클라이언트 연결 채널을 중지하는 데 사용할 방법을 선택할 수 있습니다.

메시지 채널

메시지 채널은 단방향 링크입니다. 이 링크는 메시지 및 에이전트(MCAs)를 사용하여 두 큐 관리자에 연결됩니다.

메시지 채널의 용도는 하나의 큐 관리자에서 다른 큐 관리자로 메시지를 전송하는 것입니다. 클라이언트 서버 환경에서는 메시지 채널이 필요하지 않습니다.

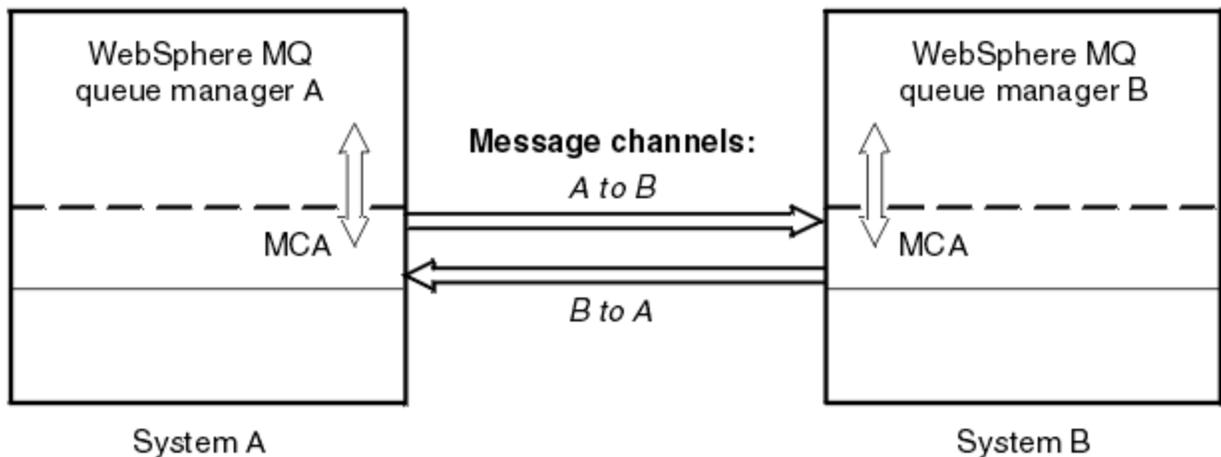


그림 28. 두 큐 관리자 간의 메시지 채널

MQI 채널

MQI 채널은 WebSphere MQ MQI 클라이언트를 서버 시스템의 큐 관리자에 연결하며, WebSphere MQ MQI 클라이언트 애플리케이션에서 MQCONN 또는 MQCONNX 호출을 실행할 때 설정됩니다.

이는 양방향 링크이고 MQI 호출 및 응답의 전송에만 사용되며, 여기에는 메시지 데이터가 포함된 MQPUT 호출 및 메시지 데이터의 리턴으로 귀결되는 MQGET 호출이 포함됩니다. 채널 정의를 작성하고 사용하는 다양한 방법이 있습니다(MQI 채널 정의 참조).

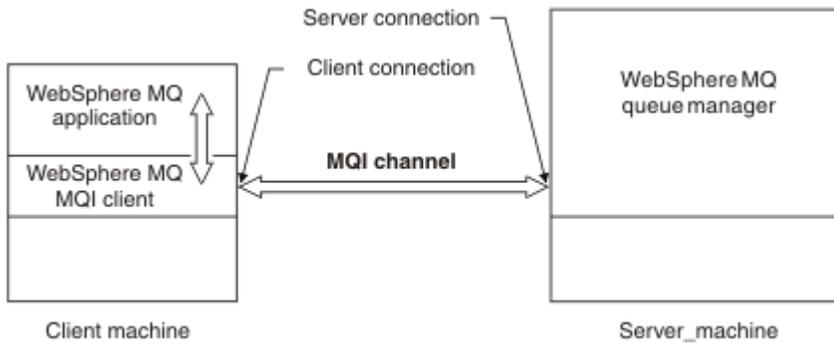


그림 29. MQI 채널에서의 클라이언트 연결 및 서버 연결

MQI 채널은 클라이언트를 단일 큐 관리자에 연결하거나 큐 공유 그룹의 일부인 큐 관리자에 연결하는 데 사용될 수 있습니다(큐 공유 그룹에 클라이언트 연결 참조).

MQI 채널 정의에는 두 가지 채널 유형이 있습니다. 양방향 MQI 채널을 정의합니다.

클라이언트 연결 채널

이 유형은 WebSphere MQ MQI 클라이언트용입니다.

서버 연결 채널

이 유형은 WebSphere MQ MQI 클라이언트 환경에서 실행되는 WebSphere MQ 애플리케이션과 통신하는 큐 관리자 실행 서버를 위한 것입니다.

채널 중지

WebSphere MQ에서 서버 연결 채널에 대해 STOP CHANNEL 명령을 실행할 때 클라이언트 연결 채널을 중지하는 데 사용할 방법을 선택할 수 있습니다.

이는 MQGET 대기 호출을 실행하는 클라이언트 채널을 제어할 수 있고 채널 중지 방법 및 시기를 결정할 수 있음을 의미합니다.

STOP CHANNEL 명령은 채널이 중지되는 방식을 표시하는 세 가지 모드로 실행될 수 있습니다.

일시정지

모든 현재 메시지가 처리된 후에 채널을 중지하십시오.

공유 대화가 사용 설정되면 WebSphere MQ MQI 클라이언트는 적절한 시간에 중지 요청을 인지하게 되며 이러한 시간은 네트워크 속도에 따라 달라집니다. 클라이언트 애플리케이션은 WebSphere MQ에 대한 후속 호출 발행의 결과로 중지 요청을 인지하게 됩니다.

강제 실행

채널을 즉시 중지하십시오.

종료

채널을 즉시 중지하십시오. 채널이 프로세스로 실행 중이면 채널의 프로세스를 종료할 수 있습니다. 또는 채널이 스레드로 실행 중이면 해당 스레드를 종료할 수 있습니다.

이는 다단계 프로세스입니다. 모드 종료를 사용하면 처음에는 일시정지 모드로 그 다음에는 강제 실행 모드로 그리고 필요한 경우에 종료 모드로 서버 연결 채널을 중지하려고 합니다. 클라이언트는 서로 다른 종료 상태에서 서로 다른 리턴 코드를 수신할 수 있습니다. 프로세스 또는 스레드가 종료되면 클라이언트는 통신 오류를 수신합니다.

애플리케이션에 리턴되는 리턴 코드는 실행된 MQI 호출 및 실행된 STOP CHANNEL 명령에 따라 다릅니다. 클라이언트가 MQRC_CONNECTION_QUIESCING 또는 MQRC_CONNECTION_BROKEN 리턴 코드를 수신합니다. 클라이언트가 MQRC_CONNECTION_QUIESCING을 감지하면 현재 트랜잭션을 완료하고 종료하려고 시도해야 합니다. 이는 MQRC_CONNECTION_BROKEN으로는 가능하지 않습니다. 클라이언트가 트랜잭션을 완료하지 않고 빨리 종결하는 경우 몇 초 후에 CONNECTION_BROKEN을 수신하게 됩니다. MODE(FORCE) 또는 MODE(TERMINATE)를 사용한 STOP CHANNEL 명령은 MODE(QUIESCE)를 사용하는 경우보다 CONNECTION_BROKEN이 발생할 가능성이 높습니다.

통신

WebSphere MQ MQI 클라이언트는 MQI 채널을 사용하여 서버와 통신합니다.

채널 정의는 WebSphere MQ MQI 클라이언트 및 서버의 연결 종료 둘 다에서 작성되어야 합니다. 채널 정의를 작성하는 방법은 [MQI 채널 정의](#)에 설명되어 있습니다.

가능한 전송 프로토콜은 다음 표에 표시됩니다.

표 8. MQI 채널용 전송 프로토콜				
클라이언트 플랫폼	LU 6.2	TCP/IP	NetBIOS	SPX
UNIX and Linux 시스템	예 ¹	예		
Windows	예	예	예	예

참고:

1. LU6.2 는 Linux (POWER® 플랫폼), Linux (x86-64 플랫폼), Linux (zSeries s390x 플랫폼) 또는 Solaris (x86-64 플랫폼) 에서 지원되지 않습니다.

전송 프로토콜 - WebSphere MQ MQI 클라이언트 및 서버 플랫폼의 조합은 이 전송 프로토콜을 사용하는 WebSphere MQ MQI 클라이언트 및 서버 플랫폼의 가능한 조합을 보여줍니다.

WebSphere MQ MQI 클라이언트의 WebSphere MQ 애플리케이션은 큐 관리자가 로컬인 경우와 동일한 방식으로 모든 MQI 호출을 사용할 수 있습니다. **MQCONN** 또는 **MQCONNX** 는 WebSphere MQ 응용프로그램을 선택한 큐 관리자와 연관시키고 연결 핸들을 작성합니다. 해당 연결 핸들을 사용하는 다른 호출은 연결된 큐 관리자가 처리합니다. WebSphere MQ MQI 클라이언트 통신은 연결 독립 및 시간 독립인 큐 관리자 간의 통신과는 반대로 클라이언트와 서버 간의 활성 연결을 요구합니다.

채널 정의를 사용하여 전송 프로토콜이 지정되며 애플리케이션에는 영향을 주지 않습니다. 예를 들어, Windows 애플리케이션은 TCP/IP로 하나의 큐 관리자에 연결되고 NetBIOS로 다른 큐 관리자에 연결할 수 있습니다.

성능 고려사항

사용하는 전송 프로토콜은 WebSphere MQ 클라이언트 및 서버 시스템에 영향을 미칠 수도 있습니다. 느린 전화 회선을 통한 전화 접속의 경우 WebSphere MQ 채널 압축 사용이 권장될 수 있습니다.

클라이언트 연결 채널

클라이언트 연결 채널 오브젝트에 대한 소개 및 추가 정보에 대한 링크입니다.

클라이언트 연결 채널은 WebSphere MQ MQI 클라이언트에서 큐 관리자로의 통신 경로를 제공하는 오브젝트입니다. 클라이언트 연결 채널은 분산 큐잉에 사용되어 큐 관리자와 클라이언트 사이에 메시지를 이동시킵니다. 이 채널은 근본적인 통신 프로토콜로부터 애플리케이션을 숨깁니다. 클라이언트는 큐 관리자와 동일한 또는 다른 플랫폼에 존재할 수 있습니다.

클라이언트 연결 채널 및 이를 사용하는 방법에 대한 정보는 [34 페이지의 『상호통신』](#)의 내용을 참조하십시오.

리스너

리스너는 다른 큐 관리자 또는 클라이언트에서 네트워크 요청을 승인하는 프로세스이며 연관된 채널을 시작합니다.

리스너는 다른 큐 관리자 또는 클라이언트에서 네트워크 요청을 승인하는 프로세스이며 연관된 채널을 시작합니다. `runmq1sr` 제어 명령을 사용하여 리스너 프로세스를 시작할 수 있습니다.

리스너 오브젝트는 WebSphere MQ 오브젝트이며 이 오브젝트를 통해 큐 관리자의 범위 내에서 리스너 프로세스의 시작 및 종단을 관리할 수 있습니다. 리스너 오브젝트의 속성을 정의하여 다음을 수행하십시오.

- 리스너 프로세스를 구성합니다.
- 큐 관리자가 시작 및 중지될 때 자동으로 리스너 프로세스를 시작 및 중지할지 여부를 지정합니다.

Listener objects are not supported on WebSphere MQ for z/OS.

서비스

서비스 오브젝트는 큐 관리자가 시작 또는 중지될 때 실행할 프로그램을 정의하는 방법입니다.

프로그램 유형은 다음 중 하나입니다.

서버

서버는 SERVER로 지정된 매개변수 SERVTYPE을 가지는 서비스 오브젝트입니다. 서버 서비스 오브젝트는 지정된 큐 관리자가 시작될 때 실행되는 프로그램의 정의입니다. 서버 프로세스의 한 인스턴스만을 동시에 실행할 수 있습니다. 실행하는 중에 MQSC 명령인 DISPLAY SVSTATUS를 사용하여 서버 프로세스의 상태를 모니터할 수 있습니다. 일반적으로 서버 서비스 오브젝트는 데드 레터 핸들러 또는 트리거 모니터와 같은 프로그램의 정의이지만 실행될 수 있는 프로그램은 WebSphere MQ와 함께 제공되는 프로그램으로 제한되지 않습니다. 추가적으로 서버 서비스 오브젝트는 지정된 큐 관리자가 종료되어 프로그램이 종료될 때 실행할 수 있는 명령을 포함하도록 정의할 수 있습니다.

명령

명령은 COMMAND로 지정된 매개변수 SERVTYPE을 가지는 서비스 오브젝트입니다. 명령 서비스 오브젝트는 지정된 큐 관리자가 시작되거나 중지될 때 실행되는 프로그램의 정의입니다. 명령 프로세스의 다중 인스턴스는 동시에 실행될 수 있습니다. 명령 서비스 오브젝트는 프로그램이 일단 실행되면 큐 관리자가 프로그램을 모니터하지 않는다는 점에서 서버 서비스 오브젝트와 다릅니다. 일반적으로 명령 서비스 오브젝트는 단 기적이고 하나 또는 둘 이상의 다른 태스크를 시작하는 것과 같은 특정 태스크를 수행하는 프로그램의 정의입니다.

관련 개념

[서비스에 대한 작업](#)

토픽 오브젝트

토픽 오브젝트는 기본이 아닌 특정 속성을 토픽에 지정할 수 있는 WebSphere MQ 오브젝트입니다.

토픽은 특정 토픽 문자열로 발행 또는 구독하는 애플리케이션에서 정의합니다. 토픽 문자열은 토픽을 슬래시(/)로 구분하여 토픽의 계층을 지정할 수 있습니다. 이러한 계층은 토픽 트리로 시각화할 수 있습니다. 예를 들어, 애플리케이션이 토픽 문자열 /Sport/American Football 및 /Sport/Soccer에 발행하는 경우 두 개의 하위, American Football 및 Soccer가 있는 상위 노드 Sport가 있는 토픽 트리가 작성됩니다.

토픽은 토픽 트리에 있는 첫 번째 상위 관리 노드에서 해당 속성을 상속합니다. 특정 토픽 트리에 관리 토픽 노드가 없는 경우, 모든 토픽은 기본 토픽 오브젝트 SYSTEM.BASE.TOPIC에서 해당 속성을 상속합니다.

해당 노드의 토픽 문자열을 토픽 오브젝트의 TOPICSTR 속성에 지정하여 토픽 트리의 모든 노드에서 토픽 오브젝트를 작성할 수 있습니다. 또한 관리 토픽 노드에 다른 속성을 정의할 수도 있습니다. 이러한 속성에 대한 자세한 정보는 [MQSC 명령어 또는 관리 태스크 자동화](#)를 참조하십시오. 각 토픽 오브젝트는 기본적으로 가장 가까운 상위 관리 토픽 노드에서 해당 속성을 상속합니다.

토픽 오브젝트는 애플리케이션 개발자로부터 전체 토픽 트리를 숨기는 데 사용될 수 있습니다. If a topic object named FOOTBALL.US is created for the topic /Sport/American Football, an application can publish or subscribe to the object named FOOTBALL.US instead of the string /Sport/American Football with the same result.

토픽 오브젝트의 토픽 문자열에 #, +, / 또는 * 문자를 입력한 경우 이 문자는 문자열 내에서 정상적인 문자로 처리되며 토픽 오브젝트와 연관된 토픽 문자열의 일부로 간주됩니다.

토픽 오브젝트에 대한 자세한 정보는 [WebSphere MQ 발행/구독 메시징 소개](#)를 참조하십시오.

IBM WebSphere MQ 오브젝트 이름 지정

WebSphere MQ 오브젝트에 채택된 이름 지정 규칙은 오브젝트에 따라 달라집니다. IBM WebSphere MQ에 대해 사용하는 시스템 및 사용자 ID의 이름 또한 일부 이름 지정 제한의 대상이 됩니다.

큐 관리자의 각 인스턴스는 그 이름으로 알려집니다. 한 큐 관리자가 제공된 메시지를 전송할 대상 큐 관리자를 분명하게 식별할 수 있도록 이 이름은 상호 연결된 큐 관리자의 네트워크 내에서 고유해야 합니다.

다른 유형 오브젝트의 경우 각 오브젝트는 오브젝트에 연관된 이름을 가지며 해당 이름으로 참조될 수 있습니다. 이러한 이름은 하나의 큐 관리자 및 오브젝트 유형에서 고유해야 합니다. 예를 들어, 동일한 이름을 가진 큐 및 프로세스를 가질 수 있지만 동일한 이름을 가진 두 개의 큐를 가질 수는 없습니다.

WebSphere MQ에서 이름은 최대 20자까지 가능한 채널의 경우를 제외하고 최대 48자까지 가질 수 있습니다. IBM WebSphere MQ 오브젝트 이름 지정에 대한 자세한 정보는 [111 페이지의 『IBM WebSphere MQ 오브젝트 이름 지정 규칙』](#)을 참조하십시오.

IBM WebSphere MQ에 대해 사용하는 시스템 및 사용자 ID의 이름 또한 다음의 일부 이름 지정 제한의 대상이 됩니다.

- 시스템 이름에 공백이 포함되지 않도록 하십시오. IBM WebSphere MQ 공백을 포함하는 시스템 이름은 지원하지 않습니다. 이러한 시스템에서 IBM WebSphere MQ를 설치할 경우 큐 관리자를 작성할 수 없습니다.
- IBM WebSphere MQ 권한 부여의 경우, 사용자 ID 및 그룹의 이름은 20자를 초과할 수 없습니다(공백 사용 불가).
- A WebSphere MQ for 윈도우 server does not support the connection of a 윈도우 client if the client is running under a user ID that contains the @ character, for example, abc@d.

관련 개념

[113 페이지의 『IBM WebSphere MQ 파일 이름 이해』](#)

각 WebSphere MQ 큐 관리자, 큐, 프로세스 정의, 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스 및 인증 정보 오브젝트는 파일로 표시됩니다. 오브젝트 이름은 꼭 올바른 파일 이름일 필요는 없기 때문에 큐 관리자는 필요한 위치에서 오브젝트 이름을 올바른 파일 이름으로 변환합니다.

관련 참조

[111 페이지의 『IBM WebSphere MQ 오브젝트 이름 지정 규칙』](#)

IBM WebSphere MQ 오브젝트 이름은 최대 길이를 갖고 있으며 대소문자가 구분됩니다. 모든 문자가 모든 오브젝트 유형에 대해 지원되지는 않으며, 많은 오브젝트가 이름의 고유성에 관한 규칙을 갖고 있습니다.

IBM WebSphere MQ 오브젝트 이름 지정 규칙

IBM WebSphere MQ 오브젝트 이름은 최대 길이를 갖고 있으며 대소문자가 구분됩니다. 모든 문자가 모든 오브젝트 유형에 대해 지원되지는 않으며, 많은 오브젝트가 이름의 고유성에 관한 규칙을 갖고 있습니다.

여러 가지 유형의 IBM WebSphere MQ 오브젝트가 있으며, 각 유형의 오브젝트는 별도의 오브젝트 네임스페이스에 존재하기 때문에 모두가 동일한 이름을 가질 수 있습니다. 예를 들어 로컬 큐와 송신자 채널은 둘 다 동일한 이름을 가질 수 있습니다. 그러나 동일한 네임스페이스에서는 한 오브젝트가 다른 오브젝트와 동일한 이름을 가질 수 없습니다. 예를 들어 로컬 큐는 모델 큐와 동일한 이름을 가질 수 없으며, 송신자 채널은 수신자 채널과 동일한 이름을 가질 수 없습니다.

다음 IBM WebSphere MQ 오브젝트는 별도의 오브젝트 네임스페이스에 존재합니다.

- 인증 정보
- 채널
- 클라이언트 채널
- 리스너
- 이름 목록
- 프로세스
- 큐
- 서비스
- 스토리지 클래스
- 구독
- 주제

오브젝트 이름의 문자 길이

일반적으로 IBM WebSphere MQ 오브젝트 이름의 길이는 최대 48자까지 가능합니다. 이 규칙은 다음 오브젝트에 적용됩니다.

- 인증 정보
- 군집

- 리스너
- 이름 목록
- 프로세스 정의
- 큐
- 큐 관리자
- 서비스
- 구독
- 주제

제한사항은 다음과 같습니다.

1. 채널 오브젝트 이름 및 클라이언트 연결 채널 이름의 최대 길이는 20문자입니다. 채널에 대한 자세한 정보는 [채널 정의를 참조하십시오](#).
2. 토픽 문자열은 최대 10240바이트일 수 있습니다. 모든 IBM WebSphere MQ 오브젝트 이름은 대소문자가 구분됩니다.
3. 스토리지 클래스 이름의 최대 길이는 8자입니다.
4. CF 구조 이름의 최대 길이는 12자입니다.

오브젝트 이름의 문자

IBM WebSphere MQ 오브젝트 이름에 대해 올바른 문자는 다음과 같습니다.

문자	제한
대문자 A - Z	• 없음
소문자 a - z	<ul style="list-style-type: none"> • MQSC 스크립트에서 소문자가 포함된 이름은 작은따옴표로 묶어야 합니다. 이렇게 하면 소문자가 대문자로 변경되지 않습니다. • EBCDIC 카타카나를 사용하는 시스템은 오브젝트 이름에서 소문자 a - z 문자를 사용할 수 없습니다.
숫자 0 - 9	• 없음
마침표(.)	• 없음
밑줄(_)	• distributed 없음
정방향 슬래시(/)	<ul style="list-style-type: none"> • Windows Windows 시스템에서는 큐 관리자 이름의 첫 번째 문자가 슬래시일 수 없습니다.
퍼센트 부호(%)	• distributed 없음

오브젝트 이름의 문자에 관한 몇 가지 규칙이 있습니다.

1. 선두 문자 또는 임베드된 공백은 허용되지 않습니다.
2. 자국어(NL) 문자는 허용되지 않습니다.
3. 전체 필드 길이보다 작은 모든 이름은 오른쪽에 공백을 채울 수 있습니다. 큐 관리자가 리턴하는 모든 짧은 이름은 항상 오른쪽이 공백으로 채워집니다.

큐 이름

큐 이름은 다음 두 파트를 갖습니다.

- 큐 관리자의 이름
- 해당 큐 관리자에게 알려진 큐의 로컬 이름

큐 이름의 각 파트는 길이가 48자입니다.

로컬 큐를 참조하기 위해(공백 문자로 바꾸거나 선행 널 문자를 사용하여) 큐 관리자의 이름을 생략할 수 있습니다. 그러나 IBM WebSphere MQ에 의해 프로그램으로 리턴되는 모든 큐 이름에 큐 관리자의 이름이 들어 있습니다.

리모트 큐를 참조하려면 프로그램이 전체 큐 이름에 큐 관리자의 이름을 포함시켜야 하거나 리모트 큐의 로컬 정의가 있어야 합니다.

애플리케이션이 큐 이름을 사용할 때 해당 이름은 로컬 큐의 이름(또는 로컬 큐에 대한 알리어스) 또는 리모트 큐의 로컬 정의의 이름일 수 있지만, 큐에서 메시지를 가져와야 하는 경우가 아니면(큐가 로컬이어야 함) 애플리케이션이 어느 것이든 알 필요는 없습니다. 애플리케이션이 큐 오브젝트를 열 때 MQOPEN 호출이 이름 해석 기능을 수행하여 후속 조작을 수행할 큐를 판별합니다. 이것의 중요성은 애플리케이션이 큐 관리자 네트워크의 특정 위치에서 정의되는 특정 큐에 대한 내장된 종속성을 갖지 않는다는 점입니다. 그러므로 시스템 관리자가 네트워크에서 큐의 위치를 변경하고 해당 정의를 변경하는 경우, 해당 큐를 사용하는 애플리케이션이 변경될 필요가 없습니다.

예약 오브젝트 이름

SYSTEM.으로 시작하는 오브젝트 이름은 큐 관리자가 정의하는 오브젝트용으로 예약됩니다. **Alter, Define** 및 **Replace** 명령을 사용하여 설치에 맞게 이러한 오브젝트 정의를 변경할 수 있습니다. IBM WebSphere MQ에 사용하도록 정의된 이름은 큐 이름에 모두 나열되어 있습니다.

관련 태스크

[설치 이름 선택](#)

IBM WebSphere MQ 파일 이름 이해

각 WebSphere MQ 큐 관리자, 큐, 프로세스 정의, 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스 및 인증 정보 오브젝트는 파일로 표시됩니다. 오브젝트 이름은 꼭 올바른 파일 이름일 필요는 없기 때문에 큐 관리자는 필요한 위치에서 오브젝트 이름을 올바른 파일 이름으로 변환합니다.

큐 관리자 디렉토리에 대한 기본 경로는 다음과 같습니다.

- 접두부(WebSphere MQ 구성 정보에 정의되어 있음)
 - 시스템에서 기본 접두어는 C:\Program Files\IBM\WebSphere MQ입니다. Windows 64비트 시스템에서 기본 접두부는 C:\Program Files\IBM\WebSphere MQ (x86)\입니다. 기본 접두부는 mqs.ini 구성 파일의 DefaultPrefix 스탠자에서 구성됩니다.
 - UNIX and Linux 시스템에서, 기본 접두부는 /var/mqm입니다. 기본 접두부는 mqs.ini 구성 파일의 DefaultPrefix 스탠자에서 구성됩니다.

사용 가능한 경우, IBM WebSphere MQ 탐색기의 WebSphere MQ 특성 페이지를 사용하여 접두부를 변경할 수 있습니다. 그렇지 않으면 mqs.ini 구성 파일을 수동으로 편집하십시오.

- 큐 관리자 이름이 올바른 디렉토리 이름으로 변환됩니다. 예를 들어, 다음 큐 관리자의 경우

```
queue.manager
```

다음과 같이 표시됩니다.

```
queue!manager
```

이 프로세스를 이름 변환이라고 합니다.

WebSphere MQ에서 큐 관리자에게 48자까지 포함하는 이름을 부여할 수 있습니다.

예를 들어, 큐 관리자의 이름을 다음과 같이 지정할 수 있습니다.

하지만 각 큐 관리자는 파일로 표시되며 파일 이름의 최대 길이 및 이름에 사용될 수 있는 문자 수에 대한 제한이 있습니다. 결과적으로 오브젝트를 나타내는 파일의 이름이 자동으로 변환되어 파일 시스템의 요구사항을 충족시킵니다.

큐 관리자 이름의 변환을 제어하는 규칙은 다음과 같습니다.

1. 개별 문자 변환:
 - .에서 예!
 - /에서 &로
2. 이름이 여전히 올바르지 않은 경우에는 다음을 수행합니다.
 - a. 여덟개 문자로 이름을 자릅니다.
 - b. 세 숫자 접미어를 첨부합니다.

예를 들어, 기본 접두부 및 큐 관리자의 이름을 `queue.manager`로 가정합니다.

- In WebSphere MQ for 윈도우 with NTFS or FAT32, the queue manager name becomes:

```
c:\Program Files\IBM\WebSphere MQ\mqgrs\queue!manager
```

- In WebSphere MQ for 윈도우 with FAT, the queue manager name becomes:

```
c:\Program Files\IBM\WebSphere MQ\mqgrs\queue!ma
```

- UNIX and Linux 시스템용 WebSphere MQ 에서 큐 관리자 이름은 다음과 같습니다.

```
/var/mqm/mqgrs/queue!manager
```

변환 알고리즘은 또한 대소문자를 구분하지 않는 파일 시스템에서만 달라지는 이름을 구별합니다.

오브젝트 이름 변환

오브젝트 이름이 올바른 파일 시스템 이름일 필요는 없습니다. 오브젝트 이름을 변환해야 합니다. 사용된 메소드는 큐 관리자 이름의 메소드와 다릅니다. 각 시스템에 소수의 큐 관리자 이름만 있더라도 각 큐 관리자에 대해 많은 수의 다른 오브젝트가 있을 수 있기 때문입니다. 큐, 프로세스 정의, 이름 목록, 채널, 클라이언트 연결 채널, 리스너, 서비스 및 인증 정보 오브젝트는 파일 시스템에 표시됩니다.

변환 프로세스가 새 이름을 생성할 때 원래 오브젝트 이름에 대한 단순 관계가 없습니다. `dspmqfls` 명령을 사용하여 실제와 변환된 오브젝트 이름 사이에서 변환할 수 있습니다.

오브젝트 관리

오브젝트 작성, 변경, 표시 및 삭제 방법의 개요

추가 정보에 대해서는 [95 페이지의 『오브젝트』](#)의 내용을 참조하십시오.

동적 큐의 경우를 제외하고 이러한 오브젝트에 대한 작업을 할 수 있으려면 먼저 큐 관리자에 대해 오브젝트를 정의해야 합니다.

다음을 사용하여 오브젝트를 정의하고 관리합니다.

- [프로그래밍 가능 명령 형식 참조 및 관리 태스크 자동화에 설명된 PCF 명령](#)
- [MQSC 명령에 설명된 MQSC 명령](#)
- WebSphere MQ 탐색기 (Windows, UNIX, Intel 시스템의 경우 Linux)

또한 다음 방법을 사용하여 오브젝트를 관리할 수도 있습니다.

- 키보드로 입력하는 제어 명령. [제어 명령](#)을 참조하십시오.
- 프로그램에서의 MQAI(IBM WebSphere MQ Administration Interface) 호출. [WebSphere MQ 관리 인터페이스\(MQAI\)](#)를 참조하십시오.
- **Windows** IBM WebSphere MQ for Windows 전용
 - 프로그램에서의 MQAI COM(Component Object Model) 호출
 - Windows 기본 구성 애플리케이션

오브젝트의 속성을 표시하거나 변경하고 또는 오브젝트를 삭제할 수 있습니다.

Windows **Linux** **UNIX** 윈도우, UNIX and Linux 시스템에서 WebSphere MQ 명령 시퀀스의 경우 MQSC 기능을 사용하여 파일에 보유된 일련의 명령을 실행할 수 있습니다.

오브젝트 속성

오브젝트의 특성은 그 속성으로 정의됩니다. 일부는 지정할 수 있고 그 외에는 보기만 가능합니다.

예를 들어, 큐가 수용할 수 있는 최대 메시지 길이는 큐의 *MaxMsgLength* 속성으로 정의할 수 있으며 큐를 작성할 때 이 속성을 지정할 수 있습니다. *DefinitionType* 속성은 큐가 작성된 방법을 지정하며 이 속성은 표시만 할 수 있습니다.

WebSphere MQ에는 다음 속성을 참조하는 두 가지 방식이 있습니다.

- 해당 PCF 이름 사용(예: *MaxMsgLength*)
- 해당 MQSC 명령 이름 사용(예: MAXMSGL)

이 안내서는 주로 MQSC 명령을 사용한 속성 지정 방법을 설명하고 있기 때문에 PCF 이름보다는 MQSC 명령 이름을 사용하는 대부분의 속성을 참조합니다.

클러스터

클러스터에서 큐 관리자를 그룹화할 수 있습니다. 클러스터에 있는 큐 관리자는 호스트하는 큐를 클러스터에 있는 다른 큐 관리자에 대해서도 사용 가능하도록 작성할 수 있습니다. 모든 큐 관리자는 표준 분산 큐잉에 필요한 여러 오브젝트 정의 없이도 동일한 클러스터 내에 있는 다른 큐 관리자에게 메시지를 송신할 수 있습니다.

분산 큐잉을 사용하는 일반적인 WebSphere MQ 네트워크에서 모든 큐 관리자는 독립적입니다. 한 큐 관리자가 다른 큐 관리자에게 메시지를 보내야 하는 경우 전송 큐, 리모트 큐 관리자에 대한 채널 및 메시지를 보내려는 모든 큐에 대한 리모트 큐 정의를 정의해야 합니다.

클러스터는 큐 관리자 그룹이며 이 그룹은 전송 큐, 채널 및 리모트 큐 정의를 필요로 하지 않고 큐 관리자가 단일 네트워크를 통해 다른 큐 관리자와 직접 통신할 수 있는 방식으로 설정됩니다.

클러스터의 각 큐 관리자는 메시지를 클러스터 내의 다른 큐 관리자에게 전송할 수 있는 단일 전송 큐를 가지고 있습니다.

관련 개념

[클러스터 설계](#)

클러스터 개념 및 작동 방식에 대한 이해

관련 태스크

[큐 관리자 클러스터 구성](#)

[새 클러스터 설정](#)

시스템 기본 오브젝트

시스템 기본 오브젝트에 대한 소개 및 추가 정보에 대한 링크입니다.

시스템 기본 오브젝트는 큐 관리자가 작성될 때마다 자동으로 작성되는 오브젝트 정의 세트입니다. 설치 시 애플리케이션에서 사용하기 위해 이러한 오브젝트 정의를 복사하고 수정할 수 있습니다.

기본 오브젝트 이름에는 어간(SYSTEM)이 포함됩니다(예를 들어, 기본 로컬 큐는 SYSTEM.DEFAULT.LOCAL.QUEUE이고 기본 수신자 채널은 SYSTEM.DEF.RECEIVER임). 이러한 오브젝트의 이름을 바꿀 수 없으며 이러한 이름의 기본 오브젝트가 필요합니다.

오브젝트를 정의할 때 명시적으로 지정하지 않은 모든 속성은 적절한 기본 오브젝트에서 복사됩니다. 예를 들어, 로컬 큐를 정의하는 경우 지정하지 않은 속성은 기본 큐 SYSTEM.DEFAULT.LOCAL.QUEUE로부터 가져옵니다.

시스템 기본값에 대한 자세한 정보는 [시스템 및 기본 오브젝트](#)를 참조하십시오.

IBM WebSphere MQ 멀티캐스트

IBM WebSphere MQ 멀티캐스트는 낮은 지연, 높은 팬아웃, 신뢰할 수 있는 멀티캐스트 메시징을 제공합니다.

멀티캐스트는 성능을 저하시키지 않고 다수의 구독자에 맞게 스케일링할 수 있기 때문에 발행/구독 메시징의 효율적인 형식입니다. WebSphere MQ는 수신확인, 부정적 수신확인 및 순서 번호를 사용하여 높은 팬아웃을 갖는 낮은 지연 메시징을 달성함으로써 신뢰할 수 있는 멀티캐스트 메시징을 가능하게 합니다.

WebSphere MQ 멀티캐스트의 공정한 전달은 거의 동시 전달을 가능하게 하므로, 어떤 수신자도 우위를 차지하지 않도록 보장합니다. WebSphere MQ 멀티캐스트가 네트워크를 사용하여 메시지를 전달하기 때문에 발행/구독 엔진은 데이터를 팬아웃하기 위해 필요하지 않습니다. 토픽이 그룹 주소에 맵핑된 후, 발행자와 구독자가 피어 투 피어 모드에서 동작할 수 있기 때문에 큐 관리자가 필요없습니다. 여기에서는 큐 관리자 서버에 대한 로드 감소하므로, 큐 관리자 서버가 더 이상 잠재적인 실패 지점이 되지 않습니다.

초기 멀티캐스트 개념

WebSphere MQ 멀티캐스트는 통신 정보(COMMINFO) 오브젝트를 사용하여 기존 시스템과 애플리케이션에 쉽게 통합할 수 있습니다. 두 개의 TOPIC 오브젝트 필드를 사용하면 기존 TOPIC 오브젝트의 빠른 구성에서 멀티캐스트 트래픽을 지원하거나 무시할 수 있습니다.

멀티캐스트에 필요한 오브젝트

다음 정보는 WebSphere MQ 멀티캐스트에 필요한 두 개의 오브젝트에 대한 간단한 개요입니다.

COMMINFO 오브젝트

COMMINFO 오브젝트에는 멀티캐스트 전송과 연관된 속성이 포함됩니다. COMMINFO 오브젝트 매개변수에 대한 자세한 정보는 [DEFINE COMMINFO](#)를 참조하십시오.

설정해야 하는 COMMINFO 필드만 COMMINFO 오브젝트의 이름입니다. 이 이름은 토픽에 대해 COMMINFO 오브젝트를 식별하는 데 사용됩니다. 값이 올바른 멀티캐스트 그룹 주소인지 확인하려면 COMMINFO 오브젝트의 **GRPADDR** 필드를 선택해야 합니다.

TOPIC 오브젝트

토픽은 발행/구독 메시지로 발행된 정보의 주제이며, 이러한 토픽은 TOPIC 오브젝트를 작성하여 정의합니다. TOPIC 오브젝트 매개변수에 대한 자세한 정보는 [DEFINE TOPIC](#)을 참조하십시오.

TOPIC 오브젝트 매개변수의 값(**COMMINFO** 및 **MCAST**)을 변경하여 멀티캐스트에서 기존 토픽을 사용할 수 있습니다.

- **COMMINFO** 이 매개변수는 멀티캐스트 통신 정보 오브젝트의 이름을 지정합니다.
- **MCAST** 이 매개변수는 토픽 트리의 이 지점에서 멀티캐스트를 허용할 수 있는지의 여부를 지정합니다. 기본적으로 **MCAST** 은 **ASPARENT** 로 설정되며 이는 주제의 멀티캐스트 속성이 상위에서 상속됨을 의미합니다. **MCAST**를 **ENABLED**로 설정하면 이 노드에서 멀티캐스트 트래픽이 가능합니다.

멀티캐스트 네트워크 및 토픽

다음 정보는 여러 유형의 구독 및 토픽 정의가 있는 구독에 발생하는 상황에 대한 개요입니다. 이러한 예에서는 모두 TOPIC 오브젝트 **COMMINFO** 매개변수가 올바른 COMMINFO 오브젝트의 이름으로 설정되어 있다고 가정합니다.

사용 가능한 멀티캐스트로 설정된 토픽

토픽 문자열 **MCAST** 매개변수를 **ENABLED**로 설정하면, 다음과 같은 경우를 제외하고는 멀티캐스트 가능 클라이언트의 구독이 허용되며 멀티캐스트 구독이 이루어집니다.

- 멀티캐스트 가능 클라이언트의 지속 가능 구독인 경우
- 멀티캐스트 가능 클라이언트의 비관리 구독인 경우

- 멀티캐스트 불가능 클라이언트의 구독인 경우

이러한 경우 멀티캐스트가 아닌 구독이 작성되며 구독이 일반 발행/구독으로 다운그레이드됩니다.

사용 불가능한 멀티캐스트로 설정된 토픽

토픽 문자열 **MCAST** 매개변수가 **DISABLED**로 설정되면, 항상 멀티캐스트가 아닌 구독이 이루어지고 구독이 일반 발행/구독으로 다운그레이드됩니다.

멀티캐스트로만 설정된 토픽

토픽 문자열 **MCAST** 매개변수를 **ONLY**로 설정하면, 다음과 같은 경우를 제외하고는 멀티캐스트 가능 클라이언트의 구독이 허용되며 멀티캐스트 구독이 이루어집니다.

- 지속 가능 구독임: 지속 가능 구독은 이유 코드 **2436 (0984) (RC2436)**: **MQRC_DURABILITY_NOT_ALLOWED**로 거부됨
- 비관리 구독임: 비관리 구독은 이유 코드 **2046 (07FE) (RC2046)**: **MQRC_OPTIONS_ERROR**로 거부됨
- 멀티캐스트가 불가능한 클라이언트의 구독임: 이러한 구독은 이유 코드 **2560 (0A00) (RC2560)**: **MQRC_MULTICAST_ONLY**로 거부됨
- 로컬로 바인드된 애플리케이션의 구독임: 이러한 구독은 이유 코드 **2560 (0A00) (RC2560)**: **MQRC_MULTICAST_ONLY**로 거부됨

보안

IBM WebSphere MQ에는 권한 서비스 인터페이스, 사용자 작성 또는 써드파티 채널 엑시트, SSL(Secure Sockets Layer)을 사용하는 채널 보안, 채널 인증 레코드 및 메시지 보안 등과 같이 보안을 제공하는 여러 가지 방법이 있습니다.

권한 서비스 인터페이스

MQI 호출에 대한 인증, 명령 및 오브젝트에 대한 액세스는 기본적으로 사용 설정된 **오브젝트 권한 관리자(OAM)**에 의해 제공됩니다. IBM WebSphere MQ 엔티티에 대한 액세스는 IBM WebSphere MQ 사용자 그룹 및 OAM을 통해 제어됩니다. 관리자는 명령행 인터페이스를 사용하여 필요에 따라 권한을 부여하거나 폐기할 수 있습니다.

권한 서비스 컴포넌트 작성에 대한 자세한 정보는 [Windows, UNIX and Linux 시스템에서 보안 설정을 참조하십시오](#).

사용자 작성 또는 써드파티 채널 엑시트

채널은 사용자 작성 또는 써드파티 채널 엑시트를 사용할 수 있습니다. 자세한 정보는 [메시지 채널의 채널-엑시트 프로그램](#)을 참조하십시오.

SSL을 사용한 채널 보안

SSL(Secure Sockets Layer) 프로토콜은 도청, 도용 및 위장에 대한 보호와 산업 표준 채널 보안을 제공합니다.

SSL은 메시지 기밀성 및 무결성을 제공하는 공용 키 및 대칭 기술과 상호 인증을 사용합니다.

SSL에 대한 세부 정보를 포함하여 IBM WebSphere MQ의 보안에 대한 종합적 검토는 [보안을 참조하십시오](#). 이 절에 설명된 명령에 대한 포인터를 포함하여 SSL의 개요는 [암호화 보안 프로토콜: SSL 및 TLS](#)를 참조하십시오.

채널 인증 레코드

채널 인증 레코드를 사용하여 채널 레벨에서 시스템 연결에 부여된 액세스 권한에 대해 정확한 제어를 실행합니다. 자세한 정보는 [채널 인증 레코드](#)를 참조하십시오.

메시지 보안

IBM WebSphere MQ의 별도로 설치되고 라이선스가 부여된 구성요소인 Advanced Message Security를 사용하여 IBM WebSphere MQ를 사용하여 전송 및 수신된 메시지에 대한 암호화 보호를 제공합니다. [WebSphere MQ Advanced Message Security](#)를 참조하십시오.

관련 태스크

보안

보안 요구사항 계획

클라이언트 및 서버

IBM WebSphere MQ가 해당 애플리케이션에 대한 클라이언트 서버 구성을 지원하는 방법에 대한 소개입니다.

IBM WebSphere MQ MQI 클라이언트는 시스템에서 실행 중인 애플리케이션이 다른 시스템에서 실행 중인 큐 관리자에 대한 MQI 호출을 실행할 수 있도록 허용하는 컴포넌트입니다. 호출의 출력은 클라이언트로 되돌아오고 이는 애플리케이션으로 다시 전달됩니다.

IBM WebSphere MQ 서버는 하나 이상의 클라이언트에게 큐잉 서비스를 제공하는 큐 관리자입니다. 모든 IBM WebSphere MQ 오브젝트(예: 큐)는 큐 관리자 시스템(IBM WebSphere MQ 서버 시스템)에만 존재하며 클라이언트에는 존재하지 않습니다. IBM WebSphere MQ 서버는 로컬 IBM WebSphere MQ 애플리케이션도 지원할 수 있습니다.

IBM WebSphere MQ 서버와 일반 큐 관리자 간의 차이점은 서버가 각 클라이언트에 대한 전용 통신 링크를 가진다는 점입니다. 클라이언트 및 서버의 채널 작성에 대한 자세한 정보는 [분산 큐잉을 사용하여 애플리케이션 연결을 참조하십시오](#).

일반적인 클라이언트에 대한 정보는 [118 페이지의 『IBM WebSphere MQ MQI 클라이언트 개요』](#)를 참조하십시오.

클라이언트 서버 환경의 IBM WebSphere MQ 애플리케이션

서버에 링크되면 클라이언트 IBM WebSphere MQ 애플리케이션은 로컬 애플리케이션과 같은 방식으로 MQI 호출의 대부분을 발행할 수 있습니다. 클라이언트 애플리케이션은 지정된 큐 관리자에 연결하기 위한 MQCONN 호출을 발행합니다. 연결 요청에서 리턴된 연결 핸들을 지정하는 추가 MQI 호출은 이 큐 관리자가 처리할 수 있습니다.

애플리케이션을 적절한 클라이언트 라이브러리로 링크해야 합니다. [WebSphere MQ MQI 클라이언트의 애플리케이션 빌드를 참조하십시오](#).

관련 개념

[124 페이지의 『트랜잭션 관리 및 지원』](#)

트랜잭션 관리 및 WebSphere MQ가 트랜잭션을 지원하는 방법 소개

[126 페이지의 『큐 관리자 기능 확장』](#)

사용자 엑시트, API 엑시트 또는 설치 가능 서비스를 사용하여 큐 관리자 기능을 확장할 수 있습니다.

IBM WebSphere MQ MQI 클라이언트 개요

WebSphere MQ MQI 클라이언트는 큐 관리자가 실행되지 않는 시스템에 설치할 수 있는 IBM WebSphere MQ 제품의 컴포넌트입니다.

IBM WebSphere MQ MQI 클라이언트를 사용하면, 클라이언트와 동일한 시스템에서 실행 중인 애플리케이션이 다른 시스템에서 실행 중인 큐 관리자에 연결할 수 있습니다. 애플리케이션은 해당 큐 관리자에게 MQI 호출을 발행할 수 있습니다. 이러한 애플리케이션은 *WebSphere MQ MQI* 클라이언트 애플리케이션이라고 하며 이 큐 관리자는 서버 큐 관리자라고 합니다.

IBM WebSphere MQ MQI 클라이언트 애플리케이션과 서버 큐 관리자는 MQI 채널을 사용하여 서로 통신합니다. MQI 채널은 클라이언트 애플리케이션이 MQCONN 또는 MQCONNX 호출을 발행하여 큐 관리자에 연결할 때 시작하고 클라이언트 애플리케이션이 MQDISC 호출을 발행하여 큐 관리자와의 연결을 끊을 때 종료됩니다. MQI 호출의 입력 매개변수는 MQI 채널에서 한 방향으로 플로우하고 출력 매개변수는 반대 방향으로 플로우합니다.

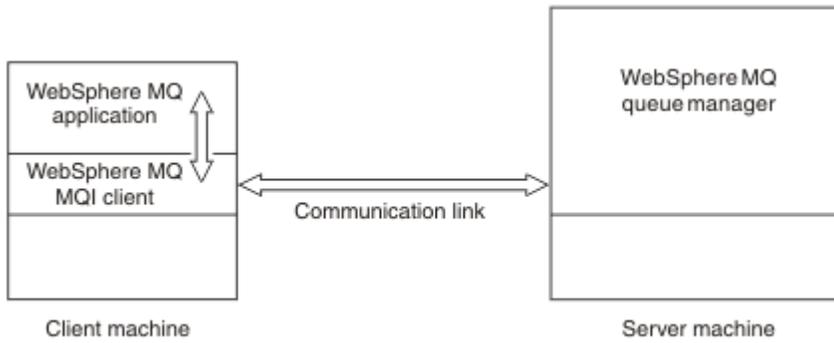


그림 30. 클라이언트와 서버 사이의 링크

다음 플랫폼을 사용할 수 있습니다. 조합은 사용 중인 IBM WebSphere MQ 제품에 따라 다르며 [121 페이지의 『IBM WebSphere MQ 클라이언트에 대한 플랫폼 지원』](#)에 설명되어 있습니다.

IBM WebSphere MQ MQI 클라이언트

UNIX and Linux
Windows

HP Integrity NonStop Server

IBM WebSphere MQ 서버

UNIX and Linux
Windows

HP Integrity NonStop Server

MQI는 클라이언트 플랫폼에서 실행 중인 애플리케이션에 사용 가능합니다. 큐 및 기타 IBM WebSphere MQ 오브젝트는 서버에 설치한 큐 관리자에 보유됩니다.

IBM WebSphere MQ MQI 클라이언트 환경에서 실행하려는 응용프로그램은 먼저 관련 클라이언트 라이브러리와 링크되어야 합니다. 애플리케이션이 MQI 호출을 실행할 때, IBM WebSphere MQ MQI 클라이언트는 요청을 큐 관리자로 전달하며, 여기서 이 요청은 처리되고 응답이 IBM WebSphere MQ MQI 클라이언트로 다시 전송됩니다.

응용프로그램과 IBM WebSphere MQ MQI 클라이언트 사이의 링크는 런타임 시 동적으로 설정됩니다.

You can also develop client applications using the IBM WebSphere MQ classes for .NET, IBM WebSphere MQ classes for Java or IBM WebSphere MQ classes for Java Message Service (JMS). You can use Java and JMS clients on UNIX, Linux and Windows platforms. Java 및 JMS의 사용은 여기에 설명되어 있지 않습니다. For full details on how to install, configure, and use IBM WebSphere MQ classes for Java and IBM WebSphere MQ classes for JMS see [Java에 대해 WebSphere MQ 클래스 사용](#) and [JMS에 대해 WebSphere MQ 클래스 사용](#).

관련 개념

[120 페이지의 『IBM WebSphere MQ 클라이언트를 사용하는 이유』](#)

IBM WebSphere MQ 클라이언트 사용은 IBM WebSphere MQ 메시징 및 큐잉을 구현하는 효율적인 방법입니다.

[121 페이지의 『IBM WebSphere MQ MQI 클라이언트를 설정하는 방법』](#)

다음 지시사항에 따라 클라이언트를 설정할 수 있습니다.

[122 페이지의 『확장된 트랜잭션 클라이언트 개념』](#)

WebSphere MQ 확장 트랜잭션 클라이언트는 외부 트랜잭션 관리자의 제어 하에 다른 자원 관리자가 관리하는 자원을 업데이트할 수 있습니다.

[123 페이지의 『클라이언트를 서버에 연결하는 방법』](#)

클라이언트는 MQCONN 또는 MQCONNX를 사용하여 서버에 연결되고 채널을 통해서 통신합니다.

IBM WebSphere MQ 클라이언트를 사용하는 이유

IBM WebSphere MQ 클라이언트 사용은 IBM WebSphere MQ 메시징 및 큐잉을 구현하는 효율적인 방법입니다.

한 시스템에서 실행 중인 MQI 및 다른 시스템에서 실행 중인 큐 관리자를 사용하는 애플리케이션을 가질 수 있습니다(실제 또는 가상). 이를 실행할 경우의 장점은 다음과 같습니다.

- 클라이언트 시스템에 전체 WebSphere MQ 구현이 필요하지 않습니다.
- 클라이언트 시스템에 대한 하드웨어 요구사항이 감소됩니다.
- 시스템 관리 요구사항이 감소됩니다.
- 클라이언트에서 실행되는 WebSphere MQ 애플리케이션을 다른 시스템의 다중 큐 관리자에 연결할 수 있습니다.
- 다른 전송 프로토콜을 사용하는 대체 채널을 사용할 수 있습니다.

관련 참조

120 페이지의 『IBM WebSphere MQ MQI 클라이언트에서 실행되는 애플리케이션은 무엇입니까?』 클라이언트 환경에서 전체 MQI가 지원됩니다.

121 페이지의 『IBM WebSphere MQ 클라이언트에 대한 플랫폼 지원』

IBM WebSphere MQ on all server platforms accepts client connections from IBM WebSphere MQ MQI clients on UNIX or Linux systems, and 윈도우.

IBM WebSphere MQ MQI 클라이언트에서 실행되는 애플리케이션은 무엇입니까? 클라이언트 환경에서 전체 MQI가 지원됩니다.

이를 통해 거의 모든 WebSphere MQ 응용프로그램이 IBM WebSphere MQ MQI 클라이언트의 응용프로그램을 MQI 라이브러리가 아닌 MQIC 라이브러리에 링크하여 IBM WebSphere MQ MQI 클라이언트 시스템에서 실행 되도록 구성할 수 있습니다. 예외사항은 다음과 같습니다.

- 신호가 있는 MQGET
- 확장된 트랜잭션 클라이언트를 사용해야 하며 다른 자원 관리자와의 동기점 조정이 필요한 애플리케이션

미리 읽기를 사용하는 경우 비지속 메시징 성능을 향상시키는 데 모든 MQGET 옵션이 사용 가능한 것은 아닙니다. 테이블은 허용되는 옵션 및 이러한 옵션이 MQGET 호출 사이에서 대체될 수 있는지 여부를 표시합니다.

	미리 읽기가 사용 가능한 경우 허용되며 MQGET 호출 간에 대체 가능	미리 읽기가 사용 가능한 경우 허용되지만 MQGET 호출 사이에 변경될 수 없음 ¹	미리 읽기가 사용 가능한 경우 허용되지 않는 MQGET 옵션 ²
MQGET MD 값	MsgId ³ CorrelId ³	Encoding CodedCharSetId	
MQGET MQGMO 옵션	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF_QUIESCING MQGMO_BROWSE_FIRST ⁴ MQGMO_BROWSE_NEXT ⁴ MQGMO_BROWSE_MESSAGE_UNDER_CURSOR ⁴	MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT MQGMO_LOGICAL_ORDER MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_AVAILABLE MQGMO_ALL_SEGMENTS_AVAILABLE MQGMO_MARK_BROWSE_HANDLE MQGMO_MARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_HANDLE MQGMO_UNMARKED_BROWSE_MSG MQGMO_PROPERTIES_FORCE_MQRFH2 MQGMO_NO_PROPERTIES MQGMO_PROPERTIES_IN_HANDLE MQGMO_PROPERTIES_COMPATIBILITY	MQGMO_SET_SIGNAL MQGMO_SYNCPOINT MQGMO_MARK_SKIP_BACKOUT MQGMO_MSG_UNDER_CURSOR ⁴ MQGMO_LOCK MQGMO_UNLOCK
MQGMO 값		MsgHandle	

1. 이 옵션이 MQGET 호출 간에 대체되면 MQRC_OPTIONS_CHANGED 이유 코드가 리턴됩니다.
2. 이 옵션을 첫 번째 MQGET 호출에 지정할 경우 미리 읽기를 사용할 수 없습니다. 이 옵션을 후속 MQGET 호출에 지정할 경우 이유 코드 MQRC_OPTIONS_ERROR가 리턴됩니다.
3. 클라이언트 애플리케이션은 MsgId 및 CorrelId 값이 MQGET 호출 간에 대체된 경우, 이전 값을 가진 메시지가 이미 클라이언트에 전송되어 이용(또는 자동으로 제거)될 때까지 클라이언트 미리 읽기 버퍼에 남아있는지 알아야 합니다.

4. 첫 번째 MQGET 호출은 메시지를 찾아볼지 아니면 미리 읽기가 가능할 때 큐에서 가져올지 여부를 판별합니다. 애플리케이션이 찾아보기와 가져오기의 결합을 사용하려고 할 경우 MQRC_OPTIONS_CHANGED 이유 코드가 리턴됩니다.
5. MQGMO_MSG_UNDER_CURSOR는 미리 읽기에 사용할 수 없습니다. 메시지를 찾아보거나 미리 읽기가 가능할 때 가져올 수 있지만, 이 둘을 결합할 수는 없습니다.

IBM WebSphere MQ MQI 클라이언트에서 실행 중인 애플리케이션은 둘 이상의 큐 관리자에 동시에 연결하거나, MQCONN 또는 MQCONNX 호출에서 큐 관리자 이름을 별표 (*) 로 사용할 수 있습니다 (큐 관리자에 IBM WebSphere MQ MQI 클라이언트 애플리케이션 연결의 예제 참조).

IBM WebSphere MQ 클라이언트에 대한 플랫폼 지원

IBM WebSphere MQ on all server platforms accepts client connections from IBM WebSphere MQ MQI clients on UNIX or Linux systems, and 윈도우.

WebSphere MQ installed as a 기본 제품 및 서버 can accept connections from the IBM WebSphere MQ MQI clients on the following platforms:

- HP Integrity NonStop Server
- UNIX and Linux 시스템
- Windows

클라이언트 연결은 CCSID(Coded Character Set ID)와 통신 프로토콜과의 차이에 따라 달라집니다.

IBM WebSphere MQ MQI 클라이언트를 설정하는 방법

다음 지시사항에 따라 클라이언트를 설정할 수 있습니다.

IBM WebSphere MQ MQI 클라이언트를 설정하려면 클라이언트가 연결할 IBM WebSphere MQ 서버가 이미 설치되어 작동 중이어야 합니다. 클라이언트 설정에 관한 단계는 다음과 같습니다.

1. IBM WebSphere MQ MQI 클라이언트에 적합한 플랫폼이 있는지 및 하드웨어와 소프트웨어가 요구사항을 충족시키는지 확인하십시오. 플랫폼 지원은 [121 페이지의 『IBM WebSphere MQ 클라이언트에 대한 플랫폼 지원』](#)에 설명되어 있습니다.
2. 클라이언트 워크스테이션에 IBM WebSphere MQ를 설치할 방법을 결정한 다음 클라이언트와 서버 플랫폼의 특정 조합에 대한 지시사항을 따르십시오. 설치에 [IBM WebSphere MQ 클라이언트 설치](#)에 설명되어 있습니다.
3. 통신 링크가 구성되고 연결되는지 확인하십시오. 통신 링크의 구성은 [서버 및 클라이언트 간의 연결 구성](#)에 설명되어 있습니다.
4. 설치가 올바르게 작동하는지 확인하십시오. 설치 확인은 [클라이언트 설치 확인](#)에 설명되어 있습니다.
5. IBM WebSphere MQ MQI 클라이언트 설치를 확인한 후에는 클라이언트에 보안 설정을 해야 하는지 여부를 고려하십시오. 클라이언트 보안은 [IBM WebSphere MQ MQI 클라이언트 보안 설정](#)에 설명되어 있습니다.
6. 클라이언트에서 실행하려는 IBM WebSphere MQ 애플리케이션이 필요로 하는 IBM WebSphere MQ MQI 클라이언트 및 서버 사이에 채널을 설정하십시오. 채널 설정은 [MQI 채널 정의](#)에 설명되어 있습니다. SSL을 사용하는 경우 일부 추가 고려사항이 있습니다. 이러한 고려사항은 [MQI 채널이 SSL을 사용하도록 지정](#)에 설명되어 있습니다. 채널을 설정하기 위해 IBM WebSphere MQ MQI 클라이언트 구성 파일 또는 IBM WebSphere MQ 환경 변수를 사용해야 할 수도 있습니다. IBM WebSphere MQ 환경 변수는 [IBM WebSphere MQ 환경 변수 사용](#)에 설명되어 있습니다.
7. IBM WebSphere MQ 애플리케이션은 [애플리케이션 개발](#)에 자세히 설명되어 있습니다.
8. IBM WebSphere MQ MQI 클라이언트 환경에서 애플리케이션을 설계, 빌드 및 실행하는 경우 고려해야 하는 큐 관리자 환경과의 일부 차이점이 있습니다. 해당 차이점에 대한 정보는 다음을 참조하십시오.
 - [클라이언트 애플리케이션에서 메시지 큐 인터페이스\(MQI\) 사용](#)
 - [WebSphere MQ MQI 클라이언트의 애플리케이션 빌드](#)
 - [큐 관리자에 IBM WebSphere MQ MQI 클라이언트 애플리케이션 연결](#)
 - [IBM WebSphere MQ MQI 클라이언트에 관련된 문제점 해결](#)

확장된 트랜잭션 클라이언트 개념

WebSphere MQ 확장 트랜잭션 클라이언트는 외부 트랜잭션 관리자의 제어 하에 다른 자원 관리자가 관리하는 자원을 업데이트할 수 있습니다.

트랜잭션 관리의 개념에 익숙하지 않은 경우 [124 페이지의 『트랜잭션 관리 및 지원』](#)의 내용을 참조하십시오.

현재 XA 트랜잭션 클라이언트가 WebSphere MQ의 일부로 제공되고 있습니다.

클라이언트 애플리케이션은 연결된 큐 관리자가 관리하는 작업 단위에 참여할 수 있습니다. 작업 단위 내에서 클라이언트 애플리케이션은 해당 큐 관리자가 소유하는 큐에 메시지를 넣고 큐에서 메시지를 가져올 수 있습니다. 클라이언트 애플리케이션은 **MQCMIT** 호출을 사용하여 작업 단위를 커밋하거나 **MQBACK** 호출을 사용하여 작업 단위를 백아웃할 수 있습니다. 하지만 동일한 작업 단위 내에서 클라이언트 애플리케이션은 다른 자원 관리자의 자원(예: DB2® 데이터베이스의 테이블)을 업데이트할 수 없습니다. WebSphere MQ 확장 트랜잭션 클라이언트를 사용하여 이 제한을 제거합니다.

WebSphere MQ 확장 트랜잭션 클라이언트는 일부 추가 기능이 있는 IBM WebSphere MQ MQI client입니다. 이 기능을 사용하면 동일한 작업 단위 내에 있는 클라이언트 애플리케이션에서 다음 태스크를 수행할 수 있습니다.

- 연결된 큐 관리자가 소유하는 큐에 메시지 넣기 및 큐에서 메시지 가져오기
- WebSphere MQ 큐 관리자가 아닌 자원 관리자의 자원 업데이트

이 작업 단위는 클라이언트 애플리케이션과 동일한 시스템에서 실행 중인 외부 트랜잭션 관리자에 의해 관리되어야 합니다. 작업 단위는 클라이언트 애플리케이션이 연결된 큐 관리자에 의해 관리될 수 없습니다. 이는 큐 관리자가 트랜잭션 관리자가 아닌 자원 관리자로서만 작업을 수행할 수 있음을 의미합니다. 또한 클라이언트 애플리케이션이 외부 트랜잭션 관리자가 제공한 API(Application Programming Interface)만을 사용하여 작업 단위를 커밋하거나 백아웃할 수 있음을 의미합니다. 따라서 클라이언트 응용프로그램은 MQI 호출 (**MQBEGIN**, **MQCMIT** 및 **MQBACK**)을 사용할 수 없습니다.

외부 트랜잭션 관리자는 큐 관리자에 연결된 클라이언트 애플리케이션이 사용하는 것과 동일한 MQI 채널을 사용하는 자원 관리자로서 큐 관리자와 통신합니다. 하지만 장애를 복구하는 상황에서, 실행되는 애플리케이션이 없을 때 트랜잭션 관리자는 전용 MQI 채널을 사용하여 큐 관리자가 실패 시에 참여하고 있었던 불완전한 작업 단위를 복구할 수 있습니다.

이 절에서 확장 트랜잭션 기능이 없는 WebSphere MQ MQI 클라이언트는 *WebSphere MQ* 기본 클라이언트로 참조됩니다. 따라서 WebSphere MQ 확장 트랜잭션 클라이언트가 확장 트랜잭션 기능이 추가된 WebSphere MQ 기본 클라이언트로 구성되는 것으로 간주할 수 있습니다.

관련 참조

[122 페이지의 『확장된 트랜잭션 클라이언트의 플랫폼 지원』](#)

IBM WebSphere MQ 확장 트랜잭션 클라이언트는 z/OS를 제외하고 기본 클라이언트를 지원하는 모든 플랫폼에 대해 사용 가능합니다.

확장된 트랜잭션 클라이언트의 플랫폼 지원

IBM WebSphere MQ 확장 트랜잭션 클라이언트는 z/OS를 제외하고 기본 클라이언트를 지원하는 모든 플랫폼에 대해 사용 가능합니다.

확장된 트랜잭션 클라이언트를 사용 중인 클라이언트 애플리케이션은 다음 IBM WebSphere MQ Version 7.5 제품의 큐 관리자에만 연결할 수 있습니다.

- AIX 의 경우 IBM WebSphere MQ
- HP-UX 의 경우 IBM WebSphere MQ
- HP Integrity NonStop Server 의 경우 IBM WebSphere MQ
- Linux 의 경우 IBM WebSphere MQ
- Solaris 의 경우 IBM WebSphere MQ
- Windows 의 경우 IBM WebSphere MQ

z/OS에서 실행되는 확장된 트랜잭션 클라이언트가 없는 경우에도 확장된 트랜잭션 클라이언트를 사용 중인 클라이언트 애플리케이션을 z/OS에서 실행되는 큐 관리자에 연결할 수 있습니다.

각 플랫폼의 경우, 확장된 트랜잭션 클라이언트에 대한 하드웨어 및 소프트웨어 요구사항은 IBM WebSphere MQ 기본 클라이언트에 대한 요구사항과 동일합니다. 프로그래밍 언어는 사용 중인 IBM WebSphere MQ 기본 클라이언트 및 트랜잭션 관리자가 지원하는 경우에는 확장된 트랜잭션 클라이언트에 의해 지원됩니다.

각 플랫폼에 대한 외부 트랜잭션 관리자는 다음 웹 페이지에 나열되어 있습니다.

확장 트랜잭션 클라이언트 플랫폼	웹 페이지
AIX	WebSphere MQ V7.5의 최소 요구사항 - AIX
HP-UX	WebSphere MQ V7.5의 최소 요구사항 - HP-UX
HP Integrity NonStop Server	HP Integrity NonStop Server에서 IBM WebSphere MQ 클라이언트 환경 계획
Linux	WebSphere MQ V7.5의 최소 요구사항 - Linux
Solaris	WebSphere MQ V7.5의 최소 요구사항 - Solaris
Windows	WebSphere MQ V7.5의 최소 요구사항 - Windows

클라이언트를 서버에 연결하는 방법

클라이언트는 MQCONN 또는 MQCONNX를 사용하여 서버에 연결되고 채널을 통해서 통신합니다.

IBM WebSphere MQ 클라이언트 환경에서 실행 중인 애플리케이션은 클라이언트와 서버 시스템 사이의 활성 연결을 유지해야 합니다.

MQCONN 또는 MQCONNX를 발행하는 애플리케이션을 통해 연결이 설정되었습니다. 클라이언트 및 서버는 MQI 채널을 통해 통신하거나 공유 대화를 사용하는 경우에는 각각의 대화가 MQI 채널 인스턴스를 공유합니다. 호출에 성공한 경우 MQI 채널 인스턴스 또는 대화는 애플리케이션이 MQDISC 호출을 발행할 때까지 연결된 상태로 유지됩니다. 이는 애플리케이션이 연결해야 하는 모든 큐 관리자에 해당됩니다.

관련 개념

[123 페이지의 『동일한 시스템의 클라이언트 및 큐 관리자』](#)

시스템에 큐 관리자가 설치된 경우 WebSphere MQ MQI 클라이언트 환경에서 애플리케이션을 실행할 수도 있습니다.

[123 페이지의 『다른 플랫폼의 클라이언트』](#)

WebSphere MQ MQI 클라이언트 및 서버 시스템의 다른 예입니다. 이 예에서는 서버 시스템이 다른 플랫폼에 있는 세 개의 WebSphere MQ MQI 클라이언트와 통신합니다.

[124 페이지의 『다른 버전의 클라이언트 및 서버 소프트웨어 사용』](#)

이전 버전의 IBM WebSphere MQ 제품을 사용하는 경우 클라이언트의 CCSID에서 코드 변환을 서버가 지원하는지 확인하십시오.

동일한 시스템의 클라이언트 및 큐 관리자

시스템에 큐 관리자가 설치된 경우 WebSphere MQ MQI 클라이언트 환경에서 애플리케이션을 실행할 수도 있습니다.

이 상황에서 큐 관리자 라이브러리에 링크할지 또는 클라이언트 라이브러리에 링크할지에 대한 선택을 할 수 있지만 클라이언트 라이브러리에 링크하는 경우 여전히 채널 연결을 정의할 필요가 있다는 점에 유의하십시오. 이는 애플리케이션의 개발 단계에서 유용할 수 있습니다. 자체 시스템에서 다른 시스템에 대한 종속성 없이 프로그램을 테스트할 수 있으며 프로그램을 독립적 WebSphere MQ MQI 클라이언트 환경으로 이동시켜도 이 프로그램이 작동하는지에 대해 확인할 수 있습니다.

다른 플랫폼의 클라이언트

WebSphere MQ MQI 클라이언트 및 서버 시스템의 다른 예입니다. 이 예에서는 서버 시스템이 다른 플랫폼에 있는 세 개의 WebSphere MQ MQI 클라이언트와 통신합니다.

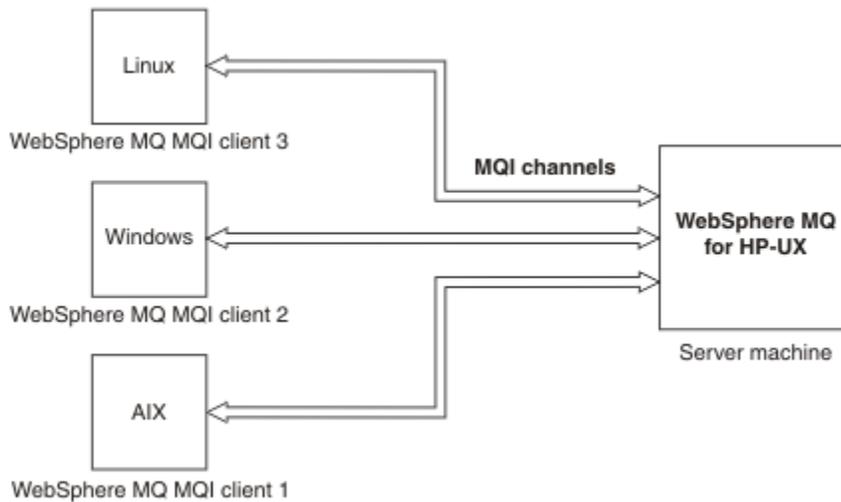


그림 31. 다른 플랫폼의 클라이언트에 연결된 WebSphere MQ 서버

다른 좀 더 복잡한 환경도 가능합니다. 예를 들어 WebSphere MQ 클라이언트는 둘 이상의 큐 관리자 또는 큐 공유 그룹의 일부로서 연결된 모든 큐 관리자에 연결할 수 있습니다.

다른 버전의 클라이언트 및 서버 소프트웨어 사용

이전 버전의 IBM WebSphere MQ 제품을 사용하는 경우 클라이언트의 CCSID에서 코드 변환을 서버가 지원하는지 확인하십시오.

IBM WebSphere MQ 클라이언트는 지원되는 모든 버전의 큐 관리자에 연결할 수 있습니다. 이전 버전 큐 관리자에 연결하는 경우, 클라이언트의 IBM WebSphere MQ 애플리케이션에서 제품의 이후 버전 기능 및 구조를 사용할 수 없습니다.

IBM WebSphere MQ 큐 관리자는 상호 지원되는 최상위 프로토콜 레벨로 협상하여 서로 다른 버전의 클라이언트와 자체적으로 통신할 수 있습니다. 이는 이전 클라이언트가 이후 큐 관리자 레벨에서 사용될 수 있음을 의미합니다. 문제점 진단을 용이하게 하고 IBM의 지원을 사용 가능하게 하기 위해 클라이언트 및 서버가 모두 현재 지원되는 IBM WebSphere MQ 버전에 있는 것이 좋습니다.

자세한 정보는 [사용할 프로그래밍 언어 결정에서 지원되는 프로그래밍 언어를 참조하십시오.](#)

트랜잭션 관리 및 지원

트랜잭션 관리 및 WebSphere MQ가 트랜잭션을 지원하는 방법 소개

자원 관리자는 애플리케이션이 액세스하고 업데이트할 수 있는 자원을 소유하고 관리하는 컴퓨터 서브시스템입니다. 다음은 자원 관리자의 예입니다.

- 큐 자원이 있는 WebSphere MQ 큐 관리자
- 테이블 자원이 있는 DB2 데이터베이스

애플리케이션이 하나 이상의 자원 관리자의 자원을 업데이트할 때, 특정 업데이트가 모두 그룹으로 완료되거나 어떠한 업데이트도 완료되지 않도록 하는 비즈니스 요구사항이 있을 수 있습니다. 이 유형의 요구사항에 대한 이유는 이러한 업데이트 중 일부가 성공적으로 완료되었지만 그 외의 업데이트는 그렇지 못한 경우에 비즈니스 데이터가 일관되지 않은 상태로 남겨지기 때문입니다.

이러한 방식으로 관리되는 자원에 대한 업데이트는 작업 단위 또는 트랜잭션 내에서 발생합니다. 애플리케이션 프로그램은 한 세트의 업데이트를 하나의 작업 단위로 그룹화할 수 있습니다.

작업 단위 중에 애플리케이션은 자원 관리자에게 요청을 발행하여 자원 관리자의 자원을 업데이트합니다. 작업 단위는 애플리케이션이 모든 업데이트 커밋 요청을 발행할 때 종료됩니다. 업데이트가 커밋될 때까지 동일한 자원에 액세스 중인 다른 애플리케이션에 아무 업데이트도 표시되지 않습니다. 그렇지 않은 경우 애플리케이션이 어떤 이유로 작업 단위를 완료할 수 없다고 결정하면 애플리케이션은 요청했던 모든 업데이트를 해당 지점까지 백아웃하도록 하는 요청을 발행할 수 있습니다. 이러한 경우에 모든 업데이트는 다른 애플리케이션에 표시

되지 않습니다. 이러한 업데이트는 보통 논리적으로 연관되어 있으며 보존되어야 하는 데이터 무결성에 대해 모두 성공적이어야 합니다. 다른 업데이트가 실패할 때 한 업데이트가 성공하면 데이터 무결성을 잃게 됩니다.

작업 단위가 성공적으로 완료되면 커미트라고 표시됩니다. 일단 커미트되면 해당 작업 단위 내에서 수행된 모든 업데이트는 영구적이 되며 되돌릴 수 없습니다. 하지만 작업 단위가 실패하면 모든 업데이트는 대신에 백아웃됩니다. 작업 단위가 커미트되거나 무결성으로 백아웃되는 이 프로세스를 동기점 조정이라고 합니다.

작업 단위 내 모든 업데이트가 커미트되거나 백아웃될 때의 지점을 동기점이라고 합니다. 작업 단위 내 업데이트는 동기점 제어 내에서 발생한다고 합니다. 애플리케이션이 동기점 제어 범위 외에 있는 업데이트를 요청하는 경우 진행 중인 작업 단위가 있고 업데이트를 나중에 백아웃할 수 없더라도 자원 관리자는 즉시 업데이트를 커미트합니다.

작업 단위를 관리하는 컴퓨터 서브시스템을 트랜잭션 관리자또는 동기화 점 코디네이터라고 합니다.

로컬 작업 단위에서는 업데이트된 유일한 자원이 WebSphere MQ 큐 관리자의 자원입니다. 여기에 1단계 커미트 프로세스를 사용하는 큐 관리자 스스로 동기점 조정을 제공합니다.

글로벌 작업 단위에서는 XA 준수 데이터베이스와 같이 다른 자원 관리자에 속하는 자원 또한 업데이트됩니다. 여기에, 2단계 커미트 프로시저가 사용되어야 하며 작업 단위는 큐 관리자 자체에서 조정할 수 있거나 IBM TXSeries®, 또는 BEA Tuxedo와 같이 외부에서 다른 XA 호환 트랜잭션 관리자에 의해 조정할 수 있습니다.

트랜잭션 관리자는 작업 단위 내에서의 자원에 대한 모든 업데이트가 성공적으로 완료되는지 또는 업데이트가 전혀 완료되지 않는지 확인해야 합니다. 애플리케이션은 작업 단위의 커미트 또는 백아웃 요청을 트랜잭션 관리자에게 발행합니다. 트랜잭션 관리자의 예는 CICS 및 WebSphere 애플리케이션 서버이며 이 두 트랜잭션 관리자는 모두 다른 기능도 가지고 있습니다.

일부 자원 관리자는 자체 트랜잭션 관리 기능을 제공합니다. 예를 들어, WebSphere MQ 큐 관리자는 자체 자원 업데이트를 포함하는 작업 단위를 관리하고 DB2 테이블로 업데이트할 수 있습니다. 큐 관리자는 사용자가 요구하는 경우 트랜잭션 관리자를 사용할 수 있기는 하지만 이 기능을 수행할 별도의 트랜잭션 관리자를 필요로 하지 않습니다. 별도의 트랜잭션 관리자가 사용되는 경우 외부 트랜잭션 관리자로 참조됩니다.

작업 단위를 관리하는 외부 트랜잭션 관리자의 경우 트랜잭션 관리자와 작업 단위에 참여하는 모든 자원 관리자 사이에는 표준 인터페이스가 있어야 합니다. 이 인터페이스를 통해 트랜잭션 관리자와 자원 관리자가 서로 통신할 수 있습니다. 이러한 인터페이스 중 하나는 XA 인터페이스이며 이 인터페이스는 다수의 트랜잭션 관리자 및 자원 관리자가 지원하는 표준 인터페이스입니다. XA 인터페이스는 분배된 트랜잭션 처리: XA 스펙에서 The Open Group에 의해 발행됩니다.

둘 이상의 자원 관리자가 작업 단위에 참여한 경우 트랜잭션 관리자는 2단계 커미트 프로토콜을 사용하여 시스템 장애가 있다 하더라도 작업 단위 내의 모든 업데이트가 성공적으로 완료되었는지 또는 업데이트가 전혀 완료되지 않았는지 확인해야 합니다. 애플리케이션이 트랜잭션 관리자에게 작업 단위 커미트 요청을 발행하면 트랜잭션 관리자는 다음을 수행합니다.

단계 1(커미트 준비)

트랜잭션 관리자가 작업 단위에 참여하고 있는 각 자원 관리자에게 자원에 대해 예정된 업데이트에 대한 모든 정보가 복구 가능한 상태에 있는지 확인할 것을 요청합니다. 자원 관리자는 일반적으로 로그에 정보를 쓰고 정보가 하드 디스크를 통해 쓰여졌는지 확인하여 이러한 작업을 수행합니다. 단계 1은 트랜잭션 관리자가 각 자원 관리자로부터 자원에 대해 예정된 업데이트에 대한 정보가 복구 가능한 상태에 있다는 알림을 수신하면 완료됩니다.

단계 2(커미트)

단계 1이 완료되면 트랜잭션 관리자는 취소 불가능한 작업 단위 커미트 결정을 내립니다. 트랜잭션 관리자는 작업 단위에 참여하는 각 자원 관리자에게 자원에 대한 업데이트 커미트를 요청합니다. 자원 관리자가 이 요청을 수신하면 업데이트를 커미트해야 합니다. 자원 관리자에게는 이 단계에서 업데이트를 백아웃하는 옵션이 없습니다. 단계 2는 트랜잭션 관리자가 각 자원 관리자로부터 자원에 대한 업데이트를 커미트했다는 알림을 수신하면 완료됩니다.

XA 인터페이스는 2단계 커미트 프로토콜을 사용합니다.

자세한 정보는 [트랜잭션 지원을 참조하십시오](#).

WebSphere MQ는 마이크로소프트 트랜잭션 서버 (COM+)에 대한 지원도 제공합니다. [마이크로소프트 트랜잭션 서버 사용 \(COM+\)](#)는 COM+ 지원을 활용하기 위해 WebSphere MQ를 설정하는 방법에 대한 정보를 제공합니다.

큐 관리자 기능 확장

사용자 엑시트, API 엑시트 또는 설치 가능 서비스를 사용하여 큐 관리자 기능을 확장할 수 있습니다.

사용자 엑시트

사용자 엑시트는 사용자가 고유 코드를 큐 관리자 기능에 삽입할 수 있도록 하는 메커니즘을 제공합니다. 지원되는 사용자 엑시트에는 다음이 포함됩니다.

채널 엑시트

이 엑시트는 채널이 작동하는 방식을 변경합니다. 채널 엑시트는 [메시지 채널에 대한 채널 엑시트 프로그램](#)에 설명되어 있습니다.

데이터 변환 엑시트

이러한 엑시트는 데이터 형식을 하나의 형식에서 다른 형식으로 변환하는 애플리케이션 프로그램에 넣을 수 있는 소스 코드 단편을 작성합니다. 데이터 변환 엑시트는 [데이터 변환 엑시트 작성](#)에 설명되어 있습니다.

클러스터 워크로드 엑시트

이 엑시트가 수행하는 기능은 엑시트의 제공자에 의해 정의됩니다. 호출 정의 정보는 [MQ CLUSTER WORKLOAD EXIT - 호출 설명](#)에 제공되어 있습니다.

API 엑시트

API 엑시트를 통해 사용자는 MQPUT 및 MQGET과 같이 WebSphere MQ API 호출의 작동을 변경하는 코드를 작성하고 해당 코드를 이러한 호출 바로 전 또는 후에 삽입할 수 있습니다. 삽입은 자동이며 큐 관리자는 등록된 지점에서 엑시트 코드를 드라이브합니다. API 엑시트에 대한 자세한 정보는 [API 엑시트 사용 및 작성](#)을 참조하십시오.

설치 가능 서비스

설치 가능 서비스가 다중 시작점이 있는 인터페이스(API)를 형식화했습니다.

설치 가능 서비스의 구현을 서비스 컴포넌트라고 합니다. WebSphere MQ에서 제공된 컴포넌트를 사용하거나 고유 컴포넌트를 작성하여 필요한 기능을 수행할 수 있습니다.

현재 다음 설치 가능 서비스가 제공됩니다.

권한 서비스

권한 서비스를 통해 고유 보안 기능을 빌드할 수 있습니다.

서비스를 구현하는 기본 서비스 컴포넌트는 오브젝트 권한 관리자(OAM)입니다. 기본적으로 OAM은 활성이며 OAM 구성을 위해 어떤 조치도 실행할 필요가 없습니다. 권한 서비스 인터페이스를 사용하여 OAM을 바꾸거나 보강시킬 다른 컴포넌트를 작성하십시오. OAM에 대한 자세한 정보는 [Windows, UNIX and Linux 시스템에서 보안 설정](#)을 참조하십시오.

이름 서비스

이름 서비스를 사용하면 애플리케이션이 리모트 큐를 로컬 큐인 것처럼 식별하여 큐를 공유할 수 있습니다.

고유의 이름 서비스 컴포넌트를 작성할 수 있습니다. 예를 들어 IBM WebSphere MQ에 대해 이름 서비스를 사용하려는 경우 이러한 작업을 수행하려고 할 수 있습니다. 이름 서비스를 사용하려면 사용자 작성 또는 다른 소프트웨어 벤더가 공급한 컴포넌트가 있어야 합니다. 기본적으로, 이름 서비스는 비활성 상태입니다.

HP Integrity NonStop Server 용 IBM WebSphere MQ 클라이언트 기술 개요

HP Integrity NonStop Server 운영 체제의 기술 개요

HP Integrity NonStop Server SupportPac 에 대한 IBM WebSphere MQ 클라이언트

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트가 SupportPac MAT1에 릴리스됩니다.

HP Integrity NonStop Server 운영 체제의 기술 개요

HP Integrity NonStop Server는 여러 하드웨어 또는 소프트웨어 장애에도 계획되거나 계획되지 않은 작동중단 시간이 발생하지 않는, 가능한 한 최고의 가용성을 위해 디자인된 운영 체제입니다. 이는 선형으로 확장 가능하며, 20퍼센트의 하드웨어를 추가하면 20퍼센트의 추가 사용 가능 성능을 얻을 수 있습니다. 데이터 무결성을 유지하기 위해, 이 운영 체제는 자체 트랜잭션 관리자 및 트랜잭션 파일 시스템을 가지고 있습니다.

HP Integrity NonStop Server 운영 체제는 일반적으로 다음 항목에서 사용됩니다.

- 금융 기관에서 ATM 네트워크, 온라인 은행 업무 지원, 신용 인증, 증권 거래 전환, 거래, 그리고 은행 간 거래 등에 사용됩니다.
- 제조업에서 웹 상점 백엔드, 재고 및 프로세스 제어 등에 사용됩니다.
- 통신에서 교환, 응급 및 기타 네트워크 서비스 등에 사용됩니다.

지원되는 HP Integrity NonStop Server 환경 및 기능에 대한 IBM WebSphere MQ 클라이언트

Provides details about the IBM WebSphere MQ client for the HP Integrity NonStop Server platform and describes supported client API and environments and client functionality specific to HP Integrity NonStop Server systems.

지원되는 클라이언트 API 및 환경

HP Integrity NonStop Server 의 IBM WebSphere MQ 클라이언트는 다음 실행 환경을 지원합니다.

표 10.		
	OSS	보호자
C	✓	✓
JMS	✓	
COBOL	✓	✓
pTAL	✓	✓

기능 요약

클라이언트 기능의 일부 측면은 호스트 운영 체제에 특정합니다. 다음 요약은 HP Integrity NonStop Server용 IBM WebSphere MQ 클라이언트에 특정한 클라이언트 기능의 측면을 설명합니다.

- C(고유), PTAL, COBOL(고유)
 - 네트워크 프로토콜: TCP(IPv4 및 IPv6)
 - 전송 유형: 클라이언트 한정
 - 전송 보안: SSL/TLS
 - 트랜잭션 지원: TMF(Transaction Management Facility)에 의해 통합된 2단계 커밋(IBM WebSphere MQ Version 7.1 이상의 큐 관리자에 대한 연결 필요)
 - 주소 지정 모드: 32비트
- JMS(Java Message Service)
 - 네트워크 프로토콜: TCP(IPv4 및 IPv6)
 - 전송 유형: 클라이언트 한정(바인딩, 다이렉트 및 다이렉트 HTTP는 지원되지 않음)
 - 전송 보안: SSL/TLS
 - 트랜잭션 지원: 1단계 커밋
 - 실행: 독립형(ASF(Application Support Facility) 및 JCA(Java Connector Architecture)는 지원되지 않음)

- 엑시트: Java 언어 한정(다른 언어로 작성된 고유 엑시트는 지원되지 않음)
- IBM WebSphere MQ 헤더 및 PCF: 다음 클래스는 지원되지 않습니다. com.ibm.mq.headers.* 및 com.ibm.mq.pcf.*

시나리오

이 절에서는 새 WebSphere MQ 버전 7.5 기능을 사용하고 결합하는 방법을 설명하는 시나리오의 정보를 제공합니다. 시나리오에는 사용자가 관심이 있는 영역을 더 잘 이해할 수 있도록 도움을 주는 InfoCenter 콘텐츠에 대한 유용한 링크가 포함되어 있습니다.

사용 가능한 시나리오는 다음 하위 주제에 설명되어 있습니다.

IBM WebSphere MQ Version 7.5 시작하기

이 시나리오에서는 Windows 플랫폼에서 IBM WebSphere MQ Version 7.5 를 시작하는 방법에 대해 설명합니다. IBM WebSphere MQ를 사용해 본 적이 없으며 빨리 시작하고 싶으면 이 시나리오를 사용하십시오.

이 시나리오에는 다음 절이 포함되어 있습니다. 이 시나리오에 나타난 바와 같이 그래픽 또는 명령행 인터페이스를 사용하여 이 단계를 완료할 수 있습니다.

기본 개념 및 핵심 용어

IBM WebSphere MQ Version 7.5 시나리오를 시작하기 전에 알아야 하는 기본 개념 및 핵심 용어에 대해 설명합니다.

기본 개념

IBM WebSphere MQ를 사용하면 큐에 대해 메시지를 읽고 쓸 수 있습니다. 메시지를 읽는 애플리케이션은 메시지를 쓰는 애플리케이션과 독립적입니다. 반드시 두 애플리케이션이 동일한 시간에 실행되어야 하는 것은 아닙니다. 메시지를 읽을 수 있는 애플리케이션을 사용할 수 없는 경우에는 애플리케이션이 읽을 수 있을 때까지 IBM WebSphere MQ 큐에 메시지가 대기됩니다.

핵심 용어

다음은 메시지 큐잉에 대한 핵심 용어 목록입니다.
메시지 큐잉에 대한 핵심 용어

용어	설명
큐 관리자	큐 관리자는 소유한 큐를 유지보수하고 적절한 큐로 수신한 모든 메시지 저장을 담당합니다.
메시지	메시지는 바이트 문자열이며 문자열을 사용하는 애플리케이션에 의미가 있습니다. 메시지는 하나의 애플리케이션 프로그램에서 다른 애플리케이션 프로그램으로 정보를 전송하는 데 사용됩니다. 애플리케이션은 동일한 컴퓨터 또는 다른 컴퓨터에서 실행될 수 있습니다.
로컬 큐	로컬 큐는 메시지를 저장하는 데 사용되는 데이터 구조입니다. 큐는 일반 큐 또는 전송 큐가 있습니다. 일반 큐는 큐 관리자로부터 직접 메시지를 읽는 애플리케이션이 읽을 메시지를 보관합니다. 전송 큐는 다른 큐 관리자로 전송되는 중인 메시지를 보관합니다.
리모트 큐	리모트 큐는 다른 큐 관리자로 메시지를 보내는 데 사용됩니다.
채널	채널은 큐 관리자 사이에서 메시지를 전송하고 수신하는 데 사용됩니다.
리스너	리스너는 기타 큐 관리자 또는 클라이언트 애플리케이션의 네트워크 요청을 승인하고 연관된 채널을 시작하는 프로세스입니다.

QM1 큐 관리자 작성

명령행 인터페이스 또는 WebSphere MQ 탐색기를 사용하여 IBM WebSphere MQ Version 7.5 로 시작하기 시나리오와 함께 사용할 큐 관리자 (QM1) 를 작성하십시오. 큐 관리자는 WebSphere MQ 메시징 네트워크의 기본 컴포넌트입니다.

시작하기 전에

IBM WebSphere MQ Version 7.5가 설치되어 있어야 합니다. 아직 설치하지 않은 경우, 설치 방법에 대한 정보는 [설치 및 설치 제거](#)를 참조하십시오.

이 태스크 정보

이 예제에서는 모든 이름이 소문자로 입력되어 있으며 IBM WebSphere MQ 이름은 대소문자를 구분하므로 모든 이름을 대문자로 입력해야 합니다.

명령행 인터페이스를 사용한 큐 관리자 작성

명령행 인터페이스를 사용하여 큐 관리자를 작성하고 시작하려면 다음 단계를 완료하십시오.

프로시저

1. 다음 명령을 입력하여 이름이 QM1인 큐 관리자를 작성하십시오.

```
crtmqm QM1
```

시스템이 큐 관리자를 작성하면 다음 출력이 표시됩니다.

```
C:\>crtmqm QM1
WebSphere MQ queue manager created.
Creating or replacing default objects for QM1.
Default objects statistics : 61 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

큐 관리자가 작성되고 중지됩니다. 큐 관리자를 관리하고 해당 큐에서 메시지를 읽고 쓰려면 먼저 큐 관리자를 시작해야 합니다.

2. 다음 명령을 입력하여 큐 관리자를 시작하십시오.

```
strmqm QM1
```

큐 관리자가 정상적으로 시작되면 다음 출력이 표시됩니다.

```
C:\>strmqm QM1
WebSphere MQ queue manager 'QM1' starting.
5 log records accessed on queue manager 'QM1' during the log replay phase.
Log replay for queue manager 'QM1' complete.
Transaction manager state recovered for queue manager 'QM1'.
WebSphere MQ queue manager 'QM1' started.
```

큐 관리자가 시작됩니다.

다음에 수행할 작업

큐를 작성하려면 [130 페이지의 『LQ1 큐 작성』](#)의 내용을 참조하십시오.

WebSphere MQ Explorer를 사용한 큐 관리자 작성

WebSphere MQ Explorer를 사용하여 큐 관리자를 작성하고 시작하려면 다음 단계를 완료하십시오.

프로시저

1. WebSphere MQ Explorer를 시작하십시오.

2. **Navigator** 보기에서 **큐 관리자** 폴더를 마우스 오른쪽 단추로 누른 다음 **새로 작성 > 큐 관리자**를 누르십시오. **큐 관리자 작성** 마법사가 시작됩니다.
3. **큐 관리자 이름** 필드에 QM1을 입력하십시오.
4. Make this the default queue manager 선택란을 선택하십시오.
5. **데드-레터 큐** 필드에 SYSTEM.DEAD.LETTER.QUEUE를 입력하십시오.
이 이름은 큐 관리자를 작성할 때 자동 작성되는 데드-레터 큐의 이름입니다.
6. 다른 필드를 비워두고 **마침**을 클릭하거나, 해당 단추를 사용할 수 없으면 **다음**을 클릭하십시오.
포트 번호가 기존 큐 관리자(예: 기본 구성의 일부로 작성된 큐 관리자)와 충돌하면 **마침** 단추를 사용할 수 없습니다. 기본 포트 번호를 변경하려면 마법사를 계속 진행해야 합니다.
7. **다음**을 클릭한 경우, 계속해서 기본값을 승인하고 마법사의 마지막 페이지에 도달할 때까지 각 페이지에서 **다음**을 클릭하십시오. 마지막 페이지에서는 **마침** 단추를 사용할 수 있습니다. 지정된 포트 번호를 변경하고(예: 1415로), **마침**을 클릭하십시오.

큐 관리자가 작성되고 시작되는 동안 WebSphere MQ는 **큐 관리자 작성** 대화 상자 창을 표시합니다.

다음에 수행할 작업

큐를 작성하려면 130 페이지의 『[LQ1 큐 작성](#)』의 내용을 참조하십시오.

LQ1 큐 작성

명령행 인터페이스 또는 WebSphere MQ Explorer를 사용하여 WebSphere IBM WebSphere MQ Version 7.5 시작하기 시나리오에서 사용할 큐를 작성합니다. 큐는 메시지 저장에 사용되는 데이터 구조이며 IBM WebSphere MQ 큐 관리자 오브젝트입니다.

이 태스크 정보

IBM WebSphere MQ 오브젝트를 작성하는 데에는 다음 세 가지 방법이 있습니다.

- 명령행
- IBM WebSphere MQ Explorer.
- 프로그래밍 가능한 인터페이스 사용

이 태스크에서 명령행 또는 IBM WebSphere MQ Explorer를 사용하여 IBM WebSphere MQ 오브젝트를 작성할 수 있습니다.

명령행 인터페이스를 사용한 큐 작성

명령행 인터페이스에는 MQSC(IBM WebSphere MQ 스크립트 명령)라는 스크립트 언어가 있습니다. 스크립트 도구 **runmqsc**는 큐 관리자에 대해 스크립트를 실행하는 데 사용됩니다. 명령행 인터페이스를 사용하여 큐를 작성하고 시작하려면 다음 단계를 완료하십시오.

프로시저

1. 다음 명령을 입력하여 스크립트 도구를 시작하십시오.

```
runmqsc QM1
```

스크립팅 도구가 시작되면 다음 출력이 표시됩니다.

```
C:\>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
```

도구가 MQSC 명령을 수락할 준비가 되었습니다.

2. 다음 MQSC 명령을 입력하여 LQ1(이)라는 로컬 큐를 작성하십시오.

```
define qlocal(LQ1)
```

큐가 작성되면 다음 출력이 표시됩니다.

```
define qlocal(LQ1)
  2 : define qlocal(LQ1)
AMQ8006: WebSphere MQ queue created.
```

3. 다음 MQSC 명령을 입력하여 스크립트 도구를 중지하십시오.

```
end
```

스크립트 도구가 종료되면 다음 출력이 표시됩니다.

```
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.

C:\>
```

다음에 수행할 작업

큐에 메시지를 넣을 준비가 되었습니다. 큐에 메시지를 넣으려면 [131 페이지의 『큐 LQ1에 메시지 넣기』](#)의 내용을 참조하십시오.

WebSphere MQ Explorer를 사용하여 큐 작성

WebSphere MQ Explorer를 사용하여 큐를 작성하고 시작하려면 다음 단계를 완료하십시오.

프로시저

1. 네비게이터 보기에서 큐 관리자 폴더를 펼치십시오.
2. 큐 관리자 **QM1**을 펼치십시오.
3. 큐 폴더를 마우스 오른쪽 단추로 클릭한 후 **새로 작성 > 로컬 큐 ...** 를 클릭하십시오. **새 로컬 큐** 마법사가 시작됩니다.
4. 이름 필드에 LQ1을 입력하십시오.
5. 완료를 누르십시오.

새 LQ1 큐가 **컨텐츠** 보기에 표시됩니다. **컨텐츠** 보기에 큐가 표시되지 않으면, **컨텐츠** 보기의 맨 위에서 **새로 고치기** 단추를 클릭하십시오.

다음에 수행할 작업

큐에 메시지를 넣을 준비가 되었습니다. 큐에 메시지를 넣으려면 [131 페이지의 『큐 LQ1에 메시지 넣기』](#)의 내용을 참조하십시오.

큐 LQ1에 메시지 넣기

명령행 인터페이스 또는 IBM WebSphere MQ Explorer를 사용하여 IBM WebSphere MQ Version 7.5 시나리오로 시작하기 시나리오와 함께 사용할 수 있도록 LQ1큐에 메시지를 넣으십시오.

이 태스크 정보

IBM WebSphere MQ는 amqsput이라는 샘플 애플리케이션과 함께 제공됩니다. 이 애플리케이션은 메시지를 사전정의된 큐에 넣습니다.

명령행 인터페이스를 사용하여 큐에 메시지 넣기

명령행 인터페이스를 사용하여 큐에 메시지를 넣으려면 다음 단계를 완료하십시오.

프로시저

1. amqsput 샘플 애플리케이션을 사용하여 다음 명령을 입력하여 메시지를 LQ1 큐에 넣으십시오.

```
amqsput LQ1 QM1
```

샘플 애플리케이션이 시작되면 다음 출력이 표시됩니다.

```
C:\>amqsput LQ1 QM1
Sample AMQSPUT0 start
target queue is LQ1
```

2. Hello World를 입력하고 Enter를 누르십시오. QM1이라는 큐 관리자가 관리하는 LQ1 큐에 "Hello World" 텍스트가 포함된 메시지를 넣었습니다.
3. amqsput을 종료하려면 Enter를 누르십시오. 다음 출력이 표시됩니다.

```
C:\>amqsput LQ1 QM1
Sample AMQSPUT0 start
target queue is LQ1
Hello World

Sample AMQSPUT0 end
```

다음에 수행할 작업

큐에서 메시지를 가져오려면 [132 페이지의 『LQ1 큐로부터 메시지 가져오기』](#)의 내용을 참조하십시오.

IBM WebSphere MQ Explorer를 사용하여 큐에 메시지 넣기

IBM WebSphere MQ Explorer를 사용하여 큐에 메시지를 넣으려면 다음 단계를 완료하십시오.

프로시저

1. 네비게이터 보기에서 큐 관리자 폴더를 펼치십시오.
2. 작성된 큐 관리자 QM1을 펼치십시오.
3. 큐 폴더를 클릭하십시오. 큐 관리자의 큐가 콘텐츠 보기에 나열됩니다.
4. 콘텐츠 보기에서 로컬 큐 LQ1을 마우스의 오른쪽 단추로 클릭한 후 **테스트 메시지 넣기..**를 클릭하십시오. **테스트 메시지 넣기** 대화 상자가 열립니다.
5. **메시지 데이터** 필드에서 일부 텍스트(예: Hello World)를 입력하고 **메시지 넣기**를 클릭하십시오. **메시지 데이터** 필드가 지워지고 메시지가 큐에 놓입니다.
6. 닫기를 클릭하십시오.
콘텐츠 보기에서 LQ1 **현재 큐 용량** 값이 이제 1임을 확인하십시오. **현재 큐 용량** 열이 표시되지 않으면 **콘텐츠 보기**의 오른쪽으로 스크롤해야 할 수도 있습니다.

다음에 수행할 작업

큐에서 메시지를 가져오려면 [132 페이지의 『LQ1 큐로부터 메시지 가져오기』](#)의 내용을 참조하십시오.

LQ1 큐로부터 메시지 가져오기

명령행 인터페이스 또는 IBM WebSphere MQ Explorer를 사용하여 IBM WebSphere MQ Version 7.5 시나리오로 시작하기 시나리오와 함께 사용할 수 있도록 LQ1큐에서 메시지를 가져오십시오.

이 태스크 정보

IBM WebSphere MQ는 amqsget이라는 샘플 애플리케이션과 함께 제공됩니다. 이 애플리케이션은 큐로부터 메시지를 읽습니다.

명령행 인터페이스를 사용하여 큐에서 메시지 가져오기

명령행 인터페이스를 사용하여 큐에서 메시지를 가져오려면 다음 단계를 완료하십시오.

프로시저

amqsget 샘플 애플리케이션을 사용하여 다음 명령을 입력하여 LQ1 큐에서 메시지를 읽으십시오.

```
amqsget LQ1 QM1
```

샘플 애플리케이션이 시작되면 다음 출력이 표시됩니다.

```
C:\>amqsget LQ1 QM1
Sample AMQSGETO start
message <Hello World>
no more messages
Sample AMQSGETO end
```

amqsget 애플리케이션은 메시지를 읽은 후 30초 후에 종료됩니다.

다음에 수행할 작업

큐잉 애플리케이션 작성, 큐 관리자에 대한 연결 및 연결 끊기, 발행/구독 및 오브젝트 열고 닫기에 대해 알아보려면 [큐잉 애플리케이션 작성](#)을 참조하십시오.

IBM WebSphere MQ Explorer를 사용하여 큐에서 메시지 가져오기

IBM WebSphere MQ Explorer를 사용하여 큐에서 메시지를 가져오려면 다음 단계를 완료하십시오.

프로시저

1. 네비게이터 보기에서 큐 관리자 폴더를 펼친 후 QM1을 펼치십시오.
2. 큐 폴더를 클릭하십시오.
3. 콘텐츠 보기에서 QM1에 마우스의 오른쪽 단추를 클릭한 후에 **메시지 찾아보기...**를 클릭하십시오. **메시지 브라우저**가 열려서 현재 QM1에 있는 메시지의 목록을 표시합니다.
4. 마지막 메시지를 두 번 클릭하여 특성 대화 상자를 여십시오.

특성 대화 상자의 **데이터** 페이지에서, **메시지 데이터** 필드에 읽을 수 있는 양식의 메시지 콘텐츠가 표시됩니다.

다음에 수행할 작업

큐잉 애플리케이션 작성, 큐 관리자에 대한 연결 및 연결 끊기, 발행/구독 및 오브젝트 열고 닫기에 대해 알아보려면 [큐잉 애플리케이션 작성](#)을 참조하십시오.

다음에 수행할 작업

IBM WebSphere MQ Version 7.5 시나리오를 시작한 후에 수행할 작업에 대해 설명합니다.

IBM WebSphere MQ에서는 특정 WebSphere 제품 오퍼링에 대한 기술을 얻을 수 있는 경로를 정의하여 사용자를 지원하기 위한 역할 기반의 훈련 경로를 제공합니다. IBM WebSphere MQ에 대한 두 개의 훈련 경로가 있습니다.

- 애플리케이션 개발자

이러한 사용자는 큐 관리자를 사용하는 애플리케이션을 작성할 책임이 있습니다. 이 시나리오에서는 amqsput 및 amqsget 애플리케이션을 작성합니다.

- 시스템 관리자

이러한 사용자는 일반적으로 이 시나리오에서 다루는 태스크와 유사한 태스크를 수행할 큐 관리자 및 해당 오브젝트를 작성할 책임이 있습니다.

IBM WebSphere MQ 훈련 경로에 대한 자세한 정보는 <https://www.ibm.com/software/websphere/education/paths/>의 내용을 참조하십시오.

IBM WebSphere MQ 코스의 전체 목록을 보려면 <https://www.ibm.com/software/websphere/education/curriculum/appint/wmq/>의 내용을 참조하십시오.

IBM WebSphere MQ에서 스킬 레벨을 획득하는 방법을 보여주는 특정 프로그램을 사용할 수 있습니다. 자세한 정보는 <https://www.ibm.com/certify/index>의 내용을 참조하십시오.

기타 사용자와 협업할 수 있습니다. 예를 들어, 다음을 참조하십시오.

- IBM WebSphere MQ 전문가의 이메일 중심 커뮤니티의 경우 <https://listserv.meduniwien.ac.at/archives/mqser-l.html>의 내용을 참조하십시오.
- 제품의 IBM WebSphere MQ 제품군 중심 토론 포럼은 <https://www.mqseries.net/>의 내용을 참조하십시오.
- 다양한 IBM 메시징 제품의 개발자가 IBM Developer 블로그를 참조하십시오 (<https://www.ibm.com/developerworks/blogs/page/messaging/>참조).
- IBM WebSphere MQ에 대한 공식 IBM호스트 포럼. <https://www.ibm.com/developerworks/forums/forum.jspa?forumid=280>의 내용을 참조하십시오.
- stackoverflow.com의 IBM WebSphere MQ 태그가 지정된 질문과 응답은 <https://stackoverflow.com/questions/tagged/websphere-mq>의 내용을 참조하십시오.

이 제품 문서에서 볼 수 있는 추가 토픽이 있습니다. 다음 절에서 참조하십시오.

- [관리IBM WebSphere MQ](#)

IBM WebSphere MQ는 사용자가 사용할 수 있는 제어 명령을 제공합니다. 이 시나리오에서는 **crtmqm** 및 **strmqm**이라는 두 명령을 사용할 수 있습니다. 또한 이 절에서는 메시지 큐잉에 대한 개요를 제공합니다.

- [MQSC 참조서](#)

이 시나리오에서는 `define qlocal('LQ1')` 명령을 사용하여 LQ1이라고 하는 로컬 큐를 정의합니다. 이 명령은 MQSC 명령입니다. IBM WebSphere MQ 시스템 관리자는 이러한 명령을 사용하여 큐 관리자를 관리합니다. 이 절에서는 명령을 알파벳순으로 상세히 설명하기 전에 명령을 소개하고 사용법을 보여줍니다.

- [큐 관리자 클러스터 구성](#)

이 절에서는 클러스터로 알려진 가상 그룹에서 큐 관리자를 조직, 사용 및 관리하는 방법에 대해 설명합니다. 클러스터를 수행하면 클러스터 내의 각 큐 관리자가 동일한 클러스터 내의 모든 기타 큐 관리자에 대해 알 수 있습니다. 또한 클러스터로 인해 복잡한 큐 관리자 네트워크를 더 간단히 관리할 수 있습니다.

제품 연결 시나리오 제품 문서에서는 다양한 시나리오에서 WebSphere Application Server를 WebSphere MQ에 연결하는 데 필요한 주요 태스크를 설명하는 정보를 제공합니다. 개별 시나리오에는 비즈니스 환경에서 솔루션을 구현하는 데 관한 지시사항이 포함되어 있어서 다른 정보 자원을 사용하지 않고도 학습할 수 있습니다. https://www.ibm.com/docs/prodconn_1.0.0/com.ibm.prodconn.doc/infocenter_homepage/ic_home.htm

스크립트를 사용하는 기본 파일 전송

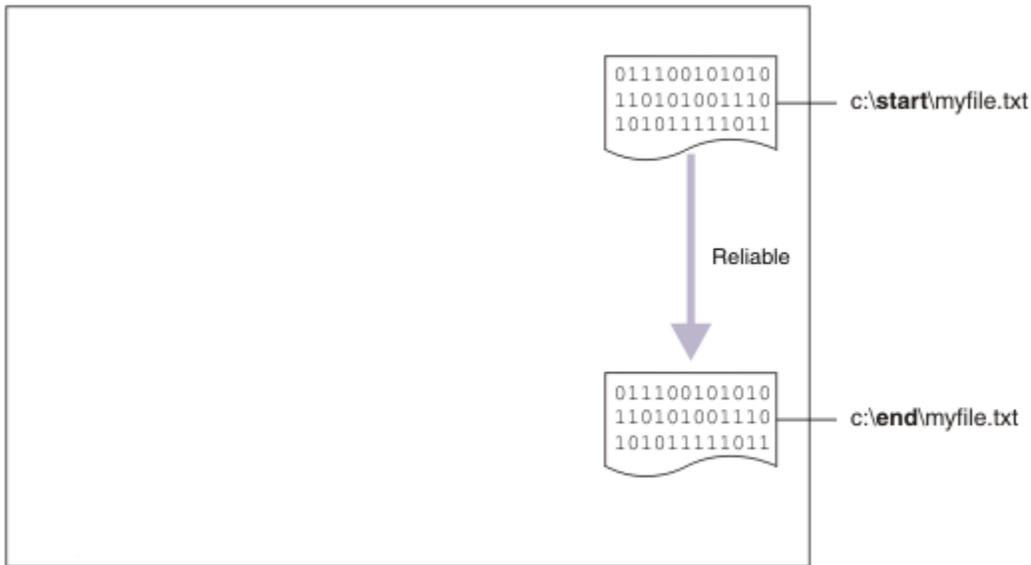
IBM WebSphere MQ Version 7.5를 사용하여 여러 가지 방법으로 파일을 전송할 수 있습니다. 이 시나리오에서 다루는 사항, 비즈니스가 시나리오를 따를 수 있는 이유, 포함된 사용자 역할, 시나리오에서 제안하는 솔루션의 개요를 이해하려면 이 절의 주제를 검토하십시오.

파일을 감사 가능한 방법으로 안전하게 제어하면서 전송하는 것은 엔터프라이즈의 기본적인 요구사항입니다. IBM WebSphere MQ Version 7.5는 통합 메시징 플랫폼의 일부로 관리 파일 전송 기능을 제공합니다. 기본 파일 전송을 통하거나 메시징에 본격적으로 참여하여 관리 파일 전송 기능을 사용하여 파일을 메시징 인프라와 유기적으로 통합할 수 있습니다. 이 기능에 대한 자세한 정보는 [WebSphere MQ Managed File Transfer 소개](#)를 참조하십시오.

이 시나리오는 가장 단순한 IBM WebSphere MQ 메시징 기술로 파일을 통합하는 기본적인 방법에 대한 이해를 돕습니다. 이를 위해 기본 IBM WebSphere MQ 시나리오 작업을 통해 파일을 한 위치에서 다른 위치로 이동하도록 디자인되었습니다. 이 초기 시나리오는 단일 컴퓨터로 제한되나 환경을 구성하고 이후 시나리오에 대한 중요한 기초를 구성하는 경험을 제공합니다. 이 시나리오는 나중에 네트워크를 통해 파일을 전송하기 위해 IBM WebSphere MQ를 사용하는 방법을 보여주며 관리 파일 전송 컴포넌트가 실제 비즈니스 문제점을 처리하는 방법을 보여줍니다.

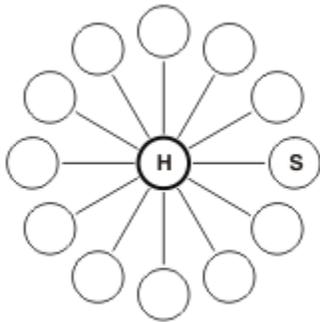
IBM WebSphere MQ, 특히 **runmqsc** 및 IBM WebSphere MQ 탐색기와 같은 명령을 사용하여 IBM WebSphere MQ의 기본 구성 및 관리 개념과 큐 관리자의 개념을 기본적으로 이해해야 합니다.

이 시나리오에서는 단일 컴퓨터의 한 위치에서 다른 위치로 파일 전송을 시작하고 추적하는 데 IBM WebSphere MQ를 사용하는 방법을 탐색하며 IBM WebSphere MQ Version 7.5에서 신뢰할 수 있는 관리 파일 전송 기능을 직접 설치, 구성 및 사용해 봅니다.



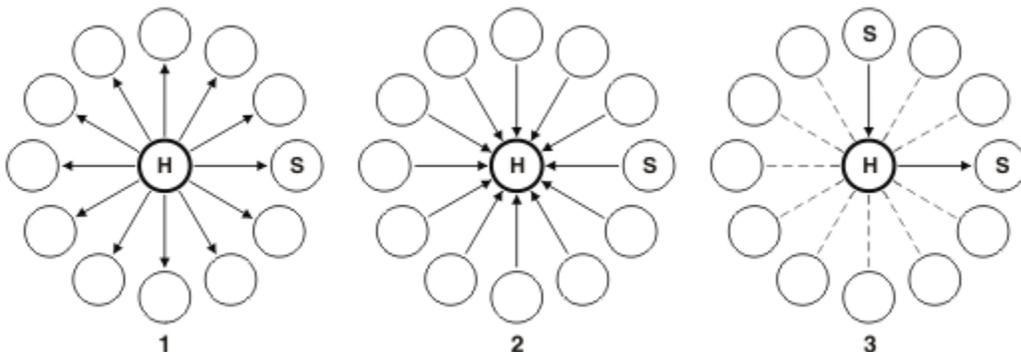
예제 파일 전송 토폴로지

이 시나리오 및 후속 시나리오는 허브 및 스포크 토폴로지를 기반으로 합니다. 다이어그램에서는 허브(H) 및 다중 스포크(S)를 구성하는 개념적인 허브 및 스포크 토폴로지를 표시합니다.



해당 토폴로지는 다양한 비즈니스 시나리오에서 애플리케이션을 찾습니다. 예를 들어, 다음과 같습니다.

1. 중앙 HQ(허브)로부터 많은 기본 노드(스포크)로 파일을 보냅니다.
2. 많은 노드(스포크)로부터 단일 위치(허브)로 파일을 조합합니다.
3. 한 스포크로부터 다른 스포크로 파일을 전송합니다.



토폴로지가 이 예와 다를 수 있습니다. 그러나 이 토폴로지 내의 원칙 및 명령은 어떠한 IBM WebSphere MQ 네트워크에도 적용할 수 있도록 확장될 준비가 되어 있습니다.

솔루션 계획

스크립트를 사용하여 컴퓨터에서 또는 컴퓨터로의 파일 전송을 계획합니다. 보안 모델, 필수조건 및 라이선스, IBM WebSphere MQ 설치 및 구성, 사용자 및 그룹 준비에 대한 이해가 포함됩니다.

파일을 A 위치에서 B 위치로 전송하려면 단일 컴퓨터에서 허브 및 스포크 토폴로지를 작성하십시오. 이 토폴로지는 허브, IBM WebSphere MQ 큐 관리자, 각 파일 전송 에이전트인 두 스포크를 포함합니다.

파일 전송 에이전트는 컴퓨터에서 실행되어 기타 에이전트에 대해 파일을 전송하는 Java 프로세스입니다. 이 시나리오에서는 두 메커니즘을 통해 샘플 파일을 한 위치에서 다른 위치로 이동하기 위해 이러한 파일 전송 에이전트를 사용하는 파일 전송을 정의합니다.

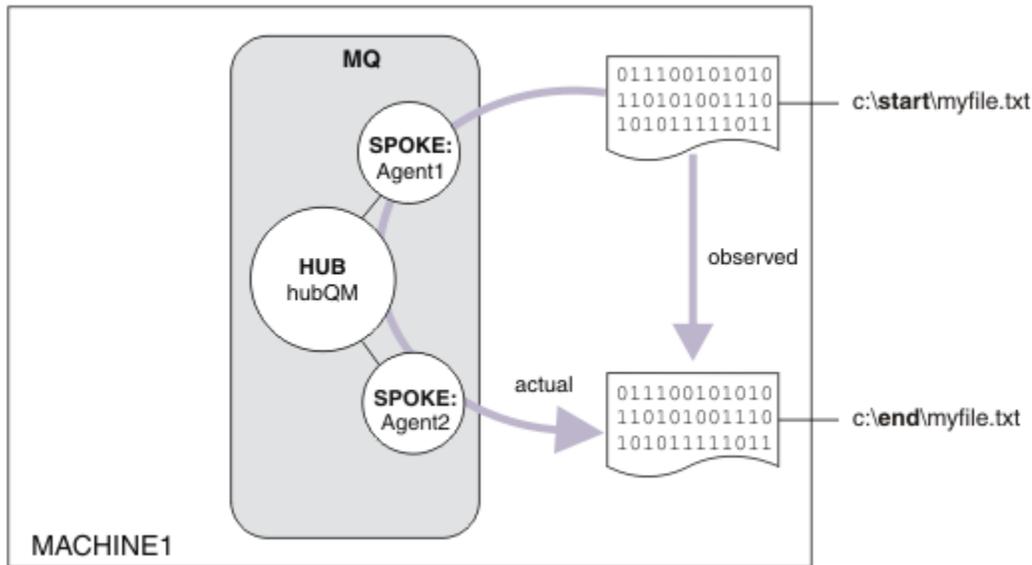
1. 그래픽 IBM WebSphere MQ Explorer 사용
2. 명령행 사용

시나리오에서는 Windows 시스템이 있다고 가정합니다. UNIX 시스템의 경우, 적절한 경로 및 명령으로 대체하십시오. 또한 모든 관련 디렉토리에 대해 읽기 및 쓰기 액세스 권한이 있어야 합니다. 또한 이 시나리오에서는 큐 관리자가 수행하는 작업에 대해 기본적으로 이해하고 있다고 가정합니다.

파일을 컴퓨터에 대해 전송하려면 파일 전송 에이전트가 해당 컴퓨터에서 실행 중이어야 합니다. 모든 에이전트가 IBM WebSphere MQ 큐 관리자에 연결되고 IBM WebSphere MQ를 사용하여 기타 에이전트와 통신합니다. 자세한 정보는 [WebSphere MQ Managed File Transfer 토폴로지 개요](#)를 참조하십시오.

IBM WebSphere MQ Version 7.5에는 파일 전송 토폴로지를 빠르고 쉽게 빌드하는 데 도움이 되는 두 개의 샘플 createHub 및 addSpoke 스크립트가 있습니다. 이러한 스크립트를 사용하여 허브로부터 바깥으로 이 토폴로지를 구성하십시오. 스크립트는 샘플이며 사용자의 요구사항에 맞게 수정할 수 있습니다. 스크립트를 수정하려면 먼저 사용자 자신의 디렉토리 내의 위치로 스크립트를 복사하십시오.

샘플 스크립트 createHub 및 addSpoke는 토폴로지를 구성하는 데 사용됩니다.



전송이 시작되면 파일이 한 위치에서 다른 위치로 복사되는 것을 관찰하십시오. 이 단순 시나리오에는 소스 및 대상 파일 둘 다를 관리하는 단일 파일 전송 에이전트를 사용하여 실행할 수 있습니다. 그러나 두 파일 전송 에이전트는 다중 컴퓨터 토폴로지로 이동하기 전에 이 설정 작업을 수행하는 방법에 대한 더 나은 예를 제공하기 위해서 사용됩니다.

보안 모델 이해

createHub 및 addSpoke 스크립트는 다음과 같은 보안 특성이 있는 파일 전송 토폴로지를 구성합니다.

- IBM WebSphere MQ에 대한 액세스 권한은 세 역할 사이에서 나뉩니다.
 1. IBM WebSphere MQ를 구성하고 createHub 및 addSpoke 스크립트를 실행하는 IBM WebSphere MQ 관리자.

2. 에이전트 시작, 중지 및 상호작용을 수행하는 FTAGENTS 사용자

3. 파일 전송을 시작한 FTUSERS

액세스 권한 구성에 대한 자세한 정보는 [WebSphere MQ Managed File Transfer에 해당되는 자원에 대한 권한](#)을 참조하십시오.

- 에이전트로부터 수신되는 모든 연결은 허브에서 단일 사용자 FTAGENT에 맵핑됩니다.
- 약한 IP 기반 인증은 개별 에이전트를 인증하는 데 사용됩니다.

명확한 설명을 위해 스크립트 내의 보안 모델은 완전히 강화되지 않습니다. 프로덕션에서 이러한 스크립트를 사용하기 전에 자체 토폴로지 요구 및 보안 위협을 이해해야 합니다. 따라서 다음과 같은 잠재적인 취약성을 처리할 것인지 여부 및 권장사항을 고려해야 합니다.

- 모든 사용자가 다른 사용자를 위장할 수 있습니다. 파일 전송 자원에 대해 오브젝트 액세스 모델에서 더 세밀한 세분성을 고려하십시오.
- 모든 에이전트가 다른 에이전트를 위장할 수 있습니다. 더 강한 인증을 고려하십시오(예: 파일 전송 자원에 대해 오브젝트 액세스 모델에서 TLS/SSL 및 더 세밀한 세분성).
- 파일 시스템 대 IBM WebSphere MQ 보안 사이의 인터페이스는 고려되지 않습니다. 파일 샌드박스를 구현하는 것을 고려하고 에이전트 구성 파일 권한의 영향에 대해 이해하십시오. 샌드박스에 대한 자세한 정보는 [Sandboxes](#)의 내용을 참조하십시오.
- 에이전트와 운영 체제 간 인터페이스에 대해서는 설명하지 않습니다. 파일 샌드박스 구현을 고려하십시오. 자세한 정보는 [샌드박스를 참조하십시오](#).

보안, IBM WebSphere MQ, 파일 전송에 대한 자세한 정보는 [다음에 수행할 작업을 참조하십시오](#).

필수조건 및 라이선스

다음 항목이 필요합니다.

- IBM WebSphere MQ의 하드웨어 및 운영 체제 전제조건을 충족시키는 테스트 컴퓨터는 IBM WebSphere MQ 또는 IBM WebSphere MQ 데이터의 기존 설치가 없는 <https://www.ibm.com/support/docview.wss?uid=swg27006467>를 참조하십시오.
- IBM WebSphere MQ Version 7.5. <https://www.ibm.com/developerworks/downloads/ws/wmq/>에서 평가판을 다운로드할 수 있습니다.

컴퓨터 준비

테스트 컴퓨터가 IBM WebSphere MQ Version 7.5의 설치 요구사항을 충족하는지 확인하십시오. [요구사항 확인](#)을 참조하십시오.

컴퓨터가 설치에 적합하게 준비되었는지 확인하십시오. [명령행을 사용하여 로컬 설치 확인](#)을 참조하십시오.

IBM WebSphere MQ Version 7.5 설치

서버, IBM WebSphere MQ 탐색기, IBM WebSphere MQ 고급 관리 파일 전송 에이전트 및 IBM WebSphere MQ 관리 파일 전송 명령행 도구 구성요소와 함께 IBM WebSphere MQ Version 7.5 서버를 설치하십시오. 자세한 내용은 [설치 항목 선택](#)을 참조하십시오.

IBM WebSphere MQ 관리 방법을 결정하십시오. 다음 방법으로 IBM WebSphere MQ를 관리할 수 있습니다.

- **setmqenv** 명령을 사용하여 적절한 환경 설정. 자세한 정보는 [setmqenv](#)를 참조하십시오.
- 완전한 IBM WebSphere MQ 관리 명령 호출.

이 시나리오에서는 IBM WebSphere MQ 또는 IBM WebSphere MQ File Transfer Edition이 아직 설치된 적이 없는 컴퓨터를 사용 중이라고 가정합니다. 그렇지 않은 경우에는 공존이 지원되는지 판별하거나 환경의 설치 메커니즘 및 구성을 조정해야 합니다. 자세한 내용은 [다중 설치를 참조하십시오](#).

사용자 및 그룹 작성

이 보안 모델은 다음 사용자 및 그룹을 작성했다고 가정합니다.

사용자

- mqmAdmin

IBM WebSphere MQ 관리자, 즉 mqm 그룹의 구성원 또는 에서 그룹의 구성원입니다. 이 사용자를 작성하고 mqm 그룹의 구성원으로 만들어야 합니다.

- ftuser

이 사용자는 반드시 작성해야 하며 FTUSERS 그룹의 구성원이어야 합니다. 관리 레벨 보안에 대한 조치가 큐 관리자에서 수행되는 가능성을 피하기 위해 이 사용자를 mqm 그룹에 추가하지 마십시오.

- ftagent

이 사용자는 반드시 작성해야 하며 FTAGENTS 그룹의 구성원이어야 합니다. 관리 레벨 보안에 대한 조치가 큐 관리자에서 수행되는 가능성을 피하기 위해 이 사용자를 mqm 그룹에 추가하지 마십시오.

그룹

- mqm

IBM WebSphere MQ 설치의 일부로 자동으로 작성됩니다. 이 그룹의 구성원은 IBM WebSphere MQ 및 해당 자원을 관리할 수 있습니다.

- FTUSERS

이 그룹은 반드시 작성해야 합니다. 이 그룹의 구성원은 파일 전송을 시작할 수 있습니다.

- FTAGENTS

이 그룹은 반드시 작성해야 합니다. 이 그룹의 구성원은 네트워크 내에서 파일의 전송을 핸들링하는 엔드포인트인 파일 전송을 시작하고 중지할 수 있습니다.

파일 전송용 IBM WebSphere MQ 구성

샘플 스크립트 createHub 및 addSpoke 를 사용하여 파일 전송에 대해 IBM WebSphere MQ 를 구성하여 스크립트 시나리오를 사용하여 기본 파일 전송에 대한 토폴로지를 빌드하십시오.

프로시저

1. (이후 시나리오에서 식별되는) 리모트 에이전트가 연결하는 데 사용할 수 있는 빈 포트를 식별하십시오. 예를 들어, 1414입니다.
2. 파일 전송 네트워크의 허브 역할을 수행하는 큐 관리자에 대해 적당한 이름을 식별하십시오. 예를 들어, hubQM과 같습니다.
3. As the user mqmAdmin, from the IBM WebSphere MQ bin directory, <MQ_INSTALL_ROOT>/bin:

- a) 허브를 작성하십시오.

다음 명령을 입력하십시오.

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\createHub hubQmgr=hubQM hubPort=1414
```

- b) 에이전트 AGENT1로 불리는 첫 번째 에이전트 스포크(큐 관리자에 대해 로컬이므로 BINDINGS 상태임)를 추가하십시오.

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT1  
hubQmgr=hubQM connectionMode=BINDINGS
```

- c) 에이전트 AGENT2로 불리는 두 번째 에이전트 스포크(큐 관리자에 대해 로컬이므로 BINDINGS 상태임)를 추가하십시오.

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT2  
hubQmgr=hubQM connectionMode=BINDINGS
```

4. ftagent 사용자로서

- a) 다음 명령을 입력하여 AGENT1을 시작하십시오.

```
fteStartAgent -p hubQM AGENT1
```

둘 이상의 허브를 구성한 경우, 명령의 -p hubQM 부분으로 인해 올바른 허브를 선택할 수 있습니다. 단일 허브인 경우 이 부분이 필요하지 않습니다.

- b) 다음 명령을 입력하여 AGENT2를 시작하십시오.

```
fteStartAgent -p hubQM AGENT2
```

- c) 에이전트가 올바르게 실행 중인지 확인하기 위해 에이전트를 나열하십시오.

```
fteListAgents -p hubQM
```

다음 출력이 표시됩니다.

Agent Name:	Queue Manager Name:	Status:
AGENT1	hubQM	READY
AGENT2	hubQM	READY

솔루션 구현

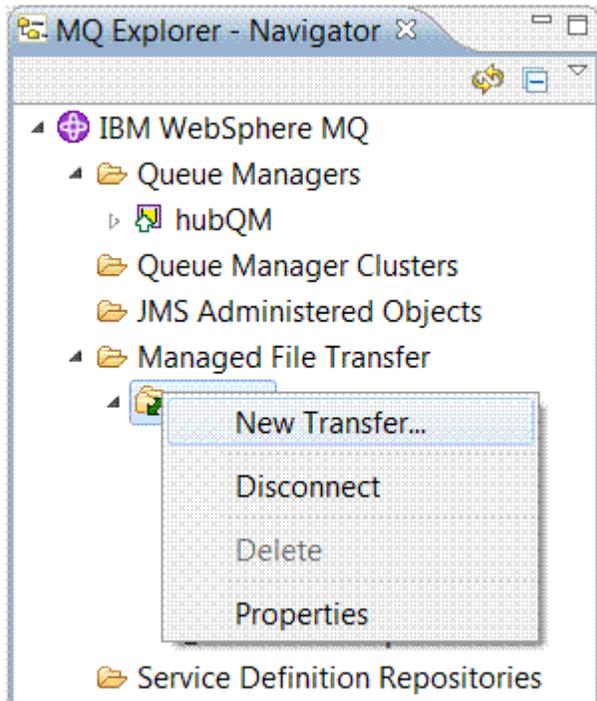
이 시나리오에서 솔루션을 구현하는 것은 IBM WebSphere MQ Explorer를 사용하여 파일 전송을 정의하고 시작하는 것과 관련됩니다. 전송 진행 과정을 모니터링하고 파일 시스템을 조사하여 성공했는지 확인하십시오.

이 태스크 정보

IBM WebSphere MQ비관리 사용자 `ftuser`에 대해 탐색기를 구성하지 않도록 하려면 이 시나리오 부분을 사용자 `mqmAdmin`으로 실행하십시오. 이 기능을 사용하려면 [보안](#) 을 참조하십시오.

프로시저

- 소스 파일 및 대상 디렉토리를 식별하십시오.
 - 전송할 샘플 파일을 작성하십시오 (예: `C:\start\myfile.txt`).
 - 이 파일을 전송할 기존 디렉토리를 식별하십시오. 예를 들어, `C:\end\`입니다.
- 이 예에서 관리자 `mqmAdmin`으로 IBM WebSphere MQ Explorer를 시작하십시오. 시작 메뉴 (또는 동등) 에서 프로그램을 시작하거나 **MQExplorer** 명령을 실행하십시오. 자세한 정보는 [WebSphere MQ Explorer 실행](#) 을 참조하십시오.
- 관리 파일 전송** 절에서 이름이 `hubQM`인 구성을 마우스의 오른쪽 단추로 클릭하고 **Connect**를 선택하십시오.
- IBM WebSphere MQ 네비게이터의 **관리 파일 전송** 섹션에서 `hubQM` 을 마우스 오른쪽 단추로 클릭하고 **새 전송** 을 선택하여 새 전송 마법사를 시작하십시오.



5. 메뉴를 사용하여 소스 에이전트로 AGENT1을 선택하고 대상 에이전트로 AGENT2를 선택하십시오.

Source agent
 Name: AGENT1
 Type: Standard

Destination agent
 Name: AGENT2
 Type: Standard

6. 다음 을 클릭하십시오.

New transfer

i Click 'Add' to begin adding items to the transfer

Source	Destination	Mode	Disposition	Overwrite

Add...
 Copy...
 Edit...
 Remove

7. 추가...를 클릭하십시오. AGENT1 에서 AGENT2 로 전송할 파일 선택을 시작합니다.

8. 소스 프레임에서 **찾아보기**를 클릭하여 앞에서 식별한 파일을 찾으십시오. 예를 들어, C:\start\myFile.txt입니다. **전송이 완료되면 소스 파일 제거 선택란**이 체크되어 있는지 확인하십시오.

Source

Agent: AGENT1
Type: Standard agent
Type: File

File name: C:\start\myFile.txt

Remove source file if the transfer is successful

9. 대상 프레임에서 Directory의 유형을 선택한 후 이전에 식별한 대상 디렉토리 (예: C:\end\)를 입력하십시오.

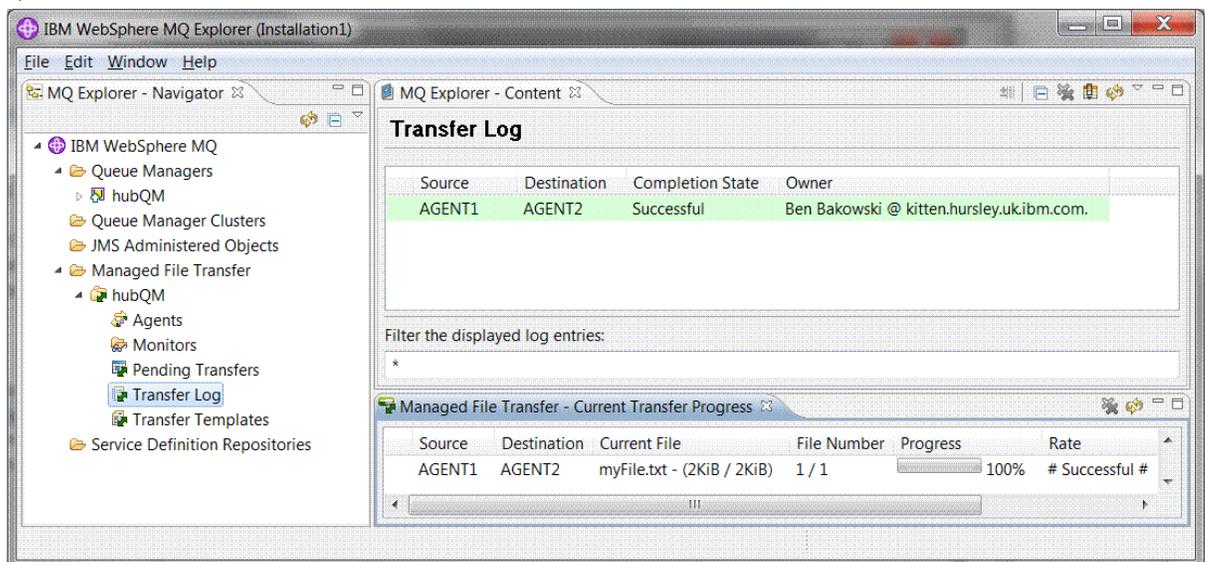
Destination

Agent: AGENT2
Type: Standard agent
Type: Directory

Directory: c:\end

Overwrite files if present

10. **확인**을 클릭한 다음 **마침**을 클릭하면 전송이 시작됩니다.
11. IBM WebSphere MQ Explorer 맨 아래에서 현재 전송 진행 과정 탭의 전송 진행 과정을 모니터링할 수 있습니다.



12. 또한 파일 시스템을 수동으로 검사하여 새 파일이 있는지 확인할 수 있습니다. 예를 들어, C:\end\myfile.txt입니다.

파일을 전송하기 위해 명령행 사용

이 절에서는 전송을 시작하기 위해 명령행을 사용하는 방법부터 시작하여 파일 전송의 유연성에 대해 설명합니다. 이 시나리오의 범위를 벗어나지만 이러한 원칙 및 Ant 스크립트 기술을 기반으로 하여 훨씬 강력한 파일 전송 시나리오를 정의하고 구현할 수 있습니다.

프로시저

1. 원래 위치로 전송한 파일을 리턴합니다.
2. `ftuser` 사용자로서 `fteCreateTransfer` 명령을 입력하여 `C:\start\myfile.txt` 에서 `C:\end\myfile.txt` 로 파일 전송을 시작하십시오.

```
fteCreateTransfer -sa AGENT1 -sm hubQM -sd delete -da AGENT2 -dm hubQM -w -dd C:\end\  
C:\start\myfile.txt
```

- `-sa AGENT1`은 소스 에이전트(즉, 파일 전송이 시작되는 에이전트)를 `AGENT1`으로 정의합니다.
- `-sm hubQM`은 소스 에이전트인 `AGENT1`이 연결되는 큐 관리자를 정의합니다.
- `-sd delete`는 전송이 완료되면 소스 파일을 삭제하도록 지정합니다.
- `-da AGENT2`는 대상 에이전트(즉, 파일이 전송되는 에이전트)를 `AGENT2`로 정의합니다.
- `-w`는 `fteCreateTransfer` 명령이 성공 확인을 기다리도록 요청합니다.
- `-dd C:\end\`는 대상 디렉토리를 `C:\end\`로 정의합니다.
- `C:\start\myfile.txt`는 전송할 파일을 정의합니다.

자세한 정보는 `fteCreateTransfer`([새 파일 전송 작성](#))를 참조하십시오.

3. IBM WebSphere MQ Explorer에서 파일 전송 로그를 조사하거나 수동으로 파일 시스템을 조사하여 파일 전송이 성공했는지 확인하십시오.

다음에 수행할 작업

외부 매체를 통해 파일 전송 기능을 추가적으로 탐색할 수 있습니다. 다음을 참조하십시오.

- [148 페이지의 『스크립트를 사용하는 두 컴퓨터 파일 전송』](#)
- 환경을 추가적으로 보안합니다. 사용자의 요구사항에 따라 이 시나리오에서 사용된 모델과 다른 액세스 모델이 필요할 수 있습니다. 이 영역의 우수 사례에 대한 자세한 정보는 [WebSphere MQ File Transfer Edition V7 보안을 참조하십시오](#).

세부적인 기본 파일 전송

IBM WebSphere MQ Version 7.5를 사용하여 여러 가지 방법으로 파일을 전송할 수 있습니다. 이 절의 주제를 읽고 이 시나리오에서 다루는 내용, 시나리오를 따르려는 비즈니스의 이유, 관련 사용자 역할 및 시나리오에서 제안하는 솔루션의 개요를 읽으십시오.

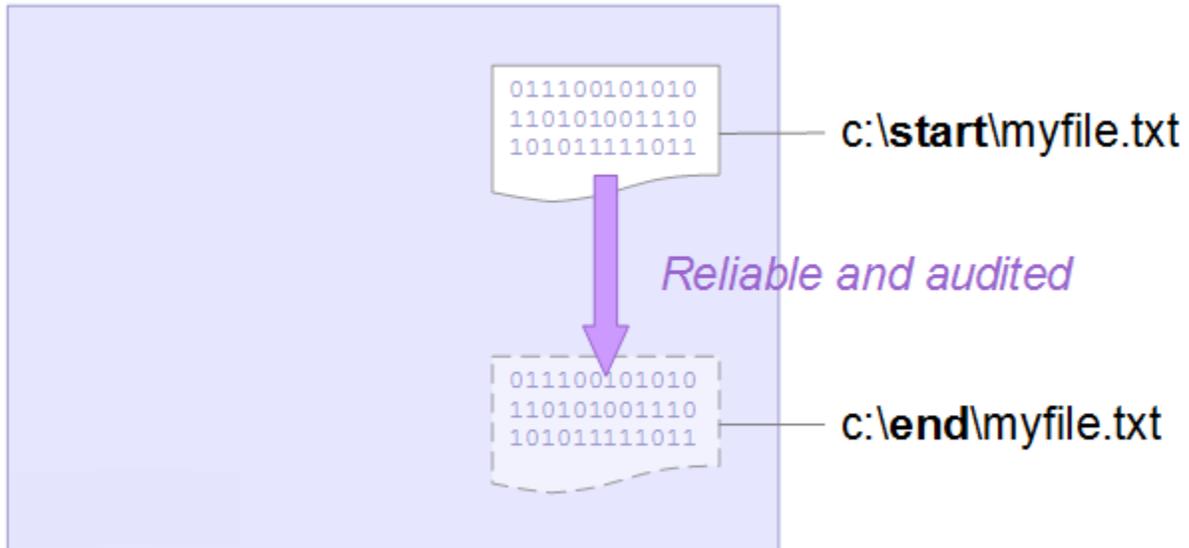
파일을 감사가 가능한 방법으로 안전하게 제어하면서 전송하는 것은 엔터프라이즈의 기본적인 요구사항입니다. IBM WebSphere MQ Version 7.5는 통합 메시징 플랫폼의 일부로 관리 파일 전송 기능을 제공합니다. 기본 파일 전송을 통하거나 메시징에 본격적으로 참여하여 관리 파일 전송 기능을 사용하여 파일을 메시징 인프라와 유기적으로 통합할 수 있습니다.

이 시나리오는 가장 단순한 IBM WebSphere MQ 메시징 기술로 파일을 통합하는 기본적인 방법에 대한 이해를 돕습니다. 이 기본 IBM WebSphere MQ 시나리오 작업은 파일을 한 위치에서 다른 위치로 이동하도록 디자인되었습니다. 이 초기 시나리오는 단일 컴퓨터로 제한되나 환경을 구성하고 이후 시나리오에 대한 중요한 기초를 구성하는 경험을 제공합니다. 이 시나리오는 네트워크를 통해 파일을 전송하기 위해 IBM WebSphere MQ를 사용하는 방법을 보여주며 관리 파일 전송 컴포넌트가 실제 비즈니스 문제점을 처리하는 방법을 보여줍니다.

IBM WebSphere MQ, 특히 `runmqsc` 및 IBM WebSphere MQ 탐색기와 같은 명령을 사용하여 IBM WebSphere MQ의 기본 구성 및 관리 개념과 큐 관리자의 개념을 기본적으로 이해해야 합니다.

개요

IBM WebSphere MQ를 사용하여 단일 컴퓨터의 한 위치에서 다른 위치로 파일을 전송하는 것을 식별하고 추적할 수 있습니다. IBM WebSphere MQ Version 7.5에서 관리 파일 전송 기능을 설치, 구성 및 사용해볼 수 있는 경험을 제공하므로 파일 전송을 사용하여 실제 비즈니스 문제점을 처리하는 방법에 대해 이해할 수 있습니다.



파일 전송 계획 기능에 대한 자세한 정보는 [WebSphere MQ Managed File Transfer 소개](#)를 참조하십시오.

솔루션 계획

컴퓨터에 대한 파일 전송, 파일 전송 에이전트, 필수조건, 라이선스, 컴퓨터 준비, 기본 파일 전송 시나리오용으로 파일을 전송할 수 있도록 IBM WebSphere MQ를 설치 및 구성하는 방법 등에 대해 설명합니다.

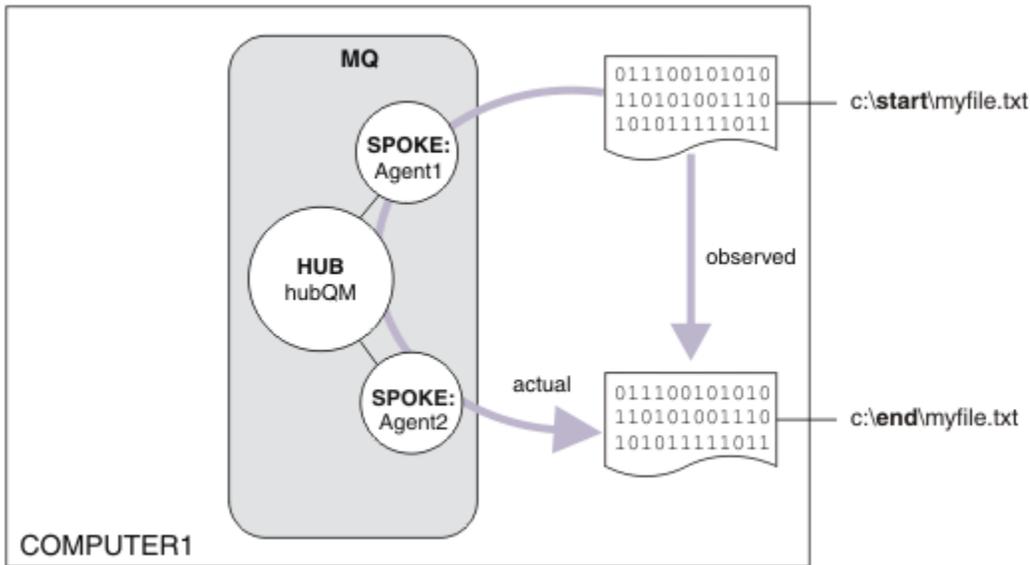
지점 A에서 지점 B로 파일을 전송하려면 IBM WebSphere MQ 큐 관리자 및 두 개의 파일 전송 에이전트를 정의하십시오. 이미 큐 관리자의 개념을 이해하고 있으나 파일 전송 에이전트에는 익숙하지 않을 수 있습니다. 파일 전송 에이전트는 컴퓨터에서 실행되어 기타 에이전트에 대해 파일을 전송하는 Java 프로세스입니다. 이 시나리오에서는 두 메커니즘을 통해 샘플 파일을 한 위치에서 다른 위치로 이동하기 위해 이러한 파일 전송 에이전트를 사용하는 파일 전송을 정의합니다.

1. 그래픽 IBM WebSphere MQ Explorer 사용
2. 명령행 사용

둘러보기에서는 사용자에게 Windows 시스템이 있다고 가정합니다. UNIX 시스템의 경우, 적절한 경로 및 명령으로 대체하십시오. 또한 모든 관련 디렉토리에 대해 읽기 및 쓰기 액세스 권한이 있어야 합니다. 또한 큐 관리자가 수행하는 작업에 대해 기본적인 사항을 이해하고 있다고 가정합니다.

파일을 컴퓨터에 대해 전송하려면 파일 전송 에이전트가 해당 컴퓨터에서 실행 중이어야 합니다. 모든 에이전트가 IBM WebSphere MQ 큐 관리자에 연결되고 IBM WebSphere MQ를 사용하여 기타 에이전트와 통신합니다. 자세한 정보는 [WebSphere MQ Managed File Transfer 토폴로지 개요](#)를 참조하십시오.

이 시나리오는 소스 및 대상 파일 둘 다를 관리하는 단일 파일 전송 에이전트를 사용할 수 있습니다. 그러나 이 예에서는 일반적인 네트워크 범위의 실제 상황에서 파일 전송이 작동하는 방법을 보다 현실적으로 이해할 수 있는 두 파일 전송 에이전트를 사용합니다.



이 다이어그램은 기본 파일 전송을 나타내기 위해 IBM WebSphere MQ가 사용되는 방법을 표시합니다. 파일이 한 위치에서 다른 위치로 복사되는 것을 관찰하십시오.

필수조건 및 라이선스

다음 항목이 필요합니다.

- IBM WebSphere MQ의 하드웨어 및 운영 체제 전제조건을 충족시키는 테스트 컴퓨터는 IBM WebSphere MQ 또는 IBM WebSphere MQ 데이터의 기존 설치가 없는 <https://www.ibm.com/support/docview.wss?uid=swg27006467>를 참조하십시오.
- IBM WebSphere MQ Version 7.5. <https://www.ibm.com/developerworks/downloads/ws/wmq/>에서 평가판을 다운로드할 수 있습니다.

컴퓨터 준비

테스트 컴퓨터가 IBM WebSphere MQ Version 7.5 설치 요구사항을 충족하는지 확인하십시오. [요구사항 확인](#)을 참조하십시오.

컴퓨터가 설치를 위해 적절한 준비가 되었는지 확인하십시오. [명령행을 사용한 로컬 설치 확인](#)을 참조하십시오.

파일 전송용 WebSphere MQ 구성

1. 큐 관리자 hubQM을 작성하십시오.

`crtmqm` hubQM 명령을 입력하십시오. 자세한 내용은 [crtmqm](#) 을 (를) 참조하십시오.

2. 큐 관리자 hubQM을 시작하십시오.

`strmqm` hubQM 명령을 입력하십시오. 자세한 정보는 [strmqm](#)을 참조하십시오.

3. 파일 전송을 통합하도록 hubQM 큐 관리자를 구성하십시오.

- a. 다음 명령을 입력하여 IBM WebSphere MQ에 대한 특성 파일 및 통합 큐 관리자 디렉토리를 작성하십시오.

```
fteSetupCoordination -coordinationQMgr hubQM
```

이 명령은 IBM WebSphere MQ에 대한 특성 파일 및 통합 큐 관리자 디렉토리를 작성합니다. 이 경우, hubQM이 통합 큐 관리자 브로드캐스트 감사 및 파일 전송 정보 역할을 수행합니다.

fteSetupCoordination 명령을 사용할 수 없는 경우, 이는 기본 설정으로 WebSphere MQ를 설치했음을 의미합니다. 추가 패키지를 설치할 경우(예: 관리 파일 전송) 이 문제가 해결됩니다.

자세한 정보는 [IBM WebSphere MQ 관리 파일 전송 토폴로지 개요](#) 및 [fteSetupCoordination\(조정 세부사항 설정\)](#)을 참조하십시오.

- b. 다음 명령을 입력하여 hubQM이 통합 큐 관리자 역할을 수행하도록 구성하십시오.

```
runmqsc hubQM < <filepath from previous statement>
```

- c. 파일 전송 명령을 핸들링하는 큐 관리자를 정의합니다. 이 경우에는 hubQM입니다.

```
fteSetupCommands -connectionQMgr hubQM
```

자세한 정보는 [fteSetupCommands\(command.properties 파일 작성\)](#)를 참조하십시오.

4. 첫 번째 파일 전송 에이전트 AGENT1을 작성하십시오.

- a. 에이전트가 연결된 큐 관리자에 대해 반드시 실행해야 하는 MQSC 스크립트를 포함하여 파일 전송 에이전트 AGENT1을 준비하십시오. 이 경우에는 다음 명령을 입력하여 hubQM을 준비하십시오.

```
fteCreateAgent -agentName AGENT1 -agentQMgr hubQM
```

자세한 정보는 [fteCreateAgent\(WebSphere MQ Managed File Transfer 에이전트 작성\)](#)을 참조하십시오.

- b. 작성한 에이전트를 핸들링하도록 hubQM을 구성하십시오.

```
runmqsc hubQM < <location of AGENT1_create.mqsc>
```

파일의 위치는 IBM WebSphere MQ를 설치한 위치에 따라 다릅니다.

5. 두 번째 파일 전송 에이전트 AGENT2를 작성하십시오.

```
fteCreateAgent -agentName AGENT2 -agentQMgr hubQM  
runmqsc hubQM < <location of AGENT2_create.mqsc>
```

6. AGENT1을 시작하십시오.

```
fteStartAgent AGENT1
```

자세한 정보는 [fteStartAgent\(WebSphere MQ Managed File Transfer 에이전트 시작\)](#)을 참조하십시오.

7. AGENT2를 시작하십시오.

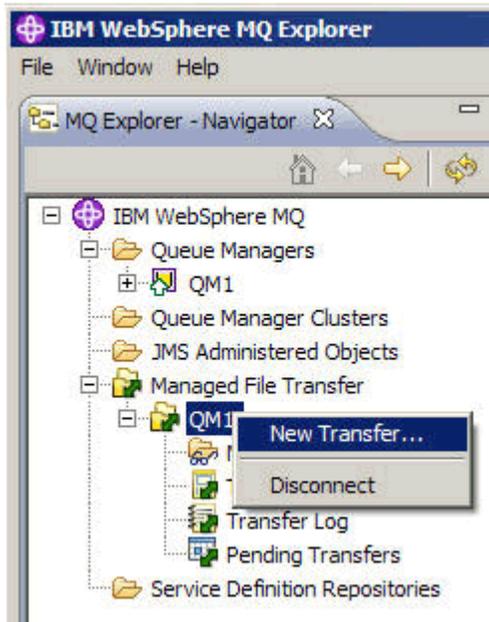
```
fteStartAgent AGENT2
```

솔루션 구현

이 시나리오에서 솔루션 구현은 IBM WebSphere MQ Explorer 를 사용하여 파일 전송을 정의하고 시작하는 것을 포함합니다. 전송 진행 과정을 모니터링하고 파일 시스템을 조사하여 성공했는지 확인하십시오.

프로시저

1. 소스 파일 및 대상 디렉토리를 식별하십시오.
 - a) 전송할 샘플 파일을 작성하십시오 (예: C:\start\myfile.txt).
 - b) 이 파일을 전송할 기존 디렉토리를 식별하십시오. 예를 들어, C:\end\입니다.
2. IBM WebSphere MQ Explorer를 시작하십시오. 시작 메뉴(또는 동등한 메뉴)에서 프로그램을 시작하거나 **MQExplorer**를 실행하십시오. 자세한 내용은 [IBM WebSphere MQ Explorer 실행](#)을 참조하십시오.
3. IBM WebSphere MQ Explorer 네비게이터에서 **관리 파일 전송**을 클릭하고 **QM**을 마우스 오른쪽 단추로 클릭하고 **새 전송**을 선택하여 새 전송 마법사를 시작하십시오.



4. 시작 섹션에서 소스 에이전트로 **AGENT1** 을 선택하십시오.

From:

Agent: AGENT1

Type: File

File:

Include subdirectories

5. 이전에 작성한 파일의 경로를 입력하십시오. 예를 들어, C:\start\myfile.txt입니다.

From:

Agent: AGENT1

Type: File

File: C:\Users\Ben Bakowski\Desktop\FTETEST\START\myDemoTransfer.txt

Include subdirectories

6. To 섹션에서 목적지 에이전트로 **AGENT2**를 선택하십시오.

7. 이전에 식별한 대상 디렉토리를 입력하십시오. 예를 들어, C:\end\입니다.

To:

Agent: AGENT2

Type: File

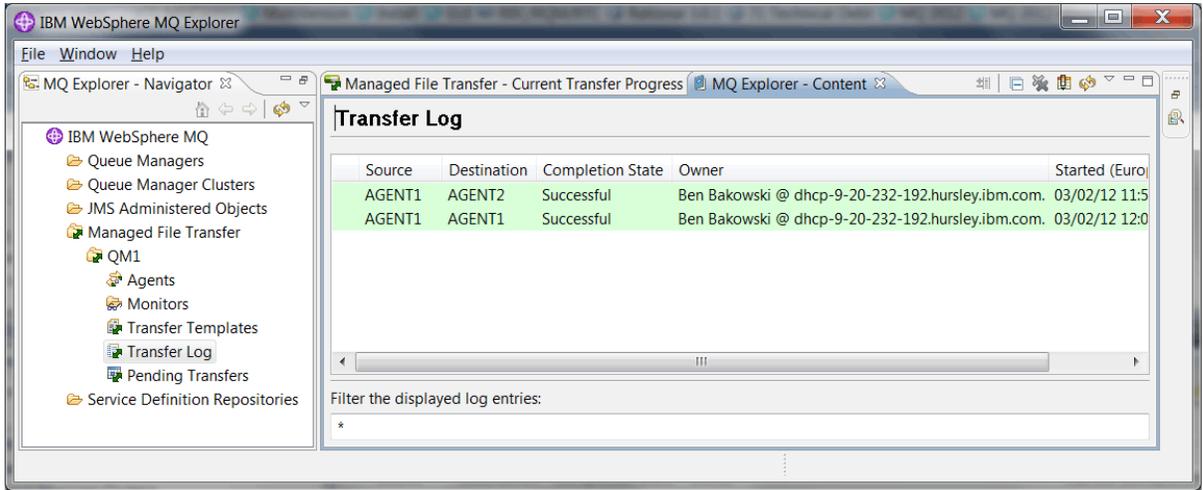
Directory: C:\Users\Ben Bakowski\Desktop\FTETEST\END

File name: myDemoTransfer.txt

Overwrite files on the destination file system that have the same name

8. 마침을 클릭하고 전송을 시작하십시오.

9. IBM WebSphere MQ Explorer의 현재 전송 진행 상태 탭에서 전송 진행 상태를 모니터할 수 있습니다.



10. 또한 파일 시스템을 수동으로 검사하여 새 파일이 있는지 확인할 수 있습니다. 예를 들어, C:\end\myfile.txt입니다.

파일을 전송하기 위해 명령행 사용

명령행을 사용하여 전송을 시작하면서 파일 전송의 유연성을 나타냅니다. 이 시나리오의 범위를 벗어나지만 이러한 원칙 및 Ant 스크립트 기술을 기반으로 하여 훨씬 강력한 파일 전송 시나리오를 정의하고 구현할 수 있습니다.

프로시저

1. 이전 데모로부터 전송된 파일을 제거하십시오. 예를 들어, C:\end\myfile.txt입니다.
2. **fteCreateTransfer** 명령을 사용하여 C:\start\myfile.txt로부터 C:\end\myfile.txt로 파일 전송을 시작하십시오.

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT2 -dm hubQM -w -dd C:\end\
C:\start\myfile.txt
```

- -sa AGENT1은 소스 에이전트(즉, 파일 전송이 시작되는 에이전트)를 AGENT1으로 정의합니다.
- -sm hubQM은 소스 에이전트인 AGENT1이 연결되는 큐 관리자를 정의합니다.
- -da AGENT2는 대상 에이전트(즉, 파일이 전송되는 에이전트)를 AGENT2로 정의합니다.
- -w는 **fteCreateTransfer** 명령이 성공 확인을 기다리도록 요청합니다.
- -dd C:\end\는 대상 디렉토리를 C:\end\로 정의합니다.
- C:\start\myfile.txt는 전송할 파일을 정의합니다.

자세한 정보는 **fteCreateTransfer(새 파일 전송 작성)**를 참조하십시오.

3. IBM WebSphere MQ Explorer에서 파일 전송 로그를 조사하거나 수동으로 파일 시스템을 조사하여 파일 전송에 성공했는지 확인하십시오.

다음에 수행할 작업

외부 매체를 통해 파일 전송 기능을 추가적으로 탐색할 수 있습니다. 다음을 참조하십시오.

- 148 페이지의 『스크립트를 사용하는 두 컴퓨터 파일 전송』
- 환경을 추가적으로 보안합니다. 사용자의 요구사항에 따라 이 시나리오에서 사용된 모델과 다른 액세스 모델이 필요할 수 있습니다. 이 영역의 우수 사례에 대한 자세한 정보는 https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html의 내용을 참조하십시오.

스크립트를 사용하는 두 컴퓨터 파일 전송

파일 전송을 다중 컴퓨터 IBM WebSphere MQ 메시징 토폴로지에 통합하기 위해 스크립트를 사용하는 기본 파일 전송 시나리오를 확장합니다.

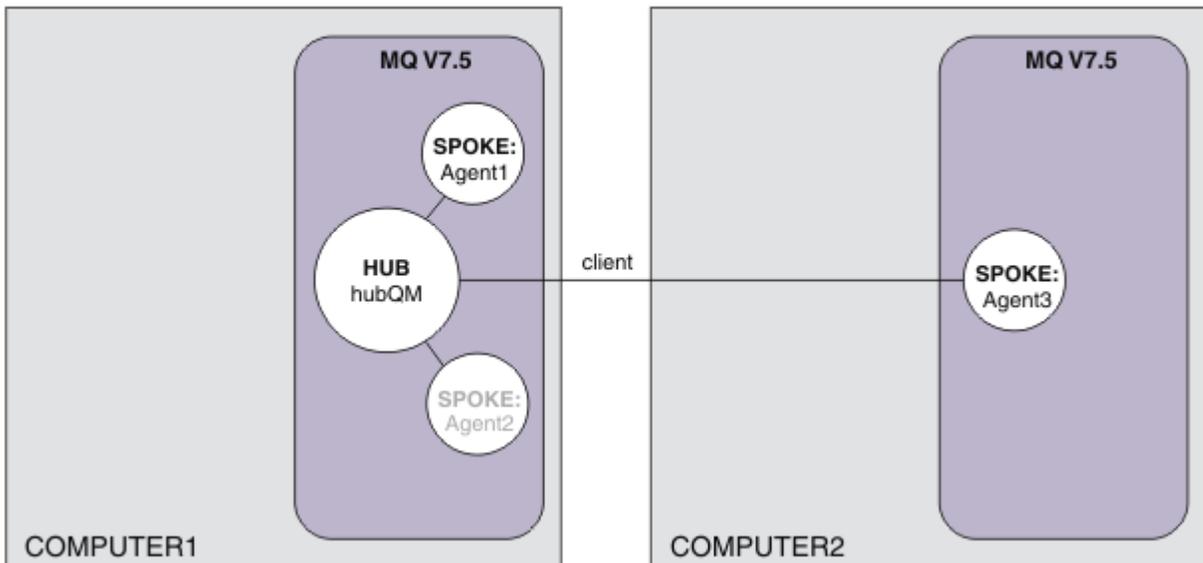
관리 파일 전송 기능의 단순 재연을 빌드하여(134 페이지의 『스크립트를 사용하는 기본 파일 전송』 참조), 이제 관리 파일 전송의 기본 원칙에 익숙해졌을 것입니다. 단일 컴퓨터에서는 이 기능이 거의 장점이 없음을 인식할 수 있으므로 이제 시나리오를 확장하여 파일 전송을 다중 컴퓨터 IBM WebSphere MQ 메시징 토폴로지로 통합하는 방법을 탐색할 것입니다.

기본적인 IBM WebSphere MQ 기술에 의해 제공되는 장점, 즉, 파일을 안정적으로 한 번만 전달하는 방법에 액세스할 수 있습니다. 이를 위해 토폴로지가 파일 전송에 참여하는 두 번째 컴퓨터를 포함하도록 확장됩니다. 이 시나리오에서는 별도로 사용 가능한 파일 전송 에이전트를 설치 및 구성하고 다중 컴퓨터 파일 전송 토폴로지에 적용되는 보안 고려사항에 대해 설명합니다. 이 시나리오는 한 컴퓨터에서 다음 시스템으로 파일을 전송하는 것을 재연하는 것으로 끝납니다. 또한 조정 기능을 추가할 수 있는 다음 시나리오에 대한 강력한 토대를 제공하며 관리 파일 전송 기능이라는 이름을 얻은 이유를 보여줍니다.

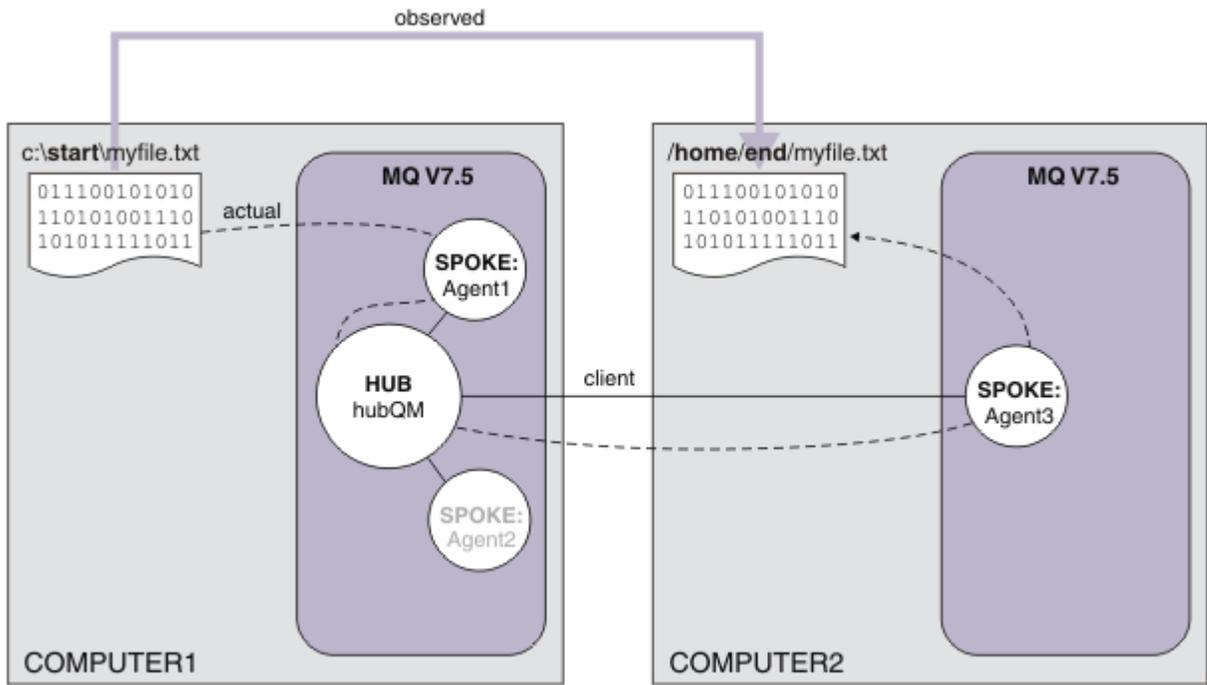
134 페이지의 『스크립트를 사용하는 기본 파일 전송』에서 지정된 태스크를 완료했다고 가정합니다.

개요

이 시나리오에서는 134 페이지의 『스크립트를 사용하는 기본 파일 전송』 시나리오에서 구성된 허브 및 스포크 토폴로지가 두 번째 컴퓨터인 컴퓨터 2를 포함하도록 확장됩니다. 컴퓨터 2에는 관리 파일 전송 에이전트 및 명령행 도구가 설치되어 있습니다. 이 두 번째 컴퓨터에는 로컬 IBM WebSphere MQ 서버 설치의 전제조건이 없습니다. 이러한 모델은 허브 및 스포크 솔루션에서 흔히 사용됩니다. 이 때, 다중 Managed File Transfer 에이전트가 각 스포크에 대해 IBM WebSphere MQ 서버 및 연관된 라이선스를 설치할 필요 없이 중앙 집중화된 IBM WebSphere MQ 큐 관리자에 대한 클라이언트 연결을 통해 상호 작용할 수 있습니다. 기타 토폴로지가 지원되며 자체 토폴로지는 사용자의 필요 및 라이선스 부여 또는 권한에 따라 결정됩니다. 이 컴퓨터에서 전송을 핸들링하는 데 AGENT1만 사용하기 위해 AGENT2는 사용되지 않습니다.



이 다이어그램은 두 컴퓨터에 걸친 허브 및 스포크 토폴로지를 표시합니다. AGENT1 및 AGENT2는 (로컬) 허브에 대해 바인딩을 사용하도록 구성되며 AGENT3은 클라이언트 연결을 통해 연결됩니다. AGENT2는 시나리오에서 추가적인 역할을 수행하지 않으므로 사용되지 않습니다. AGENT1은 컴퓨터 1에서 모든 파일 전송 활동을 처리합니다. 보안 모델 및 해당 제한사항을 이해하고 있는지 확인하십시오. 자세한 내용은 134 페이지의 『스크립트를 사용하는 기본 파일 전송』 시나리오를 참조하십시오. 이 토폴로지는 파일을 1 컴퓨터에서 2 컴퓨터로 전송하는 데 사용하기 위해 구성됩니다.



이 다이어그램은 재연되는 파일 전송 라우트를 표시합니다. 기본 파일 전송은 신뢰할 수 있으며 활성 상태인 IBM WebSphere MQ 연결을 통해 수행됩니다. 이 예제에서 컴퓨터 1은 Windows 컴퓨터이고 컴퓨터 2는 Linux 컴퓨터라고 가정합니다. 대체 플랫폼 및 아키텍처를 사용할 수 있습니다. 전체 지원 플랫폼 목록을 보려면 <https://www.ibm.com/support/docview.wss?uid=swg27006467#7.1>의 내용을 참조하십시오.

솔루션 계획

컴퓨터에 대한 파일 전송, 파일 전송 에이전트, 필수조건, 라이선스, 컴퓨터 준비, 스크립트 시나리오를 사용하여 두 컴퓨터 파일 전송용으로 IBM WebSphere MQ를 구성하는 방법 등에 대해 설명합니다.

전제조건

다음 항목이 필요합니다.

- 1 컴퓨터 - 134 페이지의 『스크립트를 사용하는 기본 파일 전송』에서 구성 작업 수행
- 2 컴퓨터 - IBM WebSphere MQ Version 7.5에 대한 하드웨어 및 운영 체제 필수조건을 충족하는 두 번째 테스트 컴퓨터. 자세한 정보는 <https://www.ibm.com/support/docview.wss?uid=swg27006467>의 내용을 참조하십시오.
- IBM WebSphere MQ Version 7.5. <https://www.ibm.com/developerworks/downloads/ws/wmq/>에서 평가판을 다운로드할 수 있습니다.
- 1 컴퓨터 및 2 컴퓨터의 IP 주소 형식은 nnn.nnn.nnn.nnn입니다.

IBM WebSphere MQ Version 7.5을 설치하십시오.

서버, IBM WebSphere MQ 탐색기, IBM WebSphere MQ 고급 관리 파일 전송 에이전트 및 IBM WebSphere MQ 관리 파일 전송 명령행 도구 구성요소와 함께 IBM WebSphere MQ Version 7.5 서버를 설치하십시오. 자세한 내용은 설치 항목 선택을 참조하십시오.

IBM WebSphere MQ 관리 방법을 결정하십시오. 다음 방법으로 IBM WebSphere MQ를 관리할 수 있습니다.

- **setmqenv** 명령을 사용하여 적절한 환경 설정. 자세한 정보는 [setmqenv](#)를 참조하십시오.
- 완전한 IBM WebSphere MQ 관리 명령 호출.

사용자 및 그룹 준비

이 보안 모델에서는 다음과 같은 그룹 및 사용자가 있다고 가정합니다.

그룹

- mqm

IBM WebSphere MQ 설치의 일부로 자동으로 작성됩니다. 이 그룹의 구성원은 IBM WebSphere MQ 및 해당 자원을 관리할 수 있습니다.

- FTUSERS

이 그룹은 반드시 작성해야 합니다. 이 그룹의 구성원은 파일 전송을 시작할 수 있습니다.

- FTAGENTS

이 그룹은 반드시 작성해야 합니다. 이 그룹의 구성원은 네트워크 내에서 파일의 전송을 핸들링하는 엔드포인트인 파일 전송을 시작하고 중지할 수 있습니다.

사용자

- mqmAdmin

mqm 그룹의 구성원이며 Windows에서는 관리자 그룹의 구성원인 IBM WebSphere MQ 관리자

- ftuser

이 사용자는 반드시 작성해야 하며 FTUSERS 그룹의 구성원이어야 합니다. 관리 보안에 대한 조치가 큐 관리자에서 수행되는 가능성을 피하기 위해 이 사용자를 mqm 그룹에 추가하지 마십시오.

- ftagent

이 사용자는 반드시 작성해야 하며 FTAGENTS 그룹의 구성원이어야 합니다. 관리 보안에 대한 조치가 큐 관리자에서 수행되는 가능성을 피하기 위해 이 사용자를 mqm 그룹에 추가하지 마십시오.

2 컴퓨터에 스포크로 AGENT3 에이전트 추가

스크립트를 사용하는 IBM WebSphere MQ 두 컴퓨터 파일 전송 시나리오를 위해 2 컴퓨터가 확장된 토폴로지를 지원하도록 준비합니다.

이 태스크 정보

1 컴퓨터에서 addSpoke 샘플 스크립트를 다시 사용하십시오. 이 때, 클라이언트 연결을 통해 연결되는 에이전트 스포크를 정의하십시오. **addSpoke** 명령을 사용하면 사용자에게 2 컴퓨터의 IBM WebSphere MQ 설치로부터 실행하는 데 필요한 명령 세트를 제시합니다.

프로시저

1. On computer 1, as user mqmAdmin, run the **addspoke** command from IBM WebSphere MQ bin directory <MQ_INSTALL_ROOT>\bin.

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT3
hubQMgr=hubQM connectionMode=CLIENT agentIPAddress=<IP address of computer 2>
hubIPAddress=<IP address of computer 1> hubPort=1414
```

2. 2 컴퓨터에서 mqmAdmin 사용자로서 1 컴퓨터의 **addSpoke** 명령에 의해 출력된 명령을 실행하십시오. 예를 들어, 다음과 같습니다.

- a) `fteSetupCoordination -coordinationQMGr hubQM-coordinationQMGrHost <IP address of computer 1> -coordinationQMGrPort 1414 -coordinationQMGrChannel FTE.USER.SVRCONN -f`
- b) `fteSetupCommands -p hubQM -connectionQMGr hubQM -connectionQMGrHost <IP address of computer 1> -connectionQMGrPort 1414 -connectionQMGrChannel FTE.USER.SVRCONN -f`

```
c) fteCreateAgent -p hubQM -agentName AGENT3 -agentQMgr hubQM -agentQMgrHost
  <IP address of computer 1> -agentQMgrPort 1414 -agentQMgrChannel
  FTE.AGENT.SVRCONN -f
```

생성된 MQSC 스크립트를 실행할 필요는 없습니다.

3. 2 컴퓨터에서 ftagent 사용자로서 AGENT3을 시작하십시오.

```
fteStartAgent -p hubQM AGENT3
```

4. 2 컴퓨터에서 ftagent 사용자로서 세 에이전트가 사용 가능한지 확인하십시오. (필요한 경우, ftagent 사용자로서 1 컴퓨터의 에이전트부터 시작하십시오.)

```
ftelistAgents -p hubQM
```

다음 출력이 표시됩니다.

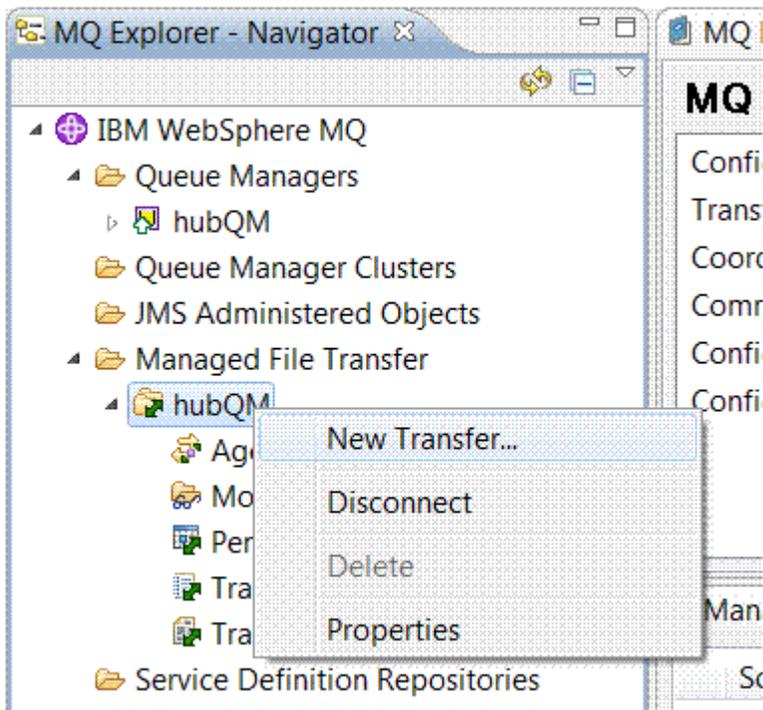
Agent Name:	Queue Manager Name:	Status:
AGENT1	hubQM	READY
AGENT2	hubQM	READY
AGENT2	hubQM	READY

솔루션 구현

파일을 전송하기 위해 IBM WebSphere MQ를 사용하여 스크립트 시나리오를 사용하는 두 컴퓨터 파일 전송 데모를 시작하십시오. 전송 진행 과정을 모니터링하고 파일 시스템을 조사하여 성공했는지 확인하십시오.

프로시저

1. 소스 파일 및 대상 디렉토리를 식별하십시오.
 - a) 1 컴퓨터에서 전송할 샘플 파일을 작성하십시오. 예를 들어, C:\start\myfile.txt입니다.
 - b) 2 컴퓨터에서 이 파일을 전송할 기존 디렉토리를 식별하십시오. 예를 들어, C:\end\입니다. ftagent 사용자가 디렉토리에 대한 쓰기 액세스 권한이 있는지 확인하십시오.
2. Start IBM WebSphere MQ Explorer on computer 1 as user mqmAdmin. 이전 시나리오에서는 IBM WebSphere MQ Explorer 구성이 아니라 파일 전송에 대한 시나리오에 초점을 맞추고 IBM WebSphere MQ 관리자로서 이 작업을 수행했습니다. 시작 메뉴(또는 동등한 메뉴)에서 프로그램을 시작하거나 **MQExplorer**를 실행하십시오. 자세한 정보는 [WebSphere MQ Explorer 실행](#)을 참조하십시오.
3. IBM WebSphere MQ 네비게이터의 **관리 파일 전송** 섹션에서 hubQM 을 마우스 오른쪽 단추로 클릭하고 **새 전송** 을 선택하여 새 전송 마법사를 시작하십시오.



4. 소스 에이전트로 AGENT1 을 선택하고 대상 에이전트로 AGENT3 을 선택하려면 이 메뉴를 사용하십시오.

Source agent
 Name: AGENT1
 Type: Standard

Destination agent
 Name: AGENT3
 Type: Standard

5. 다음 을 클릭한 후 추가 ... 를 클릭하십시오. 및 소스의 경우 전송하려는 파일의 경로를 입력하십시오 (예: C:\start\myfile.txt). 전송이 완료되면 소스 파일 제거 선택란이 체크되어 있는지 확인하십시오.

Source
 Agent: AGENT1
 Type: Standard agent
 Type: File

File name: C:\start\myFile.txt Browse...

Remove source file if the transfer is successful

6. 대상 디렉토리를 입력하십시오. 이 시나리오에서 이 목적지는 Linux 플랫폼에 있으므로 적절한 /home/end/ 표시법을 사용하십시오.

Destination

Agent: AGENT3

Type: Standard agent

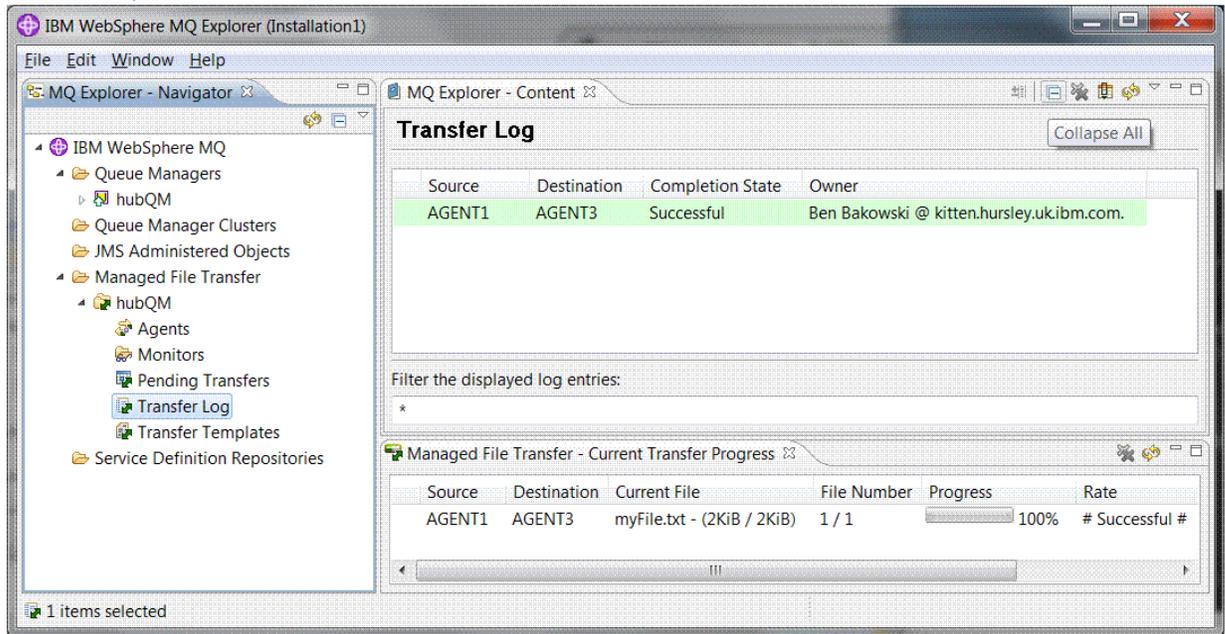
Type:

Directory:

Overwrite files if present

7. 확인을 클릭한 다음 마침을 클릭하면 전송이 시작됩니다.

8. IBM WebSphere MQ 탐색기의 현재 전송 진행 상태 탭에서 전송 진행 상태를 모니터링할 수 있습니다.



9. 또한 파일 시스템을 수동으로 검사하여 새 파일이 있는지 확인할 수 있습니다. 예를 들어, /home/end/myfile.txt입니다.

파일을 전송하기 위해 명령행 사용

명령행을 사용하여 전송을 시작하면서 파일 전송의 유연성을 나타냅니다. 이 시나리오의 범위를 벗어나지만 이러한 원칙 및 Ant 스크립트 기술을 기반으로 하여 훨씬 강력한 파일 전송 시나리오를 정의하고 구현할 수 있습니다.

프로시저

1. 이전 데모로부터 전송된 파일을 제거하십시오. 예: /home/end/myfile.txt. 그리고 원래 위치에서 이 파일을 다시 작성하십시오.
2. ftuser 사용자로서 **fteCreateTransfer** 명령을 사용하여 1 컴퓨터의 C:\start\myfile.txt로부터 2 컴퓨터의 /home/end/myfile.txt로 파일 전송을 시작하십시오.

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT3 -dm hubQM -w -dd "/home/end/"
"C:\start\myfile.txt"
```

3. IBM WebSphere MQ Explorer에서 파일 전송 로그를 조사하거나 수동으로 파일 시스템을 조사하여 파일 전송이 성공했는지 확인하십시오.

다음에 수행할 작업

IBM WebSphere MQ 보안에 대한 자세한 내용은 [보안](#) 을 참조하십시오.

외부 매체를 통해 파일 전송 기능을 추가적으로 탐색할 수 있습니다.

- 관리 파일 전송의 관리 부문을 제공하기 위해 감사 기능을 추가하는 방법에 대한 자세한 정보는 [162 페이지의 『관리 파일 전송에 감사 기능 추가』](#) 의 내용을 참조하십시오.
- 다음에 대한 자세한 정보가 제공됩니다.
 - 트리거: 새 파일이 표시되면 이를 이동.
 - 트리거: 다중 파일의 전송을 시작하기 위해 단일 파일의 모양을 구성합니다.
 - Apache Ant를 사용한 스크립트 전송

https://www.ibm.com/developerworks/websphere/library/techarticles/1003_phillips/1003_phillips.html을 참조하십시오.

- 사용자의 요구사항에 따라 이 시나리오에서 사용된 모델과 다른 액세스 모델이 필요할 수 있습니다. 사용자의 환경에 추가적인 보안을 제공하는 방법에 대한 우수 사례는 https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html의 내용을 참조하십시오.

세부적인 2 컴퓨터 파일 전송

파일 전송을 다중 컴퓨터 IBM WebSphere MQ 메시징 토폴로지에 통합하기 위해 기본 파일 전송 시나리오를 확장합니다.

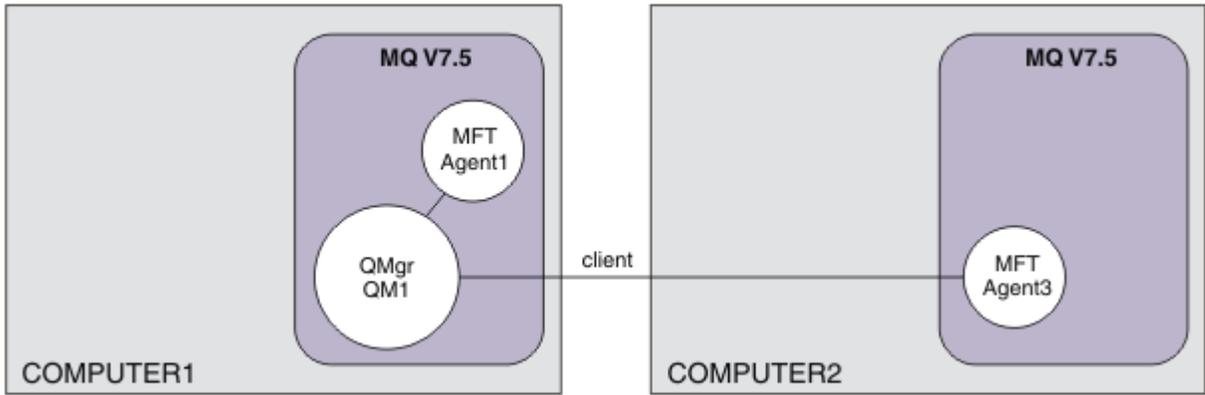
관리 파일 전송 기능의 단순 재연을 완료하여(142 페이지의 『세부적인 기본 파일 전송』 참조), 이제 관리 파일 전송의 기본 원칙에 익숙해졌을 것입니다. 단일 컴퓨터에서는 이 기능이 거의 장점이 없음을 인식할 수 있으므로 이제 시나리오를 확장하여 파일 전송을 다중 컴퓨터 IBM WebSphere MQ 메시징 토폴로지 통합하는 방법을 탐색할 것입니다.

기본적인 IBM WebSphere MQ 기술에 의해 제공되는 장점, 즉, 파일을 안정적으로 한 번만 전달하는 방법에 액세스할 수 있습니다. 이를 위해 토폴로지가 파일 전송에 참여하는 두 번째 컴퓨터를 포함하도록 확장됩니다. 이 시나리오에서는 별도로 사용 가능한 파일 전송 에이전트를 설치 및 구성하고 다중 컴퓨터 파일 전송 토폴로지에 적용되는 보안 고려사항에 대해 설명합니다. 이 시나리오는 한 컴퓨터에서 다음 시스템으로 파일을 전송하는 것을 재연하는 것으로 끝납니다. 또한 조정 기능을 추가할 수 있는 다음 시나리오에 대한 강력한 토대를 제공하며 관리 파일 전송 기능이라는 이름을 얻은 이유를 보여줍니다.

142 페이지의 『세부적인 기본 파일 전송』에서 지정된 태스크를 완료했다고 가정합니다.

개요

이 시나리오에서는 기본 파일 전송 시나리오에서 구성한 기존 Windows 컴퓨터를 계속 사용합니다. 이 첫 번째 컴퓨터에서 파일 전송 엔드 포인트로 단일 에이전트를 사용하므로 Agent2는 사용 불가능합니다. 두 번째 컴퓨터에 Managed File Transfer 에이전트를 설치합니다. 로컬 IBM WebSphere MQ 서버 설치하는 이 두 번째 컴퓨터의 필수조건이 아닙니다. 이러한 모델은 허브 및 스포크 솔루션에서 흔히 사용됩니다. 이 때, 다중 Managed File Transfer 에이전트가 각 스포크에 대해 IBM WebSphere MQ 서버 및 연관된 라이선스를 설치할 필요 없이 중앙 집중화된 IBM WebSphere MQ 큐 관리자에 대한 클라이언트 연결을 통해 상호 작용할 수 있습니다. 기타 토폴로지가 지원되며 자체 토폴로지는 사용자의 필요 및 라이선스 부여 또는 권한에 따라 결정됩니다.

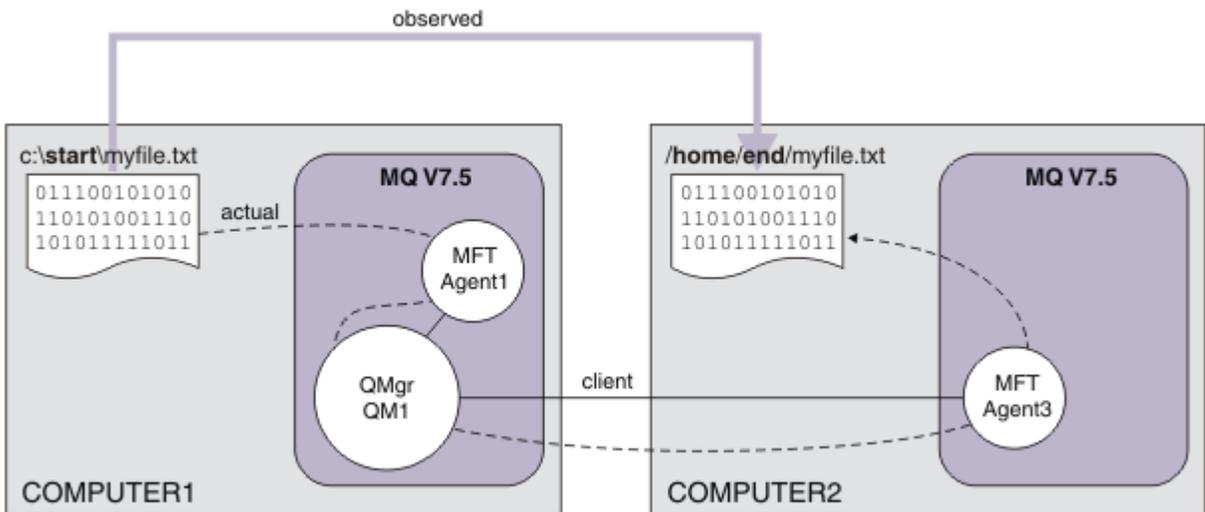


파일 전송 네트워크에서는 단일 큐 관리자가 브로드캐스트 감사 및 파일 전송 정보에 대한 네트워크 내의 단일 지점 역할을 수행하도록 선택됩니다. 이전의 142 페이지의 『세부적인 기본 파일 전송』 시나리오에서는 단일 큐 관리자가 묵시적으로 통합 큐 관리자로 사용되었습니다. 이 시나리오에서는 1 컴퓨터의 QM1 큐 관리자가 계속 통합 큐 관리자 역할을 수행하며 2 컴퓨터의 구성 중 한 파트에서 이 큐 관리자를 사용하도록 환경을 설정합니다.

이 토폴로지에서는 2 컴퓨터에서 파일 전송을 시작하는 기능이 필요하지 않으므로 선택적 관리 파일 전송 명령행 도구 컴포넌트를 이 두 번째 컴퓨터에 설치하지 않습니다.

일부 기본 보안 기능을 고려하더라도 파일 전송 토폴로지가 적합한 보안 레벨이 아니라는 점을 기억해야 합니다. IBM WebSphere MQ 보안 및 특히 파일 전송에 대한 논의는 https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html의 내용을 참조하십시오.

두 번째 컴퓨터에 설치가 수행되면 실제 에이전트 Agent3을 작성하고 파일을 1 컴퓨터에서 2 컴퓨터로 이동함으로써 솔루션을 재연합니다.



기본 파일 전송은 신뢰할 수 있고 IBM WebSphere MQ 연결을 수행하여 발생합니다. 다음 태스크는 이러한 연결을 구성하고 적절히 보안하는 방법을 보여줍니다.

이 예제에서 컴퓨터 1은 컴퓨터 2는 Linux 컴퓨터라고 가정합니다. 대체 플랫폼 및 아키텍처를 사용할 수 있습니다. 전체 지원 플랫폼 목록을 보려면 <https://www.ibm.com/support/docview.wss?uid=swg27006467#7.1>의 내용을 참조하십시오.

솔루션 계획

두 컴퓨터 파일 전송 솔루션 계획을 설명합니다. 적절한 인프라 및 작성해야 하는 사용자와 그룹에 대한 설명도 포함됩니다.

시작하기 전에

다음 항목이 필요합니다.

- 기본 파일 전송 시나리오의 구성 작업이 필요합니다. 자세한 내용은 142 페이지의 『세부적인 기본 파일 전송』의 내용을 참조하십시오.
- IBM WebSphere MQ Version 7.5. <https://www.ibm.com/developerworks/downloads/ws/wmq/>에서 평가판을 다운로드할 수 있습니다.
- IBM WebSphere MQ Version 7.5에 대한 하드웨어 및 운영 체제 필수조건을 충족하는 두 번째 테스트 컴퓨터가 필요합니다. 자세한 내용은 <https://www.ibm.com/support/docview.wss?uid=swg27006467>의 내용을 참조하십시오.

이 시나리오에서는 두 번째 컴퓨터가 클라이언트 연결 채널을 통해 큐 관리자와 상호작용합니다. 채널 인증 레코드를 사용하면 다음을 확인할 수 있습니다.

- 수신되는 연결은 추가적인 관리 파일 전송 에이전트를 호스트하는 새 컴퓨터부터 시작되는 것으로 인증됩니다.
- 수신되는 요청은 관리 파일 전송 자원에 대한 적절한 액세스 권한을 가진 사용자에게 맵핑됩니다.

파일 전송을 파일 시스템의 특정 영역 조작으로 제한하는 기술인 샌드박스(예: IBM WebSphere MQ 구성 파일)는 고려하지 않습니다.

프로시저

두 컴퓨터에서 다음을 수행하십시오.

1. FTEUSERS 및 FTEAGENTS 그룹을 작성하십시오. 자세한 정보는 운영 체제 지시사항을 참조하십시오.
2. `fteuser` 사용자를 작성하여 FTEAGENTS 그룹에 추가하십시오.
3. `fteagent` 사용자를 작성하여 FTEUSERS 그룹에 추가하십시오.

`fteuser` 사용자가 파일 전송을 시작하고 `fteagent` 사용자가 에이전트 프로세스를 시작하고 중지합니다. 이러한 사용자는 `mqm` (또는 Windows 관리자) 그룹의 구성원이 아니므로 원하지 않는 관리 공격에 대해 토폴로지를 강화하는 데 도움이 됩니다. 에이전트 기반으로 특정 사용자를 정의하여 보안 액세스 권한을 세부 단위로 구분할 수 있습니다. 예를 들어, `fteagent1`, `fteagent2` 등입니다.

확장된 토폴로지를 지원하도록 1 컴퓨터 수정

IBM WebSphere MQ 두 컴퓨터 파일 전송 시나리오를 위해 1 컴퓨터가 확장된 토폴로지를 지원하도록 준비합니다.

이 태스크 정보

이 태스크에서는 Agent2가 더 이상 필요하지 않으므로 이를 삭제합니다. 그런 다음 2 컴퓨터에서 작성된 에이전트로부터 클라이언트 연결을 허용하고 두 컴퓨터 간에 단순 IBM WebSphere MQ 네트워크를 구성하기 위해 리스너를 작성하고 시작합니다. 1 컴퓨터가 Windows 운영 체제를 실행 중인 것으로 가정합니다. 이 시나리오를 실행하기 위해 다른 플랫폼을 실행 중인 경우, 적절한 플랫폼별 명령으로 대체하십시오.

프로시저

1. Agent2 에이전트를 중지하십시오.

```
fteStopAgent AGENT2
```

fteStopAgent 명령에 대한 자세한 정보는 [fteStopAgent\(WebSphere MQ Managed File Transfer 에이전트 중지\)](#)를 참조하십시오.

2. Agent2 에이전트를 삭제하십시오.

```
fteDeleteAgent AGENT2  
runmqsc QM1 < <output>
```

fteDeleteAgent 명령에 대한 자세한 정보는 [fteDeleteAgent\(WebSphere MQ Managed File Transfer 에이전트 삭제\)](#)를 참조하십시오.

2 컴퓨터에서 구성된 새 파일 전송 에이전트가 통합 큐 관리자, QM1와 상호작용할 수 있도록 IBM WebSphere MQ 보안을 구성하십시오. 이 새 에이전트는 기존 SYSTEM.DEF.SVRCONN 채널을 통해 QM1에 연결됩니다. 사용자의 자체 보안은 상황에 따라 달라질 수 있습니다. 이 토폴로지 강화에 대한 자세한 내용은 다음에 수행할 작업을 참조하십시오.

3. QM1에 대한 MQSC 인터페이스를 시작하십시오.

```
runmqsc QM1
```

4. 사용자 및 에이전트로부터 수신되는 요청을 핸들링하기 위해 두 채널을 작성하십시오.

```
DEFINE CHANNEL(FTE.USER.SVRCONN) CHLTYPE(SVRCONN)
DEFINE CHANNEL(FTE.AGENT.SVRCONN) CHLTYPE(SVRCONN)
```

5. 작성된 사용자를 지정하고 2 컴퓨터로부터 QM1에 연결을 허용하려면 채널 인증 레코드를 작성하십시오.

```
SET CHLAUTH('FTE.USER.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('<IP address of computer2>')
USERSRC (MAP) MCAUSER('fteuser' DESCR('Rule to allow file transfer users to communicate'))
ACTION(ADD)

SET CHLAUTH('FTE.AGENT.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('<IP address of computer2>')
USERSRC(MAP) MCAUSER('fteagent') DESCR('Rule to allow file transfer agent processes to
communicate') ACTION(ADD)
```

자세한 정보는 [채널 인증 레코드](#)를 참조하십시오.

이 시나리오의 목적은 토폴로지를 잠그고 강화하는 것이 아니라 기본 파일 전송을 재연하는 것입니다. 구현된 보안 모델은 이 데모를 지원하지만 사용자는 고유 보안 위험을 파악하여 필요한 경우 적절한 조치를 취해야 합니다. 고려할 옵션에 대한 논의는 [다음에 수행할 작업](#)을 참조하십시오.

6. IBM WebSphere MQ에 대한 네트워크 통신에 사용할 빈 포트를 식별하십시오. LISTENER1이 이 빈 포트(예: 1414)를 사용하도록 정의하십시오.

```
DEFINE LISTENER(LISTENER1) TRPTYPE(TCP) CONTROL(QMGR) PORT(1414)
```

7. 리스너 LISTENER1을 시작하십시오.

```
START LISTENER(LISTENER1)
```

8. QM1에 대한 MQSC 인터페이스를 중지하십시오.

```
end
```

9. FTEAGENTS 및 FTEUSERS 그룹에 컴퓨터 2에 작성될 에이전트 AGENT3에 대한 파일 전송 조치를 수행하기 위해 IBM WebSphere MQ 오브젝트에 대한 적절한 액세스 권한이 있는지 확인하십시오. 사용자의 보안 요구 사항에 맞게 이 구성을 조정할 수 있습니다.

- a. `setmqaut -m QM1 -t qmgr -g FTEAGENTS +connect +inq`
- b. `setmqaut -m QM1 -t qmgr -g FTEUSERS +connect`
- c. `setmqaut -m QM1 -n "SYSTEM.FTE" -t q -g FTEAGENTS +get +put`
- d. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT1" -t q -g FTEUSERS +put`
- e. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT1" -t q -g FTEAGENTS +setid +get +put`
- f. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT3" -t q -g FTEUSERS +put`
- g. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT3" -t q -g FTEAGENTS +setid +get +put`
- h. `setmqaut -m QM1 -n "SYSTEM.FTE.DATA.AGENT1" -t q -g FTEAGENTS +get +put`
- i. `setmqaut -m QM1 -n "SYSTEM.FTE.DATA.AGENT3" -t q -g FTEAGENTS +get +put`

```

j. setmqaut -m QM1 -n "SYSTEM.FTE.EVENT.AGENT1" -t q -g FTEAGENTS +get +put
k. setmqaut -m QM1 -n "SYSTEM.FTE.EVENT.AGENT3" -t q -g FTEAGENTS +get +put
l. setmqaut -m QM1 -n "SYSTEM.FTE.REPLY.AGENT1" -t q -g FTEAGENTS +get +put
m. setmqaut -m QM1 -n "SYSTEM.FTE.REPLY.AGENT3" -t q -g FTEAGENTS +get +put
n. setmqaut -m QM1 -n "SYSTEM.FTE.STATE.AGENT1" -t q -g FTEAGENTS +get +put
   +inq
o. setmqaut -m QM1 -n "SYSTEM.FTE.STATE.AGENT3" -t q -g FTEAGENTS +get +put
   +inq
p. setmqaut -m QM1 -n "SYSTEM.FTE" -t topic -g FTEUSERS +sub
q. setmqaut -m QM1 -n "SYSTEM.FTE" -t topic -g FTEAGENTS +pub +sub
r. setmqaut -m QM1 -n "SYSTEM.DEFAULT.MODEL.QUEUE" -t q -g FTEUSERS +dsp
   +browse +get +put
s. setmqaut -m QM1 -n "SYSTEM.DEFAULT.MODEL.QUEUE" -t q -g FTEAGENTS +dsp
   +browse +get +put

```

setmqaut 명령에 대한 자세한 정보는 **setmqaut**의 내용을 참조하십시오.

그룹에 권한을 부여하는 방법에 대한 자세한 정보는 [특정 WebSphere MQ Managed File Transfer에 해당되는 자원에 대한 그룹 권한을 참조하십시오.](#)

파일 전송용 2 컴퓨터 준비

IBM WebSphere MQ 두 컴퓨터 파일 전송 시나리오의 파일 전송에 대해 컴퓨터 2를 준비하는 것을 설명합니다.

이 태스크 정보

이 태스크에서는 컴퓨터 2가 Linux 운영 체제를 실행 중이라고 가정합니다. 이 시나리오를 실행하기 위해 다른 플랫폼을 실행 중인 경우, 적절한 플랫폼별 명령으로 대체하십시오.

프로시저

1. 각 컴퓨터에 Managed File Transfer 에이전트 컴포넌트와 함께 IBM WebSphere MQ를 설치하십시오. 이 때, 적절한 필수 컴포넌트(예: 관리 파일 전송 명령행 도구 등)를 함께 설치하십시오. 자세한 내용은 [설치 항목 선택을 참조하십시오.](#)

이 단계에서는 이전 IBM WebSphere MQ 또는 IBM WebSphere MQ File Transfer Edition이 설치되지 않은 정리된 컴퓨터를 사용한다고 가정합니다. 그렇지 않으면, 공존이 지원되는지 여부를 판별하고 환경의 구성이나 설치 메커니즘을 적절하게 조정해야 합니다. 자세한 내용은 [다중 설치를 참조하십시오.](#)

관리 파일 전송 명령행 도구 컴포넌트를 설치하면 이 시나리오에서 보듯이 1 컴퓨터 외에 2 컴퓨터로부터 전송을 정의하고 시작할 수 있습니다.

2. mqm 그룹 내의 사용자로서, 통합 관리자 1 컴퓨터에 QM1을 사용하도록 파일 전송을 구성하십시오. 다음 명령을 입력하십시오.

```
fteSetupCoordination -coordinationQMgr QM1 -coordinationQMgrHost <computer1_hostname>
-coordinationQMgrPort 1414 -coordinationQMgrChannel FTE.USER.SVRCONN
```

생성된 MQSC 스크립트는 컴퓨터 1을 구성할 때 이미 실행했으므로 따로 실행할 필요가 없습니다.

```
fteSetupCommands -connectionQMgr QM1 -connectionQMgrHost <computer1_hostname>
-connectionQMgrPort 1414 -connectionQMgrChannel FTE.USER.SVRCONN
```

자세한 정보는 다음을 참조하십시오.

- [WebSphere MQ Managed File Transfer 토폴로지 개요](#)
- [fteSetupCoordination\(통합 설정 세부사항\)](#)
- [fteSetupCommands\(command.properties 파일 작성\)](#)

- 클라이언트 연결성에 대한 구성이 올바른지 확인하기 위해 QM1에 등록된 에이전트를 나열하십시오. 다음 명령을 입력하십시오.

```
ftelistAgents
```

다음 출력이 표시됩니다.

Agent Name:	Queue Manager Name:	Status:
AGENT1	QM1	Ready

- 두 번째 파일 전송 에이전트인 AGENT3을 작성하십시오.

```
fteCreateAgent -agentName AGENT3 -agentQMgr QM1 -agentQMgrHost <computer1_hostname>
-agentQMgrPort 1414 -agentQMgrChannel FTE.AGENT.SVRCONN
```

- 컴퓨터로 전환하고 다음 명령을 입력하십시오.

```
runmqsc QM1 < <AGENT3_create.mqsc>
```

- 2 컴퓨터로 전환하고 fteagent 사용자로서 새 파일 전송 에이전트 AGENT3을 시작하십시오.

```
fteStartAgent AGENT3
```

- 선택사항: 컴퓨터 1로 전환하고 첫 번째 시나리오에서 사용되는 권한이 있는 IBM WebSphere MQ 관리자가 아닌 fteagent 사용자로 AGENT1 을 다시 시작하십시오.

- QM1에 등록된 에이전트를 나열하여 클라이언트 연결성에 대한 구성이 올바른지 확인하십시오.

```
ftelistAgents
```

다음 출력이 표시됩니다.

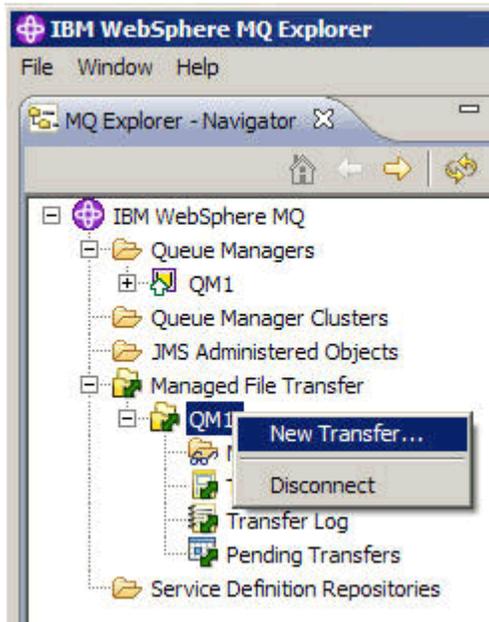
Agent Name:	Queue Manager Name:	Status:
AGENT1	QM1	READY
AGENT3	QM1	READY

솔루션 구현

이 시나리오에서 솔루션 구현은 IBM WebSphere MQ 탐색기를 사용하여 파일 전송을 정의하고 시작하는 것입니다. 전송 진행 과정을 모니터링하고 파일 시스템을 조사하여 성공했는지 확인하십시오.

프로시저

- 소스 파일 및 대상 디렉토리를 식별하십시오.
 - 1 컴퓨터에서 전송할 샘플 파일을 작성하십시오. 예를 들어, C:\start\myfile.txt입니다.
 - 2 컴퓨터에서 이 파일을 전송할 기존 디렉토리를 식별하십시오. 예를 들어, /home/end/입니다. 에이전트를 시작한 사용자가 해당 디렉토리에 대한 쓰기 액세스 권한이 있는지 확인하십시오.
- 컴퓨터 1에서 IBM WebSphere MQ 탐색기를 시작하십시오. 시작 메뉴(또는 동등한 메뉴)에서 프로그램을 시작하거나 **MQExplorer**를 실행하십시오. 자세한 정보는 [IBM WebSphere MQ Explorer 시작](#)을 참조하십시오.
- IBM WebSphere MQ 탐색기 네비게이터에서 **관리 파일 전송** 을 클릭하고 **QM**을 마우스 오른쪽 단추로 클릭하고 **새 전송** 을 선택하여 새 전송 마법사를 시작하십시오.



4. 시작 섹션에서 소스 에이전트로 **AGENT1** 을 선택하십시오.

From:

Agent: AGENT1

Type: File

File:

Include subdirectories

5. 이전에 작성한 파일의 경로를 입력하십시오. 예를 들어, C:\start\myfile.txt입니다.

From:

Agent: AGENT1

Type: File

File: C:\Users\Ben Bakowski\Desktop\FTETEST\START\myDemoTransfer.txt

Include subdirectories

6. To 섹션에서 목적지 에이전트로 **AGENT2**를 선택하십시오.

7. 이전에 식별한 대상 디렉토리를 입력하십시오. 예를 들어, /home/end/입니다.

To:

Agent: AGENT2

Type: File

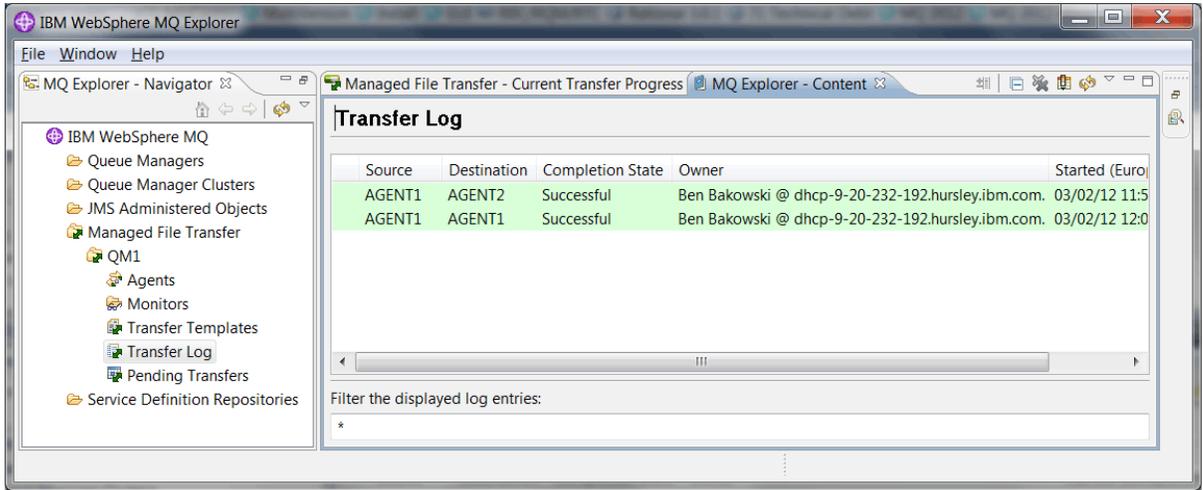
Directory: /home/ben/FTETEST/END/

File name: myDemoTransfer.txt

Overwrite files on the destination file system that have the same name

8. 마침을 클릭하고 전송을 시작하십시오.

9. IBM WebSphere MQ 탐색기의 현재 전송 진행 상태 탭에서 전송 진행 상태를 모니터링할 수 있습니다.



10. 또한 2 컴퓨터의 파일 시스템을 수동으로 검사하여 새 파일이 있는지 확인할 수 있습니다. 예를 들어, /home/end/myfile.txt입니다.

파일을 전송하기 위해 명령행 사용

이 절에서는 전송을 시작하기 위해 명령행을 사용하는 방법부터 시작하여 파일 전송의 유연성에 대해 설명합니다. 이 시나리오의 범위를 벗어나지만 이러한 원칙 및 Ant 스크립트 기술을 기반으로 하여 훨씬 강력한 파일 전송 시나리오를 정의하고 구현할 수 있습니다.

프로시저

1. 이전 데모로부터 전송된 파일을 제거하십시오. 예를 들어, /home/end/myfile.txt입니다.
2. fteuser 사용자로서 **fteCreateTransfer** 명령을 사용하여 1 컴퓨터의 C:\start\myfile.txt로부터 2 컴퓨터의 /home/end/myfile.txt로 파일 전송을 시작하십시오.

```
fteCreateTransfer -sa AGENT1 -sm QM1 -da AGENT3 -dm QM1 -w -dd "/home/end/"
"C:\start\myfile.txt"
```

3. IBM WebSphere MQ Explorer에서 파일 전송 로그를 조사하거나 수동으로 파일 시스템을 조사하여 파일 전송이 성공했는지 확인하십시오.

다음에 수행할 작업

IBM WebSphere MQ 보안에 대한 자세한 내용은 [보안](#) 을 참조하십시오.

외부 매체를 통해 파일 전송 기능을 추가적으로 탐색할 수 있습니다.

- 관리 파일 전송의 관리 부문을 제공하기 위해 감사 기능을 추가하는 방법에 대한 자세한 정보는 [162 페이지의 『관리 파일 전송에 감사 기능 추가』](#)의 내용을 참조하십시오.
- 다음에 대한 자세한 정보가 제공됩니다.
 - 트리거: 새 파일이 표시되면 이를 이동.
 - 트리거: 다중 파일의 전송을 시작하기 위해 단일 파일의 모양을 구성합니다.
 - Apache Ant를 사용한 스크립트 전송

https://www.ibm.com/developerworks/websphere/library/techarticles/1003_phillips/1003_phillips.html을 참조하십시오.

- 사용자의 요구사항에 따라 이 시나리오에서 사용된 모델과 다른 액세스 모델이 필요할 수 있습니다. 사용자의 환경에 추가적인 보안을 제공하는 방법에 대한 우수 사례는 https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html의 내용을 참조하십시오.

관리 파일 전송에 감사 기능 추가

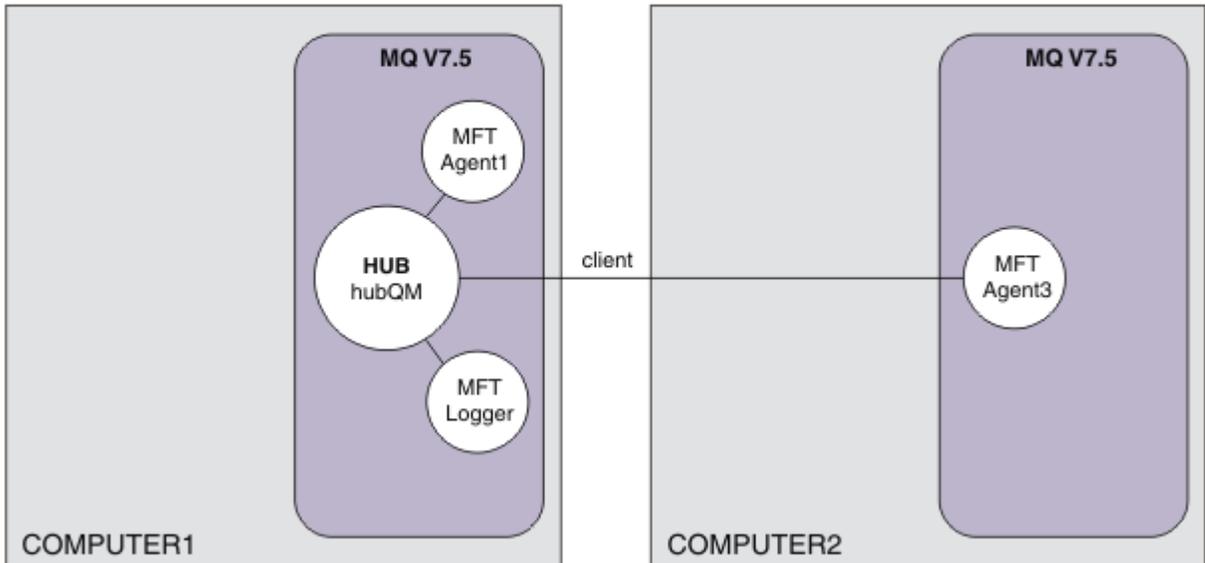
이 시나리오에서는 로거를 구성하고 이 기능을 사용하여 감사 추적을 제공하는 방법에 대해 설명합니다.

이전의 두 시나리오, 142 페이지의 『세부적인 기본 파일 전송』 및 154 페이지의 『세부적인 2 컴퓨터 파일 전송』을 통해 컴퓨터 간에 관리 파일 전송 기능 재연을 빌드했으며 파일 전송 토폴로지 구성에 익숙해졌을 것입니다. 또한 IBM WebSphere MQ Version 7.5가 파일 전송을 로그하고 감사하는 기능을 통해 관리 파일 전송의 관리 부분을 제공하는 방법을 보았을 것입니다.

154 페이지의 『세부적인 2 컴퓨터 파일 전송』 시나리오의 지시사항에 따라 두 번째 관리 파일 전송 시나리오를 완료했다고 가정합니다.

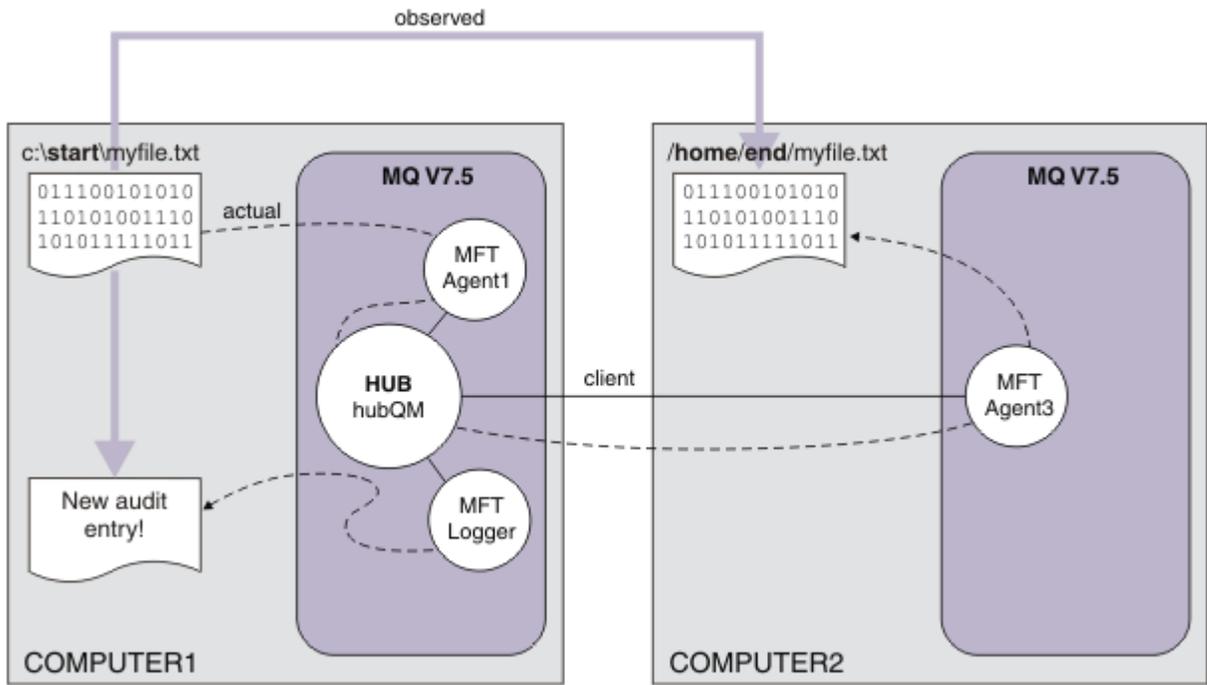
개요

이 시나리오에서는 154 페이지의 『세부적인 2 컴퓨터 파일 전송』에서 구성한 기존 Windows 및 Linux 토폴로지를 계속하고 파일 로깅 기능을 사용 가능하게 합니다. 파일 로깅 기능을 사용하기 위해 기타 컴포넌트 또는 제품을 설치할 필요는 없으므로 이 시나리오에서는 두 컴퓨터 파일 전송 시나리오로부터 라이선스 부여 또는 권한 변경을 고려할 필요가 없습니다.



확장성 및 장애 복구 등에서 프로덕션 환경에 더 적절한 데이터베이스 로거를 구현할 수도 있습니다. 그러나 기타 제품을 설치할 필요 없이 이 시나리오를 단순하게 유지하기 위해 완전히 지원되는 파일 로거를 사용하십시오. 추가적인 보안 측면은 고려하지 않습니다. 이 기능이 감사 정보를 제공하므로 보안을 강화할 필요성을 느낄 수도 있습니다.

이 시나리오에서는 파일 전송이 시작되고 세부사항은 로그 파일에 캡처됩니다.



솔루션 계획

IBM WebSphere MQ가 관리 파일 전송 시나리오에 감사 기능을 추가하기 위해 필요한 필수조건 및 라이선스 요구사항에 대해 설명합니다.

이 시나리오에서는 파일 전송을 감사하기 위한 로거를 작성합니다. 로거는 에이전트 프로세스와 대부분의 병렬을 찾으므로 기존 `ftagent` 사용자가 로거를 찾고 중지할 수 있습니다. 로거를 관리하기 위해 직접 사용자 또는 그룹을 작성할 수 있습니다.

필수조건 및 라이선스

154 페이지의 『세부적인 2 컴퓨터 파일 전송』 시나리오로부터 구성 작업을 수행해야 합니다.

솔루션 구현

IBM WebSphere MQ가 관리 파일 전송 시나리오에 감사 기능을 추가할 수 있도록 파일 로거를 구성하기 위해 1 컴퓨터를 수정합니다.

프로시저

1. 로거 프로세스를 실행할 때 `ftagent` 그룹에 IBM WebSphere MQ 오브젝트에 대한 적절한 액세스 권한이 있는지 확인하십시오. 다음 명령을 입력하십시오.

```
setmqaut -m hubQM -n "SYSTEM.FTE.LOG.RJCT.MYFILELOGGER" -t q -g FTAGENTS +put
setmqaut -m hubQM -n "SYSTEM.FTE.LOG.CMD.MYFILELOGGER" -t q -g FTAGENTS +get
```

자세한 정보는 [데이터베이스 로거에 권한을 참조하십시오](#).

2. IBM WebSphere MQ 관리자로서 파일 로거를 작성하고 로거 큐 관리자로서 `hubQM`을 사용하십시오.

```
fteCreateLogger -loggerType FILE -loggerQMgr hubQM -fileLoggerMode LINEAR -fileSize 5MB
myFileLogger
runmqsc hubQM < <MYFILELOGGER_create.mqsc>
```

자세한 정보는 [fteCreateLogger\(WebSphere MQ Managed File Transfer 로거 작성\)](#)을 참조하십시오. 프로덕션 내에서 데이터베이스 로거를 사용하는 방법을 고려하십시오.

3. ftagent 사용자로서 로거를 시작하십시오.

```
fteStartLogger MYFILELOGGER
```

4. 로거가 시작되었는지 확인하십시오.

로거가 시작되었는지 확인하려면 파일 시스템을 조사하십시오. 위의 명령을 사용하여 파일 로거를 구성한 후에는 <MQ INSTALL>/mqft/logs/hubQM/loggers/MYFILELOGGER/logs에서 로그를 찾을 수 있습니다. output0.log에 "BFGDB0023I: The logger has completed startup activities and is now running." 메시지가 포함되어 있는지 확인하십시오.

5. 이전 데모로부터 전송된 파일을 제거하십시오. 예를 들어, /home/end/myfile.txt입니다.

6. ftuser 사용자로서 1 컴퓨터에서 fteCreateTransfer(새 파일 전송 작성) 명령을 사용하여 1 컴퓨터의 C:\start\myfile.txt로부터 2 컴퓨터의 /home/end/myfile.txt로 파일 전송을 시작하십시오. 다음 명령을 입력하십시오.

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT3 -dm hubQM -w -dd "/home/end/" "C:\start\myfile.txt"
```

7. 로거가 이 전송을 캡처하는지 확인하고 로그 항목의 콘텐츠를 파악하십시오.

a. 파일 열기 <MQ INSTALL>/mqft/logs/hubQM/loggers/MYFILELOGGER/MYFILELOGGER-XXXXXXXXX.log

b. 로그 항목은 사용자가 시작한 전송을 표시하며 파일 소스 및 목적지 위치, 날짜, 시간 및 요청자 ID를 포함합니다. 예를 들면, 다음과 같습니다.

```
2012-03-23T16:42:21;414d5120514d3120202020202020202020202020207a556b4f2000aa03;[TSTR]; ;
AGENT1;hubQM;STANDARD;AGENT3;hubQM;User;;;com.ibm.wmqfte.SourceAgent=AGENT1,
com.ibm.wmqfte.DestinationAgent=AGENT3, com.ibm.wmqfte.MqmdUser=User,
com.ibm.wmqfte.OriginatingUser=User, com.ibm.wmqfte.OriginatingHost=
dhcp-9-10-123-123.hursley.ibm.com., com.ibm.wmqfte.TransferId=
414d5120514d3120202020202020202020202020207a556b4f2000aa03, com.ibm.wmqfte.Priority=0;

2012-03-23T16:42:21;414d5120514d3120202020202020202020202020207a556b4f2000aa03;[TPRO];0 ;
C:\start\myfile.txt;51447;file;leave ;;;;/home/end/myfile.txt;51447;file;
;;;;;;

2012-03-23T16:42:21;414d5120514d3120202020202020202020202020207a556b4f2000aa03;[TCOM];0 ;
AGENT1;hubQM;STANDARD;AGENT3;hubQM;STANDARD;User;;BFGRP0032I: The file transfer
request has successfully completed.;com.ibm.wmqfte.SourceAgent=AGENT1,
com.ibm.wmqfte.DestinationAgent=AGENT3, com.ibm.wmqfte.MqmdUser=User,
com.ibm.wmqfte.OriginatingUser=User, com.ibm.wmqfte.OriginatingHost=
dhcp-9-20-123-123.hursley.ibm.com.,
com.ibm.wmqfte.TransferId=414d5120514d31202020202020202020202020207a556b4f2000aa03,
com.ibm.wmqfte.Priority=0;
```

용어집

이 용어집에는 IBM WebSphere MQ의 용어 및 정의가 들어 있습니다.

다음과 같은 상호 참조가 이 용어집에서 사용됩니다.

- "참조하십시오"로 끝나는 문장은 용어에서 선호하는 동의어를 참조하거나 두문자어 또는 약어에서 전체 정의된 양식을 참조하도록 지시합니다.
- "도 참조하십시오"로 끝나는 문장은 관련되거나 대조되는 용어를 참조하도록 지시합니다.

[164 페이지의 『A』](#) [167 페이지의 『B』](#) [168 페이지의 『C』](#) [172 페이지의 『D』](#) [174 페이지의 『E』](#) [175 페이지의 『F』](#) [177 페이지의 『G』](#) [177 페이지의 『H』](#) [178 페이지의 『I』](#) [179 페이지의 『J』](#) [180 페이지의 『K』](#) [180 페이지의 『L』](#) [182 페이지의 『M』](#) [185 페이지의 『N』](#) [186 페이지의 『O』](#) [188 페이지의 『P』](#) [190 페이지의 『Q』](#) [191 페이지의 『R』](#) [194 페이지의 『S』](#) [198 페이지의 『T』](#) [200 페이지의 『U』](#) [201 페이지의 『V』](#) [201 페이지의 『W』](#) [202 페이지의 『X』](#)

A

이상종료 이유 코드(abend reason code)

z/OS 운영 체제에서 실행되는 프로그램의 문제점을 고유하게 식별하는 4바이트 16진 코드입니다.

추상 클래스(abstract class)

객체 지향 프로그래밍에서 개념을 나타내는 클래스이며 이 클래스에서 파생된 클래스는 해당 개념의 구현을 나타냅니다. 오브젝트는 추상 클래스에서 구성될 수 없습니다. 즉, 인스턴스화될 수 없습니다. 상위 클래스(parent class)도 참조하십시오.

액세스 통제(access control)

컴퓨터 보안에서 사용자가 권한이 부여된 컴퓨터 시스템의 자원에만 액세스할 수 있게 하는 프로세스입니다.

액세스 제어 목록(access control list)

컴퓨터 보안에서 오브젝트에 액세스할 수 있는 모든 주체와 이들 주체의 액세스 권한을 식별하는 오브젝트와 연관된 목록입니다.

책임성(accountability)

자신의 조치에 대해 책임을 지는 입장을 의미합니다.

ACL

액세스 제어 목록(access control list)을 참조하십시오.

활성 로그

복구 이벤트 발생 시 복구 이벤트가 기록되는 고정 크기의 데이터 세트입니다. 활성 로그가 가득 차면 활성 로그의 콘텐츠가 아카이브 로그에 복사됩니다.

활성 큐 관리자 인스턴스(active queue manager instance)

요청을 처리 중인 실행 중인 다중 인스턴스 큐 관리자의 인스턴스입니다. 다중 인스턴스 큐 관리자의 활성 인스턴스는 하나만 있습니다.

어댑터(adapter)

다른 두 소프트웨어 컴포넌트가 서로 통신할 수 있게 하는 중개 소프트웨어 컴포넌트입니다.

주소 공간(ASID, address space)

컴퓨터 프로그램 또는 프로세스에 사용할 수 있는 주소의 범위입니다. 주소 공간은 물리적 스토리지, 가상 스토리지 또는 둘 다를 나타낼 수 있습니다. 연결 주소 공간(allied address space), 버퍼 풀(buffer pool)도 참조하십시오.

관리 백(administration bag)

WebSphere MQ 관리 인터페이스(MQAI)에서 메시지 내의 선택자를 확인하고 목록을 작성하며 데이터 항목의 순서를 변경할 수 있음을 암시함으로써 WebSphere MQ를 관리하기 위해 작성된 데이터 백의 유형입니다.

관리 토픽 오브젝트(administrative topic object)

토픽에 기본이 아닌 특정 속성을 지정할 수 있게 하는 오브젝트입니다.

관리자 명령(administrator command)

큐, 프로세스 및 이름 목록과 같은 WebSphere MQ 오브젝트를 관리하기 위해 사용하는 명령.

고급 프로그램 간 통신(APPC, Advanced Program-to-Program Communication)

상호 연결된 시스템이 프로그램 처리를 통신하고 공유할 수 있게 하는 SNA LU 6.2 프로토콜의 구현입니다.

고급 텔레메트리 클라이언트(advanced telemetry client)

텔레메트리 고급 클라이언트를 참조하십시오.

연관관계(affinity)

각자에 대해 일부 관계 또는 종속성을 보유하는 오브젝트 간 연관입니다.

경보(alert)

이벤트나 임박한 이벤트를 알리는 메시지 또는 기타 표시입니다.

경보 모니터(alert monitor)

z/OS용 WebSphere MQ에서 z/OS용 WebSphere MQ에 대한 연결 요청의 결과로 발생하는 스케줄되지 않은 이벤트를 처리하는 CICS 어댑터의 컴포넌트입니다.

알리어스 큐

WebSphere MQ 오브젝트이며, 해당 이름은 로컬 큐 관리자에 정의된 기본 큐 또는 주제의 알리어스입니다. 애플리케이션 또는 큐 관리자가 알리어스 큐를 사용하면 알리어스 이름이 해석되고 요청된 조작이 연관된 기본 오브젝트에 대해 수행됩니다.

알리어스 큐 오브젝트(alias queue object)

WebSphere MQ 오브젝트이며, 해당 이름은 로컬 큐 관리자에 정의된 기본 큐의 알리어스입니다. 애플리케이션 또는 큐 관리자가 알리어스 큐를 사용하면 알리어스 이름이 해석되고 요청된 조작이 연관된 기본 큐에 대해 수행됩니다.

연결 주소 공간(allied address space)

z/OS용 WebSphere MQ에 연결되는 z/OS 주소 공간입니다.

연결 주소 공간(ALLY)

연결 주소 공간(allied address space)을 참고하십시오.

대체 사용자 권한(alternate user authority)

보안 검사를 위해 다른 사용자 ID를 제공하는 사용자 ID의 기능입니다. 애플리케이션이 WebSphere MQ 오브젝트를 열면, 이는 애플리케이션과 연관된 사용자 ID 대신 권한 확인을 위해 큐 관리자가 사용하는 MQOPEN, MQPUT1 또는 MQSUB 호출의 사용자 ID를 제공할 수 있습니다.

대체 사용자 보안(alternate user security)

z/OS에서 WebSphere MQ 오브젝트를 여는 도중 애플리케이션이 대체 사용자 권한을 요청할 때 수행되는 권한 검사입니다.

APAR

권한 부여된 프로그램 분석 보고서(APAR, authorized program analysis report)를 참고하십시오.

APF

권한 부여된 프로그램 기능(APF, authorized program facility)을 참고하십시오.

API 교차 엑시트(API-crossing exit)

개념상 API 엑시트와 비슷한 사용자 작성 프로그램입니다. z/OS용 WebSphere MQ에서 CICS 애플리케이션에서만 지원됩니다.

API 엑시트(API exit)

MQI 호출의 기능을 모니터하거나 수정하는 사용자 작성 프로그램입니다. 애플리케이션이 발행하는 각 MQI 호출마다, API 엑시트는 큐 관리자가 호출의 처리를 시작하기 전에 호출되며 큐 관리자가 호출의 처리를 완료한 이후에 다시 호출됩니다. API 엑시트는 MQI 호출의 모든 매개변수를 조사하고 수정할 수 있습니다.

APPC

고급 프로그램 간 통신(APPC, Advanced Program-to-Program Communication)을 참고하십시오.

애플리케이션 정의 형식(application-defined format)

사용자 애플리케이션이 의미를 정의하는 메시지의 애플리케이션 데이터입니다. 내장 형식(built-in format)도 참조하십시오.

애플리케이션 환경(application environment)

소프트웨어와 소프트웨어가 지원하는 서버 또는 네트워크 인프라가 포함된 환경입니다.

애플리케이션 레벨 보안

애플리케이션이 MQI 호출을 발행할 때 호출되는 보안 서비스입니다.

애플리케이션 로그(application log)

Windows 시스템에서 중요한 애플리케이션 이벤트를 기록하는 로그입니다.

애플리케이션 큐(application queue)

트리거가 설정되어 있고 트리거 조건이 충족된 경우에 트리거 메시지의 기록을 요구하는 로컬 큐입니다.

아카이브 로그(archive log)

활성 로그의 크기가 한계에 도달하면 WebSphere MQ가 각 활성 로그 데이터 세트의 콘텐츠를 복사하는 대 상 스토리지 디바이스의 데이터 세트입니다. 복구 로그(recovery log)도 참조하십시오.

ARM

자동 재시작 관리자(ARM, automatic restart manager)를 참고하십시오.

ASID

주소 공간을 참조하십시오.

비대칭 키 암호화(asymmetric key cryptography)

모든 사용자에게 알려진 공개 키와 메시지의 수신자 또는 송신자에게만 알려진 개인 키라는 두 개의 키를 사용하는 암호화 시스템입니다. 대칭 키 암호화(symmetric key cryptography)도 참조하십시오.

비동기 이용(asynchronous consumption)

애플리케이션이 큐 세트로부터의 메시지를 이용할 수 있게 하는 MQI 호출 세트를 사용하는 프로세스입니다. 애플리케이션이 식별한 코드 단위를 호출하고 메시지 또는 메시지를 표시하는 토큰을 전달하여 애플리케이션으로 메시지를 전달합니다.

비동기 메시징(asynchronous messaging)

프로그램이 메시지 큐에 메시지를 배치한 후 메시지에 대한 응답을 기다리지 않고 자체 처리를 진행하는 프로그램 간 통신 메소드입니다. [동기 메시징](#)도 참조하십시오.

비동기 Put

큐 관리자로부터의 응답을 기다리지 않고 애플리케이션이 메시지를 넣는 것입니다.

속성

- 객체 지향 프로그래밍에서 다른 특성과 분명하게 구분될 수 있는 오브젝트 또는 클래스의 특성입니다. 속성은 상태 정보에 대해 설명하기도 합니다.
- 엔티티를 설명하는 엔티티의 특성입니다. 예를 들어, 직원의 전화 번호는 직원 속성 중 하나입니다. [엔티티\(entity\)](#)도 참조하십시오.

인증

컴퓨터 시스템의 사용자가 본인임을 증명하는 증거를 제공하는 보안 서비스입니다. 이 서비스를 구현하는 일반적인 메커니즘은 비밀번호와 디지털 서명입니다.

인증 정보 오브젝트(authentication information object)

SSL(Secure Sockets Layer) 보안을 지원하기 위해 LDAP 서버를 사용하여 인증서 폐기 목록(CRL)을 검사하는 데 필요한 정의를 제공하는 오브젝트입니다.

권한 검사(authority check)

[권한 검사\(authorization check\)](#)를 참고하십시오.

권한 부여

사용자, 시스템 또는 프로세스에 오브젝트, 자원 또는 기능에 대한 완전하거나 제한된 액세스 권한을 부여하는 프로세스입니다.

권한 검사(authorization check)

사용자 또는 애플리케이션이 시스템 자원에 액세스하려고 할 때 수행되는 보안 검사입니다. 예를 들어 관리자가 WebSphere MQ 관리 명령을 실행하려고 하거나 애플리케이션에서 큐 관리자에 연결하려고 하는 경우입니다.

권한 파일(authorization file)

오브젝트, 오브젝트의 클래스 또는 오브젝트의 모든 클래스에 대한 보안 정의를 제공하는 파일입니다.

권한 서비스

UNIX 및 Linux 시스템의 WebSphere MQ 및 WebSphere MQ for Windows에서 명령 또는 호출과 연관된 사용자 ID에 대한 명령 및 MQI 호출의 권한 점검을 제공하는 서비스.

권한 부여된 프로그램 분석 보고서(APAR, authorized program analysis report)

IBM에서 제공하는 프로그램의 지원 릴리스에서 발견된 결함에 대한 정정 요청입니다.

권한 부여된 프로그램 기능(APF, authorized program facility)

z/OS 환경에서 제한된 기능을 사용하도록 권한이 부여된 프로그램의 식별을 허용하는 기능입니다.

자동 재시작 관리자(ARM, automatic restart manager)

배치 작업과 시작된 태스크 또는 이를 실행하는 시스템이 예기치 않게 종료된 후에 배치 작업과 시작된 태스크를 자동으로 재시작할 수 있는 z/OS 복구 기능입니다.

B

백아웃(backout)

현재 작업 단위 동안 작성된 자원에 대한 모든 변경사항을 되돌리는 조작입니다. [커미트\(commit\)](#)도 참조하십시오.

가방

[데이터 백\(data bag\)](#)을 참고하십시오.

막대(bar)

64비트 시스템에서는 2GB로 설정된 z/OS 메모리 한계입니다. 막대는 2GB 주소 미만의 스토리지를 2GB 주소 초과 스토리지와 분리시킵니다. 막대 위의 영역은 데이터를 위한 영역이므로 막대 위에서는 프로그램이 실행되지 않습니다.

기본 맵핑 지원(BMS, basic mapping support)

입력 및 출력 표시 데이터를 형식화하며 여러 터미널에서 사용하는 제어 문자와 상관없이 다중 페이지 출력 메시지를 라우팅하는 CICS 및 애플리케이션 프로그램 간의 인터페이스입니다.

작동(behavior)

객체 지향 프로그래밍에서 메소드에서 구현된 기능입니다.

BMS

[기본 맵핑 지원\(BMS, basic mapping support\)](#)을 참고하십시오.

부치 방법론(Booch methodology)

사용자가 객체 지향 패러다임을 사용하여 시스템을 설계할 수 있도록 도움을 주는 객체 지향 방법론입니다.

부트스트랩 데이터 세트(BSDS)

z/OS용 WebSphere MQ에 알려진 모든 활성 및 아카이브 로그 데이터 세트의 재고와 모든 최근 z/OS용 WebSphere MQ 활동의 랩어라운드 재고가 포함된 VSAM 데이터 세트입니다. z/OS용 WebSphere MQ 서버 시스템을 다시 시작하는 데 BSDS가 필요합니다.

찾아보기

메시지 큐잉에서 메시지를 큐에서 제거하지 않고 복사하는 것입니다. [가져오기\(get\)](#), [넣기\(put\)](#)도 참조하십시오.

찾아보기 커서(browse cursor)

메시지 큐잉에서 다음 순서의 메시지를 식별하기 위해 큐를 찾아볼 때 사용되는 표시기입니다.

BSDS

[부트스트랩 데이터 세트\(BSDS, bootstrap data set\)](#)를 참고하십시오.

버퍼 풀

데이터 페이지를 읽어 오고 처리 중에 데이터 페이지가 수정되고 보유되는 메모리의 영역입니다. [주소 공간](#)도 참조하십시오.

내장 형식(built-in format)

큐 관리자가 의미를 정의하는 메시지의 애플리케이션 데이터입니다. [애플리케이션 정의 형식\(application-defined format\)](#)도 참조하십시오.

C

CA

[인증 기관\(CA, certificate authority\)](#)을 참고하십시오.

CAF

[클라이언트 첨부 기능\(CAF, client attachment feature\)](#)을 참고하십시오.

콜백(callback)

메시지 이용자 또는 이벤트 핸들러 루틴입니다.

CCDT

[클라이언트 채널 정의 테이블\(CCDT, client channel definition table\)](#)을 참고하십시오.

CCF

[채널 제어 기능\(CCF, channel control function\)](#)을 참고하십시오.

CCSID

[코드화 문자 세트 ID\(CCSID, coded character set identifier\)](#)를 참고하십시오.

CDF

[채널 정의 파일\(CDF, channel definition file\)](#)을 참고하십시오.

인증 기관(CA, certificate authority)

인증서 서명 요청에 응답하여 디지털 인증서를 발행하는 신뢰하는 써드파티 조직이나 회사입니다. 인증 기관에서는 고유한 인증서를 발급 받은 개인의 ID를 확인합니다. [SSL\(Secure Sockets Layer\)](#)도 참조하십시오.

인증서 체인(certificate chain)

개인 인증서로 시작하여 체인 맨 위의 루트로 끝나는, 암호를 사용하여 서로 관련된 인증서의 계층입니다.

인증서 만기(certificate expiration)

디지털 인증서에는 인증서가 유효한 날짜 범위가 포함되어 있습니다. 유효한 날짜 범위를 벗어나면 인증서가 "만료"되었다고 합니다.

인증서 폐기 목록(CRL, certificate revocation list)

스케줄된 만기 날짜 전에 폐기된 인증서의 목록입니다. 인증서 폐기 목록은 인증 기관에 의해 관리되고 SSL(Secure Sockets Layer) 데이터 교환 중에 사용되어 관련된 인증서가 폐기되지 않았는지 확인합니다.

인증서 저장소

키 저장소의 Windows 이름입니다.

인증서 서명 요청(CSR, certificate signing request)

유틸리티 또는 조직의 공개 키와 주제 식별 이름이 포함된 요청입니다. CA가 해당 유틸리티에 디지털 서명을 발행할 수 있도록 CA에 전송됩니다.

CF

커플링 기능(CF, coupling facility)을 참고하십시오.

CFSTRUCT

큐 관리자가 사용하는 커플링 기능 목록 구조를 설명하기 위해 사용되는 WebSphere MQ 오브젝트입니다.

채널

두 큐 관리자 간에(메시지 채널) 또는 클라이언트와 큐 관리자 간에(MQI 채널) 통신 링크를 정의하는 WebSphere MQ 오브젝트입니다. 메시지 채널, MQI 채널도 참조하십시오.

채널 콜백(channel callback)

채널 연결이 올바른 시스템에 대해 설정되었는지 확인하는 메커니즘입니다. 채널 콜백에서 송신자 채널은 송신자의 정의를 사용하여 원래 요청자 채널을 콜백합니다.

채널 제어 기능(CCF, channel control function)

채널의 설정 및 제어를 허용하기 위해 운영자 패널 인터페이스와 함께 메시지를 전송 큐에서 통신 링크로 이동하고 통신 링크에서 로컬 큐로 이동하는 프로그램입니다.

채널 정의 파일(CDF, channel definition file)

전송 큐와 통신 링크를 연관시키는 통신 채널 정의가 포함된 파일입니다.

채널 이벤트(channel event)

채널 인스턴스가 시작되거나 중지되는 시기와 같이 채널 조작 중에 감지된 조건을 보고하는 이벤트입니다. 채널 이벤트는 채널의 양 끝에 있는 큐 관리자에서 생성됩니다.

채널 엑시트 프로그램(channel exit program)

메시지 채널 에이전트(MCA)의 처리 순서에서 정의된 수의 위치 중 하나에서 호출되는 사용자 작성 프로그램입니다.

채널 시작기

이니시에이션 큐를 모니터링하여 트리거링 기준이 충족되는 시점을 확인한 후에 전송자 채널을 시작하는 WebSphere MQ 분산 큐잉 컴포넌트입니다.

채널 리스너

네트워크에서 시작 요청을 모니터링한 후에 수신 채널을 시작하는 WebSphere MQ 분산 큐잉의 컴포넌트입니다.

체크포인트

프로그램에서 검사가 수행되거나 인터럽트 발생 시 프로그램이 재시작할 수 있도록 데이터 기록이 수행되는 위치입니다.

CI

제어 간격(CI, control interval)을 참고하십시오.

CipherSpec

인증이 완료된 후 SSL 메시지에 적용된 암호화 알고리즘과 해시 기능의 결합입니다.

암호 스위트(cipher suite)

안전한 데이터 교환을 위해 사용되는 인증, 키 교환 알고리즘 및 SSL(Secure Sockets Layer) 암호 스펙의 결합입니다.

암호문(ciphertext)

암호화된 데이터입니다. 키를 사용하여 일반 텍스트로 변환(복호화)할 때까지는 암호문을 읽을 수 없습니다. 일반 텍스트도 참조하십시오.

순환 로깅(circular logging)

UNIX 및 Linux 시스템의 WebSphere MQ 및 WebSphere MQ for Windows에서 로그 파일 링에 모든 재시작 데이터를 보존하는 프로세스. 선형 로깅(linear logging)도 참조하십시오.

CL

명령 언어(CL, Command Language)를 참고하십시오.

class

객체 지향 설계 또는 프로그래밍에서 공통 정의와 공통 특성, 조작 및 작동을 사용하여 오브젝트를 작성하는데 사용할 수 있는 모델 또는 템플릿입니다. 오브젝트는 클래스의 인스턴스입니다.

클래스 계층(class hierarchy)

단일 상속을 공유하는 클래스 사이의 관계입니다.

클래스 라이브러리(class library)

객체 지향 프로그래밍에서 사전 작성된 클래스 또는 코드화된 템플릿의 컬렉션이며 이들은 애플리케이션을 개발할 때 프로그래머가 지정하고 사용할 수 있습니다.

일반 텍스트(cleartext)

읽을 수 있는 양식으로 네트워크를 통해 송신된 문자열입니다. 압축할 목적으로 인코딩할 수는 있지만 디코딩하기는 쉽지 않습니다. 암호문(ciphertext)도 참조하십시오.

클라이언트

로컬 사용자 애플리케이션을 위해 서버의 큐잉 서비스에 대한 액세스를 제공하는 런타임 컴포넌트입니다. 애플리케이션이 사용하는 큐는 서버에 상주합니다. WebSphere MQ MQI 클라이언트, WebSphere MQ Java 클라이언트, WebSphere MQ 완전 관리 .NET 클라이언트도 참조하십시오.

클라이언트 애플리케이션(client application)

워크스테이션에서 실행되는 애플리케이션으로 서버의 큐잉 서비스에 대한 애플리케이션 액세스를 제공하는 클라이언트에 링크되어 있습니다.

클라이언트 첨부 기능(CAF, Client Attachment feature)

z/OS에 대한 클라이언트 첨부을 지원하는 옵션입니다.

클라이언트 채널 정의 테이블(CCDT, client channel definition table)

하나 이상의 클라이언트 연결 채널 정의가 포함된 파일입니다.

클라이언트 연결 채널 유형(client-connection channel type)

WebSphere MQ 클라이언트와 연관된 MQI 채널 정의의 유형입니다. 서버 연결 채널 유형(server-connection channel type)도 참조하십시오.

CLUSRCVR

클러스터 수신자 채널(CLUSRCVR, cluster-receiver channel)을 참고하십시오.

CLUSSDR

클러스터 송신자 채널(CLUSSDR, cluster-sender channel)을 참고하십시오.

클러스터

WebSphere MQ에서 하나 이상의 컴퓨터에 있는 둘 이상의 큐 관리자의 그룹으로서, 자동 상호연결을 제공하며 로드 밸런싱과 중복성을 위해 서로 간에 큐 및 토픽을 알릴 수 있도록 허용합니다.

클러스터 큐

클러스터 큐 관리자가 호스팅하는 로컬 큐이며 클러스터에 있는 큐 관리자에 연결된 애플리케이션에서 제공하는 메시지의 대상으로 정의됩니다. 메시지를 검색하는 모든 애플리케이션은 로컬로 연결되어야 합니다.

클러스터 큐 관리자(cluster queue manager)

클러스터의 멤버인 큐 관리자입니다. 큐 관리자는 둘 이상의 클러스터의 멤버가 될 수 있습니다.

클러스터 수신자 채널(CLUSRCVR, cluster-receiver channel)

클러스터 큐 관리자가 클러스터의 다른 큐 관리자로부터 메시지를 수신하고 저장소 큐 관리자로부터 클러스터 정보를 수신할 수 있는 채널입니다.

클러스터 송신자 채널(CLUSSDR, cluster-sender channel)

클러스터 큐 관리자가 클러스터의 다른 큐 관리자에게 메시지를 송신하고 저장소 큐 관리자에게 클러스터 정보를 송신할 수 있는 채널입니다.

클러스터 토픽(cluster topic)

클러스터 큐 관리자에 정의되어 있으며 클러스터의 다른 큐 관리자에 사용 가능한 관리 토픽입니다.

클러스터 전송 큐(cluster transmission queue)

동일한 클러스터에 있는 다른 큐 관리자를 대상으로 하는 큐 관리자의 모든 메시지를 보유하는 전송 큐입니다. 이 큐를 SYSTEM.CLUSTER.TRANSMIT.QUEUE라고 합니다.

CMS 키 데이터베이스(CMS key database)

CMS키 데이터베이스는 Windows 시스템, UNIX 시스템, Linux 및 해당 플랫폼의 클라이언트에서 지원하는 데이터베이스의 형식입니다. .kdb로 끝나는 파일은 CMS 형식입니다. .kdb 파일에는 인증서와 키가 있습니다.

CCSID(coded character set identifier)

코드화된 그래픽 문자 표현을 고유하게 식별하는 특정 인코딩 설계 ID, 문자 세트 ID, 코드 페이지 ID 및 기타 정보 세트가 포함된 16비트 숫자입니다.

공존(coexistence)

둘 이상의 서로 다른 WebSphere MQ 버전이 동일한 컴퓨터에서 작동할 수 있는 기능입니다.

명령

조치를 시작하거나 서비스를 시작하는 데 사용되는 명령문입니다. 명령은 명령어 약어와 해당 매개변수 및 플래그(해당되는 경우)로 구성됩니다.

명령 백(command bag)

MQAI에서 WebSphere MQ 오브젝트의 관리를 위해 작성된 백의 유형이지만 데이터 항목의 순서를 변경하거나 메시지 내에서 목록을 작성할 수 없습니다.

명령 이벤트(command event)

MQSC 또는 PCF 명령이 정상적으로 실행되었다는 알림입니다.

명령 언어(CL, Command Language)

iSeries용 WebSphere MQ에서 CL 프로그램을 작성하거나 명령행을 사용하여 명령을 실행하는 데 사용할 수 있는 언어입니다.

명령 접두부(command prefix)

1. 1 - 8자의 명령 ID입니다. 명령 접두부는 명령을 z/OS가 아닌 애플리케이션 또는 서브시스템에 속한 것으로 구별합니다.

2. z/OS용 WebSphere MQ에서 z/OS용 WebSphere MQ 명령이 전달되며 z/OS용 WebSphere MQ 운영자 메시지를 수신하는 큐 관리자를 식별하는 문자열.

명령 서버

시스템 명령 입력 큐에서 명령을 읽어서 확인하고 유효한 명령을 명령 프로세서에 전달하는 WebSphere MQ 컴포넌트입니다.

커미트

현재 복구 단위(UR) 또는 작업 단위(UOW) 동안 작성된 모든 변경사항을 적용하는 것입니다. 조작이 완료되면 새 UR 또는 UOW를 시작할 수 있습니다.

공용 이름(CN, common name)

일반적으로 인증서 소유자와 연관된 이름을 나타내는 X.509 인증서의 식별 이름(DN) 속성에 있는 컴포넌트입니다. 사람의 경우 CN은 일반적으로 실제 이름입니다. 웹 서버의 경우 CN은 서버의 완전한 호스트 및 도메인 이름입니다. WebSphere MQ의 경우에는 이 필드에 대한 특정 요구사항이 없지만, 다수의 관리자는 큐 관리자의 이름을 사용합니다.

식별 이름도 참조하십시오.

완료 코드

메시지 큐 인터페이스(MQI) 호출이 종료된 방식을 표시하는 리턴 코드입니다.

기밀성(confidentiality)

중요한 정보가 비정상적으로 노출되지 않도록 보호해 주는 보안 서비스입니다. 이 서비스를 구현하는 일반적인 메커니즘은 암호화입니다.

구성 이벤트(configuration event)

오브젝트의 속성에 대한 알림입니다. 오브젝트가 작성되거나 변경되거나 삭제되는 경우와 명확한 요청에 의해서도 알림이 생성됩니다.

연결 연관관계(connection affinity)

다중 연결을 사용할 수 있는 경우 클라이언트 애플리케이션이 큐 관리자에 연결하는 데 사용하는 클라이언트 채널 정의를 지정하는 채널 속성입니다.

연결 팩토리

자바 EE 컴포넌트가 자원에 액세스할 수 있도록 하는 연결을 생성하는 구성 값 세트. 연결 팩토리는 애플리케이션에서 EIS(Enterprise Information System)로의 요청 시 연결을 제공하고 애플리케이션 서버가 분산 트랜잭션에서 EIS를 등록할 수 있게 합니다.

연결 핸들

프로그램이 연결된 큐 관리자에 액세스하는 데 사용하는 ID 또는 토큰입니다.

구성자(creator)

객체 지향 프로그래밍에서 오브젝트를 초기화하는 데 사용되는 특수 메소드입니다.

이용(consume)

큐에서 메시지를 제거하고 해당 콘텐츠를 호출하는 애플리케이션에 리턴하는 것입니다.

이용자(consumer)

메시지를 수신하고 처리하는 애플리케이션입니다. [메시지 이용자](#)도 참조하십시오.

컨텍스트 보안

z/OS에서 애플리케이션이 큐를 열어서 큐에 넣는 메시지의 컨텍스트를 설정하거나 받은 메시지의 컨텍스트를 큐에 넣는 메시지로 전달함을 지정할 때 수행되는 권한 검사입니다.

제어 명령(control command)

UNIX 및 Linux 시스템의 WebSphere MQ 및 WebSphere MQ for Windows의 경우, 운영 체제 명령행에서 대화식으로 입력할 수 있습니다. 이 명령에서는 WebSphere MQ 제품의 설치만 필요하며, 특수 유틸리티나 프로그램이 이를 실행할 필요가 없습니다.

제어 간격(CI, control interval)

VSAM이 레코드를 저장하고 분산된 여유 공간을 작성하는 직접 액세스 스토리지의 고정 길이 영역입니다. 제어 간격은 VSAM이 직접 액세스 스토리지와 전송을 주고 받는 정보 단위입니다. 제어 간격에는 항상 정수의 물리적 레코드가 포함되어 있습니다.

제어된 종료(controlled shutdown)

정상 종료(quiesced shutdown)를 참고하십시오.

상관 ID

관련 메시지를 식별하는 수단을 제공하는 메시지의 필드입니다. 상관 ID는 예를 들어, 요청 메시지를 해당 응답 메시지와 일치시키는 데 사용됩니다.

커플링 기능(CF)

SYSPLEX에서 고속 캐싱, 목록 처리 및 잠금 기능을 제공하는 특수 논리 파티션입니다.

CPF

명령 접두부(CPF, command prefix)를 참고하십시오.

인증서 요청(CR, certificate request)

[인증서 서명 요청\(certificate signing request\)](#)의 동의어입니다.

CRL

[인증서 폐기 목록\(CRL, certificate revocation list\)](#)을 참고하십시오.

교차 시스템 커플링 기능(XCF, cross-system coupling facility)

SYSPLEX 내에서 실행되는 권한 부여된 프로그램 간의 협력을 지원하는 기능을 제공하는 컴포넌트입니다.

암호화(cryptography)

암호문이라고 하는 판독 불가능한 형식으로 정보를 변환(암호화)하여 정보를 보호함을 의미합니다. 비밀 키를 소유한 사람만 해당 메시지를 일반 텍스트로 해독(복호화)할 수 있습니다.

D

DAE

[덤프 분석 및 제거\(DAE, dump analysis and elimination\)](#)를 참고하십시오.

daemon

네트워크 제어와 같이 지속적이거나 주기적인 기능을 수행하기 위해 자동으로 실행되는 프로그램입니다.

데이터 백(data bag)

MQA가 큐 관리자 관리에 사용하는 오브젝트 특성의 컨테이너입니다. 사용자(사용자 데이터용), 관리자(가정 옵션을 포함한 관리용) 및 명령(가정 옵션을 포함하지 않은 관리용)이라는 세 가지 유형의 데이터 백이 있습니다.

데이터 변환 인터페이스(DCI, data-conversion interface)

서로 다른 시스템 인코딩 및 CCSID 간에 애플리케이션 데이터를 변환하는 고객 또는 벤더 작성 프로그램이 따라야 하는 WebSphere MQ 인터페이스입니다. WebSphere MQ 프레임워크의 일부입니다.

데이터 변환 서비스(data-conversion service)

다른 플랫폼의 애플리케이션이 요구하는 인코딩 및 문자 세트로 애플리케이션 데이터를 변환하는 서비스입니다.

데이터그램(datagram)

애플리케이션이 메시지를 송신하지만 응답은 요구하지 않는 비동기 메시징의 양식입니다. 요청/응답도 참조하십시오.

데이터 무결성

권한 없는 데이터 수정 또는 도용이 있었는지 여부를 감지하는 보안 서비스입니다. 이 서비스는 데이터가 수정되었는지 여부만 감지하므로 데이터가 수정된 경우에도 데이터를 원래 상태로 복원하지는 않습니다.

데이터 항목(data item)

MQAI에서 데이터 백 안에 있는 항목입니다. 정수 항목, 문자열 항목, 사용자 항목 또는 시스템 항목일 수 있습니다.

DCE

분산 컴퓨팅 환경(DCE, Distributed Computing Environment)을 참고하십시오.

DCE 프린시펄(DCE principal)

분산 컴퓨팅 환경을 사용하는 사용자 ID입니다.

DCI

데이터 변환 인터페이스(DCI, data-conversion interface)를 참고하십시오.

DCM

디지털 인증 관리자(DCM, Digital Certificate Manager)를 참고하십시오.

데드-레터 큐(DLQ, dead-letter queue)

큐 관리자 또는 애플리케이션이 올바른 목적지로 전달할 수 없는 메시지를 송신하는 큐입니다.

데드-레터 큐 핸들러

사용자 작성 규칙 테이블에 따라 데드-레터 큐(DLQ)를 모니터하고 큐에서 메시지를 처리하는 유틸리티입니다. WebSphere MQ가 샘플 데드레터 큐 핸들러를 제공합니다.

복호화(decryption)

비밀 형식으로 암호화된 데이터를 디코딩하는 프로세스입니다. 복호화를 사용하려면 보안 키 또는 비밀번호가 필요합니다.

기본 오브젝트(default object)

모든 속성이 정의된 오브젝트(예: 큐)의 정의입니다. 사용자가 오브젝트를 정의하지만 해당 오브젝트의 가능한 모든 속성을 지정하지 않는 경우 큐 관리자는 지정되지 않은 속성 대신 기본 속성을 사용합니다.

지연된 연결(deferred connection)

시작하기 전에 CICS 서브시스템에서 z/OS용 WebSphere MQ에 연결하려고 하는 경우 활성화되는 보류 이벤트입니다.

파생(derivation)

객체 지향 프로그래밍에서 한 클래스에서 다른 클래스로의 세분화 또는 확장입니다.

대상

1. JMS에서 메시지를 송수신해야 하는 위치와 방법을 지정하는 오브젝트입니다.
2. 큐 또는 주제와 같이 메시지가 전송되는 엔드 포인트입니다.

Diffie-Hellman 키 교환(Diffie-Hellman key exchange)

비보안 채널을 통해 공유 시크릿을 안전하게 설정하는 데 사용되는 공용 키 교환 알고리즘입니다.

디지털 인증서(digital certificate)

개인, 시스템, 서버, 회사 또는 몇몇 기타 엔티티를 식별하고 공개 키와 엔티티를 연관시키는 데 사용되는 전자 문서입니다. 디지털 인증서는 인증 기관에 의해 발행되며 해당 기관에 의해 디지털 방식으로 서명됩니다.

디지털 인증 관리자(DCM, Digital Certificate Manager)

IBM i 시스템에서 디지털 인증서를 관리하고 iSeries 서버의 보안 애플리케이션에서 이를 사용하는 메소드입니다. 디지털 인증 관리자는 인증 기관(CA) 또는 기타 써드파티로부터 디지털 인증서를 요청하여 처리합니다.

디지털 서명(digital signature)

수신인에게 메시지 또는 오브젝트의 무결성 및 진정성을 보장하기 위해 개인 키로 암호화되어 메시지 또는 오브젝트에 추가되는 정보입니다. 디지털 서명은 개인 키 또는 공유 비밀 대칭 키를 소유하거나 이에 대한 액세스 권한을 가진 엔티티에 의해 메시지 또는 오브젝트가 서명되었음을 증명합니다.

연결 끊기(disconnect)

애플리케이션과 큐 관리자 사이의 연결을 끊는 것입니다.

식별 이름(DN)

디지털 인증서에서 엔티티를 고유하게 식별하는 이름-값 쌍의 세트입니다(예: CN=개인 이름 및 C=국가). 참고로 식별 이름은 해당 인증 기관의 네임스페이스 내에서만 고유합니다. 식별 이름이 동일한 인증서를 서로 다른 인증 기관이 발행할 수 있습니다. 따라서 가급적 키 저장소에 최소한의 신뢰 루트 CA 인증서만 두십시오. 사실 하나만 있는 게 좋습니다. 인증 기관, 디지털 인증서, X509도 참조하십시오.

분산 애플리케이션(distributed application)

메시지 큐잉에서 각각 다른 큐 관리자에 연결될 수 있지만 총체적으로 하나의 애플리케이션을 구성할 수 있는 애플리케이션 프로그램 세트입니다.

분산 컴퓨팅 환경(DCE, Distributed Computing Environment)

네트워크 컴퓨팅에서 이기종 운영 체제 및 네트워크에서 분산 애플리케이션의 작성, 사용 및 유지보수를 지원하는 서비스 및 도구 세트입니다.

분산 큐 관리(distributed queue management)

메시지 큐잉에서 다른 시스템의 큐 관리자에 대한 메시지 채널의 설정 및 제어입니다.

분배 목록(distribution list)

단일 명령문을 사용하여 메시지를 넣을 수 있는 큐의 목록입니다.

DLQ

데드-레터 큐(DLQ, dead-letter queue)를 참조하십시오.

DN

식별 이름(DN, distinguished name)을 참조하십시오.

이중 로깅

z/OS용 WebSphere MQ 활동을 기록하는 메소드로서, 각각의 변경사항은 2개의 데이터 세트에 기록됩니다. 따라서 다시 시작이 필요하며 하나의 데이터 세트를 읽을 수 없으면 다른 데이터 세트를 사용할 수 있습니다. 단일 로깅도 참조하십시오.

이중 모드(dual mode)

이중 로깅(dual logging)을 참조하십시오.

덤프 분석 및 제거(DAE, dump analysis and elimination)

이전에 작성된 덤프를 복제하기 때문에 설치에서 필요하지 않은 SVC 덤프 및 ABEND SYSUDUMP 덤프를 억제할 수 있도록 하는 z/OS 서비스입니다.

지속 가능 구독(durable subscription)

큐 관리자에 대한 구독 애플리케이션의 연결이 닫힐 때 계속 유지되는 구독입니다. 구독 애플리케이션의 연결이 끊기면 지속 가능 구독은 그대로 보유하고 발행물이 계속 전달됩니다. 애플리케이션은 다시 연결되면 고유 구독 이름을 지정하여 동일한 구독을 사용할 수 있습니다. 지속 불가능 구독(nondurable subscription)도 참조하십시오.

동적 큐(dynamic queue)

프로그램이 모델 큐 오브젝트를 열 때 생성되는 로컬 큐입니다.

E

도청(eavesdropping)

정보가 손상되지는 않지만 해당 기밀이 누설되는 통신 보안에 대한 침해입니다. 위장, 도용도 참조하십시오.

Eclipse

ISV(Independent Software Vendor)와 다른 도구 개발자에게 플러그 형태로 호환 가능한 애플리케이션 개발 도구를 개발하는 데 필요한 표준 플랫폼을 제공하는 OSI(Open Source Initiative)입니다.

캡슐화(encapsulation)

객체 지향 프로그래밍에서 오브젝트, 기능, 또는 클래스의 내재된 세부사항을 클라이언트 프로그램에서 숨기는데 사용되는 기술입니다.

암호화

컴퓨터 보안에서 데이터를 관독할 수 없는 양식으로 변환하여 원래 데이터를 확보할 수 없게 하거나 복호화 프로세스를 통해서만 확보할 수 있게 하는 프로세스입니다.

큐에 삽입(enqueue)

큐에 메시지 또는 항목을 넣는 것입니다.

엔티티

RACF와 같이 보안 서비스에 정의되는 사용자, 그룹 또는 자원입니다.

환경 변수(environment variable)

운영 체제 또는 다른 프로그램이 실행되는 방식이나 운영 체제가 인식하는 디바이스를 지정하는 변수입니다.

ESM

[ESM\(External Security Manager\)](#)을 참고하십시오.

ESTAE

[ESTAE\(Extended Specify Task Abnormal Exit\)](#)를 참고하십시오.

이벤트 데이터(event data)

이벤트 메시지에서 이벤트에 대한 정보(예: 큐 관리자 이름과 이벤트를 발생시킨 애플리케이션)가 포함된 메시지 데이터의 부분입니다. [이벤트 헤더](#)도 참조하십시오.

이벤트 헤더(event header)

이벤트 메시지에서 이벤트에 대한 이유 코드의 이벤트 유형을 식별하는 메시지 데이터의 부분입니다. [이벤트 데이터](#)도 참조하십시오.

이벤트 메시지(event message)

WebSphere MQ 시스템의 네트워크에서 도구 이벤트의 원점과 관련한 정보(예: 이벤트의 범주, 이벤트를 발생한 애플리케이션의 이름 및 큐 관리자 통계)가 포함된 메시지입니다.

이벤트 큐(event queue)

큐 관리자가 이벤트를 감지한 후 이벤트 메시지를 넣는 큐입니다. 이벤트의 각 범주(큐 관리자, 성능, 구성, 도구 또는 채널 이벤트)에는 자체 이벤트 큐가 있습니다.

이벤트 뷰어(Event Viewer)

로그 파일을 검사하고 관리하기 위해 Windows 시스템에서 제공하는 도구입니다.

예외 리스너(exception listener)

애플리케이션이 등록할 수 있고 JMS 예외를 애플리케이션에 비동기로 전달하기 위해 `onException()` 메소드가 호출되는 클래스의 인스턴스입니다.

독점적 메소드(exclusive method)

객체 지향 프로그래밍에서 다형성을 표현하고자 하는 의도가 없는 메소드로 특정 효과를 가지고 있습니다.

ESTAE(Extended Specify Task Abnormal Exit)

재시도 주소를 지정하거나 이상종료를 진단하고 처리하는 데 필요한 사용자 지정된 종료 루틴에 제어를 제공하고 복구 기능을 제공하는 z/OS 매크로입니다.

ESM(External Security Manager)

사용자 및 자원에 대한 보안 검사를 수행하는 보안 제품입니다. RACF는 ESM의 예입니다.

F**오류 복구**

소프트웨어, 하드웨어 또는 네트워크 인터럽트가 발생한 경우에 중복 또는 대기 시스템으로 전환하는 자동 조작입니다.

FAP

[FAP\(Formats and Protocols\)](#)를 참고하십시오.

FFDC

[첫 번째 오류 데이터 캡처\(FFDC, first-failure data capture\)](#)를 참고하십시오.

FFST

[첫 번째 오류 지원 기술](#)을 참조하십시오.

FFST 파일(FFST file)

첫 번째 오류 지원 기술 파일을 참조하십시오.

FIFO

FIFO(First In, First Out)를 참고하십시오.

FIPS

미국 연방 정보 처리 표준(Federal Information Processing Standard)입니다.

첫 번째 오류 데이터 캡처(FFDC, first-failure data capture)

1. 오류를 식별하고 이 오류에 대한 정보를 수집 및 로깅하며 영향을 받는 런타임 소프트웨어에 제어를 리턴하는 문제점 진단 보조 도구입니다.
2. 문제점 인식, 진단 데이터의 선택적 덤프, 증상 문자열 생성 및 문제점 로그 항목을 제공하는 FFST 아키텍처의 IBM i 구현입니다.

FFST(First Failure Support Technology)

방어 프로그래밍 기법을 통해 오류를 감지하는 단일 접근 방법을 정의하는 IBM 아키텍처입니다. 이 기술은 예방적(필요할 때까지 수동적임) 문제점 인식과 소프트웨어 문제점을 디버그하는 데 필요한 진단 출력에 대한 설명을 제공합니다.

첫 번째 오류 지원 기술 파일(FFST file, First Failure Support Technology file)

소프트웨어 문제점을 감지하고 진단하는 데 사용하는 정보가 포함된 파일입니다. WebSphere MQ에서 FFST 파일에는 FDC의 파일 유형이 있습니다.

FIFO(First In, First Out)

가장 오랫동안 큐에서 대기하고 있던 항목을 다음 검색 항목으로 정하는 큐잉 기술입니다.

강제 종료(forced shutdown)

현재 활성 태스크의 상태와 무관하게 어댑터와 z/OS용 WebSphere MQ의 연결이 즉시 끊기는 CICS 어댑터의 시스템 종료 유형입니다. 정상 종료(quieted shutdown)도 참조하십시오.

형식

메시지 큐잉에서 메시지에 있는 애플리케이션 데이터의 네이처를 식별하는 데 사용되는 용어입니다.

FAP(Formats and Protocols)

메시지 큐잉에서 큐 관리자가 서로 통신하는 방식과 클라이언트가 서버 큐 관리자와 통신하는 방식에 대한 정의입니다.

프레임워크(Framework)

WebSphere MQ에서, 고객 또는 벤더가 WebSphere MQ 제품에 제공된 특정한 기능을 확장하거나 바꾸는 프로그램을 작성할 수 있도록 하는 프로그래밍 인터페이스의 콜렉션입니다. 해당 인터페이스는 DCI(Data Conversion Interface), MCI(Message Channel Interface), NSI(Name Service Interface), SEI(Security Enabling Interface) 및 TMI(Trigger Monitor Interface)입니다.

파생 클래스(friend class)

모든 멤버 기능에 다른 클래스의 개인용 및 보호 설정된 멤버에 대한 액세스가 부여된 클래스입니다. 파생 클래스는 다른 클래스의 선언에서 이름이 지정되고 키워드 friend를 클래스의 접두부로 사용합니다.

FRR

FRR(Functional Recovery Routine)을 참고하십시오.

전체 저장소(full repository)

클러스터의 모든 큐 관리자에 대한 전체 정보 세트입니다. 이 정보 세트는 저장소 또는 전체 저장소라고 하며 일반적으로 클러스터에서 두 개의 큐 관리자가 보유합니다. 부분 저장소(partial repository)도 참조하십시오.

함수

호출 및 평가될 수 있고 호출하는 명령문에 값을 리턴할 수 있는 이름 지정된 명령문 그룹입니다.

FRR(Functional Recovery Routine)

복구 루틴이 프로그램 인터럽트의 이벤트에서 제어를 가져올 수 있도록 하는 z/OS 복구 및 종료 관리자입니다.

G

게이트웨이 큐 관리자(gateway queue manager)

애플리케이션의 메시지를 클러스터의 다른 큐 관리자로 라우트하는 데 사용되는 클러스터 큐 관리자입니다.

GTF(Generalized Trace Facility)

I/O 인터럽트, SVC 인터럽트, 프로그램 인터럽트 및 외부 인터럽트 등 중요한 시스템 이벤트를 기록하는 z/OS 서비스 프로그램입니다.

GSS API(Generic Security Services API)

GSS API(Generic Security Services API, Generic Security Services application programming interface)를 참고하십시오.

GSS API(Generic Security Services API, Generic Security Services application programming interface)

보안 서비스에 액세스하는 데 필요한 공용 API(Application Programming Interface)입니다.

가져오기

메시지 큐잉에서 MQGET 호출을 사용하여 큐에서 메시지를 제거하고 해당 콘텐츠를 호출하는 애플리케이션에 리턴하는 것입니다. 찾아보기(browse), 넣기(put)도 참조하십시오.

글로벌 정의 오브젝트(globally defined object)

z/OS에서 해당 정의가 공유 저장소에 저장된 오브젝트입니다. 이 오브젝트는 큐 공유 그룹의 모든 큐 관리자에 사용할 수 있습니다. 로컬 정의 오브젝트(locally defined object)도 참조하십시오.

글로벌 추적

z/OS용 WebSphere MQ 추적 옵션이며 여기서 추적 데이터는 전체 z/OS용 WebSphere MQ 서브시스템에서 발생합니다.

글로벌 트랜잭션

분산 트랜잭션 환경에서 하나 이상의 자원 관리자가 수행하고 외부 트랜잭션 관리자가 조정하는 복구 가능한 작업 단위입니다.

GSS API

GSS API(Generic Security Services API, Generic Security Services application programming interface)를 참고하십시오.

GTF

GTF(Generalized Trace Facility)를 참고하십시오.

H

데이터 교환

클라이언트가 공개 키 기술을 사용하여 서버를 인증하고 선택적으로 서버가 클라이언트를 인증할 수 있게 한 후 클라이언트 및 서버가 도용의 암호화, 복호화 및 감지를 위한 대칭 키 작성 시 협력할 수 있게 하는 SSL(Secure Sockets Layer) 세션의 시작 시 메시지를 교환하는 것입니다.

기록 메시지(hardened message)

시스템 장애 시 메시지가 유실되지 않도록 보조(디스크) 스토리지에 기록되는 메시지입니다.

헤더

메시지 헤더(message header)를 참고하십시오.

heartbeat

한 엔티티가 여전히 활성 상태임을 전달하기 위해 다른 엔티티에 송신하는 신호입니다.

하트비트 플로우(heartbeat flow)

송신할 메시지가 없는 경우 송신하는 메시지 채널 에이전트(MCA)로부터 수신하는 MCA로 전달되는 펄스입니다. 이 펄스는 수신하는 MCA의 차단을 해제하며 그렇지 않을 경우에 이는 메시지가 도착하거나 연결 끊기 간격이 만료될 때까지 대기 상태로 남아 있습니다.

하트비트 간격

하트비트 플로우 사이에 경과되는 시간(초)입니다.

계층(hierarchy)

발행/구독 메시징 토폴로지에서 상위 큐 관리자에 연결된 로컬 큐 관리자입니다.

HTTP

[HTTP\(Hypertext Transfer Protocol\)](#)를 참고하십시오.

하이퍼텍스트 전송 프로토콜(HTTP, Hypertext Transfer Protocol)

웹에서 하이퍼텍스트 및 XML 문서를 전송하고 표시하는 데 사용되는 인터넷 프로토콜입니다.

I

ID 컨텍스트(identity context)

큐에 메시지를 먼저 넣는 애플리케이션의 사용자를 식별하는 정보입니다.

식별

컴퓨터 시스템의 각 사용자를 고유하게 식별할 수 있게 하는 보안 서비스입니다. 이 서비스를 구현하는 공용 메커니즘은 사용자 ID를 각각의 사용자와 연관시키는 것입니다.

ID 컨텍스트(identity context)

처음으로 메시지를 큐에 넣는 애플리케이션의 사용자를 식별하는 정보입니다.

IFCID

[IFCID\(Instrumentation Facility Component Identifier\)](#)를 참고하십시오.

ILE

[통합 언어 환경](#)을 참조하십시오.

즉시 종료(immediate shutdown)

WebSphere MQ에서 애플리케이션의 연결이 끊어질 때까지 대기하지 않는 큐 관리자의 종료입니다. 현재 메시지 큐 인터페이스(MQI) 호출은 완료할 수 있지만 새 MQI 호출은 즉시 종료가 요청된 후 실패합니다. [강제 종료\(preemptive shutdown\)](#), [정상 종료\(quieted shutdown\)](#)도 참조하십시오.

위장(impersonation)

의도된 수신자로 가장한 사용자에게 정보가 전달되거나 다른 사용자로 가장한 사용자가 정보를 송신하는 통신 보안의 침해입니다. [도청](#), [도용](#)도 참조하십시오.

인바운드 채널(inbound channel)

다른 큐 관리자로부터 메시지를 수신하는 채널입니다.

내장 형식(in-built format)

[내장 형식\(built-in format\)](#)을 참고하십시오.

index

WebSphere MQ 관리 인터페이스(MQAI)에서 데이터 항목을 참조하는 수단입니다.

인다우트 복구 단위(in-doubt unit of recovery)

동기점이 요청되었으나 아직 확인되지 않은 복구 단위 상태입니다.

인플라이트(inflight)

커밋 프로세스의 준비 단계를 아직 완료하지 않은 복구 단위 또는 자원의 상태입니다.

상속(inheritance)

기존 클래스가 다른 클래스 작성의 기초로 사용되는 객체 지향 프로그래밍 기술입니다. 상속을 통해 더 많은 특정 요소가 더 많은 일반 요소의 작동 및 구조와 통합됩니다.

초기화 입력 데이터 세트(initialization input data set)

시작 시 z/OS용 WebSphere MQ에서 사용하는 데이터 세트입니다.

이니시에이션 큐

큐 관리자가 트리거 메시지를 넣는 로컬 큐입니다.

시작기

분산 큐잉에서 다른 시스템의 네트워크 연결을 요청하는 프로그램입니다. [응답자\(responder\)](#)도 참조하십시오.

입력 매개변수

호출을 작성할 때 정보를 제공하는 MQI 호출의 매개변수입니다.

삽입 순서(insertion order)

WebSphere MQAI(MQ Administration Interface)에서 데이터 항목이 데이터 백에 배치되는 순서입니다.

설치 가능 서비스

UNIX 및 Linux 시스템의 WebSphere MQ 및 WebSphere MQ for Windows에서 독립 컴포넌트로 제공되는 추가 기능. 각 컴포넌트의 설치 선택사항이며 조직 내(in-house) 또는 써드파티 컴포넌트를 대신 사용할 수 있습니다.

인스턴스(instance)

클래스에 속하는 오브젝트의 특정 발생입니다. 오브젝트(object)도 참조하십시오.

인스턴스 데이터(instance data)

객체 지향 프로그래밍에서 오브젝트와 연관된 상태 정보입니다.

도구 이벤트(instrumentation event)

WebSphere MQ 시스템의 네트워크에서 큐 관리자 자원 정의, 성능 조건 및 채널 조건을 모니터링하는 방법입니다.

IFCID(Instrumentation Facility Component Identifier)

z/OS용 Db2에서 이벤트의 추적 레코드를 식별하고 해당 이름을 지정하는 값입니다. START TRACE 및 MODIFY TRACE 명령의 매개변수로서 이는 해당 이벤트를 추적하도록 지정합니다.

통합 언어 환경(ILE, Integrated Language Environment)

모든 ILE 공식 고급 언어의 런타임 바인드 가능 API(Application Program Interface) 및 공용 런타임 환경을 제공하는 인터페이스 및 구성 세트입니다.

IPCS(Interactive Problem Control System)

온라인 문제점 관리, 대화식 문제점 진단, 디스크 상주 이상종료 덤프의 온라인 디버깅, 문제점 추적 및 문제점 보고를 허용하는 MVS 및 z/OS의 컴포넌트입니다.

ISPF(Interactive System Productivity Facility)

전체 화면 편집기 및 대화 상자 관리자 역할을 하는 IBM 라이선스가 있는 프로그램입니다. 애플리케이션 프로그램 작성에 사용되며, 애플리케이션 프로그래머와 터미널 사용자 간의 대화식 대화 상자 및 표준 화면 패널을 생성하는 수단을 제공합니다.

중간 인증서(Intermediate certificate)

루트 인증서가 아닌 서명자 인증서입니다.

인터페이스(interface)

객체 지향 프로그래밍에서 작동의 추상 모델입니다(함수 또는 메소드의 콜렉션).

IP(Internet Protocol)

네트워크 또는 상호 연결된 네트워크를 통해 데이터를 라우트하는 프로토콜입니다. 이 프로토콜은 상위 프로토콜 계층과 물리적 네트워크 사이의 중개자 역할을 수행합니다. TCP(Transmission Control Protocol)의 내용도 참조하십시오.

프로세스 간 통신(IPC, interprocess communication)

프로그램이 서로 메시지를 송신하는 데 사용하는 프로세스입니다. 프로세스 간 통신(IPC)의 일반적인 메소드는 소켓, 세마포어, 신호 및 내부 메시지 큐입니다. 클라이언트(client)도 참조하십시오.

시스템간 통신(ISC, intersystem communication)

기타 컴퓨터 시스템에서 통신하기 위한 인바운드 및 아웃바운드 지원을 제공하는 CICS 기능입니다.

IP

IP(Internet Protocol)를 참고하십시오.

IPC

프로세스 간 통신(IPC, interprocess communication)을 참고하십시오.

IPCS

IPCS(Interactive Problem Control System)를 참고하십시오.

ISC

시스템간 통신(ISC, intersystem communication)을 참고하십시오.

ISPF

ISPF(Interactive System Productivity Facility)를 참고하십시오.

J

JAAS

JAAS(Java Authentication and Authorization Service)의 내용을 참조하십시오.

JAAS(Java Authentication and Authorization Service)

Java EE 기술에서 보안 기반 조작을 수행하기 위한 표준 API. JAAS를 통해 서비스는 애플리케이션이 기본 기술과 독립된 상태를 유지할 수 있게 하면서 사용자를 인증하고 권한 부여할 수 있습니다.

JMS(Java Message Service)

메시지를 처리하기 위한 Java 언어 기능을 제공하는 API(Application programming interface). [MQI\(Message Queue Interface\)](#)도 참조하십시오.

JRE(Java Runtime Environment)

표준 Java 플랫폼을 구성하는 코어 실행 프로그램 및 파일이 들어 있는 JDK(Java Development Kit)의 서브 세트. JRE에는 JVM(Java Virtual Machine), 코어 클래스 및 지원 파일이 포함됩니다.

JMS

[JMS\(Java Message Service\)](#)의 내용을 참조하십시오.

JMSAdmin

관리자가 JMS 오브젝트의 특성을 정의하여 JNDI 네임스페이스에 저장할 수 있게 하는 관리 도구입니다.

저널(journal)

로컬 오브젝트에 대한 업데이트를 제어하기 위해 iSeries용 WebSphere MQ에서 사용하는 OS/400의 기능입니다. 각각의 큐 관리자 라이브러리에는 해당 큐 관리자에 대한 저널이 포함되어 있습니다.

JRE

[JRE\(Java Runtime Environment\)](#)의 내용을 참조하십시오.

K

활성 유지(keepalive)

소켓이 여전히 올바르게 작동 중인지 판별하기 위해 사전정의된 간격으로 네트워크에서 작은 패킷을 송신하는 TCP/IP 메커니즘입니다.

Kerberos

대칭 키 암호화를 기반으로 하는 네트워크 인증 프로토콜입니다. Kerberos는 네트워크에 로그인하는 각각의 사용자에게 티켓이라는 고유 키를 지정합니다. 이 티켓은 네트워크를 통해 송신되는 메시지에 임베드됩니다. 메시지 수신자는 이 티켓을 사용하여 송신자를 인증합니다.

키 인증(key authentication)

[인증\(authentication\)](#)을 참고하십시오.

키 저장소(key repository)

디지털 인증서 및 이와 연관된 키의 저장소에 대한 일반 용어입니다. 서로 다른 유형의 키 저장소에는 인증서 관리 시스템(CMS), Java 키 저장소(JKS), Java 암호화 확장 키 저장소(JCEKS), 공개 키 암호화 표준 12(PKCS12) 키 저장소 및 RACF 키 링이 포함됩니다. 키 저장소 유형 간에 반드시 구분해야 하는 경우, 문서는 특정 이름으로 키 저장소 유형을 참조합니다. 다중 키 저장소 유형에 적용되는 컨텍스트에서는 일반 용어 키 저장소가 사용됩니다.

키 링(key ring)

컴퓨터 보안에서 공개 키, 개인 키, 신뢰할 수 있는 루트 및 인증서가 포함된 파일입니다.

키 저장소(key store)

개인 키 및 이와 대응되는 개인 인증서를 위한 장소입니다. [신뢰 저장소](#)도 참조하십시오.

L

이상 종료 시 메시지(last will and testament)

클라이언트가 모니터에 등록된 오브젝트이며 클라이언트가 예상치 않게 종료할 경우 모니터에서 사용됩니다.

LDAP

[LDAP\(Lightweight Directory Access Protocol\)](#)을 참고하십시오.

경량 디렉토리 액세스 프로토콜(LDAP, Lightweight Directory Access Protocol)

TCP/IP를 사용하여 X.500 모델을 지원하는 디렉토리에 대한 액세스를 제공하고 더 복잡한 X.500 DAP(Directory Access Protocol)의 자원 요구사항을 발생시키지 않는 개방형 프로토콜입니다. 예를 들어, LDAP을 사용하여 인터넷 또는 인트라넷 디렉토리에서 사용자, 조직 및 기타 자원을 찾을 수 있습니다.

선형 로깅(linear logging)

UNIX 및 Linux 시스템의 WebSphere MQ 및 WebSphere MQ for Windows에서 파일 순서로 데이터를 다시 시작하는 프로세스. 필요에 따라 새 파일이 시퀀스에 추가됩니다. 데이터가 기록되는 공간은 재사용되지 않습니다. [순환 로깅](#)도 참조하십시오.

링크 레벨 보안

메시지 채널 에이전트(MCA), 통신 서브시스템에 의해 또는 둘이 함께 결합하여 작동함으로써 직간접적으로 호출되는 보안 서비스입니다.

리스너

수신되는 요청을 감지하여 연관된 채널을 시작하는 프로그램입니다.

리모트 큐의 로컬 정의(local definition of a remote queue)

다른 큐 관리자가 소유한 큐의 속성을 정의하는 로컬 큐 관리자에 속한 WebSphere MQ 오브젝트입니다. 큐 관리자 알리어싱 및 응답 대상 큐 알리어싱에도 사용됩니다.

로케일(locale)

언어 또는 지역을 식별하고 형식 변환(예: 조합, 대소문자 변환, 문자 분류, 메시지 언어, 날짜 및 시간 표현, 숫자 표현)을 판별하는 설정입니다.

로컬 정의 오브젝트(locally defined object)

z/OS에서 해당 정의가 페이지 세트 0에 저장되는 오브젝트입니다. 이 정의는 이를 정의한 큐 관리자만 액세스할 수 있습니다. [글로벌 정의 오브젝트\(globally defined object\)](#)도 참조하십시오.

로컬 큐

로컬 큐 관리자에 속하는 큐입니다. 로컬 큐는 처리 대기 중인 메시지의 목록을 포함할 수 있습니다. [리모트 큐](#)도 참조하십시오.

로컬 큐 관리자

프로그램이 연결되고 해당 프로그램에 메시지 큐잉 서비스를 제공하는 큐 관리자입니다. [리모트 큐 관리자](#)도 참조하십시오.

로그

WebSphere MQ에서 장애 발생 시 복구할 수 있도록 메시지 수신, 전송 및 전달 도중 큐 관리자가 수행하는 작업을 기록하는 파일입니다.

로그 제어 파일(log control file)

UNIX 및 Linux 시스템의 WebSphere MQ 및 WebSphere MQ for Windows에서 로그 파일 사용을 모니터링하는 데 필요한 정보(예: 크기 및 위치, 사용 가능한 다음 파일의 이름)를 포함하는 파일.

로그 파일

UNIX의 WebSphere MQ에서는 Linux 시스템과 WebSphere MQ for Windows에서 큐 관리자가 제어하는 데이터에 대한 모든 중요한 변경사항이 기록됩니다. 기본 로그 파일이 가득 차면 WebSphere MQ는 보조 로그 파일을 할당합니다.

논리 장치(LU, logical unit)

사용자 또는 애플리케이션 프로그램이 다른 사용자 또는 애플리케이션 프로그램과 통신하기 위해 SNA 네트워킹에 액세스하는 데 사용되는 액세스 지점입니다.

논리 장치 6.2(LU 6.2, logical unit 6.2)

분산 처리 환경에서 프로그램 간 일반 통신을 지원하는 SNA 논리 장치입니다.

논리적 작업 단위 ID(LUWID, logical unit of work identifier)

네트워크에서 스레드를 고유하게 식별하는 이름입니다. 이 이름은 완전한 논리 장치 네트워크 이름, 논리적 작업 단위 인스턴스 번호 및 논리적 작업 단위 시퀀스 번호로 구성됩니다.

로그 레코드(log record)

로그 파일에서 단일 단위로 처리되는 데이터 세트입니다.

LRSN(Log Record Sequence Number)

데이터 공유 멤버와 연관된 로그 레코드의 고유 ID입니다. z/OS용 Db2는 데이터 공유 환경에서 복구를 위해 LRSN을 사용합니다.

LRSN

[LRSN\(Log Record Sequence Number\)](#)을 참조하십시오.

LU

[논리 장치\(LU, logical unit\)](#)를 참조하십시오.

LU 6.2

[논리 장치 6.2\(LU 6.2, logical unit 6.2\)](#)를 참고하십시오.

LU 6.2 대화(LU 6.2 conversation)

SNA에서 트랜잭션 프로그램 상호간에 통신을 가능하게 해 주는 LU 6.2 세션을 통한 두 트랜잭션 프로그램 간 논리적 연결입니다.

LU 6.2 대화 레벨의 보안(LU 6.2 conversation level security)

SNA에서 파트너 트랜잭션 프로그램이 대화를 시작한 트랜잭션 프로그램을 인증할 수 있게 하는 대화 레벨의 보안 프로토콜입니다. LU 6.2 대화 레벨 보안을 일반 사용자 확인이라고도 합니다.

LU 6.2 세션(LU 6.2 session)

SNA에서 6.2 유형의 두 논리 장치(LU) 간 세션입니다.

LU 이름

VTAM이 네트워크의 노드를 참조하는 이름입니다.

LUWID

[논리적 작업 단위 ID\(LUWID, logical unit of work identifier\)](#)를 참고하십시오.

M

관리 목적지(managed destination)

관리 구독을 사용하도록 선택하는 애플리케이션에 대해 발행된 메시지를 송신할 목적지로서 큐 관리자가 제공하는 큐입니다. [관리 구독](#)도 참조하십시오.

관리 핸들(managed handle)

큐 관리자가 구독에 송신되는 메시지의 스토리지를 관리하도록 지정된 경우 MQSUB 호출에서 리턴하는 ID입니다.

관리 구독(managed subscription)

애플리케이션이 특정 큐를 사용하지 않아도 되기 때문에 큐 관리자가 발행물을 수신할 구독자 큐를 작성하는 구독입니다. 관리 목적지(managed destination)도 참조하십시오.

정렬(marshalling)

[직렬화\(serialization\)](#)를 참고하십시오.

MCA

[메시지 채널 에이전트\(MCA, message channel agent\)](#)를 참고하십시오.

MCI

[메시지 채널 인터페이스\(MCI, message channel interface\)](#)를 참고하십시오.

매체 이미지(media image)

UNIX 및 Linux 시스템의 WebSphere MQ 및 WebSphere MQ for Windows에서 오브젝트의 이미지를 포함하는 로그 레코드의 순서입니다. 이 이미지에서 오브젝트를 다시 작성할 수 있습니다.

메시지

1. 개인 또는 프로그램으로부터 다른 개인 또는 프로그램에 전송된 통신입니다.
2. 시스템 프로그래밍에서 터미널 운영자 또는 시스템 관리자를 위해 작성된 정보입니다.

메시지 연관관계(message affinity)

두 애플리케이션 사이에서 교환되는 대화식 메시지 사이의 관계이며 메시지는 특정 큐 관리자에 의해 처리되거나 특정 순서로 처리되어야 합니다.

메시지 채널

분산 메시지 큐잉에서 메시지를 한 큐 관리자에서 다른 큐 관리자로 이동하는 메커니즘입니다. 메시지 채널은 두 개의 메시지 채널 에이전트(한쪽 끝의 송신자와 다른 쪽 끝의 수신자)와 통신 링크로 구성됩니다. [채널](#)도 참조하십시오.

메시지 채널 에이전트(MCA, message channel agent)

준비된 메시지를 전송 큐에서 통신 링크로 전송하거나 통신 링크에서 목적지 큐로 전송하는 프로그램입니다. [MQI\(Message Queue Interface\)](#)도 참조하십시오.

메시지 채널 인터페이스(MCI, message channel interface)

WebSphere MQ 큐 관리자 및 다른 메시징 시스템 간의 메시지를 전송하는 고객 또는 벤더 작성 프로그램에 맞는 WebSphere MQ 인터페이스입니다. WebSphere MQ 프레임워크의 일부입니다. MQI(Message Queue Interface)도 참조하십시오.

메시지 이용자(message consumer)

1. 메시지를 가져오고 처리하는 프로그램 또는 기능입니다. 이용자(consumer)도 참조하십시오.
2. JMS 에서 세션 내에서 작성되어 대상에서 메시지를 수신하는 오브젝트입니다.

메시지 컨텍스트(message context)

메시지 디스크립터의 필드에 보유되는 메시지의 진원지에 대한 정보입니다. ID 컨텍스트와 원본 컨텍스트라는 두 가지 범주의 컨텍스트 정보가 있습니다.

메시지 디스크립터

WebSphere MQ 메시지의 일부로 전달되는 메시지 형식 및 프리젠테이션을 설명하는 제어 정보입니다. 메시지 디스크립터의 형식은 MQMD 구조에 의해 정의됩니다.

메시지 엑시트(message exit)

메시지의 콘텐츠를 수정하는 데 사용되는 채널 엑시트 프로그램의 유형입니다. 메시지 엑시트는 일반적으로 쌍으로(채널의 양쪽 끝에 하나씩) 작동합니다. 채널의 송신 측에서 메시지 엑시트는 메시지 채널 에이전트(MCA)가 전송 큐로부터 메시지를 가져온 후 호출됩니다. 채널의 수신 측에서 메시지 엑시트는 메시지 채널 에이전트(MCA)가 목적지 큐에 메시지를 넣기 전에 호출됩니다.

메시지 플로우 제어(message flow control)

큐 관리자 사이의 메시지 라우트 설정 및 유지보수와 관련된 분산 큐 관리 태스크입니다.

MFS(Message Format Service)

애플리케이션에서 디바이스 종속 데이터 대신 단순 로직 메시지를 처리할 수 있도록 하여 애플리케이션 개발 프로세스를 단순화하는 IMS 편집 기능입니다.

메시지 그룹(message group)

관련 메시지의 논리 그룹입니다. 관계는 메시지를 넣는 애플리케이션에 의해 정의되며 생성자와 이용자 둘 다 그룹화를 사용할 경우 메시지는 넣어진 순서로 검색됩니다.

메시지 핸들(message handle)

메시지에 대한 참조입니다. 핸들은 메시지의 메시지 특성에 대한 액세스를 확보하는 데 사용할 수 있습니다.

메시지 헤더(message header)

제어 정보(예: 고유 메시지 ID, 메시지의 송신자 및 수신자, 메시지 우선순위 및 메시지의 유형)가 포함된 메시지의 부분입니다.

메시지 입력 디스크립터(MID, message input descriptor)

애플리케이션 프로그램에 제공된 데이터의 형식에 대해 설명하는 MFS(Message Format Service) 제어 블록입니다. 메시지 출력 디스크립터도 참조하십시오.

메시지 리스너(message listener)

비동기 메시지 이용자 역할을 수행하는 오브젝트입니다.

메시지 출력 디스크립터(MOD, message output descriptor)

애플리케이션 프로그램에 의해 생성된 출력 데이터의 형식에 대해 설명하는 MFS(Message Format Service) 제어 블록입니다. 메시지 입력 디스크립터(MID, message input descriptor)도 참조하십시오.

메시지 우선순위(message priority)

WebSphere MQ에서, 트리거 이벤트가 생성되는지 여부 및 큐의 메시지가 검색되는 순서에 영향을 미칠 수 있는 메시지의 속성입니다.

메시지 작성자(message producer)

JMS에서 세션에 의해 작성되고 목적지에 메시지를 송신하는 데 사용되는 오브젝트입니다.

메시지 특성(message property)

메시지와 연관된 이름-값 쌍 형식의 데이터입니다. 메시지 특성은 발행물을 필터링하거나 선택적으로 큐에서 메시지를 가져오기 위해 메시지 선택자로 사용할 수 있습니다. 메시지 특성은 메시지 본문을 변경하지 않고 처리에 대한 비즈니스 데이터 또는 상태 정보를 포함하는 데 사용할 수 있습니다.

메시지 큐 인터페이스(MQI, Message Queue Interface)

WebSphere MQ 큐 관리자가 제공하는 프로그래밍 인터페이스입니다. 이 프로그래밍 인터페이스를 사용하면 애플리케이션 프로그램이 메시지 큐잉 서비스에 액세스할 수 있습니다. [JMS\(Java Message Service\)](#), [메시지 채널 에이전트](#), [메시지 채널 인터페이스](#)의 내용도 참조하십시오.

메시지 큐 관리(MQM, message queue management)

HP Integrity NonStop Server용 WebSphere MQ에서 큐 관리자, 큐 및 채널을 관리할 수 있도록 PCF 명령 형식 및 제어 명령에 대한 액세스를 제공하는 기능입니다.

메시지 큐잉(message queuing)

애플리케이션의 각 프로그램이 큐에 메시지를 넣어 다른 프로그램과 통신하는 프로그래밍 기술입니다.

메시지 재시도(message-retry)

메시지를 넣을 수 없는 MCA에 사용 가능한 옵션입니다. MCA는 사전정의된 시간 동안 대기한 후 메시지 넣기를 다시 시도할 수 있습니다.

메시지 세그먼트(message segment)

너무 커서 애플리케이션 또는 큐 관리자가 처리할 수 없는 다수의 메시지 세그먼트 중 하나입니다.

메시지 선택자(message selector)

애플리케이션 프로그래밍에서 선택 문자열이 나타내는 SQL(Structured Query Language) 조회를 충족하는 특성을 갖는 메시지만 사용하도록 등록하기 위해 애플리케이션이 사용하는 가변 길이 문자열입니다. 메시지 선택자의 구문은 SQL92 조건식 구문의 서브세트를 기반으로 합니다.

메시지 순서 매기기(message sequence numbering)

통신 링크를 통해 전송되는 동안 메시지에 고유 번호를 제공하는 프로그래밍 기술입니다. 이를 통해 수신 프로세스는 모든 메시지를 수신했는지 여부를 확인하고 이를 원래 순서대로 큐에 배치하며 중복 메시지를 제거할 수 있습니다.

메시지 토큰(message token)

활성 큐 관리자에 있는 메시지의 고유 ID입니다.

메소드(method)

객체 지향 설계 또는 프로그래밍에서 조작에 의해 지정된 작동을 구현하는 소프트웨어입니다.

MFS

[MFS\(Message Format Service\)](#)를 참조하십시오.

MGAS

[대부분의 글로벌 주소 공간\(MGAS, mostly global address space\)](#)을 참조하십시오.

Microsoft 클러스터 서버(MSCS, Microsoft Cluster Server)

Windows를 실행 중인 컴퓨터를 MSCS 클러스터로 그룹화하여 고가용성을 제공하는 기술입니다. 클러스터의 한 컴퓨터에서 문제점 범위에 속하는 문제점이 발생하는 경우 MSCS는 일련의 순서대로 문제가 발생한 애플리케이션을 종료하고 해당 상태 데이터를 클러스터의 다른 컴퓨터에 전송한 후 해당 컴퓨터에서 애플리케이션을 다시 시작합니다.

Microsoft Transaction Server(MTS)

Windows 사용자가 미들 티어 서버에서 비즈니스 로직 애플리케이션을 실행할 수 있도록 도움을 주는 기능입니다. MTS는 작업을 활동으로 구분하며 이 활동은 비즈니스 로직의 독립적인 짧은 청크입니다.

MID

[메시지 입력 디스크립터\(MID, message input descriptor\)](#)를 참조하십시오.

MOD

[메시지 출력 디스크립터\(MOD, message output descriptor\)](#)를 참조하십시오.

모델 큐 오브젝트(model queue object)

프로그램이 동적 큐를 작성할 때 템플릿 역할을 수행하는 큐 속성 세트입니다.

대부분의 글로벌 주소 공간(MGAS, mostly global address space)

공유 애플리케이션을 위해 대부분의 주소 공간을 보존하는 유연한 가상 주소 공간 모델이며 HP-UX와 같은 시스템에서 사용됩니다. 이를 통해 다수의 데이터를 공유하는 프로세스의 성능을 개선할 수 있습니다. 대부분의 개인 주소 공간(MPAS, mostly private address space)도 참조하십시오.

대부분의 개인 주소 공간(MPAS, mostly private address space)

더 큰 주소 공간 블록을 프로세스에 할당할 수 있는 유연한 가상 주소 공간 모델이며 HP-UX와 같은 시스템에서 사용됩니다. 이를 통해 많은 데이터 공간이 필요한 프로세스의 성능을 개선할 수 있습니다. 대부분의 글로벌 주소 공간(MGAS, mostly global address space)도 참조하십시오.

MPAS

대부분의 개인 주소 공간(MPAS, mostly private address space)을 참고하십시오.

MQAI

MQAI(WebSphere MQ 관리 인터페이스)를 참조하십시오.

MQI

MQI(Message Queue Interface)를 참고하십시오.

MQI 채널

서버 시스템에서 WebSphere MQ 클라이언트와 큐 관리자 간의 연결. MQI 채널은 MQI 호출만 전송하고 양 방향으로 응답합니다. 채널(channel)도 참조하십시오.

MQM

메시지 큐 관리(MQM, message queue management)를 참고하십시오.

MQSC

WebSphere MQ 스크립트 명령을 참조하십시오.

MQSeries

WebSphere MQ의 이전 이름입니다.

MQ Telemetry Transport

MQTT(MQ Telemetry Transport)는 다수의 디바이스(예: 서버, 작동기, 스마트폰, 운송 수단, 가정, 의료, 원격 센서 및 제어 장치)에 연결하기 위해 TCP/IP에서 작동하는 개방형의 경량 발행/구독 프로토콜입니다. MQTT는 네트워크가 대역폭에 의해 제한을 받거나 디바이스가 메모리나 프로세서 등에 의해 제한을 받는 환경에서 작동하도록 설계되었습니다.

MQTT

MQTT(MQ Telemetry Transport)를 참고하십시오.

MQTT 클라이언트(MQTT client)

MQTT 클라이언트 애플리케이션은 MQTT 가능 서버에 연결됩니다(예: WebSphere MQ Telemetry 채널). WebSphere MQ Telemetry의 설치에서 제공하는 클라이언트 중 하나를 사용하거나 발행된 프로토콜을 사용하도록 자체 클라이언트를 작성할 수 있습니다. 일반적인 클라이언트는 텔레메트리 디바이스에서 정보를 수집하여 서버에 발행할 책임이 있습니다. 이 클라이언트는 토픽을 구독하고 메시지를 수신하고 이 정보를 사용하여 텔레메트리 디바이스를 제어할 수도 있습니다. 일부 클라이언트는 WebSphere MQ Telemetry와 함께 제공됩니다. 텔레메트리 클라이언트 및 텔레메트리 고급 클라이언트를 참조하십시오.

MQTT 서버(MQTT server)

MQTT 서버는 MQTT 프로토콜의 서버측을 핸들링합니다. 일반적으로 다수의 MQTT 클라이언트가 동시에 연결할 수 있게 하고 MQTT 클라이언트에 메시지 분배용 허브를 제공합니다. 텔레메트리(MQXR) 서비스가 있는 WebSphere MQ 큐 관리자는 MQTT 서버입니다.

MSCS

Microsoft 클러스터 서버를 참조하십시오.

MTS

Microsoft 트랜잭션 서버를 참조하십시오.

멀티홉(multi-hop)

소스 큐 관리자와 대상 큐 관리자 사이에 직접 통신 링크가 없는 경우 하나 이상의 중간 큐 관리자를 통해 전달하는 것입니다.

다중 인스턴스 큐 관리자((multi-instance queue manager)

큐 관리자 데이터의 사용을 다른 큐 관리자 인스턴스와 공유하도록 구성된 큐 관리자입니다. 실행 중인 다중 인스턴스 큐 관리자의 한 인스턴스가 활성 상태이면 다른 인스턴스는 활성 인스턴스로부터 인계받을 수 있도록 대기합니다. 단일 인스턴스 큐 관리자도 참조하십시오.

N

이름 목록

큐 이름과 같은 오브젝트 이름의 목록이 포함된 WebSphere MQ 오브젝트입니다.

이름 서비스

UNIX 및 Linux 시스템의 WebSphere MQ 및 WebSphere MQ for Windows에서 지정된 큐를 소유하는 큐 관리자를 판별하는 기능입니다.

이름 서비스 인터페이스(NSI, name service interface)

큐 이름 소유권을 해석하는 고객 또는 벤더 작성 프로그램에 맞는 WebSphere MQ 인터페이스입니다. WebSphere MQ 프레임워크의 일부입니다.

이름 변환(name transformation)

UNIX 및 Linux 시스템의 WebSphere MQ 및 WebSphere MQ for Windows에서 큐 관리자 이름이 고유하고 사용 중인 시스템에 유효하도록 변경하는 내부 프로세스. 외부적으로는 큐 관리자 이름이 변경되지 않습니다.

중첩 백(nested bag)

WebSphere MQ AI(MQ Administration Interface)에서 다른 데이터 백에 삽입된 시스템 백입니다.

중첩(nesting)

WebSphere MQ 관리 인터페이스(MQAI)에서 WebSphere MQ로부터 리턴된 정보를 그룹화하는 수단입니다.

NetBIOS(Network Basic Input/Output System)

근거리 통신망(LAN)에서 메시지, 인쇄 서버 및 파일 서버 기능을 제공하는 데 사용되는 네트워크 및 개인 컴퓨터에 대한 표준 인터페이스입니다. NetBIOS를 사용하는 애플리케이션 프로그램은 LAN 데이터 링크 제어(DLC) 프로토콜의 세부사항을 핸들링하지 않아도 됩니다.

Network Basic Input/Output System

[NetBIOS\(Network Basic Input/Output System\)](#)를 참고하십시오.

NTFS(New Technology File System)

Windows 운영 환경의 기본 파일 시스템 중 하나입니다.

노드

Microsoft 클러스터 서버(MSCS)에서 클러스터의 각 컴퓨터입니다.

지속 불가능 구독(nondurable subscription)

구독 중인 애플리케이션의 큐 관리자 연결이 열려 있는 동안만 존재하는 구독입니다. 이 구독은 고의적으로 또는 연결 유실에 의해 구독 애플리케이션과 큐 관리자의 연결이 끊길 때 제거됩니다. [지속 구독도](#) 참조하십시오.

비지속 메시지(nonpersistent message)

큐 관리자를 재시작하면 없어지는 메시지입니다. [지속 메시지\(persistent message\)](#)도 참조하십시오.

NSI

[이름 서비스 인터페이스\(NSI, name service interface\)](#)를 참고하십시오.

NTFS

[NTFS\(New Technology File System\)](#)를 참고하십시오.

NUL

[널 문자\(NUL, null character\)](#)를 참고하십시오.

널 문자(NUL, null character)

표시되거나 인쇄된 문자가 없음을 나타내는 값이 X'00'인 제어 문자입니다.

O

OAM

[오브젝트 권한 관리자\(OAM, object authority manager\)](#)를 참고하십시오.

오브젝트

1. WebSphere MQ에서 큐 관리자, 큐, 프로세스 정의, 채널, 이름 목록, 인증 정보 오브젝트, 관리 토픽 오브젝트, 리스너, 서비스 오브젝트 또는 (z/OS 전용) CF 구조 오브젝트 또는 스토리지 클래스.
2. 객체 지향 설계나 프로그래밍에서 데이터 및 해당 데이터와 연관된 조작으로 구성되는 클래스의 구체적 실현(인스턴스)입니다. 오브젝트에는 클래스에 의해 정의되는 인스턴스 데이터가 포함되어 있지만 클래스는 해당 데이터와 연관된 조작을 소유합니다.

오브젝트 권한 관리자(OAM, Object Authority Manager)

UNIX 및 Linux 시스템의 경우 WebSphere MQ, IBM i 및 WebSphere MQ for Windows의 WebSphere MQ, 명령 및 오브젝트 관리에 대한 기본 권한 부여 서비스입니다. OAM은 고객 제공 보안 서비스로 바꾸거나 이와 결합하여 실행할 수 있습니다.

오브젝트 디스크립터(object descriptor)

특정한 WebSphere MQ 오브젝트를 식별하는 데이터 구조입니다. 이 디스크립터에는 오브젝트의 이름 및 오브젝트 유형이 포함되어 있습니다.

오브젝트 핸들(object handle)

프로그램이 함께 작업하는 WebSphere MQ 오브젝트에 액세스하는 데 사용하는 ID 또는 토큰입니다.

객체 지향 프로그래밍(object-oriented programming)

데이터 추상 및 상속이라는 개념을 기반으로 하는 프로그래밍 접근 방식입니다. 절차식 프로그래밍 기술과는 달리, 객체 지향 프로그래밍은 해결 절차에 중점을 두는 대신 문제점을 구성하는 데이터 오브젝트 및 이를 조작하는 방법에 중점을 둡니다.

OCSP

온라인 인증서 상태 프로토콜(Online Certificate Status Protocol)입니다. 인증서가 폐기되는지 확인하는 메소드입니다.

오프-로딩(offloading)

z/OS용 WebSphere MQ에서 큐 관리자의 활성 로그가 해당 아카이브 로그에 전송되는 자동 프로세스입니다.

단방향 인증(one way authentication)

이 인증 방법에서 큐 관리자는 인증서를 클라이언트에 제공하지만 클라이언트에서 큐 관리자로 인증이 검사되지 않습니다.

열기(open)

큐 또는 토픽과 같은 오브젝트에 대한 액세스를 확립하는 것입니다.

개방형 시스템 상호 연결(OSI, open systems interconnection)

정보 교환을 위해 ISO(International Organization for Standardization)의 표준에 따르는 개방형 시스템의 상호 연결입니다.

OTMA(Open Transaction Manager Access)

MVS sysplex 환경에서 트랜잭션 기반의 연결이 없는 클라이언트/서버 프로토콜을 구현하는 IMS의 컴포넌트. 프로토콜의 도메인은 z/OS 시스템 간 커플링 기능(XCF)의 도메인으로 제한됩니다. OTMA는 클라이언트가 고성능을 유지하면서 대형 네트워크(또는 다수의 세션)를 지원할 수 있도록 클라이언트를 서버에 연결합니다.

OPM

기존 프로그램 모델(OPM, original program model)을 참고하십시오.

기존 프로그램 모델(OPM, original program model)

ILE(Integrated Language Environment) 모델이 도입되기 전에 소스 코드를 컴파일하고 고급 언어 프로그램을 작성하는 데 필요한 기능 세트입니다.

OSGi Alliance

음성, 데이터 및 멀티미디어 유무선 네트워크를 관리하기 위한 개방형 표준의 개요를 제시하는 스펙을 작성하는 IBM을 포함한 20개 이상의 회사로 구성된 컨소시엄입니다.

OSI

개방형 시스템 상호 연결(OSI, open systems interconnection)을 참고하십시오.

OSI 디렉토리 표준(OSI directory standard)

정보 모델, 네임스페이스, 기능 모델 및 인증 프레임워크를 포함한 포괄적인 디렉토리 서비스를 정의하는 표준입니다(X.500으로 알려져 있음). X.500은 클라이언트가 디렉토리에 액세스하는 데 사용하는 DAP(Directory Access Protocol)도 정의합니다. LDAP(Lightweight Directory Access Protocol)은 디렉토리 클라이언트에서 일부 X.500 액세스를 제거하여 해당 디렉토리를 광범위한 시스템 및 애플리케이션에 사용할 수 있게 합니다.

OTMA

OTMA(Open Transaction Manager Access)를 참고하십시오.

아웃바운드 채널(outbound channel)

전송 큐에서 메시지를 가져와서 다른 큐 관리자에 송신하는 채널입니다.

출력 로그-버퍼(output log-buffer)

z/OS용 WebSphere MQ에서 아카이브 로그에 작성되기 전에 복구 로그 레코드를 보관하는 버퍼입니다.

출력 매개변수(output parameter)

호출이 완료되거나 실패할 때 큐 관리자가 정보를 리턴하는 MQI 호출의 매개변수입니다.

오버로딩(overloading)

객체 지향 프로그래밍에서 컨텍스트에 따라 연산자 또는 메소드가 다양한 의미를 가질 수 있는 기능입니다. 예를 들어, C++에서는 함수 및 대부분의 표준 연산자가 클래스 유형과 함께 사용될 때 사용자가 함수 및 대부분의 표준 연산자를 재정의할 수 있습니다. 메소드 이름 또는 연산자는 동일하게 유지되지만 메소드 매개변수는 유형, 번호 또는 모두가 다릅니다. 이러한 차이를 총체적으로 함수 또는 연산자의 서명이라고 하며 각각의 서명은 별도로 구현해야 합니다.

P

페이지 세트

z/OS용 WebSphere MQ가 메인 스토리지의 버퍼에서 영구 백업 스토리지(DASD)로 데이터(예: 큐 및 메시지)를 이동하는 경우에 사용되는 VSAM 데이터 세트입니다.

상위 클래스(parent class)

다른 클래스가 인스턴스 메소드, 속성 및 인스턴스 변수를 상속하는 클래스입니다. 추상 클래스(abstract class)도 참조하십시오.

부분 저장소(partial repository)

클러스터의 큐 관리자에 대한 정보의 부분 세트입니다. 부분 저장소는 전체 저장소를 호스팅하지 않는 모든 클러스터 큐 관리자가 유지보수합니다. 전체 저장소(full repository)도 참조하십시오.

파트너 큐 관리자(partner queue manager)

리모트 큐 관리자(remote queue manager)를 참고하십시오.

PasTicket

RACF 보안 사인온에서 워크스테이션이나 다른 클라이언트가 네트워크를 통해 RACF 비밀번호를 보내는 대신 호스트에 사인온하기 위해 사용할 수 있는 동적으로 생성되는 임의의 1회용 대체 비밀번호.

PCF

PCF(Programmable Command Format)를 참고하십시오.

보류 이벤트(pending event)

CICS 어댑터의 연결 요청의 결과로 발생하는 스케줄되지 않은 이벤트입니다.

여과(percolation)

오류 복구에서 복구 루틴으로부터 상위 레벨 복구 루틴으로의 사전 설정된 제어 경로를 따라 통과하는 작업입니다.

성능 이벤트(performance event)

한계 조건이 발생했음을 나타내는 이벤트의 범주입니다.

성능 추적(performance trace)

추적 데이터가 성능 분석 및 조정에 사용되는 WebSphere MQ 추적 옵션입니다.

영구적 동적 큐(permanent dynamic queue)

삭제가 명확하게 요청되는 경우에만 닫힐 때 삭제되는 동적 큐입니다. 영구적 동적 큐는 큐 관리자가 실패하는 경우 복구되므로 지속 메시지를 포함할 수 있습니다. 임시 동적 큐(temporary dynamic queue)도 참조하십시오.

지속 메시지(persistent message)

큐 관리자를 재시작해도 없어지지 않는 메시지입니다. 비지속 메시지도 참조하십시오.

개인 인증서

해당 개인 키를 소유하는 인증서입니다. 큐 관리자 또는 애플리케이션과 연관되어 있습니다.

PGM

PGM(Pragmatic General Multicast)을 참고하십시오.

PID

프로세스 ID(PID, process ID)를 참고하십시오.

ping

응답 수신을 기대하면서 ICMP(Internet Control Message Protocol) 반향 요청 패킷을 게이트웨이, 라우터 또는 호스트에 송신하는 명령입니다.

PKCS

공개 키 암호 표준(PKCS, Public Key Cryptography Standards)입니다. 암호화를 위한 표준 세트이며 다음과 같이 구성되어 있습니다:

- 7은 메시지를 위한 것입니다.
- 11은 하드웨어 보안 모듈을 위한 것입니다.
- 12는 키 저장소에서 사용되는 파일 형식을 위한 것입니다.

PKI

공개 키 인프라(PKI, public key infrastructure)를 참고하십시오.

일반 텍스트(plain text)

일반 텍스트(clear text)를 참고하십시오.

복구점(point of recovery)

z/OS용 WebSphere MQ에서, z/OS용 WebSphere MQ 페이지 세트 및 이 페이지 세트의 복구에 필요한 대응되는 로그 데이터 세트의 백업 사본 세트입니다. 이 백업 사본은 페이지 세트 유실(예: 페이지 세트 I/O 오류) 발생 시 잠재적인 재시작 지점을 제공합니다.

악성 메시지(poison message)

큐에서 수신하는 애플리케이션이 처리할 수 없는 잘못된 형식의 메시지입니다. 이 메시지는 입력 큐에 반복적으로 전달되고 애플리케이션에 의해 반복적으로 백아웃될 수 있습니다.

다형성(polymorphism)

메소드를 구현하는 클래스에 따라 메소드가 다르게 수행될 수 있게 하는 객체 지향 프로그래밍 특성입니다. 다형성은 상위 클래스의 메소드에 영향을 주지 않고도 서브클래스가 상속된 메소드를 대체할 수 있도록 허용합니다. 또한 다형성을 사용하면 클라이언트가 단일 인터페이스에서 둘 이상의 오브젝트 구현에 액세스할 수 있습니다.

PGM(Pragmatic General Multicast)

여러 수신자에게 동시에 신뢰할 수 있는 패킷 시퀀스를 제공하는 신뢰할 수 있는 멀티캐스트 전송 프로토콜입니다.

강제 종료(preemptive shutdown)

WebSphere MQ에서, 연결된 애플리케이션이 연결을 끊거나 현재 MQI 호출이 완료될 때까지 대기하지 않는 큐 관리자의 시스템 종료입니다. 즉시 종료, 정상 종료도 참조하십시오.

선호 컴퓨터(preferred computer)

Microsoft 클러스터 서버의 제어 하에서 실행되는 애플리케이션이 사용하는 기본 컴퓨터입니다. 다른 컴퓨터로의 장애 복구 후 MSCS는 선호 컴퓨터가 복구될 때까지 선호 컴퓨터를 모니터하고 선호 컴퓨터가 다시 정상적으로 실행되는 즉시 애플리케이션을 다시 선호 컴퓨터로 이동합니다.

프린시펄(principal)

다른 엔티티와 안전하게 통신할 수 있는 엔티티. 프린시펄은 액세스 권한을 정의하는 연관된 보안 컨텍스트에 의해 식별됩니다.

개별 정의 오브젝트(privately defined object)

로컬 정의 오브젝트(locally defined object)를 참고하십시오.

개별 메소드 및 인스턴스 데이터(private methods and instance data)

객체 지향 프로그래밍에서 동일한 클래스의 구현에만 액세스할 수 있는 메소드 및 인스턴스 데이터입니다.

프로세스 정의 오브젝트

WebSphere MQ 애플리케이션의 정의가 포함된 WebSphere MQ 오브젝트입니다. 예를 들어, 큐 관리자는 트리거 메시지에 대해 작업할 때 이 정의를 사용합니다.

프로세스 ID(PID, process ID)

프로세스를 나타내는 고유 ID입니다. 프로세스 ID는 양의 정수이며 프로세스 수명이 종료될 때까지 재사용되지 않습니다.

작성자(producer)

메시지를 작성하고 송신하는 애플리케이션입니다. 발행자, 메시지 생산자도 참조하십시오.

PCF(Programmable Command Format)

다음과 같은 애플리케이션에서 사용하는 WebSphere MQ 메시지의 유형입니다. 해당 애플리케이션은 지정된 큐 관리자의 시스템 명령 입력 큐에 PCF 명령을 배치하기 위한 사용자 관리 애플리케이션, 지정된 큐 관리자에서 PCF 명령의 결과를 가져오기 위한 사용자 관리 애플리케이션 및 이벤트 발생 알림으로 사용되는 큐 관리자입니다. WebSphere MQ 스크립트 명령도 참조하십시오.

PTF(Program Temporary Fix)

System i, System p 및 System z 제품의 경우, 라이선스를 보유한 모든 고객이 사용할 수 있는 단일 또는 다중 수정사항을 포함하는 패키지입니다. PTF는 결함을 분석하여 개선사항을 제공할 수 있습니다.

특성(property)

오브젝트에 대해 설명하는 오브젝트의 특성입니다. 특성은 변경하거나 수정할 수 있습니다. 특성은 특히 오브젝트 이름, 유형, 값 또는 작동에 대해 설명할 수 있습니다.

보호된 메소드 및 인스턴스 데이터(protected methods and instance data)

객체 지향 프로그래밍에서 동일하거나 파생된 클래스의 구현에만 액세스하거나 파생 클래스에서만 액세스할 수 있는 메소드 및 인스턴스 데이터입니다.

프록시 구독(proxy subscription)

프록시 구독은 하나의 큐 관리자에 발행된 토픽에 대해 다른 큐 관리자가 작성하는 구독입니다. 프록시 구독은 구독에서 구독하는 각 개별 토픽 문자열에 대해 큐 관리자 사이에서 플로우됩니다. 사용자는 프록시 구독을 명확하게 작성하지 않지만 큐 관리자는 사용자를 위해 프록시 구독을 명확하게 작성합니다.

PTF

[PTF\(Program Temporary Fix\)](#)를 참고하십시오.

공개 키

모든 사용자에게 알려진 키입니다. 이 키는 일반적으로 공개 키의 소유자를 지정하는 디지털 인증서에 임베드되어 있습니다.

공개 키 암호화(public key cryptography)

모든 사용자에게 알려진 공개 키와 메시지 수신자에게만 알려진 개인 키 또는 보안 키라는 두 개의 키를 사용하는 암호화 시스템입니다. 공개 키 및 개인 키는 하나의 키로 암호화된 항목이 대응되는 개인 키로만 복호화될 수 있는 방식으로 연관됩니다.

공개 키 인프라(PKI, public key infrastructure)

네트워크 트랜잭션에 관련된 각 당사자의 유효성을 확인하고 인증하는 디지털 인증서, 인증 기관 및 기타 등록대행 기관으로 구성된 시스템입니다.

공용 메소드 및 인스턴스 데이터(public methods and instance data)

객체 지향 프로그램에서 모든 클래스에 액세스할 수 있는 메소드 및 인스턴스 데이터입니다.

발행(Publish)

지정된 토픽에 대한 정보를 발행/구독 시스템의 큐 관리자가 사용할 수 있게 하는 것입니다.

발행자(publisher)

지정된 토픽에 대한 정보를 발행/구독 시스템의 브로커가 사용할 수 있게 하는 애플리케이션입니다.

발행/구독

발행 애플리케이션이 제공하는 정보가 인프라에 의해 해당 정보 유형에 관심을 표현한 모든 구독 애플리케이션에 전달되는 메시징 상호 작용의 한 유형입니다.

발행/구독 클러스터(publish/subscribe cluster)

완전히 상호 연결되어 있고 발행/구독 애플리케이션에 대한 다중 큐 관리자 네트워크의 부분을 형성하는 큐 관리자 세트입니다.

put

메시지 큐잉에서 MQPUT 또는 MQPUT1 호출을 사용하여 메시지를 큐에 배치하는 것입니다. [찾아보기](#) ([browse](#)), [가져오기](#) ([get](#))도 참조하십시오.

Q

큐(queue)

메시지 큐잉 애플리케이션의 메시지를 보유하는 오브젝트입니다. 큐는 큐 관리자가 소유하고 유지보수합니다.

큐 색인(queue index)

z/OS용 WebSphere MQ에서 큐에서의 MQGET 조작 속도를 증가시키는 데 사용할 수 있는 상관 ID 목록 또는 메시지 ID 목록입니다.

큐 관리자

애플리케이션에 큐잉 서비스를 제공하는 메시지 큐잉 시스템의 컴포넌트입니다.

큐 관리자 이벤트(queue manager event)

다음 중 하나를 표시하는 이벤트: 큐 관리자가 사용하는 자원과 관련하여 오류 조건이 발생했습니다. 예: 큐를 사용할 수 없거나 큐 관리자에서 중요한 변경사항이 발생했습니다. 예: 큐 관리자가 중지되거나 시작되었습니다.

큐 관리자 그룹(queue manager group)

클라이언트 채널 정의 테이블(CCDT)에서 서버에 대한 연결이 설정된 경우 클라이언트가 연결을 시도하는 큐 관리자의 그룹입니다.

큐 관리자 레벨 보안

z/OS용 WebSphere MQ에서 큐 관리자에 특정한 RACF 프로파일을 사용하여 수행되는 권한 검사입니다.

큐 관리자 세트(queue manager set)

WebSphere MQ Explorer의 큐 관리자 그룹으로, 사용자가 그룹의 모든 큐 관리자에서 조치를 수행할 수 있게 합니다.

큐 공유 그룹

z/OS용 WebSphere MQ에서, 공유 저장소에 저장된 오브젝트 정의의 단일 세트 및 커플링 기능에 저장된 공유 큐의 단일 세트에 액세스할 수 있는 동일한 Sysplex의 큐 관리자 그룹입니다. 공유 큐([shared queue](#))도 참조하십시오.

큐 공유 그룹 레벨의 보안(queue-sharing group level security)

z/OS용 WebSphere MQ에서 큐 공유 그룹의 모든 큐 관리자가 공유하는 RACF 프로파일을 사용하여 수행되는 권한 검사입니다.

일시정지(quietce)

활성 조작의 정상적인 완료를 허용한 후 프로세스를 종료하거나 시스템을 종료하는 것입니다.

정상 종료(quietced shutdown)

- 어댑터가 WebSphere MQ에서 연결을 끊을 경우 CICS 어댑터의 시스템 종료 유형이며, 현재 활성 태스크가 모두 완료된 경우에만 해당됩니다. [강제 종료\(forced shutdown\)](#)도 참조하십시오.
- WebSphere MQ에서 연결된 모든 애플리케이션이 연결을 끊을 수 있도록 하는 큐 관리자의 시스템 종료입니다. 즉시 종료, [강제 종료](#)의 내용도 참조하십시오.

정지중(Quiescing)

WebSphere MQ에서 중지되기 전에 큐 관리자의 상태입니다. 이 상태에서는 프로그램이 처리를 완료할 수는 있지만 새 프로그램은 시작할 수 없습니다.

쿼럼 디스크(quorum disk)

클러스터 복구 로그를 저장하고 서버의 작동 여부를 판별하기 위해 Microsoft 클러스터 서버에서 독점적으로 액세스하는 디스크입니다. 한 번에 하나의 서버만 쿼럼 디스크를 소유할 수 있습니다. 클러스터의 서버는 소유권에 대해 협상할 수 있습니다.

R

RACF

[RACF\(Resource Access Control Facility\)](#)를 참고하십시오.

RAID

[RAID\(Redundant Array of Independent Disks\)](#)를 참고하십시오.

RBA

[RBA\(Relative Byte Address\)](#)를 참고하십시오.

리턴 코드

[리턴 코드\(RC, return code\)](#)를 참고하십시오.

미리 읽기

애플리케이션이 메시지를 요청하기 전에 메시지를 클라이언트에 송신할 수 있게 하는 옵션입니다.

이유 코드(reason code)

MQI(Message Queue Interface) 호출의 실패 또는 부분적인 성공에 대한 이유를 설명하는 리턴 코드입니다.

수신 엑시트

메시지 채널 에이전트(MCA)가 통신 수신 후 제어를 다시 확보하고 통신 연결에서 데이터 단위를 수신한 직후 호출되는 채널 엑시트 프로그램의 한 유형입니다. [송신 엑시트\(send exit\)](#)도 참조하십시오.

수신자 채널(receiver channel)

메시지 큐잉에서 송신자 채널에 응답하고 통신 링크에서 메시지를 가져와서 로컬 큐에 넣는 채널입니다.

복구 로그(recovery log)

z/OS용 WebSphere MQ에서 메시지, 큐 및 WebSphere MQ 서브시스템을 복구하는 데 필요한 정보가 포함된 데이터 세트입니다. [아카이브 로그](#)도 참조하십시오.

RTM(Recovery Termination Manager)

종료하는 기능과 연관된 복구 루틴에 제어를 전달하여 태스크의 모든 정상 및 비정상 종료를 핸들링하는 프로그램입니다.

RAID(Redundant Array of Independent Disks)

호스트에 하나 이상의 논리 디스크 드라이브 이미지를 제공하는 둘 이상의 물리적 디스크 드라이브의 콜렉션입니다. 물리적 디바이스 장애가 발생하는 경우에는 데이터 중복으로 인해 배열의 다른 디스크 드라이브에서 데이터를 읽거나 다시 생성할 수 있습니다.

참조 메시지(reference message)

전송할 데이터의 일부를 참조하는 메시지입니다. 참조 메시지는 데이터를 메시지와 연결하고 분리하여 데이터를 큐에 저장하지 않고 전송할 수 있게 하는 메시지 엑시트 프로그램에 의해 핸들링됩니다.

레지스트리(registry)

사용자, 시스템 및 소프트웨어에 대한 액세스 및 구성 정보가 포함된 저장소입니다.

레지스트리 편집기(Registry Editor)

Windows에서 사용자가 레지스트리를 편집할 수 있도록 하는 프로그램 항목입니다.

레지스트리 구조(registry hive)

Windows 시스템에서 레지스트리에 저장된 데이터의 구조입니다.

RBA(Relative Byte Address)

속하는 데이터 세트 또는 파일에 할당되는 스토리지 공간의 시작과 데이터 레코드 또는 제어 간격의 오프셋입니다.

신뢰할 수 있는 멀티캐스트 메시징(RMM, reliable multicast messaging)

메시지 지향 미들웨어 발행/구독 방식으로 일대다 데이터 전달 또는 다대다 데이터 교환을 위해 설계된 처리량이 많고 지연 시간이 짧은 전송 구조입니다. RMM은 IP 멀티캐스트 인프라를 활용하여 확장 가능한 자원 보존 및 시기적절한 정보 배포를 보장할 수 있습니다.

리모트 큐

리모트 큐 관리자에 속하는 큐입니다. 프로그램은 메시지를 리모트 큐에 넣을 수는 있지만 리모트 큐에서 메시지를 가져올 수는 없습니다. [로컬 큐\(local queue\)](#)도 참조하십시오.

리모트 큐 관리자

프로그램과 동일한 시스템에서 실행 중인 경우에도 프로그램이 연결되어 있지 않은 큐 관리자입니다. [로컬 큐 관리자](#)도 참조하십시오.

리모트 큐 오브젝트(remote queue object)

로컬 큐 관리자에 속한 WebSphere MQ 오브젝트입니다. 이 오브젝트는 다른 큐 관리자가 소유하는 큐의 속성을 정의합니다. 큐 관리자 알리어싱 및 응답 대상 큐 알리어싱에도 사용됩니다.

리모트 큐잉

메시지 큐잉에서 애플리케이션이 다른 큐 관리자에 속하는 큐에 메시지를 넣을 수 있게 하는 서비스를 제공하는 것입니다.

응답 메시지(reply message)

요청 메시지에 대한 응답에 사용되는 메시지의 유형입니다. [보고 메시지](#), [요청 메시지](#)도 참조하십시오.

응답 대상 큐(reply-to queue)

MQPUT 호출을 발행한 프로그램이 응답 메시지 또는 보고 메시지를 송신하려는 큐의 이름입니다.

보고 메시지(report message)

다른 메시지에 대한 정보를 제공하는 메시지의 유형입니다. 보고 메시지는 메시지가 전달되었거나 목적지에 도착했거나 만료되었거나 어떤 이유로 처리될 수 없음을 나타낼 수 있습니다. [응답 메시지\(reply message\)](#), [요청 메시지\(request message\)](#)도 참조하십시오.

저장소(repository)

클러스터의 멤버인 큐 관리자에 대한 정보의 콜렉션입니다. 이 정보에는 큐 관리자 이름, 해당 위치, 해당 채널 및 큐 관리자 이름이 호스팅하는 큐가 포함됩니다.

저장소 큐 관리자(repository queue manager)

클러스터에 대한 정보의 전체 저장소를 호스팅하는 큐 관리자입니다.

요청자 채널

메시지 큐잉에서 서버 채널의 조작을 시작하기 위해 로컬로 시작할 수 있는 채널입니다. [서버 채널\(server channel\)](#)도 참조하십시오.

요청 메시지(request message)

다른 프로그램에서 응답을 요청하는 데 사용되는 메시지의 유형입니다. [응답 메시지\(reply message\)](#), [보고 메시지\(report message\)](#)도 참조하십시오.

요청/응답(request/reply)

다른 애플리케이션에서 응답을 요청하는 데 요청 메시지가 사용되는 메시징 애플리케이션의 유형입니다. [데 이터그램](#)도 참조하십시오.

RESLEVEL

z/OS용 WebSphere MQ에서 API 자원 보안이 확인된 사용자 ID 수를 제어하는 옵션입니다.

해석 경로(resolution path)

애플리케이션이 MQOPEN 호출에 대한 입력에서 알리어스 또는 리모트 큐를 지정할 때 열리는 큐의 세트입니다.

자원(resource)

작업, 태스크 또는 실행 중인 프로그램에 필요한 컴퓨팅 시스템 또는 운영 체제의 기능입니다. 자원에는 주기역장치, 입/출력 디바이스, 처리 장치, 데이터 세트, 파일, 라이브러리, 폴더, 애플리케이션 서버 및 제어 또는 처리 프로그램이 포함됩니다.

RACF(resource access control facility)

시스템에 대한 사용자 식별, 시스템의 사용자 확인, 보호 자원에 대한 액세스 권한 부여, 권한이 없는 사용자의 시스템 진입 시도 로깅 및 보호 자원에 대한 액세스 로깅을 통해 액세스 제어를 제공하는 IBM 라이선스가 있는 프로그램입니다.

자원 어댑터(resource adapter)

애플리케이션 서버에서 실행 중인 JMS 애플리케이션 및 메시지 구동 Bean이 WebSphere MQ 큐 관리자의 자원에 액세스할 수 있도록 허용하는 Java Enterprise Edition 커넥터 아키텍처의 구현입니다.

자원 관리자

메모리 버퍼 및 데이터 세트와 같은 공유 자원에 대한 액세스를 관리하고 제어하는 애플리케이션, 프로그램 또는 트랜잭션입니다. WebSphere MQ, CICS 및 IMS는 자원 관리자입니다.

RRS(Resource Recovery Services)

동기점 관리자를 사용하여 참여 중인 자원 관리자 간의 변경사항을 조정하는 z/OS의 컴포넌트입니다.

응답자(responder)

분산 큐잉에서 다른 시스템의 네트워크 연결을 요청하는 프로그램입니다. [시작기\(initiator\)](#)도 참조하십시오.

Resynch

WebSphere MQ에서 메시지 전송을 다시 시작하지 않고 인다우트 상태 메시지를 시작하여 해석하기 위해 채널의 방향을 지시하는 데 필요한 옵션입니다.

리턴 코드(RC, return code)

처리 결과를 표시하기 위해 프로그램에 의해 리턴되는 값입니다. 완료 코드 및 이유 코드가 리턴 코드의 예입니다.

송신자로 돌아가기(return-to-sender)

메시지를 전달할 수 없는 MCA에 사용할 수 있는 옵션입니다. MCA는 메시지를 다시 진원지에 송신할 수 있습니다.

RSA 알고리즘(Rivest-Shamir-Adleman algorithm)

RSA Data Security, Inc에서 개발하여 SSL의 IBM 구현에 사용되는 공개 키 암호화 기술입니다.

RMM

[신뢰할 수 있는 멀티캐스트 메시징\(reliable multicast messaging\)](#)을 참고하십시오.

롤백(rollback)

[백아웃\(backout\)](#)을 참고하십시오.

루트 인증서(root certificate)

체인의 맨 위 인증서입니다. 이 인증서는 자체 서명 인증서인 경우에는 다른 인증서에 서명하는 용도로만 사용됩니다. [자체 서명 인증서](#)도 참조하십시오.

RRS

RRS(Resource Recovery Services)를 참고하십시오.

RSA

RSA 알고리즘(Rivest-Shamir-Adleman algorithm)을 참고하십시오.

RTM

RTM(Recovery Termination Manager)을 참고하십시오.

규칙 테이블(rules table)

데드-레터 큐 핸들러가 데드-레터 큐(DLQ)의 메시지에 적용하는 하나 이상의 규칙이 포함된 제어 파일입니다.

S

SP2(Scalable Parallel 2)

IBM의 병렬 UNIX 시스템: 고속 네트워크의 실제 병렬 AIX 시스템입니다.

SDK

소프트웨어 개발 키트(SDK, software development kit)을 참고하십시오.

SDWA

SDWA(System Diagnostic Work Area)를 참고하십시오.

SECMEC

보안 메커니즘(SECMEC, security mechanism)을 참고하십시오.

SSL(Secure Sockets Layer)

통신 개인정보 보호를 제공하는 보안 프로토콜입니다. SSL을 사용하면 도청, 도용 및 메시지 위조를 방지하도록 설계된 방식으로 클라이언트/서버 애플리케이션이 통신할 수 있습니다. [인증 기관도 참조하십시오.](#)

SEI(Security Enabling Interface)

권한을 확인하고 사용자 ID를 제공하거나 인증을 수행하는 고객 또는 벤더 작성 프로그램에 맞는 WebSphere MQ 인터페이스입니다. WebSphere MQ 프레임워크의 일부입니다.

보안 엑시트

채널 시동 시 초기 데이터 협상이 완료된 후 즉시 호출되는 채널 엑시트 프로그램입니다. 보안 엑시트는 일반적으로 쌍으로 작동하며 메시지 채널과 MQI 채널 모두에서 호출될 수 있습니다. 보안 엑시트의 1차 목적은 채널의 양쪽 끝에 있는 메시지 채널 에이전트(MCA)가 해당 파트너를 인증할 수 있게 하는 것입니다.

보안 ID(SID, security identifier)

Windows 시스템에서, 사용자가 정의된 Windows 보안 계정 관리자 데이터베이스의 전체 사용자 계정 세부 사항을 식별하는 사용자 ID를 보완합니다.

보안 메커니즘(SECMEC, security mechanism)

보안 서비스를 구현하는 데 사용되는 기술 도구 또는 기술입니다. 메커니즘은 특정 서비스를 제공하기 위해 자체적으로 또는 다른 메커니즘과 결합하여 작동될 수 있습니다. 보안 메커니즘의 예로는 액세스 제어 목록(ACL), 암호화, 디지털 서명이 있습니다.

보안 메시지(security message)

서로 통신하기 위해 채널의 양쪽 끝에서 호출되는 보안 엑시트에서 송신하는 메시지 중 하나입니다. 보안 메시지의 형식은 정의되어 있지 않고 사용자가 결정합니다.

보안 서비스(security service)

자원을 보호하는 컴퓨터 시스템 내의 서비스입니다. 액세스 제어는 보안 서비스의 예입니다.

보안 지원 제공자 인터페이스(SPI, Security Support Provider Interface)

네트워크로 연결된 애플리케이션이 몇몇 보안 지원 제공자(SSP) 중 하나를 호출하여 인증된 연결을 설정하고 해당 연결을 통해 데이터를 안전하게 교환하게 하는 수단입니다. 이는 Windows 시스템에서 사용할 수 있습니다.

자체 서명 인증서(self-signed certificate)

인증서의 디지털 서명은 인증서의 공개 키에 대응되는 개인 키를 사용하여 생성됩니다.

세그먼트화(segmentation)

큐 관리자, 큐 또는 애플리케이션에 대해 너무 큰 메시지를 다수의 작은 물리적 메시지로 분할한 후 수신하는 큐 관리자 또는 애플리케이션이 이를 리어셈블링하는 것입니다.

SEI

SEI(Security Enabling Interface)를 참고하십시오.

선택기

데이터 항목의 ID입니다. WebSphere MQAI(MQ Administration Interface)에서 2개 유형의 선택자(사용자 선택자 및 시스템 선택자)가 있습니다.

세마포어(semaphore)

UNIX 및 Linux 시스템에서 신호의 기능을 확장하는 두 프로세스 간의 일반적인 통신 방법입니다.

송신자 채널

메시지 큐잉에서 전송을 시작하고 전송 큐에서 메시지를 제거하고 통신 링크를 통해 해당 메시지를 수신자 또는 요청자 채널로 이동하는 채널입니다.

송신 엑시트(send exit)

통신 연결을 통해 데이터 단위를 송신하기 위해 메시지 채널 에이전트(MCA)가 통신 송신을 발행하기 직전에 호출되는 채널 엑시트 프로그램의 한 유형입니다. 수신 엑시트도 참조하십시오.

SPX 프로토콜(Sequenced Packet Exchange protocol)

네트워크의 두 노드 사이에 연결 지향 서비스를 제공하고 클라이언트/서버 애플리케이션이 주로 사용하는 세션 지향 네트워크 프로토콜입니다. 이 프로토콜은 IPX(Internet Packet Exchange) 프로토콜에 의존하고 플로우 제어 및 오류 복구를 제공하며 물리적 네트워크의 신뢰성을 보장합니다.

순서 번호 줄바꿈 값(sequence number wrap value)

WebSphere MQ에서, 통신 링크의 양 끝이 현재 메시지 시퀀스 번호를 동시에 재설정하도록 하는 방법입니다. 순서 번호와 함께 메시지를 전송하면 수신하는 채널이 메시지를 저장할 때 메시지 순서를 재설정할 수 있습니다.

직렬화(serialization)

객체 지향 프로그래밍에서 프로그램 메모리로부터 통신 미디어에 순차적으로 데이터를 기록하는 것입니다.

SERVER

1. 원격 워크스테이션에서 실행되는 클라이언트 애플리케이션에 큐 서비스를 제공하는 큐 관리자입니다.
2. 다른 소프트웨어 프로그램이나 다른 컴퓨터에 서비스를 제공하는 소프트웨어 프로그램이나 컴퓨터입니다. 클라이언트(client)도 참조하십시오.

서버 채널

메시지 큐잉에서 요청자 채널에 응답하고 전송 큐에서 메시지를 제거하여 통신 링크를 통해 요청자 채널로 이동하는 채널입니다. 요청자 채널도 참조하십시오.

서버 연결 채널 유형(server-connection channel type)

큐 관리자를 실행하는 서버와 연관된 MQI 채널 정의의 유형입니다. 클라이언트 연결 채널 유형(client-connection channel type)도 참조하십시오.

서비스 간격(service interval)

서비스 간격 이벤트에 대한 조건이 충족되었는지 여부를 결정하기 위해 큐 관리자가 넣기 또는 가져오기와 후속 가져오기 사이의 경과 시간을 비교하는 시간 간격입니다. 큐의 서비스 간격은 큐 속성에 의해 지정됩니다.

서비스 간격 이벤트(service interval event)

서비스 간격과 관련된 이벤트입니다.

서비스 오브젝트(service object)

큐 관리자가 시작될 때 추가적인 프로세스를 시작하고 큐 관리자가 중지될 때 프로세스를 중지할 수 있는 오브젝트입니다.

세션

두 요소가 통신하여 세션 지속 기간 동안 데이터를 교환할 수 있게 하는 네트워크의 두 스테이션, 소프트웨어 프로그램 또는 디바이스 간 논리적 연결 또는 가상 연결입니다.

세션 ID(session ID)

z/OS용 WebSphere MQ에서 전송 큐의 메시지를 링크로 옮길 때 메시지 채널 에이전트에서 사용할 통신 링크를 정의하는 CICS 고유 ID입니다.

세션 레벨의 인증(session-level authentication)

SNA(Systems Network Architecture)에서 두 논리 장치(LU)가 세션을 활성화하는 동안 서로를 인증할 수 있게 하는 세션 레벨의 보안 프로토콜입니다. 세션 레벨의 인증은 LU-LU 확인으로도 알려져 있습니다.

세션 레벨의 암호화(session-level cryptography)

SNA(Systems Network Architecture)에서 두 논리 장치(LU) 사이의 세션에서 흐르는 데이터를 암호화하고 복호화하는 방법입니다.

공유 인바운드 채널(shared inbound channel)

z/OS용 WebSphere MQ에서 그룹 포트를 사용하여 리스너에서 시작한 채널입니다. 공유 채널의 채널 정의는 페이지 세트 영(0)(개인용) 또는 공유 저장소(글로벌)에 저장될 수 있습니다.

공유 아웃바운드 채널(shared outbound channel)

z/OS용 WebSphere MQ에서 공유 전송 큐의 메시지를 옮기는 채널입니다. 공유 채널의 채널 정의는 페이지 세트 영(0)(개인용) 또는 공유 저장소(글로벌)에 저장될 수 있습니다.

공유 큐

z/OS용 WebSphere MQ에서 로컬 큐의 유형입니다. 큐에 있는 메시지는 커플링 기능에 저장되며 큐 공유 그룹에 있는 하나 이상의 큐 관리자가 액세스할 수 있습니다. 큐의 정의는 공유 저장소에 저장됩니다. 큐 공유 그룹도 참조하십시오.

공유 저장소

z/OS용 WebSphere MQ에서 글로벌로 정의된 오브젝트 정의를 보유하는 데 사용되는 공유 Db2 데이터베이스입니다.

대화 공유

채널 인스턴스를 공유할 둘 이상의 대화 또는 채널 인스턴스를 공유하는 대화를 위한 기능입니다.

셸(shell)

사용자와 운영 체제 사이의 소프트웨어 인터페이스입니다. 셸은 일반적으로 명령행 셸(운영 체제에 명령행 인터페이스를 제공함)과 그래픽 셸(그래픽 사용자 인터페이스(GUI)를 제공함)이라는 두 가지 범주 중 하나로 구분됩니다.

SID

보안 ID(security identifier)를 참조하십시오.

신호(signal)

시스템에서 발생하는 이벤트를 프로세스에 알리거나 이 이벤트가 프로세스에 영향을 미치는 데 사용하는 메커니즘입니다. 이러한 이벤트의 예로는 프로세스에 의한 특정 조치 및 하드웨어 예외가 있습니다.

신호보내기(signaling)

z/OS용 WebSphere MQ 및 Windows용 WebSphere MQ에서 예상된 메시지가 큐에 도착할 때 운영 체제가 프로그램에 알릴 수 있도록 허용하는 기능입니다.

서명(signature)

메소드와 연관된 유형의 콜렉션입니다. 서명에는 리턴값의 유형이 포함되며 해당되는 경우에는 메소드의 인수 각각의 유형, 순서 및 번호도 포함됩니다.

서명자 인증서

암호화 또는 서명에 사용되는 인증서입니다.

단일 인스턴스 큐 관리자(single instance queue manager)

다중 인스턴스를 가지고 있지 않은 큐 관리자입니다. 멀티 인스턴스 큐 관리자도 참조하십시오.

단일 로깅(single logging)

각 변경이 하나의 데이터 세트에만 기록되는 z/OS용 WebSphere MQ 활동을 기록하는 방법입니다. 이중 로깅도 참조하십시오.

1단계 백아웃(single-phase backout)

진행 중인 조치를 완료할 수 없고 해당 조치의 일부인 모든 변경사항을 실행 취소해야 하는 메소드입니다.

1단계 커밋(single-phase commit)

프로그램이 커밋 자원에 대한 업데이트를 다른 자원 관리자가 제어하는 자원에 대해 프로그램이 작성한 업데이트와 통합하지 않고 해당 업데이트를 커밋할 수 있는 메소드입니다.

SIT

SIT(System Initialization Table)를 참조하십시오.

SMF

시스템 관리 기능(SMF, System Management Facilities)을 참조하십시오.

SNA

SNA(Systems Network Architecture)를 참조하십시오.

소프트웨어 개발 키트(SDK, software development kit)

특정 컴퓨터 언어 또는 특정 운영 환경을 위한 소프트웨어 개발을 지원하는 도구, API 및 문서 세트입니다.

소스 큐 관리자(source queue manager)

로컬 큐 관리자(local queue manager)를 참고하십시오.

SP2

SP2(Scalable Parallel 2)를 참고하십시오.

SPX

SPX 프로토콜(Sequenced Packet Exchange protocol)을 참고하십시오.

SSI

보안 지원 제공자 인터페이스(SSI, Security Support Provider Interface)를 참고하십시오.

SSL

SSL(Secure Sockets Layer)을 참고하십시오.

SSLPeer

발행인의 값은 원격 개인 인증서의 식별 이름을 나타냅니다.

SSL 또는 TLS 클라이언트(SSL or TLS client)

연결을 시작하는 한쪽 끝입니다. 큐 관리자로부터의 아웃바운드 채널 하나도 SSL 또는 TLS 클라이언트입니다.

대기 큐 관리자 인스턴스(standby queue manager instance)

활성 인스턴스로부터 인계할 준비가 된 실행 중인 다중 인스턴스 큐 관리자의 인스턴스입니다. 다중 인스턴스 큐 관리자의 대기 인스턴스는 하나 이상이 있습니다.

스탠자

파일에서 함께 공용 기능을 가지거나 시스템의 부분을 정의하는 행 그룹입니다. 스탠자는 일반적으로 빈 줄 또는 콜론으로 구분되며 각각의 스탠자에는 이름이 있습니다.

별모양 통신 네트워크(star-connected communications network)

모든 노드가 중앙 노드에 연결된 네트워크입니다.

스토리지 클래스

z/OS용 WebSphere MQ에서 특정한 큐의 메시지를 보유할 페이지 세트입니다. 스토리지 클래스는 큐가 정의될 때 지정됩니다.

저장 후 전달(store and forward)

데이터 네트워크의 패킷, 메시지 또는 프레임을 해당 목적지로 재전송하기 전에 임시로 저장하는 것입니다.

스트리밍(streaming)

객체 지향 프로그래밍에서 클래스 정보 및 오브젝트 인스턴스 데이터를 직렬화하는 것입니다.

구독(subscribe)

토픽에 대한 정보를 요청하는 것입니다.

서브시스템(subsystem)

z/OS에서 하나 또는 다수의 기능을 수행하지만 요청을 받아야 작업을 수행하는 서비스 제공자입니다. 예를 들어, 각 z/OS용 WebSphere MQ 큐 관리자 또는 z/OS용 Db2 데이터베이스 관리 시스템의 인스턴스는 z/OS 서브시스템입니다.

SVC(SuperVisor Call)

지시사항에 따라 특정 서비스를 수행할 수 있도록 실행 중인 프로그램을 인터럽트하고 감독자에게 제어를 전달하는 지시사항입니다.

SVC

SVC(SuperVisor Call)를 참고하십시오.

전환(switchover)

활성 다중 인스턴스 큐 관리자 인스턴스에서 대기 인스턴스로의 변경입니다. 운영자가 의도적으로 활성 다중 인스턴스 큐 관리자 인스턴스를 중지하면 전환이 발생합니다.

스위치 프로파일(switch profile)

z/OS용 WebSphere MQ에서 WebSphere MQ가 시작되거나 보안 새로 고치기 명령 실행 시 사용되는 RACF 프로파일입니다. WebSphere MQ에서 발견하는 각 스위치 프로파일에서는 지정된 자원 확인을 끕니다.

대칭 키 암호화(symmetric key cryptography)

메시지의 송신자와 수신자가 메시지를 암호화하거나 복호화하는 데 사용되는 하나의 공용 보안 키를 공유하는 암호화 시스템입니다. 이 시스템은 인증을 제공하지 않습니다. [비대칭 키 암호화\(asymmetric key cryptography\)](#)도 참조하십시오.

증상 문자열(symptom string)

IBM 소프트웨어 지원 데이터베이스를 검색하기 위해 설계된 구조화된 형식에 표시된 진단 정보입니다.

동기 메시징(synchronous messaging)

프로그램이 메시지 큐에 메시지를 넣은 후 자체 처리를 계속하기 전에 해당 메시지에 대한 응답을 기다리는 프로그램 간 통신 메소드입니다. [비동기 메시징](#)도 참조하십시오.

동기점

트랜잭션 처리 중 보호된 자원이 일치하는 지점입니다.

SYSPLX

특정 멀티시스템 하드웨어 컴포넌트 및 소프트웨어 서비스를 통해 서로 통신하는 z/OS 시스템 세트.

시스템 백(system bag)

MQAI에 의해 작성되는 데이터 백의 한 유형입니다.

시스템 제어 명령(system control commands)

버퍼 풀, 스토리지 클래스 및 페이지 세트와 같은 플랫폼별 엔티티를 조작하는 데 사용되는 명령입니다.

SDWA(System Diagnostic Work Area)

z/OS 환경에서 프로그램 또는 하드웨어 오류를 설명하는 SYS1.LOGREC 항목에 기록되는 데이터입니다.

SIT(System Initialization Table)

시작 시 CICS에서 사용하는 매개변수가 포함된 테이블입니다.

시스템 항목(system item)

MQAI에 의해 작성되는 데이터 항목의 한 유형입니다.

시스템 관리 기능(SMF, System Management Facilities)

여러 시스템 및 작업 관련 정보를 수집하여 기록하는 z/OS의 컴포넌트입니다.

시스템 선택자(system selector)

WebSphere MQAI(MQ Administration Interface)에서 작성 시 데이터 백에 포함되는 시스템 항목 ID입니다.

SNA(Systems Network Architecture)

네트워크의 구성 및 작동을 제어하고 이를 통해 정보를 전송하는 데 필요한 논리적 구조, 형식, 프로토콜 및 작동 순서에 대한 설명입니다.

T**도용(tampering)**

전송 중인 정보가 변경되거나 바뀐 후 수신자에게 송신되는 통신 보안 침해입니다. [도청](#), [위장](#)도 참조하십시오.

THLQUAL(target library high-level qualifier)

z/OS 대상 데이터 세트 이름의 상위 레벨 규정자입니다.

대상 큐 관리자(target queue manager)

[리모트 큐 관리자\(remote queue manager\)](#)를 참고하십시오.

TCB(Task Control Block)

서브시스템에 연결된 주소 공간의 태스크에 대한 정보를 통신하는 데 사용되는 z/OS 제어 블록입니다.

태스크 전환(task switching)

여러 태스크 사이에서 I/O 조작 및 처리가 겹치는 것입니다.

TCB

[TCB\(Task Control Block\)](#)를 참고하십시오.

TCP

[TCP\(Transmission Control Protocol\)](#)를 참고하십시오.

TCP/IP

[TCP/IP\(Transmission Control Protocol/Internet Protocol\)](#)를 참고하십시오.

기술 노트(technote)

단일 토픽에 대한 짧은 문서입니다.

텔레메트리 채널(telemetry channel)

텔레메트리 채널은 WebSphere MQ의 큐 관리자와 MQTT 클라이언트 간의 통신 링크입니다. 각 채널에는 하나 이상의 텔레메트리 디바이스가 연결되어 있을 수 있습니다.

텔레메트리 고급 클라이언트(telemetry advance client)

고급 텔레메트리 클라이언트는 기본 WebSphere MQ 설치의 mqxr 서브폴더에 설치되어 있습니다. 이는 여러 MQTT 클라이언트가 자신에 연결될 수 있도록 허용하는 소형 풋프린트, MQTT 서버이며 WebSphere MQ에 대한 업링크나 브릿지를 제공합니다. 고급 클라이언트는 업링크 연결이 끊어질 때 클라이언트를 대신하여 메시지를 시작할 수 있습니다.

텔레메트리 클라이언트(telemetry client)

텔레메트리 클라이언트는 기본 WebSphere MQ 설치의 mqxr 서브폴더 내에 설치된 MQTT 클라이언트입니다. 텔레메트리 클라이언트는 MQTT 프로토콜을 사용하여 MQ에 연결합니다.

텔레메트리(MQXR) 서비스

MQTT 프로토콜을 대신하여 서버를 핸들링하는 MQ 서비스입니다(MWTT 서버 참조). 텔레메트리(MQXR) 서비스는 텔레메트리 채널을 호스트합니다.

임시 동적 큐(temporary dynamic queue)

닫힐 때 삭제되는 동적 큐입니다. 임시 동적 큐는 큐 관리자가 실패하면 복구되지 않으므로 비지속 메시지만 포함할 수 있습니다. 영구적 동적 큐(permanent dynamic queue)도 참조하십시오.

테라스페이스(teraspace)

프로세스에 개인용 스토리지를 제공하는 1테라바이트의 임시 스토리지 영역입니다.

중단 알림(termination notification)

CICS 서브시스템이 z/OS용 WebSphere MQ에 정상적으로 연결되는 경우에 활성화되는 보류 이벤트입니다.

THLQUAL

THLQUAL(target library high-level qualifier)을 참고하십시오.

스레드

프로세스의 제어를 받는 컴퓨터 명령어 스트림입니다. 일부 운영 체제에서는 스레드가 프로세스의 가장 작은 조작 단위입니다. 여러 스레드를 동시에 실행하여 다양한 작업을 수행할 수 있습니다.

TID

트랜잭션 ID(transaction identifier)를 참고하십시오.

시간 독립 메시징(time-independent messaging)

비동기 메시징(asynchronous messaging)을 참고하십시오.

TLS

전송 계층 보안(Transport Layer Security) - SSL에 대한 후속입니다.

TMF

TMF(Transaction Manager Facility)를 참고하십시오.

TMI

TMI(trigger monitor interface)를 참고하십시오.

TP

트랜잭션 프로그램(TP, transaction program)을 참고하십시오.

추적

컴퓨터 프로그램 또는 트랜잭션의 처리 레코드입니다. 추적에서 수집되는 정보는 문제점 및 성능을 평가하는데 사용할 수 있습니다.

트랜잭션 ID(transaction ID)

트랜잭션 ID(transaction identifier)를 참고하십시오.

트랜잭션 ID(TID, transaction identifier, transaction ID, XID)

트랜잭션에 지정되고 해당 트랜잭션과 연관된 조치를 식별하는 데 사용되는 고유 이름입니다.

트랜잭션 관리자(transaction manager)

글로벌 트랜잭션을 관리하고 의사결정을 통합하여 글로벌 트랜잭션을 커밋하거나 롤백하여 자원 관리자의 활동을 통합하는 소프트웨어 단위입니다.

TMF(Transaction Manager Facility)

HP Integrity NonStop Server에 대해 IBM WebSphere MQ 에서, 서브시스템은 비즈니스 트랜잭션 및 데이터베이스의 무결성을 보호합니다. 종종 NonStop 트랜잭션 관리자/MP와 동의어로 사용됩니다.

트랜잭션 프로그램(TP, transaction program)

SNA 네트워크에서 트랜잭션을 처리하는 프로그램입니다.

TCP(Transmission Control Protocol)

네트워크 간 프로토콜을 위해 IETF(Internet Engineering Task Force) 표준을 따르는 네트워크와 인터넷에서 사용되는 통신 프로토콜입니다. TCP는 패킷 교환 통신 네트워크 및 이 네트워크의 상호연결된 시스템에서 신뢰할 수 있는 호스트 간 프로토콜을 제공합니다. [IP\(Internet Protocol\)](#)도 참조하십시오.

TCP/IP(Transmission Control Protocol/Internet Protocol)

다양한 유형의 상호연결된 네트워크를 통해 애플리케이션 사이에 신뢰할 수 있는 엔드-투-엔드 연결을 제공하는 업계 표준 비독점 통신 프로토콜 세트입니다.

전송 프로그램(transmission program)

[메시지 채널 에이전트\(MCA, message channel agent\)](#)를 참조하십시오.

전송 큐

리모트 큐 관리자를 목적지로 하는 준비된 메시지가 임시로 저장되는 로컬 큐입니다.

트리거된 큐(triggered queue)

트리거가 설정되어 있고 트리거 조건이 충족된 경우에 트리거 메시지의 기록을 요구하는 로컬 큐입니다.

트리거 이벤트(trigger event)

큐 관리자가 이니시에이션 큐에서 트리거 메시지를 작성하게 하는 이벤트입니다(예: 큐에 메시지 도착).

트리거

WebSphere MQ에서 큐의 사전 결정된 조건에 맞으면 큐 관리자가 자동으로 애플리케이션을 시작할 수 있도록 하는 기능입니다.

트리거 메시지

트리거 모니터가 시작할 프로그램에 대한 정보가 포함된 메시지입니다.

트리거 모니터

하나 이상의 이니시에이션 큐를 제공하는 지속적으로 실행되는 애플리케이션입니다. 트리거 메시지가 이니시에이션 큐에 도착하면 트리거 모니터가 메시지를 검색합니다. 트리거 모니터는 트리거 메시지의 정보를 사용하여 트리거 이벤트가 발생한 큐를 제공하는 프로세스를 시작합니다.

TMI(trigger monitor interface)

고객 또는 벤더 작성 트리거 모니터 프로그램에 맞는 WebSphere MQ 인터페이스입니다. WebSphere MQ 프레임워크의 일부입니다.

신뢰 저장소(trust store)

원격 시스템의 인증서를 유효성 검증하기 위해 CA 인증서가 놓여지는 위치입니다. [키 저장소](#)도 참조하십시오.

양방향 인증(two way authentication)

이 인증 방법에서는 큐 관리자와 클라이언트가 서로에게 인증서를 제공합니다. 상호 인증으로도 알려져 있습니다.

2단계 커미트(two-phase commit)

복구 가능한 자원과 외부 서브시스템을 커미트하는 2단계 프로세스입니다. 첫 번째 단계 동안 데이터베이스 관리자 서브시스템은 커미트 준비가 되었음을 확인하기 위해 플링됩니다. 모든 서브시스템이 긍정적으로 응답하면 데이터베이스 관리자는 커미트하도록 지시합니다.

유형

데이터의 내부 형식을 지정하고 데이터를 사용하는 방법을 판별하는 특성입니다.

U**UDP**

[UDP\(User Datagram Protocol\)](#)를 참조하십시오.

비인가 액세스(unauthorized access)

권한 없이 컴퓨터 시스템 내 자원에 대한 액세스를 확보하는 것입니다.

전달되지 않은 메시지 큐(undelivered message queue)

데드-레터 큐(DLQ, dead-letter queue)를 참고하십시오.

실행 취소/다시 실행 레코드(undo/redo record)

복구에서 사용되는 로그 레코드입니다. 레코드의 다시 실행 파트는 WebSphere MQ 오브젝트에 작성할 변경 사항을 설명합니다. 실행 취소 부분은 작업이 커밋되지 않은 경우 변경사항을 백아웃하는 방법을 기술합니다.

복구 단위

단일 자원 관리자 내에서 복구 가능한 일련의 조작(예: z/OS용 DB2의 인스턴스)입니다. 작업 단위도 참조하십시오.

작업 단위(UOW, unit of work)

일관성의 두 지점 사이에서 애플리케이션이 수행하는 복구 가능한 일련의 조작입니다. 작업 단위는 트랜잭션이 시작하는 시점 또는 사용자가 요청한 동기점에서 시작합니다. 이는 사용자가 요청한 동기점에서 또는 트랜잭션이 끝나는 시점에 종료됩니다.

UOW

작업 단위(UOW, unit of work)를 참고하십시오.

사용자 백(user bag)

MQAI에서 사용자가 작성하는 데이터 백의 한 유형입니다.

UDP(User Datagram Protocol)

신뢰할 수 없는 비연결 데이터그램 서비스를 제공하는 인터넷 프로토콜입니다. 이는 하나의 시스템 또는 프로세스에 있는 애플리케이션 프로그램이 다른 시스템 또는 프로세스에 있는 애플리케이션 프로그램에 데이터그램을 송신할 수 있게 합니다.

사용자 항목(user item)

MQAI에서 사용자가 작성하는 데이터 항목의 한 유형입니다.

사용자 선택자(user selector)

WebSphere MQ 관리 인터페이스(MQAI)에서 데이터 항목을 식별하기 위해 데이터 백에 데이터 항목과 함께 놓인 ID입니다. WebSphere MQ는 WebSphere MQ 오브젝트에 대한 사전정의된 사용자 선택자를 제공합니다.

사용자 토큰(UTOKEN, user token)

사용자의 보안 특성을 캡슐화하거나 표시하는 RACF 보안 토큰입니다. RACF는 UTOKEN을 시스템의 각 사용자에게 지정합니다.

유틸리티

WebSphere MQ에서, WebSphere MQ 명령이 제공하는 기능과 함께 시스템 운영자나 시스템 관리자에게 기능을 부여하는 제공된 프로그램 세트입니다.

UTOKEN

사용자 토큰(UTOKEN, user token)을 참고하십시오.

V

값

데이터 항목의 콘텐츠입니다. 이는 정수, 문자열 또는 다른 데이터 백의 핸들일 수 있습니다.

가상 메소드(virtual method)

객체 지향 프로그래밍에서 다형성을 표현하는 메소드입니다.

W

WebSphere MQ

메시지 큐잉 서비스를 제공하는 IBM 라이선스가 있는 프로그램 제품군입니다.

WebSphere MQ 관리 인터페이스(MQAI, WebSphere MQ Administration Interface)

데이터 백을 사용하여 WebSphere MQ 큐 관리자에서 관리 태스크를 수행하는 프로그래밍 인터페이스입니다. 데이터 백을 통해 사용자는 WebSphere MQ 오브젝트의 특성(또는 매개변수)을 처리할 수 있습니다.

.NET용 WebSphere MQ 클래스(WebSphere MQ classes for .NET)

.NET 프로그래밍 프레임워크에 작성된 프로그램이 WebSphere MQ 클라이언트로서 WebSphere MQ에 연결되거나 직접 WebSphere MQ 서버에 연결될 수 있도록 하는 클래스 세트입니다.

C++용 WebSphere MQ 클래스(WebSphere MQ classes for C++)

C++ 프로그래밍 언어에서 WebSphere MQ 메시지 큐 인터페이스(MQI)를 캡슐화하는 클래스 세트입니다.

Java용 WebSphere MQ 클래스

Java 프로그래밍 언어에서 WebSphere MQ 메시지 큐 인터페이스(MQI)를 캡슐화하는 클래스 세트입니다.

WebSphere MQ 완전 관리 .NET 클라이언트(WebSphere MQ fully-managed .NET client)

전체 큐 관리자를 설치하지 않고 시스템에 설치할 수 있는 WebSphere MQ 제품의 파트입니다. WebSphere MQ .NET 클라이언트는 완전 관리 .NET 애플리케이션에서 사용되며 서버 시스템의 큐 관리자와 통신합니다. 완전히 관리되지 않는 .NET 애플리케이션은 WebSphere MQ MQI 클라이언트를 사용합니다. 클라이언트, WebSphere MQ MQI 클라이언트, WebSphere MQ Java 클라이언트도 참조하십시오.

WebSphere MQ Java 클라이언트(WebSphere MQ Java client)

전체 큐 관리자를 설치하지 않고 시스템에 설치할 수 있는 WebSphere MQ 제품의 파트입니다. WebSphere MQ Java 클라이언트는 Java 애플리케이션(Java용 WebSphere MQ 클래스 및 JMS용 WebSphere MQ 클래스 둘 다)에서 사용되며 서버 시스템의 큐 관리자와 통신합니다. 클라이언트, WebSphere MQ MQI 클라이언트, WebSphere MQ 완전 관리 .NET 클라이언트도 참조하십시오.

WebSphere MQ MQI 클라이언트

전체 큐 관리자를 설치하지 않고 시스템에 설치할 수 있는 WebSphere MQ 제품의 파트입니다. WebSphere MQ MQI 클라이언트는 애플리케이션의 MQI 호출을 승인하고 서버 시스템의 큐 관리자와 통신합니다. 클라이언트, WebSphere MQ Java 클라이언트, WebSphere MQ 완전 관리 .NET 클라이언트도 참조하십시오.

WebSphere MQ 스크립트 명령(MQSC, WebSphere MQ script commands)

WebSphere MQ 오브젝트를 조작하는 데 사용되는 육안으로 읽을 수 있는 명령이며 모든 플랫폼에서 동일합니다. PCF(programmable command format)도 참조하십시오.

WebSphere MQ 서버

하나 이상의 클라이언트에 큐잉 서비스를 제공하는 큐 관리자입니다. 모든 WebSphere MQ 오브젝트(예: 큐)는 큐 관리자 시스템, 즉 MQI 서버 시스템에서만 존재할 수 있습니다. 서버는 일반적인 로컬 MQI 애플리케이션도 지원할 수 있습니다.

WebSphere MQ Telemetry

WebSphere MQ Telemetry는 다양한 디바이스 플랫폼 상에서 실행되는 스마트 디바이스에 임베드되는 소형 클라이언트 라이브러리를 제공합니다. 이 클라이언트로 빌드한 애플리케이션은 MQTT(MQ Telemetry Transport)와 WebSphere MQ Telemetry 서비스를 사용하여 WebSphere MQ로 신뢰할 수 있는 메시지를 발행하고 구독합니다. 텔레메트리를 설치하는 WebSphere MQ 사용자 설치 옵션을 선택하면, 1) 텔레메트리(MQXR) 서비스 2) 텔레메트리 클라이언트 및 3) 텔레메트리 고급 클라이언트가 설치됩니다.

디바이스용 WebSphere MQ Telemetry 디먼(WebSphere MQ Telemetry daemon for devices)

디바이스용 WebSphere MQ Telemetry 디먼은 고급 MQTT V3 클라이언트입니다. 이는 임베드된 시스템을 위해 설계된 매우 작은 풋프린트 MQTT 서버입니다.

Windows NT 인증 확인/응답(Windows NT Challenge/Response)

독립형 시스템 및 Windows NT 시스템이 포함된 네트워크에서 사용되는 인증 프로토콜입니다.

회선 도청(wiretapping)

통신에서 사용되는 선 또는 기타 유형의 전도체를 따라 전달되는 정보에 대한 액세스를 확보하는 행위입니다. 회선 도청의 목적은 감지되지 않고 정보에 대한 비인가 액세스를 확보하는 것입니다.

X

X509

PKI에 대한 국제 전기 통신 연합(ITU) 표준입니다. 공개 키 인증서 및 공개 키 암호화의 형식을 지정합니다.

XCF

교차 시스템 커플링 기능(XCF, cross-system coupling facility)을 참고하십시오.

XID

트랜잭션 ID(transaction identifier)를 참고하십시오.

X/Open XA

X/Open 분산 트랜잭션 처리 XA 인터페이스입니다. 분산 트랜잭션 통신을 위해 제안된 표준입니다. 이 표준은 트랜잭션 내 공유 자원에 대한 액세스를 제공하는 자원 관리자 사이의 양방향 인터페이스와 트랜잭션을 모니터링하고 해석하는 트랜잭션 서비스 사이의 양방향 인터페이스를 지정합니다.

IBM WebSphere MQ의 접근성 기능

내게 필요한 옵션 기능을 사용하면 거동이 불편하거나 시각 장애가 있는 사용자도 정보 기술 콘텐츠를 정상적으로 사용할 수 있습니다.

내게 필요한 옵션 기능

IBM WebSphere MQ에는 다음의 주요 내게 필요한 옵션 기능이 포함되어 있습니다.

- 키보드 전용 조작
- 스크린 리더를 사용하는 조작

IBM WebSphere MQ는 최신 W3C 표준, [WAI-ARIA 1.0](https://www.w3.org/TR/wai-aria/)을 사용하여, [US Section 508](https://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards) 및 [Web Content Accessibility Guidelines\(WCAG\) 2.0](https://www.w3.org/TR/WCAG20/) 준수를 보장합니다. 내게 필요한 옵션 기능을 이용하려면 이 제품이 지원하는 최신 웹 브라우저와 함께 최신 릴리스의 스크린 리더를 사용하십시오.

IBM Documentation의 IBM WebSphere MQ 온라인 제품 정보에서 내게 필요한 옵션을 사용할 수 있습니다. IBM Documentation의 내게 필요한 옵션 기능은 <https://www.ibm.com/docs/about/releasesnotes.html>에 설명되어 있습니다.

키보드 탐색

이 제품은 표준 탐색 키를 사용합니다.

인터페이스 정보

IBM WebSphere MQ를 사용하는 완전히 액세스 가능한 방법은 명령행 인터페이스를 사용하는 것입니다. 명령 사용에 대한 자세한 정보는 [IBM WebSphere MQ 제어 명령 사용 방법 및 MQSC 명령을 사용하여 관리를 참조하십시오](#).

Windows의 경우, IBM WebSphere MQ 설치를 위해 가능한 방법은 대화식이 아닌 설치를 사용하는 것입니다. 자세한 정보는 [msiexec를 사용한 고급 설치를 참조하십시오](#).

IBM WebSphere MQ 사용자 인터페이스는 초당 2 - 55번 깜박이는 콘텐츠를 포함하지 않습니다.

IBM WebSphere MQ 웹 사용자 인터페이스는 콘텐츠를 올바르게 렌더링하고 유용한 경험을 제공하는 데 캐스케이딩 스타일시트를 사용하지 않습니다. 그러나 제품 문서는 캐스케이딩 스타일시트를 사용합니다. IBM WebSphere MQ에서는 시력이 좋지 않은 사용자가 고대비 모드를 포함하여 사용자의 시스템 표시 설정을 사용할 수 있는 동등한 방법을 제공합니다. 디바이스 또는 브라우저 설정을 사용하여 글꼴 크기를 제어할 수 있습니다.

관련된 내게 필요한 옵션 정보

표준 IBM 헬프 데스크 및 지원 웹 사이트 외에도 IBM은 청각 장애가 있는 고객이 영업 및 지원 서비스에 액세스하는 데 사용할 수 있도록 TTY 전화 서비스를 개설했습니다.

TTY 서비스
800-IBM-3383(800-426-3383)
(북미)

IBM과 내게 필요한 옵션

IBM이 내게 필요한 옵션에 대해 제공하는 약속에 대한 자세한 정보는 [IBM Accessibility\(www.ibm.com/able\)](http://www.ibm.com/able)의 내용을 참조하십시오.

Windows의 내게 필요한 옵션

IBM WebSphere MQ 사용자 인터페이스는 특수 키를 사용하지 않지만, 대신 컨텍스트 메뉴, 대화 상자 및 단추와 같은 대화 상자 제어 항목 등에 대한 단축키의 경우 Windows 사용자 인터페이스 지침을 따릅니다. 평소대로 단축키로 액세스하십시오. 자세한 정보는 Windows 도움말을 참조하십시오. (Windows 도움말 색인에서 키보드를 참조하십시오. 액세스 용이성의 경우 내게 필요한 옵션을 참조하십시오.)

액세스 용이성의 특수 기능

IBM WebSphere MQ에 있는 일부 사용자 인터페이스는 일반적으로 시각적이지만, 다음과 같이 액세스 가능성 기능이 활성화된 경우 다르게 작동합니다.

- 고선명 모드

이 모드에서는 런치패드, IBM WebSphere MQ 준비 마법사, Postcard 및 기본 구성이 모두 백그라운드 비트맵을 숨기고 시스템 텍스트 색상을 사용하여 쉽게 표시되고 읽을 수 있습니다.

- 화면 판독기 모드

화면 판독기가 활성화되면, IBM WebSphere MQ 준비 마법사, 기본 구성 및 Postcard는 백그라운드 비트맵, 상승된 효과, 음영 상자 및 기타 화면 판독기를 복잡하게 만들 수 있는 효과를 숨겨 화면을 단순하게 처리합니다.

- 탐색기 오브젝트 상태

IBM WebSphere MQ의 탐색기 컴포넌트에서는 큐 관리자와 같이 오브젝트의 상태를 나타내는 아이콘을 사용합니다. 스크린 리더는 이러한 아이콘을 해석할 수 없으므로 아이콘에 대한 텍스트 설명을 표시하는 옵션이 있습니다. 이 옵션을 선택하려면 탐색기에서 **창 > 환경 설정 > WebSphere MQ 탐색기** 를 클릭하고 **오브젝트 이름 뒤에 오브젝트 상태 표시** 를 선택하십시오.

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 디버거를 돕기 위해 제공된 것입니다.

중요사항: 이 진단, 수정 및 튜닝 정보는 변경될 수 있으므로 프로그래밍 인터페이스로 사용하지 마십시오.

상표

IBM, IBM 로고, ibm.com는 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: